A FRANK-WOLFE FRAMEWORK FOR EFFICIENT AND EFFECTIVE ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Depending on how much information an adversary can access to, adversarial attacks can be classified as white-box attack and black-box attack. In both cases, optimization-based attack algorithms can achieve relatively low distortions and high attack success rates. However, they usually suffer from poor time and query complexities, thereby limiting their practical usefulness. In this work, 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 the non-convex Frank-Wolfe algorithm. We show in theory that the proposed attack algorithms are efficient with an $O(1/\sqrt{T})$ convergence rate, which, to our knowledge, is the first convergence rate analysis for the zeroth-order non-convex Frank-Wolfe type algorithm. The empirical results on attacking Inception V3 model with the ImageNet dataset also verify the efficiency and effectiveness of the proposed algorithms. They attain a 100% attack success rate in both white-box and black-box attacks, and are more time and query efficient than the state-of-the-art baseline algorithms.

1 INTRODUCTION

Deep Neural Networks (DNNs) have made many breakthroughs in different areas of artificial intelligence such as image classification (Krizhevsky et al., 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 can be vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 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: 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., 2018b), reinforcement learning (Pattanaik et al., 2018), and even on systems that operate in the physical world (Kurakin et al., 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 et al., 2015) and black-box attack (Papernot et al., 2016a; 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. Among the approaches proposed for white-box and black-box attacks, optimization-based methods (Carlini & Wagner, 2017; Chen et al., 2017b;c; Ilyas et al., 2018; Cheng et al., 2018a) are the most effective: they usually achieve relatively low distortions and high attack success rates. However, these methods are not that efficient. In the white-box setting, they need to solve constrained optimization problems (Carlini & Wagner, 2017), and are usually significantly slower than the gradient-based methods such as Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) or Iterative FGSM (I-FGM) (Kurakin et al., 2016). In the black-box setting, it becomes even more severe since they need to make coordinate-wise gradient estimations (Chen et al., 2017c). Therefore, a large number of queries are needed for them to perform a successful attack, especially when the data dimension is large. For example, attacking a $299 \times 299 \times 3$ Imagenet image may take them hundreds of thousands of queries. This significantly limits their practical usefulness since they can be easily defeated by limiting the number of queries that an adversary can make to the target model.

In this study, we aim to examine the following questions in this study:

Can we improve the efficiency of the optimization-based attack algorithms? In other words, can we use less time and queries to conduct adversarial attacks?

In this work, we provide an affirmative answer to this question by proposing an efficient Frank-Wolfe optimization framework for both white-box and black-box attacks. In summary, we make the following main contributions:

- We propose a novel Frank-Wolfe based adversarial attack framework. The white-box attack algorithm is an iterative first-order method which admits the fast gradient sign method (FGSM) as the one-step special case. And the corresponding black-box attack algorithm adopts zeroth-order optimization with two sensing vector options (either from the Euclidean unit sphere or from the standard Gaussian distribution) provided.
- We show that the proposed white-box and black-box attack algorithms enjoy an $O(1/\sqrt{T})$ convergence rate. To the best of our knowledge, we provide the first convergence rate analysis for the zeroth-order non-convex Frank-Wolfe type algorithm.
- We show that the query complexity of the proposed black-box attack algorithm is linear in data dimension d.
- Our empirical results on attacking Inception V3 model with the ImageNet dataset show that (i) the proposed white-box attack algorithm is more efficient than all the baseline white-box algorithms evaluated here, and (ii) the proposed black-box attack algorithm is highly efficient and is also the only one algorithm that achieves a 100% attack success rate.

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 et al. (2015) proposed the Fast Gradient Sign Method (FGSM) based on linearization of the network as a simple alternative to L-BFGS. Kurakin et al. (2016) proposed to iteratively perform one-step FGSM (Goodfellow et al., 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_∞ norm case, BIM/I-FGM is equivalent to Projected Gradient Descent (PGD), which is a standard tool for constraint optimization. Papernot et al. (2016b) proposed JSMA to greedily attack the most significant pixel based on the Jacobian-based saliency map. Moosavi-Dezfooli et al. (2016) proposed attack methods by projecting the data to the closest separating hyperplane. Carlini & 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.

Black-box Attacks: One popular family of black-box attacks (Hu & Tan, 2017; Papernot et al., 2016a; 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.

Non-convex Frank-Wolfe Algorithms: The Frank-Wolfe algorithm (Frank & Wolfe, 1956), also known as the conditional gradient method, is an iterative optimization method for constrained op-

timization problem. Jaggi (2013) revisited Frank-Wolfe algorithm in 2013 and provided a stronger and more general convergence analysis in the convex setting. Yu et al. (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-sun optimization setting. Very recently, Staib & Jegelka (2017) proposed to use Frank-Wolfe for distributionally robust training (Sinha et al., 2018).

3 METHODOLOGY

3.1 NOTATIONS

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 $\|\mathbf{x}\|_p = (\sum_{i=1}^d \theta_i^p)^{1/p}$. Specially, for $p = \infty$, the L_∞ 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, which is considerably harder, requires to mislead the classifier to a specific target class. In this work, we follow the literature (Carlini & Wagner, 2017; Ilyas et al., 2018) and focus on the strictly harder targeted attack setting. It is worth noting that our proposed algorithm can be extended to untargeted attack straightforwardly.

