Can Stochastic Zeroth-Order Frank-Wolfe Method Converge Faster for Non-Convex Problems?

Hongchang Gao [1 2] **Heng Huang** [1 3]

Abstract

Frank-Wolfe algorithm is an efficient method for optimizing non-convex constrained problems. However, most of existing methods focus on the first-order case. In real-world applications, the gradient is not always available. To address the problem of lacking gradient in many applications, we propose two new stochastic zeroth-order Frank-Wolfe algorithms and theoretically proved that they have a faster convergence rate than existing methods for non-convex problems. Specifically, the function queries oracle of the proposed faster zeroth-order Frank-Wolfe (FZFW) method is $O(\frac{n^{1/2}d}{\epsilon^2})$ which can match the iteration complexity of the first-order counterpart approximately. As for the proposed faster zeroth-order conditional gradient sliding (FZCGS) method, its function queries oracle is improved to $O(\frac{n^{1/2}d}{\epsilon})$, indicating that its iteration complexity is even better than that of its first-order counterpart NCGS-VR. In other words, the iteration complelxity of the accelerated first-order Frank-Wolfe method NCGS-VR is suboptimal. Then, we proposed a new algorithm to improve its IFO (incremental first-order oracle) to $O(\frac{n^{1/2}}{\epsilon})$. At last, the empirical studies on benchmark datasets validate our theoretical results.

1. Introduction

In this paper, we consider the following *constrained* finite-sum minimization problem:

$$\min_{\mathbf{x}\in\Omega} F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \ , \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ denotes a closed convex feasible set, each component function f_i is smooth and non-convex, and n represents the number of component functions. Many problems in machine learning can be represented by Eq. (1). A representative example is the robust low-rank matrix completion problem, which is defined as follows:

$$\min_{\|X\|_*\leq R} \sum_{(i,j)\in\mathcal{O}} \Big(1 - \exp\Big\{-\frac{(x_{i,j}-y_{i,j})^2}{\sigma^2}\Big\}\Big) \ , \tag{2}$$

where $\mathcal{O}$ denotes the observed elements, σ is a hyper-parameter, and $\|X\|_* \leq R$ stands for the low-rank constraint. Here, the component function is a non-convex function which is less sensitive to the residual than the least square loss.

Compared with the non-constraint finite-sum minimization problem, optimizing Eq. (1) has to deal with the constraint, which introduces new challenges. A straightforward method to optimize the large-scale Eq. (1) is the projected gradient descent method which first takes a step along the gradient direction and then performs the projection to satisfy the constraint. However, the computational overhead of the projection step is usually large. For instance, the low-rank constraint in Eq. (2) requires the time-consuming singular value decomposition. Unlike the projected gradient descent method, Frank-Wolfe method (Frank & Wolfe, 1956) is more efficient when dealing with the constraint. Specifically, rather than performing projection, it solves an efficient linear subproblem to make the solution lie in the feasible set Ω. Thus, Frank-Wolfe method has been popularly used in optimizing Eq. (1).

The classical Frank-Wolfe method (a.k.a. Conditional Gradient method) is first proposed for convex constrained problems. It attracts increasing attention in machine learning community (Clarkson, 2010; Jaggi; Lacoste-Julien & Jaggi, 2015; Mokhtari et al., 2018; Hassani et al., 2019; Zhang et al., 2019) since it is efficient to optimize some difficult machine learning problems, such as the low-rank constraint problem. As for the convex problem, (Jaggi; Lacoste-Julien & Jaggi, 2015; Hazan & Luo, 2016; Lan & Zhou, 2016) analyzed and improved the convergence rate under different settings. As for the non-convex problem, (Lacoste-Julien, 2016) demonstrated the convergence rate of non-convex FW

[1] Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, USA [2] Department of Computer and Information Sciences, Temple University, Philadelphia, USA [3] JD Finance America Corporation. Correspondence to: Heng Huang <heng.huang@pitt.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

under the batched setting. Later, (Reddi et al., 2016) proposed the stochastic Frank-Wolfe (SFW) method and the variance reduced variant (SVFW). Specifically, to achieve ϵ-solution, SFW requires $O(\frac{1}{\epsilon^4})$ incremental first-order oracle (IFO) and SVFW improves it to $O(n + \frac{n^{2/3}}{\epsilon^2})$. Recently, (Shen et al., 2019; Yurtsever et al., 2019) proposed a new variant SPFW which employs a biased estimator (Fang et al., 2018; Nguyen et al., 2017; Wang et al., 2018) for the gradient rather than an unbiased estimation like SFW and SVFW so that the IFO is improved to $O(\frac{n^{1/2}}{\epsilon^2})$. Compared with SVFW, the improvement is significant when n is huge for the large-scale data.

