Zeroth-Order Optimization and Its Application to Adversarial Machine Learning

Sijia Liu and Pin-Yu Chen

***Abstract*—Many big data problems deal with complex data generating processes that cannot be described by analytical forms but can provide function evaluations, such as measurements from physical environments or predictions from deployed machine learning models. These types of problems fall into zeroth-order (gradient-free) optimization with respect to black-box models. In this paper, we provide a comprehensive introduction to recent advances in zeroth-order (ZO) optimization methods in both theory and applications. On the theory side, we will elaborate on ZO gradient estimation and the convergence rate of various ZO algorithms. The existing studies suggest that ZO algorithms typically agree with the iteration complexity of first-order algorithms up to a small-degree polynomial of the problem size. On the application side, we will delve into applications of ZO algorithms on studying the robustness of deep neural networks against adversarial perturbations. In particular, we will illustrate how to formulate the design of black-box adversarial attacks as a ZO optimization problem and how adversarial attacks can benefit from advanced ZO optimization techniques, such as providing query-efficient approaches to generating adversarial examples from a black-box image classifier.**

***Index Terms*—Zeroth-order optimization, adversarial machine learning, black-box adversarial example, gradient estimation.**

I. Introduction

ZEROTH-order (ZO) optimization is increasingly embraced for solving big data and machine learning problems when explicit expressions of the gradients are difficult or infeasible to obtain. It achieves gradient-free optimization by approximating the full gradient via efficient gradient estimators. One recent application of particular interest is to generate prediction-evasive adversarial examples using only the input-output correspondence of the target machine learning model, e.g., deep neural networks (DNNs) [1]–[4]. Additional classes of applications include network control and management with time-varying constraints and limited computation capacity [5], [6], parameter inference of black-box systems [7]–[9], and bandit optimization in which a player receives partial feedback in terms of loss function values revealed by her adversary [10], [11].

Spurred by application demands for ZO optimization, many types of ZO algorithms were developed for convex and non-convex optimization. In these algorithms, a full gradient is typically approximated using either a one-point or a two-point gradient estimator, where the former acquires a gradient estimate by querying the (black-box) objective function $f(\mathbf{x})$ at a single random location close to $\mathbf{x}$ [10], [12], and the latter computes a finite difference using two random function queries [13], [14]. Compared to the one-point gradient estimator, the *two-point* gradient estimator has a lower variance and thus improves the complexity bounds of ZO algorithms.

Despite the meteoric rise of two-point based ZO algorithms, most of the work is restricted to convex problems [6], [11], [15]–[18]. For example, a ZO mirror descent algorithm proposed by [15] has an exact rate $O(\sqrt{d}/\sqrt{T})$, where d is the number of optimization variables, and T is the number of iterations. The same rate is obtained by bandit convex optimization [11] and ZO online alternating direction method of multipliers [6].

In contrast to the convex setting, non-convex ZO optimization introduces a large amount of recent attention [8], [14], [19]–[21]. Different from convex optimization, the stationary condition is used to measure the convergence of nonconvex methods. In [14], the ZO gradient descent (ZO-GD) algorithm was proposed for deterministic nonconvex programming, which yields $O(d/T)$ convergence rate. A stochastic version of ZO-GD (namely, ZO-SGD) studied in [19] achieves the rate of $O(\sqrt{d}/\sqrt{T})$. In [20], a ZO distributed algorithm was developed for multi-agent optimization, leading to $O(1/T + d/q)$ convergence rate. Here q is the number of random directions used to construct a gradient estimate. In [8], an asynchronous ZO stochastic coordinate descent (ZO-SCD) was derived for parallel optimization and achieved the rate of $O(\sqrt{d}/\sqrt{T})$. In [9], [21], a stochastic variance reduced technique was used to achieve the improved convergence rate of $O(d/T)$.

Current studies suggested that ZO algorithms typically agree with the iteration complexity of first-order algorithms up to a small-degree polynomial of the problem size. In this paper, we will investigate how (two-point) random gradient estimate fits into ZO optimization. We will also survey the convergence rate of existing ZO optimization algorithms. Lastly, we will delve into applications of ZO algorithms to study the robustness of deep neural networks against adversarial perturbations.

