One Sample Stochastic Frank-Wolfe

Mingrui Zhang[1] **Zebang Shen**[2] **Aryan Mokhtari**[3] **Hamed Hassani**[2] **Amin Karbasi**[1]
[1]Yale University [2]University of Pennsylvania [3]University of Texas at Austin

Abstract

One of the beauties of the projected gradient descent method lies in its rather simple mechanism and yet stable behavior with inexact, stochastic gradients, which has led to its wide-spread use in many machine learning applications. However, once we replace the projection operator with a simpler linear program, as is done in the Frank-Wolfe method, both simplicity and stability take a serious hit. The aim of this paper is to bring them back without sacrificing the efficiency. In this paper, we propose the first one-sample stochastic Frank-Wolfe algorithm, called `1-SFW`, that avoids the need to carefully tune the batch size, step size, learning rate, and other complicated hyper parameters. In particular, `1-SFW` achieves the best known convergence rate of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)-\epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. Moreover, in a general non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations, achieving the current best known convergence rate. All of this is possible by designing a novel unbiased momentum estimator that governs the stability of the optimization process while using a single sample at each iteration.

1 Introduction

Projection-free methods, also known as conditional gradient methods or Frank-Wolfe (FW) methods, have been widely used for solving constrained optimization problems [Frank and Wolfe, 1956, Jaggi, 2013, Lacoste-Julien and Jaggi, 2015]. Indeed, extending such methods to the stochastic setting is a challenging task as it is known that FW-type methods are highly sensitive to stochasticity in gradient computation [Hazan and Kale, 2012]. To resolve this issue several stochastic variants of FW methods have been studied in the literature [Hazan and Kale, 2012, Hazan and Luo, 2016, Reddi et al., 2016, Lan and Zhou, 2016, Braun et al., 2017, Hassani et al., 2019, Shen et al., 2019a, Yurtsever et al., 2019]. In all these stochastic methods, the basic idea is to provide an accurate estimate of the gradient by using some variance-reduction techniques that typically rely on large mini-batches of samples where the size grows with the number of iterations or is reciprocal of the desired accuracy. A growing mini-batch, however, is undesirable in practice as requiring a large collection of samples per iteration may easily prolong the duration of each iterate without updating optimization parameters frequently enough Defazio and Bottou [2018]. A notable exception to this trend is the the work of Mokhtari et al. [2018b] which employs a momentum variance-reduction technique requiring only one sample per iteration; however, this method suffers from suboptimal convergence rates. At the heart of this paper is the answer to the following question:

> *Can we achieve the best known complexity bounds for a stochastic variant of Frank-Wolfe while using a single stochastic sample per iteration?*

We show that the answer to the above question is positive and present the first projection-free method that requires only one sample per iteration to update the optimization variable and yet achieves the best known complexity bounds for convex, nonconvex, and monotone DR-submodular settings.

More formally, we focus on a general *non-oblivious* constrained stochastic optimization problem

$$\min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}) \triangleq \min_{x\in\mathcal{K}} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{F}(\mathbf{x};\mathbf{z})], \qquad (1)$$

where $\mathbf{x}\in\mathbb{R}^d$ is the optimization variable, $\mathcal{K}\subseteq\mathbb{R}^d$ is the convex constraint set, and the objective function $F:\mathbb{R}^d\to\mathbb{R}$ is defined as the expectation over a set of functions $\tilde{F}$. The function $\tilde{F}:\mathbb{R}^d\times\mathcal{Z}\to\mathbb{R}$ is determined by $\mathbf{x}$ and a random variable $\mathbf{z}\in\mathcal{Z}$ with

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

Table 1: Convergence guarantees of stochastic Frank-Wolfe methods for constrained **convex** minimization, **non-convex** minimization, and stochastic **monotone continuous DR-submodular** function maximization

function	Ref.	batch	complexity	non-oblivious	utility
Convex	[Hazan and Kale, 2012]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	✗	-
Convex	[Hazan and Luo, 2016]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Convex	[Mokhtari et al., 2018b]	1	$\mathcal{O}(1/\epsilon^3)$	✗	-
Convex	[Yurtsever et al., 2019]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✗	-
Convex	[Hassani et al., 2019]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✓	-
Convex	**This paper**	1	$\mathcal{O}(1/\epsilon^2)$	✓	-
Non-convex	[Hazan and Luo, 2016]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	✗	-
Non-convex	[Hazan and Luo, 2016]	$\mathcal{O}(1/\epsilon^{4/3})$	$\mathcal{O}(1/\epsilon^{10/3})$	✗	-
Non-convex	[Shen et al., 2019a]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Non-convex	[Yurtsever et al., 2019]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Non-convex	[Hassani et al., 2019]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✓	-
Non-convex	**This paper**	1	$\mathcal{O}(1/\epsilon^3)$	✓	-
Submodular	[Hassani et al., 2017]	1	$O(1/\epsilon^2)$	✗	$(1/2)\text{OPT}-\epsilon$
Submodular	[Mokhtari et al., 2018b]	1	$O(1/\epsilon^3)$	✗	$(1-1/e)\text{OPT}-\epsilon$
Submodular	[Hassani et al., 2019]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✓	$(1-1/e)\text{OPT}-\epsilon$
Submodular	**This paper**	1	$\mathcal{O}(1/\epsilon^2)$	✓	$(1-1/e)\text{OPT}-\epsilon$

distribution $\mathbf{z} \sim p(\mathbf{z}; \mathbf{x})$. We refer to problem (1) as a non-oblivious stochastic optimization problem as the distribution of the random variable $\mathbf{z}$ depends on the choice of $\mathbf{x}$. When the distribution p is independent of $\mathbf{x}$, we are in the standard oblivious stochastic optimization regime where the goal is to solve

$$\min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}) \triangleq \min_{\mathbf{x}\in\mathcal{K}} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\tilde{F}(\mathbf{x};\mathbf{z})]. \tag{2}$$

Hence, the oblivious problem (2) can be considered as a special case of the non-oblivious problem (1). Note that non-oblivious stochastic optimization has broad applications in machine learning, including multi-linear extension of a discrete submodular function [Hassani et al., 2019], MAP inference in determinantal point processes (DPPs) [Kulesza et al., 2012], and reinforcement learning [Du et al., 2017, Sutton and Barto, 2018, Papini et al., 2018, Shen et al., 2019b].