However, all aforementioned results are based on the availability of the gradient ∇f_i. In many applications, only the function value is available so that we cannot apply aforementioned methods. Fortunately, the zeroth-order optimization algorithm can address this challenge. Its basic idea is to approximate the gradient by the difference of function values $f(\mathbf{x})$ under small disturbance on $\mathbf{x}$. In (Duchi et al., 2015; Shamir, 2017; Gao et al., 2018; Dvurechensky et al., 2018; Wang et al., 2017), the convergence rate of zeroth-order optimization for convex problems has been explored. In addition, different works (Balasubramanian & Ghadimi, 2018; Lian et al., 2016; Ghadimi & Lan, 2013; Hajinezhad et al., 2017; Liu et al., 2018; Nesterov & Spokoiny, 2017; Ji et al., 2019) have been proposed to analyze the convergence rate for the non-convex problem. For instance, a string of works (Balasubramanian & Ghadimi, 2018; Sahu et al., 2019; Chen et al., 2020) have been proposed to study the non-convex zeroth-order Frank-Wolfe method recently. In particular, (Chen et al., 2020) proposed a zeroth-order Frank-Wolfe method for adversarial attack based on full gradient. (Balasubramanian & Ghadimi, 2018) shows that the zeroth-order stochastic conditional gradient (ZSCG) method requires $O(\frac{d}{\epsilon^4})$ function queries and (Sahu et al., 2019) proposes stochastic gradient free Frank-Wolfe (SGFFW) method whose function queries oracle is $O(\frac{d^{4/3}}{\epsilon^4})$. However, it is commonly agreed that zeroth-order methods can share the same iteration complexity with the first-order counterparts besides some constant with respect to the input dimension (Liu et al., 2018). As discussed earlier, the iteration complexity of the first-order stochastic Frank-Wolfe method is improved to $O(\frac{n^{1/2}}{\epsilon^2})$, which is much better than $O(\frac{d}{\epsilon^4})$. Thus, a natural question follows: Is it possible to improve the stochastic zeroth-order Frank-Wolfe method to this level? In this paper, we obtain a positive answer. Specifically, to improve the convergence rate, we resort to a biased variance reduction technique (Fang et al., 2018; Nguyen et al., 2017; Wang et al., 2018) as SPFW to reduce the variance when estimating the gradient. However, although the biased estimator for gradient has been applied to some unconstrained optimization methods (Fang et al., 2018), yet there are new challenges in analyzing the convergence of the stochastic zeroth-order Frank-Wolfe method. On one hand, most existing works only apply it to *unconstrained* problems while Frank-Wolfe optimizes the *constrained* problem. As a result, the convergence criterion of Frank-Wolfe method is totally different from that of unconstrained problems, which intrigues the difficulty in the convergence analysis. On the other hand, most existing works only apply this biased variance reduction technique to first-order methods rather than zeroth-order methods. The intrinsic properties of the zeroth-order gradient also introduce difficulty in the convergence analysis. Thus, it is challenging to improve the convergence rate of the stochastic zeroth-order Frank-Wolfe method. In this paper, we have addressed these challenges and improved the number of function queries to $O(\frac{n^{1/2}d}{\epsilon^2})$.

On the other hand, the acceleration technique has shown superior performance in different kinds of first-order optimization methods. Inspired by that, (Qu et al., 2017) proposed the accelerated non-convex FW. In particular, (Qu et al., 2017) shows that the accelerated stochastic conditional gradient sliding method (NCGS-VR) enjoys the IFO of $O(\frac{n^{2/3}}{\epsilon})$. Following the previous argument about the consistency between the first-order and zeroth-order methods, another natural question follows: Is it possible to further improve the stochastic zeroth-order Frank-Wolfe method to have similar iteration complexity with the accelerated first-order counterpart? In this paper, we also give a positive answer. Specifically, with the same biased variance reduction technique, we proposed a faster zeroth-order stochastic conditional gradient sliding method which improves the number of function queries to $O(\frac{n^{1/2}d}{\epsilon})$. In fact, the iteration complexity of this stochastic zeroth-order method is even better than its first-order counterpart in terms of the order regarding n. In other words, the convergence rate of NCGS-VR is suboptimal. Thus, in this paper, we also proposed a faster first-order conditional gradient descent method to further improve NCGS-VR. In particular, our proposed method enjoys the IFO of $O(\frac{n^{1/2}}{\epsilon})$, which is better than $O(\frac{n^{2/3}}{\epsilon})$. At last, we compare the iteration complexity of different methods is Tables 1, 2 and summarize the contributions of this paper as follows:

- We proposed a faster stochastic zeroth-order Frank-Wolfe method with FQO as $O(\frac{n^{1/2}d}{\epsilon^2})$.
- We proposed a faster stochastic zeroth-order conditional gradient sliding method and improved the function queries oracle to $O(\frac{n^{1/2}d}{\epsilon})$.
- We proposed a new stochastic first-order conditional gradient sliding method with IFO as $O(\frac{n^{1/2}}{\epsilon})$.

2. Preliminaries

In this section, we will list assumptions and some important definitions for the convergence analysis.

Table 1. FQO of different zeroth-order algorithms.

Zeroth-Order	FQO
ZSCG(Balasubramanian & Ghadimi, 2018)	$O(\frac{d}{\epsilon^4})$
SGFFW (Sahu et al., 2019)	$O(\frac{d^{4/3}}{\epsilon^4})$
FZFW (this work)	$O(\frac{n^{1/2}d}{\epsilon^2})$
FZCGS(this work)	$O(\frac{n^{1/2}d}{\epsilon})$

Table 2. IFO of different first-order algorithms.

First-Order	IFO
SFW (Reddi et al., 2016)	$O(\frac{1}{\epsilon^4})$
SVFW (Reddi et al., 2016)	$O(n + \frac{n^{2/3}}{\epsilon^2})$
SPFW (Shen et al., 2019)	$O(\frac{n^{1/2}}{\epsilon^2})$
NCGS-VR (Qu et al., 2017)	$O(\frac{n^{2/3}}{\epsilon})$
FCGS (this work)	$O(\frac{n^{1/2}}{\epsilon})$

Assumption 1. *The component function f_i $(i \in [n])$ is L-smooth as follows:*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \Omega \, . \tag{3}$$

Assumption 2. *The diameter of the feasible set Ω is D.*

Assumption 3. *Assume that the variance of the stochastic gradient $\nabla f_i(\mathbf{x})$ is bounded as follows:*

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \sigma^2 \, , \tag{4}$$

where $\sigma > 0$.

Convergence Criterion. Following (Lacoste-Julien, 2016; Reddi et al., 2016), the convergence criterion used for the standard Frank-Wolfe method is the *Frank-Wolfe gap* which is defined as follows:

$$\mathcal{G}(\mathbf{x}) = \max_{\mathbf{u} \in \Omega} \langle \mathbf{u} - \mathbf{x}, -\nabla F(\mathbf{x}) \rangle \, . \tag{5}$$

For the conditional gradient sliding method which incorporates the Nesterov's acceleration technique, following (Qu et al., 2017; Lan & Zhou, 2016), we employ the following gradient mapping as the convergence criterion:

$$\mathcal{G}(\mathbf{x}, \nabla F(\mathbf{x}), \gamma) = \frac{1}{\gamma}(\mathbf{x} - \psi(\mathbf{x}, \nabla F(\mathbf{x}), \gamma)) \, , \tag{6}$$

where $\psi(\mathbf{x}, \nabla F(\mathbf{x}), \gamma)$ denotes a prox-mapping function which is defined as follows:

$$\psi(\mathbf{x}, \nabla F(\mathbf{x}), \gamma) = \arg\min_{\mathbf{y} \in \Omega} \langle \nabla F(\mathbf{x}), \mathbf{y} \rangle + \frac{1}{2\gamma}\|\mathbf{y} - \mathbf{x}\|^2 \, , \tag{7}$$

where $\gamma > 0$ is a hyper-parameter.