II. Random Gradient Estimation via Zeroth-Order Oracle

We consider a finite-sum optimization problem of the form

$$\underset{\mathbf{x}\in\mathcal{C}}{\text{minimize}} \quad f(\mathbf{x}) := (1/n)\textstyle\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the optimization variable, $\mathcal{C} \in \mathbb{R}^d$ is a convex constraint set, and $\{f_i(\mathbf{x})\}$ are n component functions (not necessarily convex). In (1), if $\mathcal{C} = \mathbb{R}^d$, then we study an unconstrained finite-sum problem.

Compared to first-order optimization, ZO optimization requires to approximate the first-order gradient of $f(\mathbf{x})$ only

The authors are from the MIT-IBM Watson AI Lab, IBM Research, USA. Corresponding Author e-mail: ({sijia.liu, pin-yu.chen}@ibm.com).

through function values. Given a component function f_i, a two-point based average random gradient estimator $\hat{\nabla} f_i(\mathbf{x})$ is defined by [9], [11], [14], [16]

$$\hat{\nabla} f_i(\mathbf{x}) = \frac{d}{q}\sum_{j=1}^{q}\left[\frac{f_i(\mathbf{x}+\mu\mathbf{u}_j) - f_i(\mathbf{x})}{2\mu}\mathbf{u}_j\right], \tag{2}$$

where d is the number of optimization variables, $\mu > 0$ is a smoothing parameter, and $\{\mathbf{u}_j\}_{j=1}^q$ are i.i.d. random directions drawn from a uniform distribution over a unit sphere [11], [16]. Notably, the random direction vector $\mathbf{u}_j$ can also be drawn from the standard Gaussian distribution [14], [20], [22]. However, we argue that the uniform distribution could be more useful in practice since it is defined in a *bounded* space rather than the *whole* real space required for Gaussian. In (2), the larger q is, the smaller the variance of ZO gradient estimate is. Also, the gradient estimate (2) requires $(q+1)$ function queries. Clearly, the parameter q plays a trade-off between the variance of ZO gradient estimate and the function query complexity.

We highlight that unlike the first-order stochastic gradient estimate, the ZO gradient estimate (2) is a biased approximation to the true gradient of f_i. Instead, it becomes unbiased to the gradient of the randomized smoothing version of f_i [15], [16],

$$f_{i,\mu}(\mathbf{x}) = \mathbb{E}_{\mathbf{v}}[f_i(\mathbf{x}+\mu\mathbf{v})], \tag{3}$$

where $f_{i,\mu}$ is called the randomized smoothing version of f_i with smoothing parameter μ, and the random variable $\mathbf{v}$ follows a uniform distribution over the unit Euclidean ball. Although there exists a gap between a ZO gradient estimate and the true gradient of f_i, such a gap can be measured through its smoothing function.

In what follows, we derive the key statistical properties of the ZO gradient estimate (2).

Lemma 1: The ZO gradient estimate (2) yields:

1) For any $\mathbf{x} \in \mathbb{R}^d$

$$\mathbb{E}\left[\hat{\nabla} f_i(\mathbf{x})\right] = \nabla f_{i,\mu}(\mathbf{x}). \tag{4}$$

2) Suppose that f_i has L-Lipschitz continuous gradient, then for any $\mathbf{x} \in \mathbb{R}^d$

$$\begin{aligned}&\mathbb{E}\left[\|\hat{\nabla} f_i(\mathbf{x}) - \nabla f_{i,\mu}(\mathbf{x})\|_2^2\right]\\ \leq& 2\left(1+\frac{d}{q}\right)\|\nabla f_i(\mathbf{x})\|_2^2 + \left(1+\frac{1}{q}\right)\frac{\mu^2 L^2 d^2}{2}.\end{aligned} \tag{5}$$

Proof: see [9, Lemma 2]. □

Lemma 1 uncovers important properties of ZO gradient estimation. First, the use of *multiple* ($q > 1$) random direction vectors $\{\mathbf{u}_j\}$ does not reduce the bias of $\hat{\nabla} f_i$ (with respect to ∇f_i) since $\hat{\nabla} f$ is unbiased only with respect to ∇f_μ. Second, the variance of the random gradient estimator is reduced as q increases. In particular, a large q mitigates the dimension (d) dependency on the second-order moment of (2). This is crucial to improve the convergence performance of ZO optimization algorithms.