Let us define $f(\cdot)$ as the classification loss function of the targeted DNN. For targeted attacks, we aim to learn an adversarial example $\mathbf{x}$ that is close enough to the original input $\mathbf{x}_{\text{ori}}$ and can be misclassified to the target class y_{tar}. The corresponding optimization problem [1] is defined as:

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

Evidently, the constraint set $\mathcal{X} := \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_{\text{ori}}\|_p \leq \epsilon\}$ is a bounded convex set when $p \geq 1$. Normally, $p = 2$ and $p = \infty$ are used to measure the distortions $\|\mathbf{x} - \mathbf{x}_{\text{ori}}\|_p$, resulting in L_2 attack model and L_∞ attack model respectively. In this work, we study both models.

3.3 FRANK-WOLFE WHITE-BOX ATTACKS

Frank-Wolfe algorithm (Frank & Wolfe, 1956), also known as the conditional gradient descent, is a popular optimization tool for constraint optimization. Different from PGD that first performs gradient descent followed by a projection step at each iteration, Frank-Wolfe algorithm calls a Linear Minimization Oracle (LMO) over the the constraint set $\mathcal{X}$ at each iteration, i.e.,

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

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

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

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.

We present our proposed Frank-Wolfe white-box attack algorithm in Algorithm 1, which is built upon the original Frank-Wolfe algorithm. The key difference between Algorithm 1 and the standard Frank-Wolfe algorithm is in Line 4, where the LMO is called over a slightly relaxed constraint set $\mathcal{X}_\lambda := \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_{\text{ori}}\|_p \leq \lambda\epsilon\}$ with $\lambda \geq 1$, instead of the original constraint set $\mathcal{X}$. When $\lambda = 1$, set $\mathcal{X}_\lambda$ reduces to $\mathcal{X}$, and Algorithm 1 reduces to standard Frank Wolfe. We argue that this modification makes our algorithm more general, and gives rise to better attack results.

The LMO solution itself can be expensive to obtain in general. Fortunately, applying Frank-Wolfe to solve (3.1) actually gives us a close-form LMO solution. We provide the solutions of LMO (Line

[1] 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.

Algorithm 1 Frank-Wolfe White-box Attack Algorithm

input: number of iterations T, step sizes $\{\gamma_t\}$, $\lambda > 0$, original image $\mathbf{x}_{\text{ori}}$, target label y_{tar};
$\mathbf{x}_0 = \mathbf{x}_{\text{ori}}$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{v}_t = \text{argmin}_{\mathbf{v} \in \mathcal{X}_\lambda} \langle \mathbf{v}, \nabla f(\mathbf{x}_t, y_{\text{tar}}) \rangle$ // LMO
  $\mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$
  $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t \mathbf{d}_t$
  if $\lambda > 1$ **then**
    $\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}(\mathbf{x}_{t+1})$
  end if
end for
output: $\mathbf{x}_T$

4 in Algorithm 1) for L_2 norm and L_∞ norm cases respectively:

$$\mathbf{v} = -\frac{\lambda\epsilon \cdot \nabla f(\mathbf{x}_t, y_{\text{tar}})}{\|\nabla f(\mathbf{x}_t, y_{\text{tar}})\|_2} + \mathbf{x}_{\text{ori}}, \qquad (L_2 \text{ norm})$$

$$\mathbf{v} = -\lambda\epsilon \cdot \text{sign}(\nabla f(\mathbf{x}_t, y_{\text{tar}})) + \mathbf{x}_{\text{ori}}. \qquad (L_\infty \text{ norm})$$

The derivation can be found in the supplemental materials.

Note that when $T = 1, \lambda = 1$, substituting the above LMO solutions into Algorithm 1 yields the final update of $x_1 = x_0 - \gamma_t \epsilon \cdot \nabla f(\mathbf{x}_t, y_{\text{tar}})$, which reduces to FGSM [2] when $\gamma_t = 1$. Similar derivation also applies to L_2 norm case. Therefore, just like PGD, our proposed Frank-Wolfe white-box attack also includes FGSM (FGM) as a one-step special instance.

3.4 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. The label-only setting is doable under our framework, but will incur extra difficulty such as designing new loss functions. For simplicity, here we consider the confidence score output.

We propose a zeroth-order Frank-Wolfe based algorithm to solve this problem. Algorithm 2 show our proposed Frank-Wolfe black-box attack algorithm. 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 note that for the final output, we provide two options. While option II is the common choice in practice, option I is also provided for the ease of theoretical analysis.

Algorithm 2 Frank-Wolfe Black-box Attack Algorithm

1: **input:** number of iterations T, step sizes $\{\gamma_t\}$, $\lambda > 0$, original image $\mathbf{x}_{\text{ori}}$, target label y_{tar};
2: $\mathbf{x}_0 = \mathbf{x}_{\text{ori}}$
3: **for** $t = 0, \ldots, T-1$ **do**
4: $\mathbf{q}_t = \text{ZERO_ORD_GRAD_EST}(\mathbf{x}_t, y_{\text{tar}})$ // Algorithm 3
5: $\mathbf{v}_t = \text{argmin}_{\mathbf{v} \in \mathcal{X}_\lambda} \langle \mathbf{v}, \mathbf{q}_t \rangle$
6: $\mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$
7: $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t \mathbf{d}_t$
8: **if** $\lambda > 1$ **then**
9: $\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}(\mathbf{x}_{t+1})$
10: **end if**
11: **end for**
12: **Option I:** $\mathbf{x}_a$ is uniformly random chosen from $\{\mathbf{x}_t\}_{t=1}^T$
13: **Option II:** $\mathbf{x}_a = \mathbf{x}_T$
14: **output:** $\mathbf{x}_a$