Oracle Model. Here, we use the following oracle models to compare the iteration complexity of different algorithms.

- Function Query Oracle (FQO): FQO samples a component function and returns its function value $f_i(\mathbf{x})$.
- Incremental First-Order Oracle (IFO): IFO samples a component function and returns its gradient $\nabla f_i(\mathbf{x})$.
- Linear Oracle (LO): LO solves a linear programming problem and returns $\arg\max_{\mathbf{u} \in \Omega} \langle \mathbf{u}, \mathbf{v} \rangle$.

3. Faster Zeroth-Order Method for Constrained Non-Convex Problems

In this section, we will present the faster zeroth-order Frank-Wolfe (FZFW) method and the faster zeroth-order conditional gradient sliding (FZCGS) method.

3.1. Zeroth-Order Gradient Estimator

When the gradient of a function is not available, we can utilize the difference of the function value with respect to two random points to estimate it. Specifically, the widely used methods are the two-point Gaussian random gradient estimator (Nesterov & Spokoiny, 2017) and the coordinate-wise gradient estimator (Lian et al., 2016). In this paper, we only consider the coordinate-wise gradient estimator since existing literature (Liu et al., 2018) shows that it has better convergence performance than the two-point Gaussian random gradient estimator. Specifically, the coordinate-wise gradient estimator is defined as follows:

$$\hat{\nabla} f_i(\mathbf{x}) = \sum_{j=1}^{d} \frac{f_i(\mathbf{x} + \mu_j \mathbf{e}_j) - f_i(\mathbf{x} - \mu_j \mathbf{e}_j)}{2\mu_j} \mathbf{e}_j \, , \tag{8}$$

where $\mu_j > 0$ is the smoothing parameter, and $\mathbf{e}_j \in \mathbb{R}^d$ denotes the basis vector where only the j-th element is 1 and all the others are 0. This estimator for the gradient can be used for the Frank-Wolfe method when the standard gradient is not available.

3.2. Faster Zeroth-Order Frank-Wolfe Method

In (Balasubramanian & Ghadimi, 2018), a zeroth-order stochastic conditional gradient (ZSCG) method is proposed and the number of function queries $O(\frac{d}{\epsilon^4})$ under the stochastic setting is provided. However, the zeroth-order method is supposed to share the same order of iteration complexity with its first-order counterpart if ignoring the term about the dimension of the model parameter (Liu et al., 2018). Currently, the first-order stochastic Frank-Wolfe method for non-convex contrained problems has been improved to

Algorithm 1 Faster Zeroth-Order Frank-Wolfe Method (FZFW)

```
Input: $\mathbf{x}_0$, $q > 0$, $\mu > 0$, $K > 0$, $n$
for $k = 0, \cdots, K-1$ do
    if mod(k, q) = 0 then
        Sample $S_1$ without replacement to compute $\hat{\mathbf{v}}_k = \hat{\nabla} f_{S_1}(\mathbf{x}_k)$
    else
        Sample $S_2$ with replacement to compute $\hat{\mathbf{v}}_k = \frac{1}{|S_2|}\sum_{i\in S_2}[\hat{\nabla} f_i(\mathbf{x}_k) - \hat{\nabla} f_i(\mathbf{x}_{k-1}) + \hat{\mathbf{v}}_{k-1}]$
    end if
    $\mathbf{u}_k = \arg\max_{\mathbf{u}\in\Omega}\langle \mathbf{u}, -\hat{\mathbf{v}}_k\rangle$
    $\mathbf{d}_k = \mathbf{u}_k - \mathbf{x}_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{d}_k$
end for
Output: Randomly choose $\mathbf{x}_\alpha$ from $\{\mathbf{x}_k\}$ and return it
```

$O(\frac{n^{1/2}}{\epsilon^2})$ (Shen et al., 2019). Thus, the iteration complexity of ZSCG is suboptimal. In this subsection, we will propose a new algorithm to improve it. The pseudo code of our proposed faster zeroth-order Frank-Wolfe (FZFW) method is summarized in Algorithm 1. In detail, to obtain the gradient required in the linear oracle, we estimate the gradient at every q iterations as follows:

$$\hat{\nabla} f_{S_1}(\mathbf{x}_k) = \sum_{j=1}^{d} \frac{f_{S_1}(\mathbf{x}_k + \mu_j \mathbf{e}_j) - f_{S_1}(\mathbf{x}_k - \mu_j \mathbf{e}_j)}{2\mu_j}\mathbf{e}_j \, , \tag{9}$$

and estimate the gradient at other iterations as follows:

$$\hat{\mathbf{v}}_k = \frac{1}{|S_2|}\sum_{i\in S_2}[\hat{\nabla} f_i(\mathbf{x}_k) - \hat{\nabla} f_i(\mathbf{x}_{k-1}) + \hat{\mathbf{v}}_{k-1}] \, , \tag{10}$$

where S_1 and S_2 denote the randomly selected samples. Compared with ZSCG (Balasubramanian & Ghadimi, 2018), our proposed FZFW has two improvement. On one hand, in (Balasubramanian & Ghadimi, 2018), the gradient is estimated by the averaged Gaussian random gradient estimator as follows:

$$\hat{\mathbf{v}}_k = \frac{1}{|S|}\sum_{j\in S} \frac{f_j(\mathbf{x}_k + v\mathbf{u}_j) - f_j(\mathbf{x}_k)}{v}\mathbf{u}_j \, , \tag{11}$$

where $\mathbf{u}_j \sim N(\mathbf{0}, \mathbf{I})$ is a Gaussian random vector. (Balasubramanian & Ghadimi, 2018) shows that when using Eq. (11) to estimate $\hat{\mathbf{v}}_k$ in each iteration, the number of function queries is in the order of $O(d/\epsilon^2)$. On the contrary, the coordinate-wise gradient estimator in Eq. (9) only needs $O(d)$ function queries. Thus, our method using the coordinate-wise gradient estimator is better than ZSCG. On the other hand, compared with ZSCG, the proposed FZFW employs the variance reduction technique (Fang et al., 2018; Nguyen et al., 2017; Wang et al., 2018) to estimate the gradient, which is shown in Eq. (10). This estimator can reduce the variance introduced by the randomly selected component functions. Thus, based on these two points, our method is supposed to converge faster than ZSCG. In particular, we established the convergence of Algorithm 1 as follows.