III. Convergence Analysis of Zeroth-Order Optimization Algorithms

In this section, we review the existing ZO algorithms that can be used to solve problem (1) and elaborate on their convergence rates. We divide the studied algorithms into two categories for unconstrained optimization and constrained optimization, respectively. Moreover, if problem (1) is convex, we use the optimality gap $f(\mathbf{x}) - f(\mathbf{x}^*)$ to measure the convergence rate, where $\mathbf{x}^*$ is the globally optimal solution. When problem (1) is nonconvex and unconstrained, we measure the stationarity in terms of $\|\nabla f(\mathbf{x})\|_2^2$. For constrained non-convex problems, a fitting alternative of $\|\nabla f(\mathbf{x})\|_2^2$, called gradient mapping, is then used for convergence evaluation [22]–[24].

A. ZO algorithms for unconstrained optimization

1) ZO gradient descent (ZO-GD) [14]: At the kth iteration, ZO-GD updates the solution as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \hat{\nabla} f(\mathbf{x}_k), \tag{6}$$

where $\eta_k > 0$ is learning rate, and $\hat{\nabla} f(\mathbf{x}_k) = \frac{1}{n}\sum_{i=1}^n \hat{\nabla} f_i(\mathbf{x}_k)$.

2) ZO stochastic gradient descent (ZO-SGD) [19] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)\right), \tag{7}$$

where $\mathcal{B}$ is a mini-batch of size $|\mathcal{B}|$, and $\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)$ is an estimate of stochastic gradient under mini-batch $\mathcal{B}$.

3) ZO stochastic coordinate descent (ZO-SCD) [8] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \tilde{\nabla} f_{i_k}(\mathbf{x}_k), \tag{8}$$

where i_k is a component function index randomly picked from $[n] := \{1, 2, \ldots, n\}$, and $\tilde{\nabla} f_{i_k}(\mathbf{x}_k)$ is an estimate of a block coordinate stochastic gradient given by $\frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}}\left(\frac{d}{2\mu}\left(f_{i_k}(\mathbf{x}_k+\mu\mathbf{e}_j) - f_{i_k}(\mathbf{x}_k-\mu\mathbf{e}_j)\right)\mathbf{e}_j\right)$. Here $\mathcal{S}$ is a mini-batch of coordinates randomly selected from $[d]$.

4) ZO sign-based stochastic gradient descent (ZO-signSGD) [25] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathrm{sign}\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)\right), \tag{9}$$

where $\mathrm{sign}(\cdot)$ takes element-wise signs of $\mathbf{x}$. It is shown in [25] that the convergence of ZO-signSGD can be measured via $\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2]$, a stricter criterion than $\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2]$.

For ease of comparison, we do not incorporate variance reduced versions of ZO algorithms, e,g., ZO-SVRG and ZO-SVRC, which are ZO stochastic variance reduced gradient/coordinate descent algorithms in [9], [21]. That is because those algorithms require extra query complexity in order to achieve better convergence rates.

B. Constrained optimization

1) ZO stochastic mirror descent (ZO-SMD) [15] :

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{C}}\left\{\langle \hat{\mathbf{g}}_k, \mathbf{x}\rangle + \frac{1}{\eta_k}\|\mathbf{x}-\mathbf{x}_k\|_2^2\right\}, \tag{10}$$

where for east of notation, let $\hat{\mathbf{g}}_k = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)$.

TABLE I: Summary of convergence rate and query complexity of various ZO algorithms.