[2] 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 many other zeroth-order optimization algorithms (Shamir, 2017; Flaxman et al., 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 Zeroth-Order Gradient Estimation (ZERO_ORD_GRAD_EST)

parameters: number of gradient estimation samples b, sampling parameter δ_t;
$\mathbf{q} = \mathbf{0}$
for $i = 1, \ldots, b$ **do**
  Option I: Sample $\mathbf{u}_i$ uniformly from the Euclidean unit sphere with $\|\mathbf{u}_i\|_2 = 1$, $\lambda = d$
  Option II: Sample $\mathbf{u}_i$ uniformly from the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, $\lambda = 1$
  $\mathbf{q} = \mathbf{q} + \frac{\lambda}{2\delta_t b}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y_{\text{tar}}) - f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y_{\text{tar}})\big)\mathbf{u}_i$
end for
output: $\mathbf{q}$

4 Main Theory

In this section, we establish the convergence guarantees for our proposed Frank-Wolfe adversarial attack algorithms described in Section 3. 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):

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

Note that for the Frank-Wolfe gap, 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 $g(\mathbf{x}_t) = 0$. Also the Frank-Wolfe gap is affine invariant and do not tie to any specific choice of norm, which makes itself 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

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

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 [3]. This justifies the validity of Assumption 4.1.

[3] 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 backpropagation) as the network inputs. Therefore, the argument in Santurkar et al. (2018) can also be applied here with respect to the network inputs.

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 ϵ is a small constant, this assumption trivially holds.

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 4.3. Under Assumptions 4.1 and 4.2, let $\gamma = \sqrt{2(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))/(LD^2T)}$, denote $\widetilde{g}_T = \min_{1\leq k\leq T} g(\mathbf{x}_k)$ where $\{\mathbf{x}_k\}_{k=1}^T$ are iterates in Algorithm 1 with $\lambda = 1$, we have:

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

where $\mathbf{x}^*$ is the optimal solution to (3.1).

Remark 4.4. Theorem 4.3 suggests that our proposed Frank-Wolfe white-box attack algorithm achieves a $O(1/\sqrt{T})$ rate of convergence.

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 $\|\nabla f(\mathbf{0}, y)\|_2$ is bounded.

Assumption 4.5. Gradient of $f(\cdot)$ at zero point $\nabla f(\mathbf{0}, y)$ satisfies $\max_y \|\nabla f(\mathbf{0}, y)\|_2 \leq C_g$.

Following the analysis in Shamir (2017), let $f_\delta(\mathbf{x}, y) = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x} + \delta\mathbf{u}, y)]$, which is the smoothed version of $f(\mathbf{x}, y)$. 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 4.6. For the gradient estimator $\mathbf{q}_t$ in Algorithm 3, its expectation and variance satisfy

$$\mathbb{E}[\mathbf{q}_t] = \nabla f_\delta(\mathbf{x}_t, y_{\text{tar}}), \qquad \mathbb{E}\|\mathbf{q}_t - \mathbb{E}[\mathbf{q}_t]\|_2^2 \leq \frac{1}{b}\bigg(2d(C_g + LD)^2 + \frac{1}{2}\delta_t^2 L^2 d^2\bigg).$$

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

Theorem 4.7. Under Assumptions 4.1, 4.2 and 4.5, let $\gamma = \sqrt{2(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))/(LD^2T)}$, $b = Td$ and $\delta_t = \sqrt{2/Td^2}$, suppose we use Option I in Algorithm 2 and option II for Algorithm 3, then the output $\mathbf{x}_a$ from Algorithm 2 with $\lambda = 1$ satisfies:

$$g(\mathbf{x}_a) \leq \frac{D}{\sqrt{2T}}\bigg(\sqrt{L(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))} + 2(L + C_g + LD)\bigg),$$

where $\mathbf{x}^*$ is the optimal solution to (3.1).

Remark 4.8. Theorem 4.7 suggests that Algorithm 2 also enjoys a $O(1/\sqrt{T})$ rate of convergence. In terms of query complexity, the total number of queries needed is $Tb = T^2d$, which is linear in the data dimension d. In fact, in the experiment part, we observed that this number can be substantially smaller than d, e.g., $b = 25$, which is much lower than the theorem suggests. Note that although we only prove for option II in Algorithm 3, our result can be readily extended to Option I (the Gaussian sensing vector case).

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.

Table 1: Comparison of L_2 norm based white-box attacks on Inception V3 model with $\epsilon = 5$. We report attack success rate, average time and average distortion.

METHODS	SUCCESS RATE (%)	AVERAGE TIME (s)	AVERAGE DISTORTION
FGM	0	-	-
BIM/ I-FGM	100	197.67	0.79
CW	100	167.07	**0.55**
EAD	100	175.20	1.07
FW-White	100	**28.39**	0.87

5.1 Evaluation Setup and Metrics

We test the attack effectiveness of all algorithms by evaluating on a pre-trained Inception V3 model (Szegedy et al., 2016) that is trained on ImageNet dataset (Deng et al., 2009). The pre-trained model is reported to have a 78% top-1 accuracy. We randomly choose 200 images from the ImageNet validation set that are verified to be correctly classified by the pre-trained model and also randomly choose a target class for each image. Each image has a dimension of $299 \times 299 \times 3$ and we test all attack algorithms through the same randomly chosen data samples and target labels.