Theorem 1. *Under Assumption 1, if the parameters are chosen as $S_1 = n$, $q = |S_2| = \sqrt{n}$, $\gamma_k = \gamma = \frac{1}{D\sqrt{K}}$, and $\mu = \frac{1}{\sqrt{dK}}$, then Algorithm 1 satisfies:*

$$\mathbb{E}[\mathcal{G}(\mathbf{x}_\alpha)] \leq \frac{D\Big(F(\mathbf{x}_0) - F(\mathbf{x}_*) + 11L\Big)}{\sqrt{K}} \, . \tag{12}$$

Here, we present the proof sketch due to the space limitation. The detailed proof can be found in the appendix.

Proof. At first, in terms of the smoothness of the loss function, we can prove the following inequality:

$$\begin{aligned} \mathbb{E}[F(\mathbf{x}_{k+1})] &\leq \mathbb{E}[F(\mathbf{x}_k)] - \gamma\mathbb{E}[\mathcal{G}(\mathbf{x}_k)] \\ &+ \frac{1}{2L}\mathbb{E}[\|\hat{\mathbf{v}}_k - \nabla F(\mathbf{x}_k)\|^2] + L\gamma^2 D^2 \, . \end{aligned} \tag{13}$$

Then, in terms of Lemma 4, 5, we have

$$\begin{aligned} &\mathbb{E}[F(\mathbf{x}_{k+1})] \leq \mathbb{E}[F(\mathbf{x}_k)] - \gamma\mathbb{E}[\mathcal{G}(\mathbf{x}_k)] \\ &+ \frac{3L\gamma^2}{|S_2|}\sum_{t=(n_k-1)q}^{k-1}\mathbb{E}\|\mathbf{d}_t\|^2 + \frac{6Ld\mu^2(k-(n_k-1)q)}{|S_2|} \\ &+ \frac{3I(|S_1|<n)(2L^2\mu^2 d + \sigma^2)}{L|S_1|} + Ld\mu^2 + L\gamma^2 D^2 \, . \end{aligned} \tag{14}$$

Furthermore, telescoping the above inequality over k from $(n_k-1)q$ to k where $k \leq n_k q - 1$, we have

$$\begin{aligned} &\mathbb{E}[F(\mathbf{x}_{k+1})] \leq \mathbb{E}[F(\mathbf{x}_{(n_k-1)q})] - \gamma\sum_{j=(n_k-1)q}^{k}\mathbb{E}[\mathcal{G}(\mathbf{x}_j)] \\ &+ \frac{3L\gamma^2(k-(n_k-1)q+1)}{|S_2|}\sum_{i=(n_k-1)q}^{k}\mathbb{E}\|\mathbf{d}_i\|^2 \\ &+ \sum_{j=(n_k-1)q}^{k}\Big(\frac{3I(|S_1|<n)(2L^2\mu^2 d+\sigma^2)}{L|S_1|} + Ld\mu^2 \\ &+ L\gamma^2 D^2\Big) + \frac{6Ld\mu^2(k-(n_k-1)q)(k-(n_k-1)q+1)}{|S_2|} \, . \end{aligned} \tag{15}$$

Based on the definition of $\mathbf{x}_\alpha$, we have

$$\begin{aligned} \mathbb{E}[\mathcal{G}(\mathbf{x}_\alpha)] &\leq \frac{F(\mathbf{x}_0) - F(\mathbf{x}_*)}{K\gamma} + \frac{3\gamma LqD^2}{|S_2|} + \frac{6Ld\mu^2 q}{|S_2|\gamma} \\ &+ \frac{3I(|S_1|<n)(2L^2\mu^2 d+\sigma^2)}{L|S_1|\gamma} + \frac{Ld\mu^2}{\gamma} + L\gamma D^2 \, . \end{aligned} \tag{16}$$

By setting $|S_1| = n$, we have $I(|S_1| < n) = 0$. In addition, if we set $\mu = \frac{1}{\sqrt{dK}}$, $\gamma = \frac{1}{D\sqrt{K}}$, and $q = |S_2| = \sqrt{n}$, we can get the desired result. □

Corollary 1. *With the same setting as Theorem 1, the amortized function queries oracle is $O(\frac{n^{1/2}d}{\epsilon^2})$ and the linear oracle is $O(\frac{1}{\epsilon^2})$.*

Proof. Since $q = |S_2| = \sqrt{n}$, then the total number of estimating the gradient in every q iterations is $n + q \times |S_2| = 2n$. In addition, at each estimation, the coordinate-wise gradient estimator evaluates the function value for d times. Thus, the amortized function queries of each iteration is $2nd/q = 2n^{1/2}d$. Then, the total FQO of FZFW is $O(\frac{n^{1/2}d}{\epsilon^2})$. As for LO, it is easy to obtain $O(\frac{1}{\epsilon^2})$. □

Remark 1. *In (Balasubramanian & Ghadimi, 2018), ZSCG requires the number of function queries in each iteration as much as $O(\frac{d}{\epsilon^2})$, while our method only needs $O(n^{1/2}d)$. Thus, our method improves its convergence rate significantly. On the other hand, our result can approximately match the first-order counterpart (Shen et al., 2019).*