Method	Problem setting	Gradient estimator	Smoothing parameter μ	Convergence rate	Query complexity (T iterations)
ZO-GD [14]	nonconvex, unconstrained	GauSGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{d}{T}\right)$	$O\left(\|\mathcal{B}\|qT\right)$
ZO-SGD [19]	nonconvex, unconstrained	GauSGE	$O\left(\frac{1}{d\sqrt{T}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O\left(\|\mathcal{B}\|qT\right)$
ZO-SCD [8]	nonconvex, unconstrained	CooGE	$O\left(\frac{1}{\sqrt{T}} + \min\{\frac{1}{(dT)^{-1/4}}, \frac{1}{\sqrt{d}}\}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O\left(\|\mathcal{B}\|\|\mathcal{S}\|T\right)$
ZO-signSGD [25]	nonconvex, unconstrained	GauSGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2] = O\left(\frac{\sqrt{d}}{\sqrt{T}} + \frac{\sqrt{d}}{\sqrt{\|\mathbf{B}\|}} + \frac{d}{\sqrt{q\|\mathcal{B}\|}}\right)$	$O\left(\|\mathcal{B}\|qT\right)$
ZO-SVRG [9]	nonconvex, unconstrained	UniSGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{d}{T} + \frac{1}{\sqrt{\|\mathbf{B}\|}}\right)$	$O\left(qnS + q\|\mathcal{B}\|Sm\right)$, $T = Sm$**
ZO-SMD [15]	convex, constrained	GauSGE/UniSGE	$O\left(\frac{1}{dt}\right)$	$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-PSGD [26]	nonconvex, constrained	UniSGE	$O\left(\frac{1}{\sqrt{dq\|\mathcal{B}\|}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{1}{\sqrt{T}} + \frac{d+q}{bq}\right)$	$O\left(\|\mathcal{B}\|qT\right)$
ZO-FW [27]	nonconvex, constrained	GauSGE/UniSGE	$O\left(\frac{1}{d^{1.5}t^{1/3}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{(d/q)^{1/3}}{T^{1/4}}\right)$	$O\left(qT\right)$
ZO-ProxSGD [22]	nonconvex, composite***	GauSGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2] = O\left(\frac{d}{\|\mathcal{B}\|qT} + \frac{d^2}{\|\mathcal{B}\|qT} + \frac{d}{\|\mathcal{B}\|q}\right)$	$O\left(\|\mathcal{B}\|qT\right)$
ZO-OADMM [6]	convex, composite	GauSGE/UniSGE	$O\left(\frac{1}{d^{1.5}t}\right)$	$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] = O\left(\frac{\sqrt{d}}{\sqrt{T\|\mathcal{B}\|q}}\right)$	$O\left(\|\mathcal{B}\|qT\right)$

* GauSGE and UniSGE represents the ZO gradient estimator using random direction vectors generated from the standard normal distribution and the uniform distribution over a unit sphere, respectively.
** ZO-SVRG contains two iteration loops, where the number of outer iterations is S and the number of inner iterations is m.
*** Composite optimization can handle smooth + nonsmooth objective functions.

2) ZO projected stochastic gradient descent (ZO-PSGD) [26] :

$$\mathbf{x}_{k+1} = \Pi_{\mathcal{C}}\left[\mathbf{x}_k - \eta_k \hat{\mathbf{g}}_k\right], \tag{11}$$

where $\Pi_{\mathcal{C}}$ denotes the projection operator with respect to $\mathcal{C}$, i.e., $\Pi_{\mathcal{C}}(\mathbf{a}) = \arg\min_{\mathbf{z}\in\mathcal{C}} \|\mathbf{z} - \mathbf{a}\|^2$. We remark that ZO-PSGD can be regarded as a special case of ZO proximal stochastic gradient descent (ZO-ProxSGD), which is proposed to solve constrained composite optimization problems [22]. However, the complexity of ZO-ProxSGD is dominated by the computation of the proximal operation with respect to all nonsmooth regularization functions. To overcome this issue, reference [6] developed a ZO online alternating direction method of multipliers (ZO-OADMM) algorithm, which can split the original complex optimization problem into a sequence of easily-solved subproblems in a flexible manner.