We test for both L_2 norm based and L_∞ norm based attacks. For L_2 norm based attack, we set $\epsilon = 5$ and for L_∞ based attack, we set $\epsilon = 0.05$. For white-box attack, we restrict a maximum of $1,000$ iterations per attack for each method. And for black-box attack, we set a maximum query limit of $500,000$ per attack per image for each method.

For all algorithms, we stop the algorithm when a successful attack is found. For our proposed black-box attack, we use option II in Algorithm 2 and test both options in Algorithm 3. We set the number of gradient estimation samples $b = 25$ for Algorithm 2. More detailed description on parameter settings can be found in the supplemental materials.

We evaluate the final performance through attack success rate where the success is defined as making the classifier output the exact target class label (not any incorrect labels). We also measure average attack time per image, average distortion (only on successful attacked samples) and average number of queries needed (only for black-box attack) per image. For a fair time comparison, even though some of the algorithms including ours can be written in batch form (attack multiple images at one time), all algorithms are set to attack one image at a time.

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) FGM (FGSM) (Goodfellow et al., 2015) (ii) BIM / I-FGM (Kurakin et al., 2016) (iii) CW attack (Carlini & Wagner, 2017) and (iv) EAD attack (Chen et al., 2017b). We compare the proposed black-box attack algorithm with (i) ZOO attack (Chen et al., 2017c) and (ii) NES-PGD attack (Ilyas et al., 2018).

5.3 White-box Attack Experiments

In this subsection, we present the white-box attack experiments on Inception V3 model. Tables 1 and 2 present our experimental results for L_2 norm and L_∞ norm based white-box attacks respectively. As we can observe from the tables, the attack success rate is basically 100% for every method except for the simplest FGM (FGSM) attack. For the other baselines in the L_2 norm case, CW method achieves the smallest average distortion, yet it comes with an expansive time cost. EAD method does not have either time advantage or distortion advantage in this experiment, probably due to its different motivation in attacking. BIM/ I-FGM has relatively small average distortion, yet it also costs the most attack time among all methods. On the other hand, our proposed algorithm achieves the shortest attack time with moderate distortion. It significantly reduces the time complexity needed for attacking data with large dimensionality. For the L_∞ norm case, CW method takes significantly longer time and does not perform very well on average distortion either. This is largely due to the special design of CW method to make it possible for L_∞ norm attack. Again, our proposed white-box attack algorithm achieves the shortest average attack time and a moderate average distortion.

Table 2: Comparison of L_∞ norm based white-box attacks on Inception V3 model with $\epsilon = 0.05$. We report attack success rate, average time and average distortion.

METHODS	SUCCESS RATE (%)	AVERAGE TIME (s)	AVERAGE DISTORTION
FGM	0	-	-
BIM/ I-FGM	100	99.44	**0.0026**
CW	100	722.90	0.0067
FW-White	100	**4.41**	0.0045

5.4 Black-box Attack Experiments

In this subsection, we present the black-box experiments on Inception V3 model. For black-box attacks, attack success rate, time needed and number of queries needed are more meaningful evaluation metrics than distortion distances. Therefore, we omit all the grid search / binary search steps that are used in the white-box setting since extra time / queries are needed for finding parameters that can obtain better distortion distances.

Tables 3 and 4 present our experimental results for L_2 norm and L_∞ norm based black-box attacks respectively. For ZOO method, note that it only has the L_2 norm version and it follows CW's framework and thus uses different loss function and problem formulation (cannot exactly control the adversarial example to fit the distortion limit, we manage to keep the average distortion around ϵ for ZOO while other methods have average distortions very close to ϵ). Furthermore, we can observe that ZOO is quite slow in this task. Attack on a single image can take up to 2 hours for ZOO and it is only able to achieve a 74.5% success rate (comparing with the 88.9% success rate in the original paper, we think main cause is the query limit here is only half of the query limit in the origin paper). NES-PGD method, while greatly improving ZOO's performance, still cannot achieve 100% success rate in both settings and takes relatively more time and queries. In sharp contrast, our proposed Frank-Wolfe black-box attacks (both option I and option II) achieve 100% success rate in both L_2 norm and L_∞ norm based attacks and further improve the attack efficiency to another level (around 100s per image for L_2 norm case and less than 50s per image for L_∞ norm case).

Table 3: Comparison of L_2 norm based black-box attacks on Inception V3 model with $\epsilon = 5$. We report attack success rate, average time and average number of queries needed per image. Opt I and Opt II refer to the two options in Algorithm 2.

METHODS	SUCCESS RATE (%)	AVERAGE TIME (s)	AVERAGE QUERIES
ZOO	74.5	5825.89	294735.87
NES-PGD	96.5	130.45	58957.55
FW-Black (Opt I)	**100**	**100.75**	45410.50
FW-Black (Opt II)	**100**	100.91	**44999.75**

Table 4: Comparison of L_∞ norm based black-box attacks on Inception V3 model with $\epsilon = 0.05$. We report attack success rate, average time and average number of queries needed per image. Opt I and Opt II refer to the two options in Algorithm 2.

METHODS	SUCCESS RATE (%)	AVERAGE TIME (s)	AVERAGE QUERIES
NES-PGD	98	73.58	32741.32
FW-Black (Opt I)	**100**	47.56	21129.25
FW-Black (Opt II)	**100**	**46.48**	**20694.75**

6 Conclusions

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 $O(1/\sqrt{T})$ 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 Inception V3 model with ImageNet dataset yields a 100% attack success rate for our proposed algorithms, even in the setting of black-box attack.