Our goal is to propose an efficient FW-type method for the non-oblivious optimization problem (1). Here, the efficiency is measured by the number of stochastic oracle queries, i.e., the sample complexity of $\mathbf{z}$. As we mentioned earlier, among the stochastic variants of FW, the momentum stochastic Frank-Wolfe method proposed in [Mokhtari et al., 2018a,b] is the only method that requires only one sample per iteration. However, the stochastic oracle complexity of this algorithm is suboptimal, i.e., $\mathcal{O}(1/\epsilon^3)$ stochastic queries are required for both convex minimization and monotone DR-submodular maximization problems. This suboptimal rate is due to the fact that the gradient estimator in momentum FW is biased and it is necessary to use a more conservative averaging parameter to control the effect of the bias term.

To resolve this issue, we propose a one-sample stochastic Frank-Wolfe method, called `1-SFW`, which modifies the gradient approximation in momentum FW to ensure that the resulting gradient estimation is an unbiased estimator of the gradient (Section 3). This goal has been achieved by adding an unbiased estimator of the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$ to the gradient approximation vector (Section 3.1). We later explain why coming up with an unbiased estimator of the gradient difference Δ_t could be a challenging task in the non-oblivious setting and show how we overcome this difficulty (Section 3.2). We also characterize the convergence guarantees of `1-SFW` for convex minimization, nonconvex minimization, and monotone DR-submodular maximization (Section 4). In particular, we show that `1-SFW` achieves the optimal convergence rate of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)-\epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. Moreover, in a general non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations, achieving the current best known convergence rate. Finally, we study the oblivious problem in (2) and show that our proposed `1-SFW` method becomes significantly simpler and the corresponding theoretical results hold under less strict assumptions. For example, in the non-oblivious setting, we require second-order information as the nature of the problems requires; while in the oblivious setting, we only need access to first-order information (Theorem 4). We further highlight the similarities between the variance reduced method in [Cutkosky and Orabona, 2019] also known as STORM and the oblivious variant of `1-SFW`. Indeed, our algorithm has been originally inspired by STORM.

Theoretical results of `1-SFW` and other related works are summarized in Table 1. The complexity shows the required number of stochastic queries to obtain an ϵ-suboptimal solution in convex case; an ϵ-first order stationary point in non-convex case; and an $\alpha\cdot\mathrm{OPT}-\epsilon$ utility in monotone DR-submodular case, where $\alpha = 1/2$ or $(1-1/e)$. These results show that `1-SFW` attains the best known complexity bounds in all the considered settings, while requiring only *one single* stochastic oracle query per iteration and avoiding large batch sizes altogether. Even though the focus of this paper is the fundamental theory behind `1-SFW`, we provide some empirical evidence in Appendix A.

2 Related Work

As a projection-free algorithm, Frank-Wolfe method [Frank and Wolfe, 1956] has been studied for both convex optimization [Jaggi, 2013, Lacoste-Julien and Jaggi, 2015, Garber and Hazan, 2015, Hazan and Luo, 2016, Mokhtari et al., 2018b] and non-convex optimization problems [Lacoste-Julien, 2016, Reddi et al., 2016, Mokhtari et al., 2018c, Shen et al., 2019b, Hassani et al., 2019]. In large-scale settings, distributed FW methods were proposed to solve specific problems, including optimization under block-separable constraint set [Wang et al., 2016], and learning low-rank matrices [Zheng et al., 2018]. The communication-efficient distributed FW variants were proposed for specific sparse learning problems in Bellet et al. [2015], Lafond et al. [2016], and for general constrained optimization problems in [Zhang et al., 2019]. Zeroth-order FW methods were studied in [Balasubramanian and Ghadimi, 2018, Sahu et al., 2018, Chen et al., 2019b].

Several works have studied different ideas for reducing variance in stochastic cases. The SVRG method was proposed by Johnson and Zhang [2013] for the convex setting and then extended to the nonconvex setting in [Allen-Zhu and Hazan, 2016, Reddi et al., 2016, Zhou et al., 2018]. The StochAstic Recursive grAdient algoritHm (SARAH) was studied in [Nguyen et al., 2017a,b]. Then as a variant of SARAH, the Stochastic Path-Integrated Differential Estimator (SPIDER) technique was proposed by Fang et al. [2018]. Based on SPIDER, various algorithms for convex and non-convex optimization problems have been studied [Shen et al., 2019a, Hassani et al., 2019, Yurtsever et al., 2019].

In this paper, we also consider optimizing an important subclass of non-convex objectives, known as continuous DR-submodular functions that generalize the diminishing returns property to the continuous domains. Continuous DR-submodular functions can be minimized exactly [Bach, 2015, Staib and Jegelka, 2017], and maximized approximately [Bian et al., 2017b,a, Hassani et al., 2017, Mokhtari et al., 2018a, Niazadeh et al., 2018, Hassani et al., 2019, Chen et al., 2019a]. They have interesting applications in machine learning, including experimental design [Chen et al., 2018], MAP inference in determinantal point processes (DPPs) [Kulesza et al., 2012], and mean-field inference in probabilistic models [Bian et al., 2019].

3 One Sample SFW Algorithm

3.1 Stochastic gradient approximation

In our work, we build on the momentum variance reduction approach proposed in [Mokhtari et al., 2018a,b] to reduce the variance of the one-sample method. To be more precise, in the momentum FW method [Mokhtari et al., 2018a], we update the gradient approximation $\mathbf{d}_t$ at round t as follows

$$\mathbf{d}_t = (1-\rho_t)\mathbf{d}_{t-1} + \rho_t \nabla\tilde{F}(\mathbf{x}_t;\mathbf{z}_t), \tag{3}$$

where ρ_t is the averaging parameter and $\nabla\tilde{F}(\mathbf{x}_t;\mathbf{z}_t)$ is a *one-sample* estimation of the gradient. Since $\mathbf{d}_t$ is a weighted average of the previous gradient estimation $\mathbf{d}_{t-1}$ and the newly updated stochastic gradient, it has a lower variance comparing to one-sample estimation $\nabla\tilde{F}(\mathbf{x}_t;\mathbf{z}_t)$. In particular, it was shown by Mokhtari et al. [2018a] that the variance of gradient approximation in (3) approaches zero at a sublinear rate of $O(t^{-2/3})$. The momentum approach reduces the variance of gradient approximation, but it leads to a *biased* gradient approximation, i.e., $\mathbf{d}_t$ is not an unbiased estimator of the gradient $\nabla F(\mathbf{x}_t)$. Consequently, it is necessary to use a conservative averaging parameter ρ_t for momentum FW to control the effect of the bias term which leads to a sublinear error rate of $\mathcal{O}(t^{-1/3})$ and overall complexity of $\mathcal{O}(1/\epsilon^3)$.