3) ZO Frank-Wolfe (ZO-FW) [27] : The ZO Frank-Wolfe algorithm calls the following linear minimization oracle (LMO) at each iteration

$$\begin{aligned} \mathbf{v}_k &= \arg\min_{\mathbf{x}\in\mathcal{C}} \langle \hat{\mathbf{g}}_k, \mathbf{x}\rangle \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \eta_k(\mathbf{v}_k - \mathbf{x}_k) \end{aligned} \tag{12}$$

where the ZO gradient estimate $\hat{\mathbf{g}}_k$ has been defined in (10). We note that the LMO is equivalent to the minimization of the first-order Taylor expansion of f at point $\mathbf{x}_k$ using the ZO gradient estimate $\hat{\mathbf{g}}_k$.

As a concluding remark, we summarize the settings, ZO gradient estimators, and convergence rates of various ZO algorithms in Table I.

IV. Black-Box Adversarial Attacks: An ZO Optimization Perspective

In this section, we will illustrate how to formulate black-box adversarial attacks as a ZO optimization problem and how adversarial attacks can benefit from advanced ZO optimization techniques, such as providing query-efficient approaches to generating adversarial examples from a black-box image classifier.

Generally speaking, given a natural input $\mathbf{x}_0$ to a machine learning model, its adversarial example $\mathbf{x}$ refers to a modified input which is (semantically) close to $\mathbf{x}_0$ but the model outputs of $\mathbf{x}$ and $\mathbf{x}_0$ are drastically different, e.g., classifying $\mathbf{x}_0$ as a label t_0 but classifying $\mathbf{x}$ as another label $t \neq t_0$. The adversarial modification can be accomplished by considering the additive perturbation model $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}$, and the level of distortion is often measured by the ℓ_p norm ($p \geq 1$) of the perturbation $\boldsymbol{\delta}$, particularly the ℓ_1, ℓ_2 and ℓ_∞ norms [28]–[30]. When the distortion is small, the adversarial perturbation is visually imperceptible but can cause the target machine learning model to misbehave, resulting in increasing concerns in safety-critical and cost-sensitive applications, as well as new challenges in training robust machine learning models.

Typically, the adversarial perturbations are crafted in the "white-box" setting, where the adversary has full access to the target model such as model parameters and neural network structures. Take neural network classifiers as an example, adversarial perturbations for misclassification can be found by performing back-propagation through network layers from the model output to the model input. With some designed attack loss function (e.g., cross entropy), back-propagation provides the direction of making the perturbed input adversarial and can be applied successively to perturbed inputs.

While in the white-box setting crafting adversarial examples are shown to be plausible in many machine learning tasks, spanning from image classification [31], speech recognition [32], machine translation [33], image captioning [34], text sentiment analysis [35] to sparse regression [36], the need for requiring back-propagation of the target model renders white-box adversarial attacks less practical when attacking a deployed machine learning service, such as Google Cloud Vision API[1] and Clarifai.com[2]. In this case, one only has access to the model output (e.g., class prediction scores) of a queried input but is completely agnostic about the target model, which is known as the black-box attack setting. The target model can be a neural network, a support vector machine, a decision tree,

[1] https://cloud.google.com/vision
[2] https://clarifai.com

or any other classifier. One question that naturally arises is: *How can we generate adversarial examples in the black-box setting?* Notably, this problem setup of black-box adversarial attacks fits into the framework of ZO optimization. One can formulate the process of finding an adversarial example of a black-box model as a ZO optimization problem, where the objective function is associated with the model output and the gradient is infeasible to obtain (e.g., back-propagation is inadmissible in the black-box setting).

Without lose of generality, we denote a target machine learning model as a classification function $F : [0, 1]^d \mapsto \mathbb{R}^K$ that takes a d-dimensional scaled data sample as its input and yields a vector of prediction scores of all K image classes, such as the prediction probabilities for each class. We further consider the case of applying an entry-wise monotonic transformation $M(F)$ to the output of F for black-box attacks, since monotonic transformation preserves the ranking of the class predictions and can alleviate the problem of large score variation in F (e.g., probability to log probability). As an illustration, we use the black-box targeted attack loss function proposed in [3], which aims to minimize the following objective function

$$\text{minimize}_{\boldsymbol{\delta}:\mathbf{x}_0+\boldsymbol{\delta}\in[0,1]^d} \ \|\boldsymbol{\delta}\|_2^2 + \lambda \cdot \text{Loss}(\mathbf{x}, M(F(\mathbf{x})), t), \tag{13}$$