REFERENCES

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4945–4949. IEEE, 2016.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. *arXiv preprint arXiv:1712.09491*, 2017.

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

Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security Symposium*, pp. 513–530, 2016.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. Show-and-fool: Crafting adversarial examples for neural image captioning. *arXiv preprint arXiv:1712.02051*, 2017a.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017b.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 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*, pp. 15–26. ACM, 2017c.

Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018a.

Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*, 2018b.

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

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

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

Xiang Gao, Bo Jiang, and Shuzhong Zhang. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76(1):327–363, 2018.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

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

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

Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin, Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning,{ICML} 2018*, 2018.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

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

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

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Data Mining (ICDM)*, 2018.

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

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

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

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

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

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

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2040–2042. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pp. 1244–1251. IEEE, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?(no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*, 2018.

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

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. *International Conference on Learning Representations*, 2018.

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

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

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

Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *CoRR*, abs/1805.11770, 2018.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darell, and Dawn Song. Can you fool ai with adversarial examples on a visual turing test? *arXiv preprint arXiv:1709.08693*, 2017.

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

A LINEAR MINIMIZATION ORACLE (LMO) SOLUTIONS

Denote $\mathbf{u} = (\mathbf{v} - \mathbf{x}_{\text{ori}})/(\lambda\epsilon)$, the linear minimization problem can be written as

$$\begin{aligned}\min_{\|\mathbf{v}-\mathbf{x}_{\text{ori}}\|_p \le \lambda\epsilon} \langle \mathbf{v}, \nabla f(\mathbf{x}_t, y_{\text{tar}})\rangle &= \min_{\|\mathbf{u}\|_p \le 1} \lambda\epsilon \cdot \langle \mathbf{u}, \nabla f(\mathbf{x}_t, y_{\text{tar}})\rangle \\ &= \max_{\|\mathbf{u}\|_p \le 1} \lambda\epsilon \cdot \langle \mathbf{u}, -\nabla f(\mathbf{x}_t, y_{\text{tar}})\rangle \\ &= \lambda\epsilon \cdot \|\nabla f(\mathbf{x}_t, y_{\text{tar}})\|_{p*},\end{aligned}$$

where $\|\cdot\|_{p*}$ denotes the dual norm of $\|\cdot\|_p$. For $p = 2$ case, we have

$$\langle (\mathbf{v} - \mathbf{x}_{\text{ori}})/(\lambda\epsilon), -\nabla f(\mathbf{x}_t, y_{\text{tar}})\rangle = \|\nabla f(\mathbf{x}_t, y_{\text{tar}})\|_2.$$

It immediately implies that

$$\mathbf{v} = -\frac{\lambda\epsilon \cdot \nabla f(\mathbf{x}_t, y_{\text{tar}})}{\|\nabla f(\mathbf{x}_t, y_{\text{tar}})\|_2} + \mathbf{x}_{\text{ori}}.$$

For $p = \infty$ case, we have

$$\langle (\mathbf{v} - \mathbf{x}_{\text{ori}})/(\lambda\epsilon), -\nabla f(\mathbf{x}_t, y_{\text{tar}})\rangle = \|\nabla f(\mathbf{x}_t, y_{\text{tar}})\|_1.$$

It immediately implies that

$$\mathbf{v} = -\lambda\epsilon \cdot \text{sign}(\nabla f(\mathbf{x}_t, y_{\text{tar}})) + \mathbf{x}_{\text{ori}}.$$

B PROOF OF THE MAIN THEORY IN SECTION 4

B.1 PROOF OF THEOREM 4.3

Proof. For simplicity, we denote $f(\mathbf{x}_t, y_{\text{tar}})$ by $f(\mathbf{x}_t, y)$ for the rest of the proof. First by Assumption 4.1, we have

$$\begin{aligned}f(\mathbf{x}_{t+1}, y) &\le f(\mathbf{x}_t, y) + \nabla f(\mathbf{x}_t, y)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2 \\ &= f(\mathbf{x}_t, y) + \gamma \nabla f(\mathbf{x}_t, y)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{L\gamma^2}{2}\|\mathbf{v}_t - \mathbf{x}_t\|_2^2 \\ &\le f(\mathbf{x}_t, y) + \gamma \nabla f(\mathbf{x}_t, y)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{LD^2\gamma^2}{2},\end{aligned}$$

where the last inequality uses the bounded domain condition in Assumption 4.2. Note that by definition of the Frank-Wolfe gap, we have

$$f(\mathbf{x}_{t+1}, y) \le f(\mathbf{x}_t, y) - \gamma g(\mathbf{x}_t) + \frac{LD^2\gamma^2}{2}.$$

Do telescope sum over t of the above inequality, we obtain

$$\begin{aligned}f(\mathbf{x}_T, y) &\le f(\mathbf{x}_0, y) - \sum_{k=0}^{T-1} \gamma g(\mathbf{x}_k) + \frac{TLD^2\gamma^2}{2} \\ &\le f(\mathbf{x}_0, y) - \gamma T \widetilde{g}_T + \frac{TLD^2\gamma^2}{2},\end{aligned}$$

where the second inequality follows from the definition of $\widetilde{g}_t$. Note that by optimality we easily have $f(\mathbf{x}_{t+1}, y) \ge f(\mathbf{x}^*, y)$. Rearrange the above inequality we have

$$\begin{aligned}\widetilde{g}_T &\le \frac{f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y)}{T\gamma} + \frac{LD^2\gamma}{2} \\ &\le \sqrt{\frac{LD^2(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))}{2T}},\end{aligned}$$

where the second inequality is achieved when $\gamma = \sqrt{2(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))/(LD^2T)}$. □