To resolve this issue and come up with a faster momentum based FW method for the non-oblivious problem in (1), we slightly modify the gradient estimation in (3) to ensure that the resulting gradient estimation is an unbiased estimator of the gradient $\nabla F(\mathbf{x}_t)$. Specifically, we add the term $\tilde{\Delta}_t$, which is an unbiased estimator of the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$, to $\mathbf{d}_{t-1}$. This modification leads to the following gradient approximation

$$\mathbf{d}_t = (1-\rho_t)(\mathbf{d}_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla\tilde{F}(\mathbf{x}_t;\mathbf{z}_t). \tag{4}$$

To verify that $\mathbf{d}_t$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$ we can use a simple induction argument. Assuming that $\mathbf{d}_{t-1}$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$ and $\tilde{\Delta}_t$ is an unbiased estimator of $\nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$ we have $\mathbb{E}[\mathbf{d}_t] = (1-\rho_t)(\nabla F(\mathbf{x}_{t-1}) + (\nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1}))) + \rho_t \nabla F(\mathbf{x}_t) = \nabla F(\mathbf{x}_t)$. Hence, the gradient approximation in (4) leads to an unbiased approximation of the gradient. Let us now explain how to compute an unbiased estimator of the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$ for a non-oblivious setting.

3.2 Gradient variation estimation

The most natural approach for estimating the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$ using only one sample $\mathbf{z}$ is computing the difference of two consecutive stochastic gradients, i.e., $\nabla \tilde{F}(\mathbf{x}_t; \mathbf{z}) - \nabla \tilde{F}(\mathbf{x}_{t-1}; \mathbf{z})$. However, this approach leads to an unbiased estimator of the gradient variation Δ_t only in the oblivious setting where $p(\mathbf{z})$ is independent of the choice of $\mathbf{x}$, and would introduce bias in the more general non-oblivious case. To better highlight this issue, assume that $\mathbf{z}$ is sampled according to distribution $p(\mathbf{z}; \mathbf{x}_t)$. Note that $\nabla \tilde{F}(\mathbf{x}_t; \mathbf{z})$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$, i.e., $\mathbb{E}[\nabla F(\mathbf{x}_t; \mathbf{z})] = \nabla F(\mathbf{x}_t)$, however, $\nabla \tilde{F}(\mathbf{x}_{t-1}; \mathbf{z})$ is not an unbiased estimator of $\nabla F(\mathbf{x}_{t-1})$ since $p(\mathbf{z}; \mathbf{x}_{t-1})$ may be different from $p(\mathbf{z}; \mathbf{x}_t)$.

To circumvent this obstacle, an *unbiased* estimator of Δ_t was introduced in Hassani et al. [2019]. To explain their proposal for approximating the gradient variation using only one sample, note that the difference $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$ can be written as

$$\begin{aligned}\Delta_t &= \int_0^1 \nabla^2 F(\mathbf{x}_t(a))(\mathbf{x}_t - \mathbf{x}_{t-1}) \mathrm{d}a \\ &= \left[\int_0^1 \nabla^2 F(\mathbf{x}_t(a)) \mathrm{d}a\right](\mathbf{x}_t - \mathbf{x}_{t-1}),\end{aligned}$$

where $\mathbf{x}_t(a) = a\mathbf{x}_t + (1-a)\mathbf{x}_{t-1}$ for $a \in [0,1]$. According to this expression, one can find an unbiased estimator of $\int_0^1 \nabla^2 F(\mathbf{x}_t(a)) \mathrm{d}a$ and use its product with $(\mathbf{x}_t - \mathbf{x}_{t-1})$ to find an unbiased estimator of Δ_t. It can be easily verified that $\nabla^2 F(\mathbf{x}_t(a))(\mathbf{x}_t - \mathbf{x}_{t-1})$ is an unbiased estimator of Δ_t if a is chosen from $[0,1]$ uniformly at random. Therefore, all we need is to come up with an unbiased estimator of the Hessian $\nabla^2 F$.

By basic calculus, we can show that $\forall \mathbf{x} \in \mathcal{K}$ and $\mathbf{z}$ with distribution $p(\mathbf{z}; \mathbf{x})$, the matrix $\tilde{\nabla}^2 F(\mathbf{x}; \mathbf{z})$ defined as

$$\begin{aligned}\tilde{\nabla}^2 F(\mathbf{x}; \mathbf{z}) = &\tilde{F}(\mathbf{x}; \mathbf{z})[\nabla \log p(\mathbf{z}; \mathbf{x})][\nabla \log p(\mathbf{z}; \mathbf{x})]^\top \\ &+ \nabla^2 \tilde{F}(\mathbf{x}; \mathbf{z}) + [\nabla \tilde{F}(\mathbf{x}; \mathbf{z})][\nabla \log p(\mathbf{z}; \mathbf{x})]^\top \\ &+ \tilde{F}(\mathbf{x}; \mathbf{z}) \nabla^2 \log p(\mathbf{z}; \mathbf{x}) \\ &+ [\nabla \log p(\mathbf{z}; \mathbf{x})][\nabla \tilde{F}(\mathbf{x}; \mathbf{z})]^\top, \end{aligned} \quad (5)$$

is an *unbiased* estimator of $\nabla^2 F(\mathbf{x})$. Note that the above expression requires only one sample of $\mathbf{z}$. As a result, we can construct $\tilde{\Delta}_t$ as an unbiased estimator of Δ_t using only one sample

$$\tilde{\Delta}_t \triangleq \tilde{\nabla}_t^2(\mathbf{x}_t - \mathbf{x}_{t-1}), \quad (6)$$

where $\tilde{\nabla}_t^2 = \tilde{\nabla}^2 F(\mathbf{x}_t(a); \mathbf{z}_t(a))$, and $\mathbf{z}_t(a)$ follows the distribution $p(\mathbf{z}_t(a); \mathbf{x}_t(a))$. By using this procedure, we can indeed compute the vector $\mathbf{d}_t$ in (4) with only one sample of $\mathbf{z}$ per iteration. Through a completely different analysis from the ones in [Mokhtari et al., 2018a, Hassani et al., 2019], we show that the modified $\mathbf{d}_t$ is still a good gradient estimation (Lemma 2), which allows the establishment of the best known stochastic oracle complexity for our proposed algorithm.