where $\|\boldsymbol{\delta}\|_2^2$ measures the distortion between $\mathbf{x}$ and $\mathbf{x}_0$ in the squared ℓ_2 norm and $\text{Loss}(\cdot)$ is an attack objective reflecting the likelihood of predicting $t = \arg\max_{k\in\{1,\ldots,K\}}[M(F(\mathbf{x}))]_k$, λ is a regularization coefficient, and the constraint $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta} \in [0, 1]^d$ confines the adversarial example $\mathbf{x}$ to the valid sample space. Specifically, the Loss term is defined as

$$\text{Loss} = \max\{\max_{j\neq t} \log[F(\mathbf{x})]_j - \log[F(\mathbf{x})]_t\}, -\kappa\}, \tag{14}$$

where the monotonic transformation $M(\cdot) = \log(\cdot)$ is applied to the model output $F(\cdot)$, the constant parameter $\kappa \geq 0$ controls the gap between the confidence of target class label $\log[F(\mathbf{x})]_t$ and the second highest class label $\max_{j\neq t} \log[F(\mathbf{x})]_j$, and the hinge-like term $\max\{\cdot, -\kappa\}$ ensures this term is a constant $-\kappa$ once $\log[F(\mathbf{x})]_t - \max_{j\neq t} \log[F(\mathbf{x})]_j \geq \kappa$. For untargeted attacks that aim at classifying $\mathbf{x}$ as any label other than the original top-1 label t_0 of $\mathbf{x}_0$, the loss term can be defined as

$$\text{Loss} = \max\{\log[F(\mathbf{x})]_{t_0} - \max_{j\neq t} \log[F(\mathbf{x})]_j, -\kappa\}. \tag{15}$$

The constraint $\mathbf{x}_0 + \boldsymbol{\delta} \in [0, 1]^d$ in (13) can be eliminated via change-of-variable (e.g., using tanh transformation) such that the black-box attack formulation becomes an unconstrained zeroth-order optimization problem.

Here we discuss two ZO optimization based black-box adversarial attacks on the Inception-v3 model [37] trained on ImageNet: (i) the ZOO attack [3] and (ii) the AutoZOOM attack [4]. The ZOO attack adopts random block coordinate descent for solving (13), whereas AutoZOOM adopts the two-point based average random gradient estimator as in (2) and uses dimension reduction on the perturbation $\boldsymbol{\delta}$ (either an off-line trained autoencoder or a bilinear resizer) to improve the efficiency in model query. The parameter settings and the displayed images are adopted from these two papers.

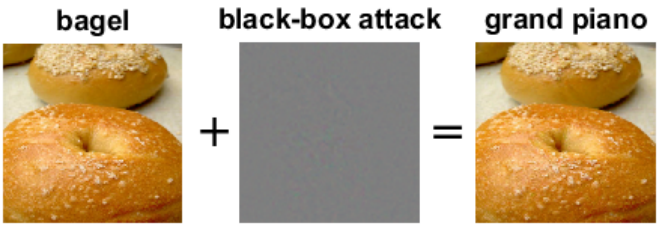

Fig. 1: Targeted black-box adversarial example (original class: bagel; targeted class: grand piano) using the ZOO attack [3] on the black-box Inception-v3 model. Left: original image ($\mathbf{x}_0$). Middle: adversarial perturbation ($\boldsymbol{\delta}$). Right: adversarial example ($\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}$).