3.3. Faster Zeroth-Order Conditional Gradient Sliding Method

In this subsection, we will present the faster zeroth-order conditional gradient sliding (FZCGS) method. Specifically, the acceleration technique is widely used in the first-order optimization method. Especially, (Qu et al., 2017) proposed the accelerated conditional sliding method NCGS-VR which incorporates the idea of the acceleration method to the non-convex Frank-Wolfe method. Inspired by that, we propose the accelerated stochastic zeroth-order stochastic Frank-Wolfe method to further accelerate FZFW. The pseudo code is summarized in Algorithm 2. In detail, to estimate the gradient, we employ the same method as Algorithm 1. The difference between Algorithm 1 and Algorithm 2 lies in the updating of $\mathbf{x}_k$. Here, Algorithm 2 employs the conditional gradient sliding algorithm (Lan & Zhou, 2016; Qu et al., 2017) which is defined in Algorithm 3. If we define $\phi(\mathbf{y}; \mathbf{x}, \nabla F(\mathbf{x}), \gamma) = \min_{\mathbf{y}\in\Omega}\langle \nabla F(\mathbf{x}), \mathbf{y}\rangle + \frac{1}{2\gamma}\|\mathbf{y} - \mathbf{x}\|^2$, step 2 in Algorithm 3 is equivalent to optimize $\max_{\mathbf{x}\in\Omega}\langle \phi'(\mathbf{u}_t; \mathbf{u}, \mathbf{g}, \gamma), \mathbf{u}_t - \mathbf{x}\rangle$. In fact, it is the Wolfe gap. As shown in Algorithm 3, it terminates when the Wolfe gap is smaller than the predefined tolerance η. More details about the conditional gradient sliding algorithm can be found in (Lan & Zhou, 2016; Qu et al., 2017).

Theorem 2. *Under Assumption 1, if the parameters are chosen as $|S_1| = n$, $q = |S_2| = \sqrt{n}$, $\mu = \frac{1}{\sqrt{dK}}$, $\gamma_k = \gamma = \frac{1}{3L}$, and $\eta_k = \eta = \frac{1}{K}$, then Algorithm 2 has the following convergence rate:*

$$\mathbb{E}[\|\mathcal{G}(\mathbf{x}_\alpha, \nabla F(\mathbf{x}_\alpha), \gamma)\|^2] \le \frac{\Big(3(F(\mathbf{x}_0) - F(\mathbf{x}_*) + 1) + 7L\Big)6L}{K} \,. \tag{17}$$

Similarly, we present the proof sketch of this algorithm. More details can be found in the appendix.

Proof. Firstly, we prove the following inequality:

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{k+1})] &\le \mathbb{E}[F(\mathbf{x}_k)] + \frac{\gamma}{2}\mathbb{E}[\|\nabla F(\mathbf{x}_k) - \hat{\mathbf{v}}_k\|^2] \\ &+ (\frac{L}{2} - \frac{1}{2\gamma})\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2] \\ &+ (L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2] + \eta \,.\end{aligned} \tag{18}$$

Secondly, with Lemma 4 and 5, we have

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{k+1})] &\le \mathbb{E}[F(\mathbf{x}_k)] + \frac{3L^2\gamma}{|S_2|}\sum_{t=(n_k-1)q+1}^{k}\mathbb{E}\|\mathbf{d}_{t-1}\|^2 \\ &+ \frac{6L^2\mu^2 d(k-(n_k-1)q)\gamma}{|S_2|} + L^2 d\mu^2\gamma \\ &+ \frac{3I(|S_1|<n)(2L^2\mu^2 d + \sigma^2)\gamma}{|S_1|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 \\ &+ (L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2] + \eta \,.\end{aligned} \tag{19}$$

Thirdly, telescoping it over k from $(n_k-1)q$ to k where $k \le n_k q - 1$, we have

$$\begin{aligned}&\mathbb{E}[F(\mathbf{x}_{k+1})] \le \mathbb{E}[F(\mathbf{x}_{(n_k-1)q})] \\ &+ \sum_{j=(n_k-1)q}^{k}\Big((L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{j+1} - \mathbf{x}_j\|^2] \\ &+ \frac{3\gamma L^2 D^2(k-(n_k-1)q+1)}{|S_2|} + L^2 d\mu^2\gamma \\ &+ \frac{3I(|S_1|<n)(2L^2\mu^2 d + \sigma^2)\gamma}{|S_1|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 + \eta\Big) \\ &+ \frac{6L^2 d\mu^2(k-(n_k-1)q)(k-(n_k-1)q+1)\gamma}{|S_2|} \,.\end{aligned} \tag{20}$$

Then, we have the following inequality:

$$\begin{aligned}&\gamma^2(\frac{1}{2\gamma} - L)\mathbb{E}[\|\mathcal{G}(\mathbf{x}_\alpha, \nabla F(\mathbf{x}_\alpha), \gamma)\|^2] \\ &\le \frac{F(\mathbf{x}_0) - F(\mathbf{x}_*) + 1}{K} + \frac{3\gamma q L^2 D^2}{|S_2|} + \frac{6L^2 d\mu^2\gamma q}{|S_2|} \\ &+ \frac{3I(|S_1|<n)(2L^2\mu^2 d + \sigma^2)\gamma}{|S_1|} \\ &+ L^2 d\mu^2\gamma + (\frac{L}{2} - \frac{1}{2\gamma})D^2 \,.\end{aligned} \tag{21}$$

By setting $|S_1| = n$, then $I(|S_1| < n) = 0$. Besides, if we set $q = |S_2| = \sqrt{n}$, $\mu = \frac{1}{\sqrt{dK}}$, and $\gamma = \frac{1}{3L}$, we have $\frac{3\gamma q L^2 D^2}{|S_2|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 = 0$, then we complete the proof. □

Corollary 2. *With the same setting as Theorem 2, the amortized function queries oracle is $O(\frac{n^{1/2}d}{\epsilon})$ and the linear oracle is $O(\frac{1}{\epsilon^2})$.*

Proof. Similar with the proof of Corollary 1, by setting $q = |S_2| = \sqrt{n}$ and $|S_1| = n$, we have the amortized gradient calling as $O(\frac{n^{1/2}}{\epsilon})$. Thus, the total FQO is $O(\frac{n^{1/2}d}{\epsilon})$. Similarly, the LO is $O(\frac{1}{\epsilon^2})$. □

Remark 2. *Compared with FZFW in Algorithm 1, FZCGS has better FQO since it employs the conditional gradient sliding algorithm to accelerate the convergence speed.*

Remark 3. *Compared with the first-order NCSG-VR (Qu et al., 2017) whose IFO is $O(\frac{n^{2/3}}{\epsilon})$, the iteration complexity of our proposed FZCGS is even better than this first-order counterpart if we ignore the multiplicative parameter d. Thus, we argue that the iteration complexity of NCSG-VR is suboptimal. In the next section, we will propose a new algorithm to improve the first-order counterpart.*