Another issue of this scheme is that in (5) and (6), we need to calculate $\nabla^2 \tilde{F}(\mathbf{x}_t(a); \mathbf{z}_t(a))(\mathbf{x}_t - \mathbf{x}_{t-1})$ and $\nabla^2 \log p(\mathbf{x}_t(a); \mathbf{z}_t(a))(\mathbf{x}_t - \mathbf{x}_{t-1})$, where computation of Hessian is involved. When exact Hessian is not accessible, however, we can resort to an approximation by the difference of two gradients. Precisely, for any function $\psi : \mathbb{R}^d \to \mathbb{R}$, any vector $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\| \leq D$, and some $\delta > 0$ small enough, we have

$$\phi(\delta; \psi) \triangleq \frac{\nabla \psi(\mathbf{x} + \delta \mathbf{u}) - \nabla \psi(x - \delta \mathbf{u})}{2\delta} \approx \nabla^2 \psi(\mathbf{x}) \mathbf{u}.$$

If we assume that ψ is L_2-second-order smooth, i.e., $\|\nabla^2 \psi(\mathbf{x}) - \nabla^2 \psi(\mathbf{y}))\| \leq L_2 \|\mathbf{x} - \mathbf{y}\|,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we can upper bound the approximation error quantitatively:

$$\begin{aligned}\|\nabla^2 \psi(\mathbf{x})\mathbf{u} - \phi(\delta; \psi)\| &= \|\nabla^2 \psi(\mathbf{x})\mathbf{u} - \nabla^2 \psi(\tilde{\mathbf{x}})\mathbf{u})\| \\ &\leq D^2 L_2 \delta, \end{aligned} \quad (7)$$

where $\tilde{\mathbf{x}}$ is obtained by the mean-value theorem. In other words, the approximation error can be sufficiently small for proper δ. So we can estimate Δ_t by

$$\begin{aligned}\tilde{\Delta}_t = &\tilde{F}(\mathbf{x}; \mathbf{z})[\nabla \log p(\mathbf{z}; \mathbf{x})][\nabla \log p(\mathbf{z}; \mathbf{x})]^\top \mathbf{u}_t \\ &+ \phi(\delta_t, \tilde{F}(\mathbf{x}; \mathbf{z})) + [\nabla \tilde{F}(\mathbf{x}; \mathbf{z})][\nabla \log p(\mathbf{z}; \mathbf{x})]^\top \mathbf{u}_t \\ &+ \tilde{F}(\mathbf{x}; \mathbf{z}) \phi(\delta_t, \log p(\mathbf{z}, \mathbf{x})) \\ &+ [\nabla \log p(\mathbf{z}; \mathbf{x})][\nabla \tilde{F}(\mathbf{x}; \mathbf{z})]^\top \mathbf{u}_t, \end{aligned} \quad (8)$$

where $\mathbf{u}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$, $\mathbf{x}, \mathbf{z}, \delta_t$ are chosen appropriately. We also note that since computation of gradient difference has a computational complexity of $\mathcal{O}(d)$, while that for Hessian is $\mathcal{O}(d^2)$, this approximation strategy can also help to accelerate the optimization process.

3.3 Variable update

Once the gradient approximation $\mathbf{d}_t$ is computed, we can follow the update of conditional gradient methods for computing the iterate $\mathbf{x}_t$. In this section, we introduce two different schemes for updating the iterates depending on the problem that we aim to solve.

Algorithm 1 One-Sample SFW (`1-SFW`)

Input: Step sizes $\rho_t \in (0,1), \eta_t \in (0,1)$, initial point $\mathbf{x}_1 \in \mathcal{K}$, total number of iterations T

Output: $\mathbf{x}_{T+1}$ or $\mathbf{x}_o$, where $\mathbf{x}_o$ is chosen from $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$ uniformly at random

for $t = 1, 2, \dots, T$ **do**
    if $t = 1$ **then**
        Sample a point $\mathbf{z}_1$ according to $p(\mathbf{z}_1, \mathbf{x}_1)$
        Compute $\mathbf{d}_1 = \nabla \tilde{F}(\mathbf{x}_1; \mathbf{z}_1)$
    else
        Choose a uniformly at random from $[0,1]$
        Compute $\mathbf{x}_t(a) = a\mathbf{x}_t + (1-a)\mathbf{x}_{t-1}$
        Sample a point $\mathbf{z}_t$ according to $p(\mathbf{z}; \mathbf{x}_t(a))$
        Compute $\tilde{\Delta}_t$ either by $\tilde{\nabla}_t^2 = \tilde{\nabla}^2 F(\mathbf{x}_t(a); \mathbf{z}_t)$ based on (5) and $\tilde{\Delta}_t = \tilde{\nabla}_t^2(\mathbf{x}_t - \mathbf{x}_{t-1})$ (Exact Hessian Option); or by Eq. (8) with $\mathbf{x} = \mathbf{x}_t(a), \mathbf{z} = \mathbf{z}_t$ (Gradient Difference Option)
        $\mathbf{d}_t = (1-\rho_t)(\mathbf{d}_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla \tilde{F}(\mathbf{x}_t, \mathbf{z}_t)$
    end if
    (non-)convex min.: Update $\mathbf{x}_{t+1}$ based on (9)
    DR-sub. max.: Update $\mathbf{x}_{t+1}$ based on (10)
end for

For minimizing a general (non-)convex function using one sample stochastic FW, we update the iterates according to

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_t(\mathbf{v}_t - \mathbf{x}_t), \tag{9}$$

where $\mathbf{v}_t = \arg\min_{v \in \mathcal{K}} \{\mathbf{v}^\top \mathbf{d}_t\}$. In this case, we find the direction that minimizes the inner product with the current gradient approximation $\mathbf{d}_t$ over the constraint set $\mathcal{K}$, and update the variable $\mathbf{x}_{t+1}$ by descending in the direction of $\mathbf{v}_t - \mathbf{x}_t$ with step size η_t.

For monotone DR-submodular maximization, the update rule is slightly different, and a stochastic variant of the continuous greedy method [Vondrák, 2008] can be used. Using the same stochastic estimator $\mathbf{d}_t$ as in the (non-)convex case, the update rule for DR-Submodular optimization is given by

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_t \mathbf{v}_t, \tag{10}$$

where $\mathbf{v}_t = \arg\max_{v \in \mathcal{K}} \{\mathbf{v}^\top \mathbf{d}_t\}, \eta_t = 1/T, T$ is the total number of iterations. Hence, if we start from the origin, after T steps the outcome will be a feasible point as it can be written as the average of T feasible points.