Fig. 1 shows an adversarial bagel image with a target label "grand piano" using the ZOO attack. It can be observed that the adversarial perturbation is indeed visually imperceptible but will cause the resulting adversarial example to be misclassified as grand piano. Notably, it has been shown in [3] that even without using back-propagation, the distortion level of black-box adversarial attacks can be comparable to that of white-box adversarial attacks, suggesting the effectiveness of ZO optimization. Intuitively, in the context of black-box adversarial attacks, the success of ZO optimization with gradient estimates can be anticipated as it is performing a "psuedo back-propagation" of the target model. Furthermore, its reliable attack performance is assured by the convergence analysis. Fig. 2 compares the performance of the ZOO attack and the AutoZOOM attack on the same image and target label. With the use of two point based average random gradient estimator in AutoZOOM instead of the coordinate-wise gradient estimator in ZOO, the AutoZOOM attack significantly reduces the number of queries (about 83%) required to generate a visually similar adversarial bagel image from the black-box Inception-v3 model. The remarkable improvement in query efficiency is consistent with the query complexity analysis between ZO-SCD and ZO-SGD as discussed in the previous sections and Table I. It is also worth noting that even in the stringent attacking scenario where the target black-box classifier only outputs the top-1 prediction label of a queried input, ZO optimization with some additional objective function smoothing techniques can still be used to craft adversarial examples [38], [39].

V. Conclusion

This paper provides a systematic and comprehensive overview of zeroth-order (ZO) optimization, which only requires function evaluations to solve for a finite-sum minimization problem with optionally convex set constraints. We discuss several gradient estimation based ZO optimization methods and compare their performance in terms of convergence rate and query complexity. As a motivating example, we highlight how ZO optimization can be used to craft adversarial examples of a black-box machine learning model in an efficient and principled manner.

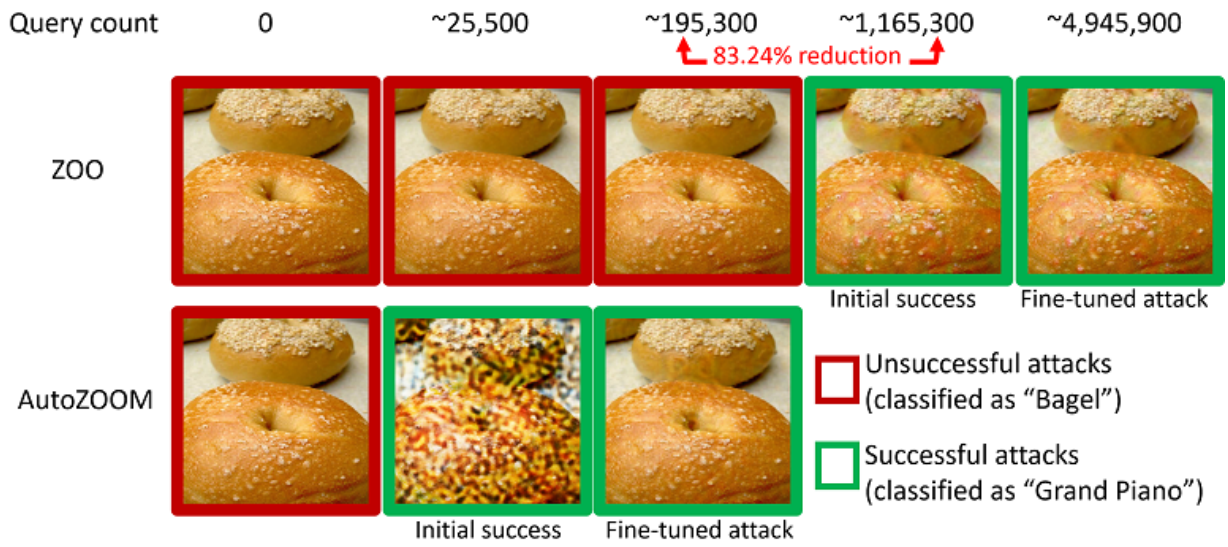

Fig. 2: Comparison of ZOO and AutoZOOM black-box adversarial attacks. With the use of the two point based average random gradient estimator in AutoZOOM instead of the coordinate-wise gradient estimator in ZOO, AutoZOOM significantly reduces the number of queries required to generate a visually similar adversarial bagel image from the black-box Inception-v3 model.

ACKNOWLEDGMENT

The authors thank the MIT-IBM Watson AI Lab which fully for fully supporting this research. The authors also would like to thank Mingyi Hong for the helpful discussion on zeroth-order optimization.

REFERENCES

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

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[3] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. 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*. ACM, 2017, pp. 15–26.

[4] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," *arXiv preprint arXiv:1805.11770*, 2018.

[5] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet of Things Journal*, 2018.