Algorithm 2 Faster Zeroth-Order Conditional Gradient Method (FZCGS)

```
Input: $\mathbf{x}_0, q > 0, \mu > 0, K > 0, \eta > 0, \gamma > 0, n$
for $k = 0, \cdots, K-1$ do
    if mod(k, q) = 0 then
        Sample $S_1$ without replacement to compute $\hat{\mathbf{v}}_k = \hat{\nabla} f_{S_1}(\mathbf{x}_k)$
    else
        Sample $S_2$ with replacement to compute $\hat{\mathbf{v}}_k = \frac{1}{|S_2|}\sum_{i \in S_2}[\hat{\nabla} f_i(\mathbf{x}_k) - \hat{\nabla} f_i(\mathbf{x}_{k-1}) + \hat{\mathbf{v}}_{k-1}]$
    end if
    $\mathbf{x}_{k+1} = \text{condg}(\hat{\mathbf{v}}_k, \mathbf{x}_k, \gamma_k, \eta_k)$
end for
Output: Randomly choose $\mathbf{x}_\alpha$ from $\{\mathbf{x}_k\}$ and return it
```

Algorithm 3 $\mathbf{u}^+ = \text{condg}(\mathbf{g}, \mathbf{u}, \gamma, \eta)$ (Qu et al., 2017)

```
1: $\mathbf{u}_1 = \mathbf{u}, t = 1$
2: $\mathbf{v}_t$ be an optimal solution for
       $V_{\mathbf{g},\mathbf{u},\gamma}(\mathbf{u}_t) = \max_{\mathbf{x} \in \Omega} \langle \mathbf{g} + \frac{1}{\gamma}(\mathbf{u}_t - \mathbf{u}), \mathbf{u}_t - \mathbf{x} \rangle$
3: If $V_{\mathbf{g},\mathbf{u},\gamma}(\mathbf{u}_t) \leq \eta$, return $\mathbf{u}^+ = \mathbf{u}_t$.
4: Set $\mathbf{u}_{t+1} = (1 - \alpha_t)\mathbf{u}_t + \alpha_t \mathbf{v}_t$ where $\alpha_t = \min\{1, \frac{\langle \frac{1}{\gamma}(\mathbf{u} - \mathbf{u}_t) - \mathbf{g}, \mathbf{v}_t - \mathbf{u}_t \rangle}{\frac{1}{\gamma}\|\mathbf{v}_t - \mathbf{u}_t\|^2}\}$ .
5: Set $t \leftarrow t + 1$ and goto step 2.
```

3.4. Faster First-Order Conditional Gradient Sliding Method

As discussed in the last subsection, we found that the iteration complexity of the existing first-order conditional gradient sliding method is even worse than that of our proposed FZCGS in Algorithm 2 if we ignore the multiplicative parameter d. Thus, it is necessary to further improve the existing first-order methods. To address this problem, we propose a new faster conditional gradient sliding (FCGS) method in Algorithm 4. Here, similar with Algorithm 2, we also reduce the variance of the estimator for the full gradient by utilizing Eq. (10). The only difference is that we use the standard gradient rather than the zeroth-order gradient estimator.

Algorithm 4 Faster First-Order Conditional Gradient Sliding Method (FCGS)

```
Input: $\mathbf{x}_0, q > 0, K > 0, \eta > 0, \gamma > 0, n$
for $k = 0, \cdots, K-1$ do
    if mod(k, q) = 0 then
        Compute the full gradient $\mathbf{v}_k = \nabla F(\mathbf{x}_k)$
    else
        Sample $S_2$ to compute $\mathbf{v}_k = \frac{1}{|S_2|}\sum_{i \in S_2}[\nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}]$
    end if
    $\mathbf{x}_{k+1} = \text{condg}(\mathbf{v}_k, \mathbf{x}_k, \gamma_k, \eta_k)$
end for
Randomly choose $\mathbf{x}_\alpha$ from $\{\mathbf{x}_k\}$ and return it
```

Theorem 3. *Under Assumption 1, if the parameters are chosen as $q = |S_2| = \sqrt{n}$, $\gamma_k = \gamma = \frac{1}{3L}$, $\eta_k = \eta$, then Algorithm 4 has the following convergence rate:*

$$\mathbb{E}[\|\mathcal{G}(\mathbf{x}_\alpha, \nabla F(\mathbf{x}_\alpha), \gamma)\|^2] \leq \frac{18L(F(\mathbf{x}_0) - F(\mathbf{x}_*) + 1)}{K} . \tag{22}$$

The proof sketch about Theorem 3 is shown as follows.

Proof. Similar with the proof of Theorem 2, we first have the following inequality:

$$\begin{aligned} & \mathbb{E}[F(\mathbf{x}_{k+1})] \leq \mathbb{E}[F(\mathbf{x}_k)] + \frac{\gamma}{2}\mathbb{E}[\|\nabla F(\mathbf{x}_k) - \hat{\mathbf{v}}_k\|^2] \\ & + (\frac{L}{2} - \frac{1}{2\gamma})\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2] \\ & + (L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2] + \eta . \end{aligned} \tag{23}$$

According to Lemma 4 and 5, we have

$$\begin{aligned} & \mathbb{E}[F(\mathbf{x}_{k+1})] \leq \mathbb{E}[F(\mathbf{x}_k)] + (L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2] \\ & + \frac{\gamma}{2}\sum_{i=(n_k-1)q}^{k} \frac{L^2\mathbb{E}[\|\mathbf{d}_i\|^2]}{|S_2|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 + \eta . \end{aligned} \tag{24}$$

Then, telescoping it over k from $(n_k - 1)q$ to k where $k \leq n_k q - 1$, we have

$$\begin{aligned} & \mathbb{E}[F(\mathbf{x}_{k+1})] \leq \mathbb{E}[F(\mathbf{x}_{(n_k-1)q})] \\ & + \sum_{j=(n_k-1)q}^{k} \Big((L - \frac{1}{2\gamma})\mathbb{E}[\|\tilde{\mathbf{x}}_{j+1} - \mathbf{x}_j\|^2] \\ & + \frac{\gamma q L^2 D^2}{2|S_2|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 + \eta\Big) . \end{aligned} \tag{25}$$