The description of our proposed `1-SFW` method for smooth (non-)convex minimization as well as monotone DR-submodular maximization is outlined in (1).

4 Main Results

Before presenting the convergence results of our algorithm, we first state our assumptions on the constraint set $\mathcal{K}$, the stochastic function $\tilde{F}$, and the distribution $p(\mathbf{z}; \mathbf{x})$.

Assumption 1. *The constraint set $\mathcal{K} \subseteq \mathbb{R}^d$ is compact with diameter $D = \max_{x,y \in \mathcal{K}} \|x - y\|$, and radius $R = \max_{x \in \mathcal{K}} \|x\|$.*

Assumption 2. *The stochastic function $\tilde{F}(\mathbf{x}; \mathbf{z})$ has uniformly bounded function value, i.e., $|\tilde{F}(\mathbf{x}; \mathbf{z})| \leq B$ for all $\mathbf{x} \in \mathcal{K}, \mathbf{z} \in \mathcal{Z}$.*

Assumption 3. *The stochastic gradient $\nabla \tilde{F}$ has uniformly bound norm: $\|\nabla \tilde{F}(\mathbf{x}; \mathbf{z})\| \leq G_{\tilde{F}}, \forall \mathbf{x} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$. The norm of the gradient of $\log p$ has bounded fourth-order moment: $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}; \mathbf{x})} \|\nabla \log p(\mathbf{z}; \mathbf{x})\|^4 \leq G_p^4$. We also define $G = \max\{G_{\tilde{F}}, G_p\}$.*

Assumption 4. *The stochastic Hessian $\nabla^2 \tilde{F}$ has uniformly bounded spectral norm: $\|\nabla^2 \tilde{F}(\mathbf{x}; \mathbf{z})\| \leq L_{\tilde{F}}, \forall \mathbf{x} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$. The spectral norm of the Hessian of $\log p$ has bounded second-order moment: $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}; \mathbf{x})} \|\nabla^2 \log p(\mathbf{z}; \mathbf{x})\|^2 \leq L_p^2$. We also define $L = \max\{L_{\tilde{F}}, L_p\}$.*

We note that in Assumptions 2-4, we assume that the stochastic function $\tilde{F}$ has uniformly bounded function value, gradient norm, and second-order differential. We also note that all these assumptions are necessary, and not restrictive. We elaborate on the reasons as below:

- Assumption 1: The compactness of the feasible set has been assumed in all projection-free papers. It is indeed needed for the convergence of the linear optimization subroutine in the Frank-Wolfe method, otherwise, $\mathbf{v}_t$ in (9) can be unbounded.

- Assumptions 3 and 4 about $\tilde{F}$: Bounded gradient and Hessian of the stochastic function $\tilde{F}$ are the customary assumptions for all the variance reduction methods when we solve the problem over a compact set. The boundedness of the function values (Assumption 2) is a direct implication of bounded gradient and compact constraint set.

- Assumptions 3 and 4 about the distribution p: We emphasize these assumptions hold trivially for the oblivious setting (2), where p is not a function of the variable $\mathbf{x}$. For the non-oblivious case (1), consider the reinforcement learning as an example where p is the distribution of a trajectory given the policy parameter $\mathbf{x}$. It can be verified that for common Gaussian policy with bounded mean and variance, the smoothness of the parameterization of the policy(*e.g.*, neural network with smooth activation function) can imply Assumptions 3 and 4.

Now with these assumptions, we can establish an upper bound for the second-order moment of the spectral norm of the Hessian estimator $\tilde{\nabla}^2 F(\mathbf{x}; \mathbf{z})$ in (5).

Lemma 1. *[Lemma 7.1 of [Hassani et al., 2019]] Under Assumptions 2-4, for all $\mathbf{x} \in \mathcal{K}$, we have*

$$\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\|\tilde{\nabla}^2 F(\mathbf{x};\mathbf{z})\|^2] \leq 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2 \triangleq \bar{L}.$$

Note that the result in Lemma (1) also implies the $\bar{L}$-smoothness of F, since

$$\|\nabla^2 F(\mathbf{x})\|^2 = \|\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{\nabla}^2 F(\mathbf{x};\mathbf{z})]\|^2 \leq \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\|\tilde{\nabla}^2 F(\mathbf{x};\mathbf{z})\|^2] \leq \bar{L}^2.$$

In other words, the conditions in Assumptions 2-4 implicitly imply that the objective function F is $\bar{L}$-smooth.

To establish the convergence guarantees for our proposed `1-SFW` algorithm, the key step is to derive an upper bound on the errors of the estimated gradients. To do so, we prove the following lemma, which provides the required upper bounds in different settings of parameters.

Lemma 2. *Consider the gradient approximation $\mathbf{d}_t$ defined in* (4)*. Under Assumptions 1-4, if we run Algorithm 1 with Exact Hessian Option in Line 9, and with parameters $\rho_t = (t-1)^{-\alpha}$ ($\forall t \geq 2$), and $\eta_t \leq t^{-\alpha}$ ($\forall t \geq 1$ and for some $\alpha \in (0,1]$), then the gradient estimation $\mathbf{d}_t$ satisfies*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_t) - \mathbf{d}_t\|^2] \leq Ct^{-\alpha}, \tag{11}$$

where $C = \max\left\{\frac{2(2G+D\bar{L})^2}{2-2^{-\alpha}-\alpha}, \left[\frac{2}{2-2^{-\alpha}-\alpha}\right]^4, [2D(\bar{L}+L)]^4\right\}$.

Lemma (2) shows that with an appropriate parameter setting, the gradient error converges to zero at a rate of $\mathcal{O}(t^{-\alpha})$. With this unifying upper bound, we can obtain the convergence rates of our algorithm for different kinds of objective functions.

If in the update of `1-SFW` we use the Gradient Difference Option in Line 9 of Algorithm 1 to estimate $\tilde{\Delta}_t$, as pointed out above, we need one further assumption on second-order smoothness of the functions $\tilde{F}$ and $\log p$.