[6] S. Liu, J. Chen, P.-Y. Chen, and A. O. Hero, "Zeroth-order online ADMM: Convergence analysis and applications," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84, April 2018, pp. 288–297.

[7] M. C. Fu, "Optimization for simulation: Theory vs. practice," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.

[8] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Advances in Neural Information Processing Systems*, 2016, pp. 3054–3062.

[9] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," *Advances in Neural InformationProcessing Systems*, 2018.

[10] O. Shamir, "On the complexity of bandit and derivative-free stochastic convex optimization," in *Conference on Learning Theory*, 2013, pp. 3–24.

[11] ——, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *Journal of Machine Learning Research*, vol. 18, no. 52, pp. 1–11, 2017.

[12] A. D. Flaxman, A. T. Kalai, and H. B. 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*, 2005, pp. 385–394.

[13] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *COLT*, 2010, pp. 28–40.

[14] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 2, no. 17, pp. 527–566, 2015.

[15] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[16] X. Gao, B. Jiang, and S. Zhang, "On the information-adaptive variants of the ADMM: an iteration complexity perspective," *Optimization Online*, vol. 12, 2014.

[17] P. Dvurechensky, A. Gasnikov, and E. Gorbunov, "An accelerated method for derivative-free smooth stochastic convex optimization," *arXiv preprint arXiv:1802.09022*, 2018.

[18] Y. Wang, S. Du, S. Balakrishnan, and A. Singh, "Stochastic zeroth-order optimization in high dimensions," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84. PMLR, April 2018, pp. 1356–1365.

[19] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[20] D. Hajinezhad, M. Hong, and A. Garcia, "Zeroth order nonconvex multi-agent optimization over networks," *arXiv preprint arXiv:1710.09997*, 2017.

[21] B. Gu, Z. Huo, and H. Huang, "Zeroth-order asynchronous doubly stochastic algorithm with variance reduction," *arXiv preprint arXiv:1612.01425*, 2016.

[22] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Mathematical Programming*, vol. 155, no. 1-2, pp. 267–305, 2016.

[23] S. J. Reddi, S. Sra, B. Poczos, and A. J. Smola, "Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 1145–1153.

[24] E. Hazan, K. Singh, and C. Zhang, "Efficient regret minimization in non-convex games," *arXiv preprint arXiv:1708.00075*, 2017.

[25] Anonymous, "signsgd via zeroth-order oracle," in *Submitted to International Conference on Learning Representations*, 2019, under review. [Online]. Available: https://openreview.net/forum?id=BJe-DsC5Fm

[26] S. Liu, X. Li, P.-Y. Chen, B. Vinzamuri, J. Haupt, and L. Amini, "Zeroth-

order stochastic projected gradient descent for nonconvex optimization," in *GlobalSIP*. IEEE, 2018.
[27] A. K. Sahu, M. Zaheer, and S. Kar, "Towards gradient free and projection free stochastic optimization," *arXiv preprint arXiv:1810.03233*, 2018.
[28] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: elastic-net attacks to deep neural networks via adversarial examples," *AAAI*, 2018.
[29] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.
[30] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *ICLR*, 2017.
[31] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *ECCV*, 2018, pp. 631–648.
[32] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," *arXiv preprint arXiv:1801.01944*, 2018.
[33] M. Cheng, J. Yi, H. Zhang, P.-Y. Chen, and C.-J. Hsieh, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," *arXiv preprint arXiv:1803.01128*, 2018.
[34] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, "Attacking visual language grounding with adversarial examples: A case study on neural image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 2587–2597.
[35] Q. Lei, L. Wu, P.-Y. Chen, A. G. Dimakis, I. S. Dhillon, and M. Witbrock, "Discrete attacks and submodular optimization with applications to text classification," *arXiv preprint arXiv:1812.00151*, 2018.
[36] P.-Y. Chen, B. Vinzamuri, and S. Liu, "Is ordered weighted ℓ_1 regularized regression robust to adversarial perturbation? a case study on oscar," *arXiv preprint arXiv:1809.08706*, 2018.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
[38] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *ICML*, 2018.
[39] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv preprint arXiv:1807.04457*, 2018.