B.2 PROOF OF LEMMA 4.6

Proof. For simplicity we denote $f(\cdot, y_{\text{tar}})$ by $f(\cdot, y)$ for the rest of the proof. Let us denote $\boldsymbol{\psi}_i = \frac{d}{2\delta_t b}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y) - f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y)\big)\mathbf{u}_i$. For the first part, we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\psi}_i] &= \mathbb{E}_{\mathbf{u}}\bigg[\frac{d}{2\delta_t b}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y) - f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y)\big)\mathbf{u}_i\bigg] \\
&= \mathbb{E}_{\mathbf{u}}\bigg[\frac{d}{2\delta_t b} f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y)\mathbf{u}_i\bigg] + \mathbb{E}_{\mathbf{u}}\bigg[\frac{d}{2\delta_t b} f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y)(-\mathbf{u}_i)\bigg] \\
&= \mathbb{E}_{\mathbf{u}}\bigg[\frac{d}{\delta_t b} f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y)\mathbf{u}_i\bigg] \\
&= \frac{1}{b}\nabla f_\delta(\mathbf{x}_t, y),
\end{aligned}$$

where the third equality holds due to symmetricity of $\mathbf{u}_i$ and the last eqaulity follows from Lemma 4.1(a) in Gao et al. (2018). Therefore, we have

$$\mathbb{E}[\mathbf{q}_t] = \mathbb{E}\bigg[\sum_{i=1}^{b} \boldsymbol{\psi}_i\bigg] = \nabla f_\delta(\mathbf{x}_t, y).$$

For second part, note that $\boldsymbol{\psi}_i$'s are independent from each other due to the independence of $\mathbf{u}_i$, we have

$$\mathbb{E}\|\mathbf{q}_t - \mathbb{E}[\mathbf{q}_t]\|_2^2 = \mathbb{E}\bigg\|\sum_{i=1}^{b}\Big[\boldsymbol{\psi}_i - \mathbb{E}\boldsymbol{\psi}_i\Big]\bigg\|_2^2 = \sum_{i=1}^{b}\mathbb{E}\big\|\boldsymbol{\psi}_i - \mathbb{E}\boldsymbol{\psi}_i\big\|^2 \le \sum_{i=1}^{b}\mathbb{E}\big\|\boldsymbol{\psi}_i\big\|^2.$$

Now take a look at $\mathbb{E}\big\|\boldsymbol{\psi}_i\big\|^2$:

$$\begin{aligned}
\mathbb{E}\big\|\boldsymbol{\psi}_i\big\|^2 &= \mathbb{E}_{\mathbf{u}}\bigg\|\frac{d}{2\delta_t b}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y) - f(\mathbf{x}_t, y) + f(\mathbf{x}_t, y) - f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y)\big)\mathbf{u}_i\bigg\|_2^2 \\
&\le \frac{1}{2b^2}\mathbb{E}_{\mathbf{u}}\bigg\|\frac{d}{\delta_t}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y) - f(\mathbf{x}_t, y)\big)\mathbf{u}_i\bigg\|_2^2 + \frac{1}{2b^2}\mathbb{E}_{\mathbf{u}}\bigg\|\frac{d}{\delta_t}\big(f(\mathbf{x}_t, y) - f(\mathbf{x}_t - \delta_t \mathbf{u}_i, y)\big)\mathbf{u}_i\bigg\|_2^2 \\
&= \frac{1}{b^2}\mathbb{E}_{\mathbf{u}}\bigg\|\frac{d}{\delta_t}\big(f(\mathbf{x}_t + \delta_t \mathbf{u}_i, y) - f(\mathbf{x}_t, y)\big)\mathbf{u}_i\bigg\|_2^2 \\
&\le \frac{1}{b^2}\bigg(2d\|\nabla f(\mathbf{x}_t, y)\|_2^2 + \frac{1}{2}\delta_t^2 L^2 d^2\bigg),
\end{aligned}$$

where the first inequality is due to the fact that $(a+b)^2 \le 2a^2 + 2b^2$, the second equality follows from the symmetricity of $\mathbf{u}_i$ and the last inequality is by Lemma 4.1(b) in Gao et al. (2018). Also note that by Assumption 4.1 and 4.5 we have

$$\|\nabla f(\mathbf{x}_t, y)\|_2^2 \le (\|\nabla f(\mathbf{0}, y))\|_2 + L\|\mathbf{x}_t\|_2)^2 \le (C_g + LD)^2.$$

Combine all above results, we obtain

$$\mathbb{E}\|\mathbf{q}_t - \mathbb{E}[\mathbf{q}_t]\|_2^2 \le \frac{1}{b}\bigg(2d(C_g + LD)^2 + \frac{1}{2}\delta_t^2 L^2 d^2\bigg).$$