Afterwards, we have the following inequality:

$$\gamma^2(\frac{1}{2\gamma} - L)\mathbb{E}[\|\mathcal{G}(\mathbf{x}_\alpha, \nabla F(\mathbf{x}_\alpha), \gamma)\|^2] \\ \leq \frac{F(\mathbf{x}_0) - F(\mathbf{x}_*) + 1}{K} + \frac{\gamma q L^2 D^2}{2|S_2|} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 \, . \tag{26}$$

Furthermore, by setting $q = |S_2| = \sqrt{n}$ and $\gamma = \frac{1}{3L}$, then $\frac{\gamma L^2 D^2}{2} + (\frac{L}{2} - \frac{1}{2\gamma})D^2 < 0$, we can get the desired result. □

Corollary 3. *With the same setting as Theorem 3, the amortized IFO is $O(\frac{n^{1/2}}{\epsilon})$ and the linear oracle is $O(\frac{1}{\epsilon^2})$.*

Proof. Since $q = |S_2| = \sqrt{n}$, then the total number of IFO in every q iterations is $n + q \times |S_2| = 2n$. Thus, the amortized IFO of each iteration is $2n/q = 2n^{1/2}$. Then, the total IFO of FCGS is $\frac{n^{1/2}}{\epsilon}$. In addition, it is easy to obtain $O(\frac{1}{\epsilon^2})$ for LO. □

4. Experiments

4.1. Experimental Settings

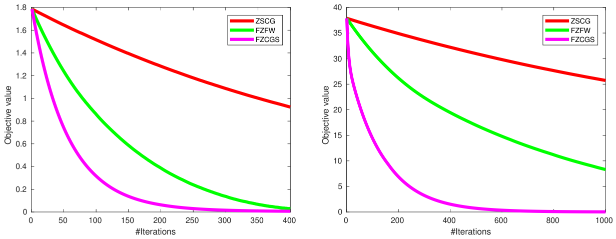
In our experiment, we focus on the non-convex maximum correntropy criterion induced regression (MCCR) (Feng et al., 2015) model as follows:

$$\min_{\|\mathbf{x}\|_1 \leq s} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \sigma^2 \Big(1 - \exp\Big\{ - \frac{(\mathbf{b}_i - \mathbf{x}^T \mathbf{a}_i)^2}{\sigma^2} \Big\}\Big) \tag{27}$$

where σ and s are hyper-parameters. As for the experiment for zeroth-order methods, we view the loss function as a black-box function, which means that only function value is available. As for the experiment for first-order methods, both function value and gradient are available.

To investigate the performance of optimization algorithms for Eq. (27), we synthesize two datasets. In detail, for the data matrix $A = \{\mathbf{a}_i\}_{i=1}^n \in \mathbb{R}^{d \times n}$, each data point $a_i \in \mathbb{R}^d$ is generated independently from a Gaussian distribution $\mathrm{N}(0, \Sigma)$. Then, we construct the response vector by $b = Ax^* + z$ where x^* is a sparse vector with sparsity as s^*, and z is the random noise. Specifically, we use a uniform distribution $U[-1, 1]$ to generate the non-zero entries of x^*. In addition, we employ a $\mathcal{X}^2$-distribution whose degrees of freedom is 2 to generate the noise z. Following these settings, we construct two datasets. Specifically, the first synthetic data (Syn-1) is configured with $n = 10,000$, $d = 100$, $s^* = 20$, and Σ being an identity matrix. The second synthetic data (Syn-2) is configured with $n = 25,000$, $d = 200$, $s^* = 50$, and the off-diagonal entries of the covariance matrix Σ are set to 0.1 and the diagonal entries are set to 1.

To evaluate the performance of our proposed algorithms, we compare them with different baseline methods. Specifically, for the zeroth-order method, the baseline method includes zeroth-order stochastic conditional gradient method (ZSCG) (Balasubramanian & Ghadimi, 2018). Since the FQO of another stochastic zeroth-order method (Sahu et al., 2019) is $O(d^{4/3}/\epsilon^4)$, which is even worse than that of ZSCG, we only compare our method with ZSCG. For the first-order method, the baseline methods include stochastic Frank-Wolfe (SFW) method (Reddi et al., 2016), stochastic variance-reduced Frank-Wolfe (SVFW) method (Reddi et al., 2016), variance reduction non-convex conditional gradient sliding (NCGS-VR) method (Qu et al., 2017). All parameters of these methods are set following the original paper. As for our methods, the parameters are also set in terms of our theoretical analysis.



(a) Syn-1

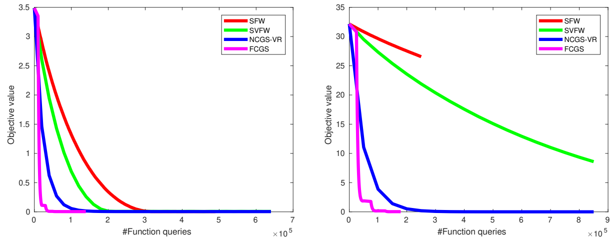
(b) Syn-2

Figure 1. The objective value with respect to the number of iterations obtained by zeroth-order methods.

4.2. Experimental Results

Zeroth-Order Method The convergence result of the zeroth-order method is reported in Figure 1(a) and 1(b). Here, we show the objective function value with respect to the number of iterations. It can be found that our proposed methods outperform the baseline method significantly. Specifically, FZFW converges faster than ZSCG. The reason is two fold. On one hand, FZFW employs the coordinate-wise gradient estimator while ZSCG uses the averaged Gaussian random gradient estimator. On the other hand, FZFW utilizes a variance reduced gradient estimator while ZSCG not. As a result, our proposed FZFW can converge faster than ZSCG. Furthermore, the proposed FZCSG can outperform FZFW. The reason is that FZCSG incorporates the acceleration technique.

First-Order Method In Figure 2(a) and 2(b), we demonstrate the convergence result of different first-order Frank-Wolfe methods. Specifically, we report the objective function value with respect to the number of function queries. It is easy to find that the proposed FCGS is the fastest one among these methods, which means that FCGS has better IFO.