Assumption 5. *The stochastic function $\tilde{F}$ is uniformly $L_{2,\tilde{F}}$- second-order smooth: $\|\nabla^2 \tilde{F}(\mathbf{x};\mathbf{z}) - \nabla^2 \tilde{F}(\mathbf{y};\mathbf{z})\| \leq L_{2,\tilde{F}}\|\mathbf{x}-\mathbf{y}\|$, $\forall \mathbf{x},\mathbf{y} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$. The log probability $\log p(\mathbf{z};\mathbf{x})$ is uniformly $L_{2,p}$-second-order smooth: $\|\nabla^2 \log p(\mathbf{z};\mathbf{x}) - \nabla^2 \log p(\mathbf{z};\mathbf{y})\| \leq L_{2,p}\|\mathbf{x}-\mathbf{y}\|$, $\forall \mathbf{x},\mathbf{y} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$. We also define $L_2 = \max\{L_{2,\tilde{F}}, L_{2,p}\}$.*

We note that under (5), the approximation bound in (7) holds for both $\tilde{F}$ and $\log p$. So for δ_t sufficiently small, the error introduced by the Hessian approximation can be ignored. Thus similar upper bound for errors of estimated gradient still holds.

Lemma 3. *Consider the gradient approximation $\mathbf{d}_t$ defined in* (4)*. Under Assumptions 1-5, if we run Algorithm 1 with Gradient Difference Option in Line 9, and with parameters $\rho_t = (t-1)^{-\alpha}, \delta_t = \frac{\sqrt{3}\eta_{t-1}\bar{L}}{DL_2(1+B)}$ ($\forall t \geq 2$), and $\eta_t \leq t^{-\alpha}$ ($\forall t \geq 1$ and for some $\alpha \in (0,1]$), then the gradient estimation $\mathbf{d}_t$ satisfies*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_t) - \mathbf{d}_t\|^2] \leq Ct^{-\alpha}, \tag{12}$$

where $C = \max\left\{\frac{8(D^2\bar{L}^2+G^2+GD\bar{L})}{2-2^{-\alpha}-\alpha}, \left(\frac{2}{2-2^{-\alpha}-\alpha}\right)^4, (4D(\bar{L}+L))^4\right\}$.

Lemma 3 shows that with Gradient Difference Optionin Line 9 of Algorithm 1, the error of estimated gradient has the same order of convergence rate as that with Exact Hessian Option. So in the following three subsections, we will present the theoretical results of our proposed `1-SFW` algorithm with Exact Hessian Option, for convex minimization, non-convex minimization, and monoton DR-submodular maximization, respectively. The results of Gradient Difference Option only differ in constant factors.

4.1 Convex Minimization

For convex minimization problems, to obtain an ϵ-suboptimal solution, (1) only requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries, and $\mathcal{O}(1/\epsilon^2)$ linear optimization oracle calls. Or precisely, we have

Theorem 1 (Convex)**.** *Consider the `1-SFW` method outlined in Algorithm 1 with Exact Hessian Option in Line 9. Further, suppose the conditions in Assumptions 1-4 hold, and assume that F is convex on $\mathcal{K}$. If we set the algorithm parameters as $\rho_t = (t-1)^{-1}$ and $\eta_t = t^{-1}$, then the output $\mathbf{x}_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(\mathbf{x}_{T+1}) - F(\mathbf{x}^*)] \leq \frac{2\sqrt{C}D}{\sqrt{T}} + \frac{\bar{L}D^2(1+\ln T)}{2T},$$

where $C = \max\{4(2G+D\bar{L})^2, 256, [2D(\bar{L}+L)]^4\}$, and $\mathbf{x}^*$ is a minimizer of F on $\mathcal{K}$.

The result in Theorem 1 shows that the proposed one sample stochastic Frank-Wolfe method, in the convex setting, has an overall complexity of $\mathcal{O}(1/\epsilon^2)$ for finding an ϵ-suboptimal solution. Note that to prove this claim we used the result in Lemma 2 for the case that $\alpha = 1$, i.e., the variance of gradient approximation converges to zero at a rate of $\mathcal{O}(1/t)$. We also highlight that `1-SFW` is *parameter-free*, as the learning rate η_t and the momentum parameter ρ_t do not depend on the parameters of the problem.

4.2 Non-Convex Minimization

For non-convex minimization problems, showing that the gradient norm approaches zero, i.e., $\|\nabla F(\mathbf{x}_t)\| \to 0$, implies convergence to a stationary point in the *unconstrained* setting. Thus, it is usually used as a measure for convergence. In the constrained setting, however, the norm of gradient is not a proper measure for defining stationarity and we instead use the Frank-Wolfe Gap [Jaggi, 2013, Lacoste-Julien, 2016], which is defined by

$$\mathcal{G}(\mathbf{x}) = \max_{\mathbf{v}\in\mathcal{K}} \langle \mathbf{v} - \mathbf{x}, -\nabla F(\mathbf{x})\rangle.$$

We note that by definition, $\mathcal{G}(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathcal{K}$. If some point $\mathbf{x} \in \mathcal{K}$ satisfies $\mathcal{G}(\mathbf{x}) = 0$, then it is a first-order stationary point.

In the following theorem, we formally prove the number of iterations required for one sample stochastic FW to find an ϵ-first-order stationary point in expectation, i.e., a point $\mathbf{x}$ that satisfies $\mathbb{E}[\mathcal{G}(\mathbf{x})] \leq \epsilon$.

Theorem 2 (Non-Convex)**.** *Consider the `1-SFW` method outlined in Algorithm 1 with Exact Hessian Option in Line 9. Further, suppose the conditions in Assumptions 1-4 hold. If we set the algorithm parameters as $\rho_t = (t-1)^{-2/3}$, and $\eta_t = T^{-2/3}$, then the output $\mathbf{x}_o \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[\mathcal{G}(\mathbf{x}_o)] \leq \frac{2B + 3\sqrt{C}D/2}{T^{1/3}} + \frac{\bar{L}D^2}{2T^{2/3}},$$

where $C = \max\left\{ \frac{2(2G+D\bar{L})^2}{\frac{4}{3}-2^{-\frac{2}{3}}}, \left[\frac{2}{\frac{4}{3}-2^{-\frac{2}{3}}}\right]^4, [2D(\bar{L}+L)]^4 \right\}.$

We remark that Theorem (2) shows that Algorithm 1 finds an ϵ-first order stationary points after at most $\mathcal{O}(1/\epsilon^3)$ iterations, while uses exactly one stochastic gradient per iteration. Note that to obtain the best performance guarantee in Theorem (2), we used the result of Lemma 2 for the case that $\alpha = 2/3$, i.e., the variance of gradient approximation converges to zero at a rate of $\mathcal{O}(T^{-2/3})$. Again, we highlight that `1-SFW` is a *parameter-free* algorithm.