□

B.3 PROOF OF THEOREM 4.7

Proof. For simplicity we denote $f(\mathbf{x}_t, y_{\text{tar}})$ by $f(\mathbf{x}_t, y)$ for the rest of the proof. First by Assumption 4.1, we have

$$\begin{aligned}
f(\mathbf{x}_{t+1}, y) &\leq f(\mathbf{x}_t, y) + \nabla f(\mathbf{x}_t, y)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2 \\
&= f(\mathbf{x}_t, y) + \gamma \nabla f(\mathbf{x}_t, y)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{L\gamma^2}{2}\|\mathbf{v}_t - \mathbf{x}_t\|_2^2 \\
&\leq f(\mathbf{x}_t, y) + \gamma \nabla f(\mathbf{x}_t, y)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{LD^2\gamma^2}{2} \\
&= f(\mathbf{x}_t, y) + \gamma \mathbf{q}_t^\top (\mathbf{v}_t - \mathbf{x}_t) + \gamma (\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{LD^2\gamma^2}{2},
\end{aligned}$$

where the second inequality uses the bounded domain condition in Assumption 4.2. Now define an auxialiary quantity:

$$\widehat{\mathbf{v}}_t = \operatorname*{argmin}_{\mathbf{v} \in \mathcal{X}} \langle \mathbf{v}, \nabla f(\mathbf{x}_t, y) \rangle.$$

According to the definition of $g(\mathbf{x}_t)$, this immediately implies

$$g(\mathbf{x}_t) = \langle \widehat{\mathbf{v}}_t, \nabla f(\mathbf{x}_t, y) \rangle.$$

Then we further have

$$\begin{aligned}
f(\mathbf{x}_{t+1}, y) &\leq f(\mathbf{x}_t, y) + \gamma \mathbf{q}_t^\top (\widehat{\mathbf{v}}_t - \mathbf{x}_t) + \gamma (\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{LD^2\gamma^2}{2} \\
&= f(\mathbf{x}_t, y) + \gamma \nabla f(\mathbf{x}_t, y)^\top (\widehat{\mathbf{v}}_t - \mathbf{x}_t) + \gamma (\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t)^\top (\mathbf{v}_t - \widehat{\mathbf{v}}_t) + \frac{LD^2\gamma^2}{2} \\
&= f(\mathbf{x}_t, y) - \gamma g(\mathbf{x}_t) + \gamma (\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t)^\top (\mathbf{v}_t - \widehat{\mathbf{v}}_t) + \frac{LD^2\gamma^2}{2} \\
&\leq f(\mathbf{x}_t, y) - \gamma g(\mathbf{x}_t) + \gamma D \cdot \|\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t\|_2 + \frac{LD^2\gamma^2}{2},
\end{aligned}$$

where the first inequality follows from the optimally of $\mathbf{v}_t$ in Algorithm 2 and the last inequality holds due to Cauchy-Schwarz inequality. Take expectations for both sides of the above inequality, we have

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{x}_{t+1}, y)] \\
&\leq \mathbb{E}[f(\mathbf{x}_t, y)] - \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma D \cdot \mathbb{E}\|\nabla f(\mathbf{x}_t, y) - \mathbf{q}_t\|_2 + \frac{LD^2\gamma^2}{2} \\
&\leq \mathbb{E}[f(\mathbf{x}_t, y)] - \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma D \cdot \big(\|\nabla f(\mathbf{x}_t, y) - \mathbb{E}[\mathbf{q}_t]\|_2 + \mathbb{E}\|\mathbf{q}_t - \mathbb{E}[\mathbf{q}_t]\|_2\big) + \frac{LD^2\gamma^2}{2}, \\
&\leq \mathbb{E}[f(\mathbf{x}_t, y)] - \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma D \cdot \Big(\|\nabla f(\mathbf{x}_t, y) - \mathbb{E}[\mathbf{q}_t]\|_2 + \sqrt{\mathbb{E}\|\mathbf{q}_t - \mathbb{E}[\mathbf{q}_t]\|_2^2}\Big) + \frac{LD^2\gamma^2}{2} \\
&\leq \mathbb{E}[f(\mathbf{x}_t, y)] - \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma D \cdot \left(\|\nabla f(\mathbf{x}_t, y) - \nabla f_\delta(\mathbf{x}_t, y)\|_2 + \sqrt{\frac{4d(C_g + LD)^2 + \delta_t^2 L^2 d^2}{2b}}\right) \\
&\qquad + \frac{LD^2\gamma^2}{2}, \\
&\leq \mathbb{E}[f(\mathbf{x}_t, y)] - \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma D \cdot \left(\frac{\delta_t L d}{2} + \frac{2\sqrt{d}(C_g + LD) + \delta_t L d}{\sqrt{2b}}\right) + \frac{LD^2\gamma^2}{2},
\end{aligned}$$

where the second inequality follows from triangle inequality, the third inequality is due to Jenson's inequality and the last inequality holds due to Lemma 4.6.

Do telescope sum over t of the above inequality, we obtain

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{x}_T, y)] \\
&\leq f(\mathbf{x}_0, y) - \sum_{t=0}^{T-1} \gamma \mathbb{E}[g(\mathbf{x}_t)] + \gamma DT \left(\frac{\delta_t L d}{2} + \frac{2\sqrt{d}(C_g + LD) + \delta_t L d}{\sqrt{2b}} \right) + \frac{TLD^2\gamma^2}{2} \\
&\leq f(\mathbf{x}_0, y) - \gamma T g_a + \gamma DT \left(\frac{\delta_t L d}{2} + \frac{2\sqrt{d}(C_g + LD) + \delta_t L d}{\sqrt{2b}} \right) + \frac{TLD^2\gamma^2}{2},
\end{aligned}$$

where the second inequality follows from the definition of $\widetilde{g}_a$. Note that by the zeroth-order optimality, we have $f(\mathbf{x}_{t+1}, y) \geq f(\mathbf{x}^*, y)$. Rearrange the above inequality we obtain

$$\begin{aligned}
g_a &\leq \frac{f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y)}{T\gamma} + \frac{LD^2\gamma}{2} + D \left(\frac{\delta_t L d}{2} + \frac{2\sqrt{d}(C_g + LD) + \delta_t L d}{\sqrt{2b}} \right) \\
&\leq \frac{D}{\sqrt{2T}} \Big(\sqrt{L(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))} + 2(L + C_g + LD) \Big),
\end{aligned}$$

where the second inequality is achieved by setting $\gamma = \sqrt{2(f(\mathbf{x}_0, y) - f(\mathbf{x}^*, y))/(LD^2T)}$, $b = Td$ and $\delta_t = \sqrt{2/Td^2}$.