(a) Syn-1 (b) Syn-2

Figure 2. The objective value with respect to the number of function queries obtained by first-order methods

4.3. Additional Experiments

Low-Rank Matrix Completion In this experiment, we use the low-rank matrix completion task, which is defined in Eq. (2), to verify the performance of our proposed stochastic zeroth-order methods. The dataset used in this experiment is MovieLens100k[1]. It is a movie rating matrix. There are 1,682 users and 943 movies. The task is to predict the missing value in the given rating matrix, which can be used for movie recommendation. In this experiment, the trace norm constraint R is set to 7,000.

In Figure 3, we demonstrate the loss function value regarding the number of iterations. It can be seen that our proposed zeroth-order methods converges much faster than ZSCG. Meanwhile, for our proposed two zeroth-order methods, FZCGS converges faster than FZFW. These results confirm the correctness of our theoretical results and the effectiveness of our proposed algorithms.

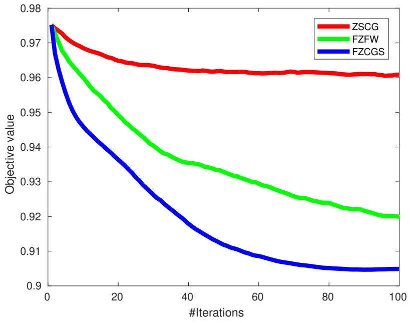


Figure 3. The objective value with respect to the number of iterations obtained by zeroth-order methods for low-rank matrix completion.

Generation of Adversarial Examples In this experiment, we verify the performance of our proposed zeroth-order methods on the task of adversarial attack on black-box DNNs. In particular, given a black-box DNN $f : \mathbb{R}^d \to \mathbb{R}^c$ and a dataset $\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{0, 1, \cdots, c\}\}_{i=1}^n$, the task is to find the adversarial perturbation $\delta \in \mathbb{R}^d$ for sample $\mathbf{x}_i$ such that the DNN model makes the incorrect prediction $\hat{y}_i \neq y_i$. To this end, we optimize the following problem:

$$\min_{\|\delta\|_\infty \leq s} \frac{1}{n} \sum_{i=1}^{n} \max\{f_{y_i}(\mathbf{x}_i + \delta) - \max_{j \neq y_i} f_j(\mathbf{x}_i + \delta), 0\} \tag{28}$$

where $f(\mathbf{x}) = [f_1(\mathbf{x}), f_1(\mathbf{x}), \cdots, f_c(\mathbf{x})]$ denotes the output of the last layer before conducting the softmax operation.

Following (Liu et al., 2018; Ji et al., 2019), we use the same pretrained DNN[2] for MNIST dataset as the black-box model. The hyperparameter s is set to 0.1. The convergence result of different zeroth-order methods is shown in Figure 4. It can be seen that our proposed FZFW converges faster than ZSCG, confirming the correctness of our theoretical result. In addition, our proposed FZCGS method outperforms the non-accelerated FZFW, which also confirms the correctness of our theoretical results.

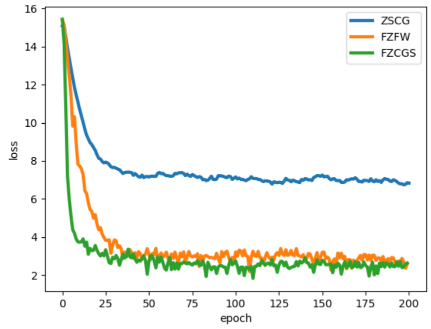


Figure 4. The objective value with respect to the number of iterations obtained by zeroth-order methods for generation of adversarial examples.

5. Conclusion

In this paper, we improved the convergence rate of stochastic zeroth-order Frank-Wolfe method. Specifically, we proposed two algorithms for the zeroth-order Frank-Wolfe methods. Both of them improve the function queries oracle significantly over existing methods. In addition, we also improved the accelerated stochastic zeroth-order Frank-Wolfe method to a better IFO. Experimental results have confirmed the effectiveness of our proposed methods.

[1] https://grouplens.org/datasets/movielens/

[2] https://github.com/IBM/ZOSVRG-BlackBox-Adv

Acknowledgements

This work was partially supported by U.S. NSF IIS 1836945, IIS 1836938, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956.

References

Balasubramanian, K. and Ghadimi, S. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in Neural Information Processing Systems*, pp. 3455–3464, 2018.

Chen, J., Zhou, D., Yi, J., and Gu, Q. A frank-wolfe framework for efficient and effective adversarial attacks. *AAAI*, 2020.

Clarkson, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

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

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Feng, Y., Huang, X., Shi, L., Yang, Y., and Suykens, J. A. Learning with the maximum correntropy criterion induced losses for regression. *Journal of Machine Learning Research*, 16:993–1034, 2015.

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

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

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

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

Hassani, H., Karbasi, A., Mokhtari, A., and Shen, Z. Stochastic conditional gradient++. *arXiv preprint arXiv:1902.06992*, 2019.

Hazan, E. and Luo, H. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pp. 1263–1271, 2016.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization.

Ji, K., Wang, Z., Zhou, Y., and Liang, Y. Improved zeroth-order variance reduced algorithms and analysis for non-convex optimization. *arXiv preprint arXiv:1910.12166*, 2019.

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

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pp. 496–504, 2015.

Lan, G. and Zhou, Y. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pp. 2348–2358, 2017.

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

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., and Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 3727–3737, 2018.

Mokhtari, A., Hassani, H., and Karbasi, A. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2613–2621. JMLR. org, 2017.

Qu, C., Li, Y., and Xu, H. Non-convex conditional gradient sliding. *arXiv preprint arXiv:1708.04783*, 2017.

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

Sahu, A. K., Zaheer, M., and Kar, S. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3468–3477, 2019.

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

Shen, Z., Fang, C., Zhao, P., Huang, J., and Qian, H. Complexities in projection-free stochastic non-convex minimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2868–2876, 2019.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. *arXiv preprint arXiv:1710.10551*, 2017.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

Yurtsever, A., Sra, S., and Cevher, V. Conditional gradient methods via stochastic path-integrated differential estimator. In *Proceedings of the International Conference on Machine Learning-ICML 2019*, number CONF, 2019.

Zhang, M., Shen, Z., Mokhtari, A., Hassani, H., and Karbasi, A. One sample stochastic frank-wolfe. *arXiv preprint arXiv:1910.04322*, 2019.