4.3 Monotone DR-Submodular Maximization

In this section, we focus on the convergence properties of one-sample stochastic Frank-Wolfe or one-sample stochastic Continuous Greedy for solving a monotone DR-submodular maximization problem. Consider a differentiable function $F : \mathcal{X} \to \mathbb{R}_{\geq 0}$, where the domain $\mathcal{X} \triangleq \prod_{i=1}^d \mathcal{X}_i$, and each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}_{\geq 0}$. We say F is continuous DR-submodular if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ that satisfy $\mathbf{x} \leq \mathbf{y}$ and every $i \in \{1, 2, \cdots, d\}$, we have $\frac{\partial F}{\partial x_i}(\mathbf{x}) \geq \frac{\partial F}{\partial x_i}(\mathbf{y})$.

An important property of continuous DR-submodular function is the concavity along the non-negative directions [Calinescu et al., 2011, Bian et al., 2017b]: for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$, we have $F(\mathbf{y}) \leq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle$. We say F is monotone if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$, we have $F(\mathbf{x}) \leq F(\mathbf{y})$.

For continuous DR-submodular maximization, it has been shown that approximated solution within a factor of $(1 - e^{-1} + \epsilon)$ can not be obtained in polynomial time [Bian et al., 2017b]. To achieve a $(1-e^{-1})\text{OPT} - \epsilon$ approximation guarantee, `1-SFW` requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries, and $\mathcal{O}(1/\epsilon^2)$ linear optimization oracle calls, which are the lower bounds of the complexity established in Hassani et al. [2019].

Theorem 3 (Submodular)**.** *Consider the `1-SFW` method outlined in Algorithm 1 with Exact Hessian Option in Line 9 for maximizing DR-Submodular functions. Further, suppose the conditions in Assumptions 1-4 hold, and further assume that F is monotone and continuous DR-submodular on the positive orthant. If we set the algorithm parameters as $\mathbf{x}_1 = 0, \rho_t = (t-1)^{-1}, \eta_t = T^{-1}$, then the output $\mathbf{x}_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(\mathbf{x}_{T+1})] \geq (1 - e^{-1})F(\mathbf{x}^*) - \frac{4R\sqrt{C}}{T^{1/2}} - \frac{\bar{L}R^2}{2T},$$

where $C = \max\{4(2G + R\bar{L})^2, 256, [2R(\bar{L} + L)]^4\}$.

Finally, we note that Algorithm 1 can also be used to solve stochastic discrete submodular maximization [Karimi et al., 2017, Mokhtari et al., 2018a]. Precisely, we can apply Algorithm 1 on the multilinear extension of the discrete submodular functions, and round the output to a feasible set by lossless rounding schemes like pipage rounding [Calinescu et al., 2011] and contention resolution method [Chekuri et al., 2014].

5 Oblivious Setting

In this section, we specifically study the oblivious problem introduced in (2) which is a special case of the non-oblivious problem defined in (1). In particular, we show that our proposed `1-SFW` method becomes significantly simpler and the corresponding theoretical results hold under less strict assumptions.

5.1 Algorithm

As we discussed in Section 3, a major challenge that we face for designing a variance reduced Frank-Wolfe method for the non-oblivious setting is computing an unbiased estimator of the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$. This is indeed not problematic in the oblivious setting, as in this case $\mathbf{z} \sim p(\mathbf{z})$ is independent of $\mathbf{x}$ and therefore $\nabla \tilde{F}(\mathbf{x}_t; \mathbf{z}) - \nabla \tilde{F}(\mathbf{x}_{t-1}; \mathbf{z})$

Algorithm 2 One-Sample SFW (Oblivious Setting)

Input: Step sizes $\rho_t \in (0,1), \eta_t \in (0,1)$, initial point $\mathbf{x}_1 \in \mathcal{K}$, total number of iterations T
Output: $\mathbf{x}_{T+1}$ or $\mathbf{x}_o$, where $\mathbf{x}_o$ is chosen from $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$ uniformly at random

```
for t = 1, 2, ..., T do
    Sample a point z_t according to p(z)
    if t = 1 then
        Compute d_1 = ∇F̃(x_1; z_1)
    else
        Δ̃_t = ∇F̃(x_t; z_t) − ∇F̃(x_{t−1}; z_t)
        d_t = (1 − ρ_t)(d_{t−1} + Δ̃_t) + ρ_t ∇F̃(x_t, z_t)
    end if
    (non-)convex min.: Update x_{t+1} based on (9)
    DR-sub. max.: Update x_{t+1} based on (10)
end for
```

is an unbiased estimator of the gradient variation $\Delta_t = \nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_{t-1})$. Hence, in the oblivious setting, our proposed one sample FW uses the following gradient approximation

$$\mathbf{d}_t = (1-\rho_t)(\mathbf{d}_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla \tilde{F}(\mathbf{x}_t; \mathbf{z}_t),$$

where $\tilde{\Delta}_t$ is given by

$$\tilde{\Delta}_t = \nabla \tilde{F}(\mathbf{x}_t; \mathbf{z}_t) - \nabla \tilde{F}(\mathbf{x}_{t-1}; \mathbf{z}_t).$$

The rest of the algorithm for updating the variable $\mathbf{x}_t$ is identical to the one for the non-oblivious setting. The description of our proposed algorithm for the oblivious setting is outlined in Algorithm 2.

Remark 1. *We note that by rewriting our proposed* `1-SFW` *method for the oblivious setting, we recover the variance reduction technique STORM [Cutkosky and Orabona, 2019] with different sets of parameters. In [Cutkosky and Orabona, 2019], however, the STORM algorithm was combined with SGD to solve* unconstrained *non-convex minimization problems, while our proposed* `1-SFW` *method solves convex minimization, non-convex minimization, and DR-submodular maximization in a* constrained *setting.*

5.2 Theoretical results

In this section, we show that the variant of one sample stochastic FW for the oblivious setting (described in Algorithm 2) recovers the theoretical results for the non-oblivious setting with less assumptions. In particular, we only require the following condition for the stochastic functions $\tilde{F}$ to prove our main results.

Assumption 6. *The function $\tilde{F}$ has uniformly bound gradients, i.e., $\forall \mathbf{x} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$*

$$\|\nabla \tilde{F}(\mathbf{x}; \mathbf{z})\| \leq G.$$

Moreover, the function $\tilde{F}$ is uniformly L-smooth, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}, \forall \mathbf{z} \in \mathcal{Z}$

$$\|\nabla \tilde{F}(\mathbf{x}; \mathbf{z}) - \nabla \tilde{F}(\mathbf{y}; \mathbf{z})\| \leq L \|\mathbf{x} - \mathbf{y}\|$$

We note that as direct corollaries of Theorems 1 to 3, Algorithm 2 achieves the same convergence rates, which is stated in Theorem 4 formally.