□

C Parameters Settings for Section 5

For Frank-Wolfe white-box attack algorithm, we list the parameters we use in Section 5 at Table 5.

Table 5: Parameters used in Frank-Wolfe white-box attack.

PARAMETER	L_2 **CASE**	L_{inf} **CASE**
T	1000	1000
$\{\gamma_t\}$	0.03	0.005
λ	20	5

Similarly, for Frank-Wolfe black-box attack algorithm, we also list the parameters we use in Section 5 at Table 6.

Table 6: Parameters used in Frank-Wolfe black-box attack.

PARAMETER	L_2 **CASE**	L_{inf} **CASE**
T	10000	10000
$\{\gamma_t\}$	$0.05/\sqrt{t}$	$0.03/\sqrt{t}$
λ	30	30
b	25	25
δ_t	0.001	0.01

D Additional Experiments

For the completeness, we also provide some visual illustrations on the adversarial examples generated by various algorithms. Figure 1 shows some examples adversarial examples generated through different L_2 norm based white-box attacks. Figure 2 shows some adversarial examples generated through different L_∞ norm based black-box attacks.

Figure 1: Sample adversarial examples generated through different L_2 norm based white-box attacks. Left side labels denote the original class label and right side labels denote the target class label.

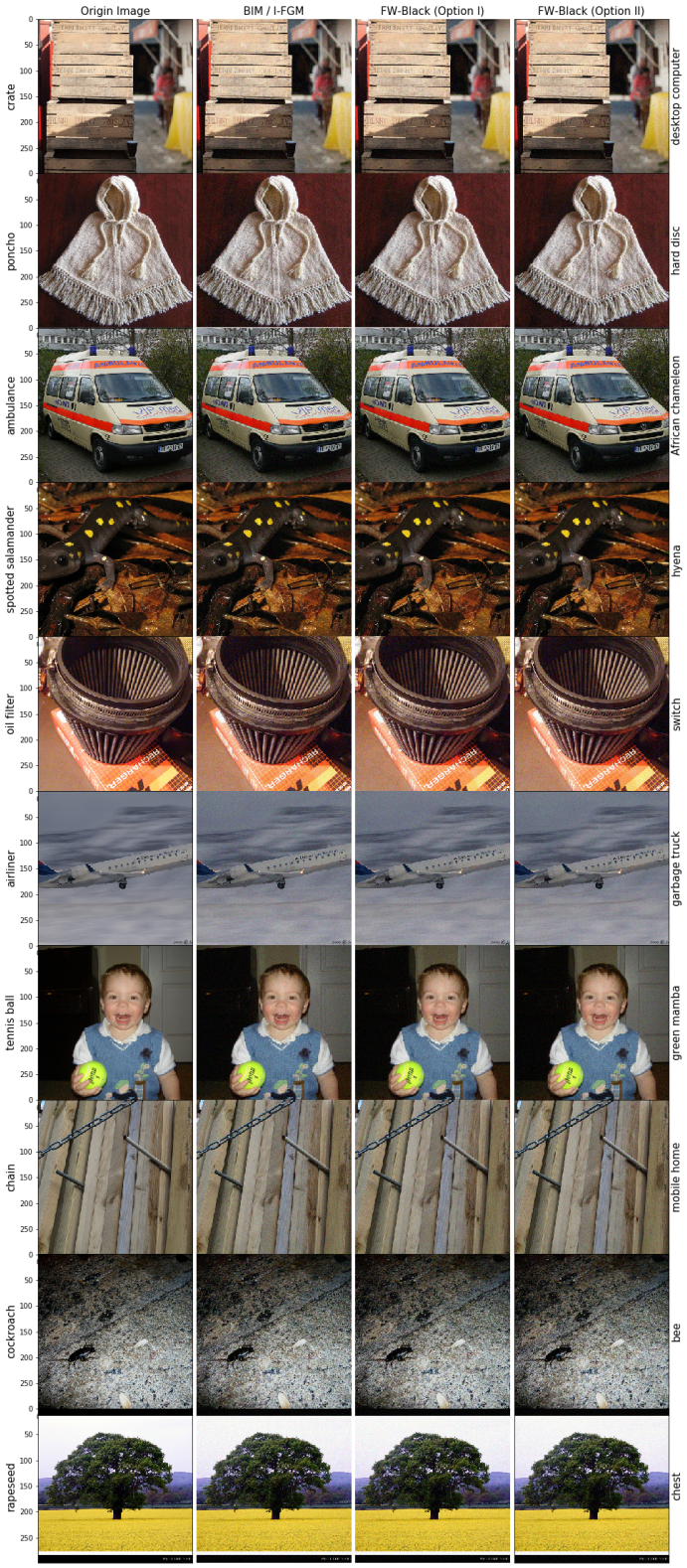

Figure 2: Sample adversarial examples generated through different L_∞ norm based black-box attacks. Left side labels denote the original class label and right side labels denote the target class label.