Theorem 4. *Consider the oblivious variant of* `1-SFW` *outlined in Algorithm 2, and assume that the conditions in Assumptions 1, 2 and 6 hold. Then we have*

1. *If F is convex on $\mathcal{K}$, and we set $\rho_t = (t-1)^{-1}$ and $\eta_t = t^{-1}$, then the output $\mathbf{x}_{T+1} \in \mathcal{K}$ is feasible and satisfies*
$$\mathbb{E}[F(\mathbf{x}_{T+1}) - F(\mathbf{x}^*)] \leq \mathcal{O}(T^{-1/2}).$$
2. *If F is non-convex, and we set $\rho_t = (t-1)^{-2/3}$, and $\eta_t = T^{-2/3}$, then the output $\mathbf{x}_o \in \mathcal{K}$ is feasible and satisfies*
$$\mathbb{E}[\mathcal{G}(\mathbf{x}_o)] \leq \mathcal{O}(T^{-1/3}).$$
3. *If F is monotone DR-submodular on $\mathcal{K}$, and we set $\mathbf{x}_1 = 0, \rho_t = (t-1)^{-1}$ and $\eta_t = T^{-1}$, then the output $\mathbf{x}_{T+1} \in \mathcal{K}$ is feasible and satisfies*
$$\mathbb{E}[F(\mathbf{x}_{T+1})] \geq (1 - e^{-1})F(\mathbf{x}^*) - \mathcal{O}(T^{-1/2}).$$

Theorem 4 shows that the oblivious version of `1-SFW` requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries to find an ϵ-suboptimal solution for convex minimization, at most $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations to achieve a $(1-1/e)-\epsilon$ approximate solution for monotone DR-submodular maximization, and at most $\mathcal{O}(1/\epsilon^3)$ stochastic oracle queries to find an ϵ-first-order stationary point for nonconvex minimization.

6 Conclusion

In this paper, we studied the problem of solving constrained stochastic optimization programs using projection-free methods. We proposed the first stochastic variant of the Frank-Wolfe method, called `1-SFW`, that requires only one stochastic sample per iteration while achieving the best known complexity bounds for (non-)convex minimization and monotone DR-submodular maximization. In particular, we proved that `1-SFW` achieves the best known oracle complexity of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)\mathrm{OPT} - \epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. Moreover, in a non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations, achieving the best known overall complexity.

Acknowledgements

Amin Karbasi is partially supported by NSF (IIS-1845032), ONR (N00014-19-1-2406), and AFOSR (FA9550-18-1-0160). The research of H. Hassani is supported by AFOSR Award 19RT0726, NSF HDR TRIPODS award 1934876, NSF award CPS-1837253, NSF award CIF-1910056, and NSF CAREER award CIF-1943064.

References

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707, 2016.

Francis Bach. Submodular functions: from discrete to continous domains. *arXiv preprint arXiv:1511.00394*, 2015.

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

Aurélien Bellet, Yingyu Liang, Alireza Bagheri Garakani, Maria-Florina Balcan, and Fei Sha. A distributed frank-wolfe algorithm for communication-efficient sparse learning. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 478–486. SIAM, 2015.

An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems*, pages 486–496, 2017a.

Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 111–120. PMLR, 2017b. URL `http://proceedings.mlr.press/v54/bian17a.html`.

Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous dr-submodular maximization and applications to provable mean field inference. In *International Conference on Machine Learning*, pages 644–653, 2019.

Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 566–575, 2017.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *AISTATS*, pages 1896–1905, 2018.

Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–113, 2019a.

Lin Chen, Mingrui Zhang, Hamed Hassani, and Amin Karbasi. Black box submodular maximization: Discrete and continuous settings. *arXiv preprint arXiv:1901.09515*, 2019b.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*, 2019.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1049–1058. JMLR. org, 2017.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 687–697, 2018.

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

Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *ICML*, volume 15, pages 541–549, 2015.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *arXiv preprint arXiv:1708.03949*, 2017.

Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy++: When upper and lower bounds match. In *Advances in Neural Information Processing Systems*, pages 13066–13076, 2019.

Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, pages 1843–1850, 2012.

Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *ICML*, pages 1263–1271, 2016.

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

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. In *NIPS*, page to appear, 2017.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

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

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

Jean Lafond, Hoi-To Wai, and Eric Moulines. D-fw: Communication efficient distributed algorithms for high-dimensional sparse optimization. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4144–4148. IEEE, 2016.

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

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *AISTATS*, pages 1886–1895, 2018a.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018b.

Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. In *Advances in Neural Information Processing Systems*, pages 3629–3639, 2018c.

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

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017b.

Rad Niazadeh, Tim Roughgarden, and Joshua Wang. Optimal algorithms for continuous non-monotone submodular and dr-submodular maximization. In *Advances in Neural Information Processing Systems*, pages 9594–9604, 2018.

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1806.05618*, 2018.

Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. *arXiv preprint arXiv:1708.04783*, 2017.

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

Anit Kumar Sahu, Manzil Zaheer, and Soummya Kar. Towards gradient free and projection free stochastic optimization. *arXiv preprint arXiv:1810.03233*, 2018.

Zebang Shen, Cong Fang, Peilin Zhao, Junzhou Huang, and Hui Qian. Complexities in projection-free stochastic non-convex minimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2868–2876, 2019a.

Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *International Conference on Machine Learning*, pages 5729–5738, 2019b.

Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. In *ICML*, pages 3230–3240, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74. ACM, 2008.

Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. Parallel and distributed block-coordinate frank-wolfe algorithms. In *International Conference on Machine Learning*, pages 1548–1557, 2016.

Kilian Q Weinberger and Lawrence K Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1):77–90, 2006.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in neural information processing systems*, pages 2301–2309, 2013.

Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *International Conference on Machine Learning*, pages 7282–7291, 2019.

Mingrui Zhang, Lin Chen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Quantized frank-wolfe: Communication-efficient distributed optimization. *arXiv preprint arXiv:1902.06332*, 2019.

Wenjie Zheng, Aurélien Bellet, and Patrick Gallinari. A distributed frank–wolfe framework for learning low-rank matrices with the trace norm. *Machine Learning*, 107(8-10):1457–1475, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3921–3932, 2018.