SIAM J. OPTIM.
Vol. 30, No. 4, pp. 3315–3344

STOCHASTIC CONDITIONAL GRADIENT++: (NON)CONVEX MINIMIZATION AND CONTINUOUS SUBMODULAR MAXIMIZATION*

HAMED HASSANI†, AMIN KARBASI‡, ARYAN MOKHTARI§, AND ZEBANG SHEN†

Abstract. In this paper, we consider the general nonoblivious stochastic optimization where the underlying stochasticity may change during the optimization procedure and depends on the point at which the function is evaluated. We develop Stochastic Frank–Wolfe++ (SFW++), an efficient variant of the conditional gradient method for minimizing a smooth nonconvex function subject to a convex body constraint. We show that SFW++ converges to an ϵ-first order stationary point by using $O(1/\epsilon^3)$ stochastic gradients. Once further structures are present, SFW++'s theoretical guarantees, in terms of the convergence rate and quality of its solution, improve. In particular, for minimizing a convex function, SFW++ achieves an ϵ-approximate optimum while using $O(1/\epsilon^2)$ stochastic gradients. It is known that this rate is optimal in terms of stochastic gradient evaluations. Similarly, for maximizing a monotone continuous DR-submodular function, a slightly different form of SFW++, called Stochastic Continuous Greedy++ (SCG++), achieves a tight $[(1-1/e)\mathrm{OPT}-\epsilon]$ solution while using $O(1/\epsilon^2)$ stochastic gradients. Through an information theoretic argument, we also prove that SCG++'s convergence rate is optimal. Finally, for maximizing a nonmonotone continuous DR-submodular function, we can achieve a $[(1/e)\mathrm{OPT}-\epsilon]$ solution by using $O(1/\epsilon^2)$ stochastic gradients. We should highlight that our results and our novel variance reduction technique trivially extend to the standard and easier oblivious stochastic optimization settings for (non)convex and continuous submodular settings.

Key words. nonconvex minimization, submodular maximization, stochastic optimization, conditional gradient method, first order method, variance reduction

AMS subject classifications. 49M05, 49M15, 49M37, 90C06, 90C30

DOI. 10.1137/19M1304271

1. Introduction.

In this paper, we consider the following *nonoblivious* stochastic maximization problem:

$$\max_{\mathbf{x}\in\mathcal{C}} F(\mathbf{x}) \;:=\; \max_{\mathbf{x}\in\mathcal{C}} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{F}(\mathbf{x};\mathbf{z})], \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^d_+$ is the decision variable, $\mathcal{C}\subseteq\mathbb{R}^d$ is a feasible set, $\mathbf{z}\in\mathcal{Z}$ is a random variable with distribution $p(\mathbf{z};\mathbf{x})$, and the function $F:\mathbb{R}^d\to\mathbb{R}$ is defined as the expectation over a set of smooth stochastic functions $\tilde{F}:\mathbb{R}^d\times\mathcal{Z}\to\mathbb{R}$. Problem (1) is called nonoblivious as the underlying distribution depends on the variable $\mathbf{x}$ and may change during the optimization procedure. Note that the standard stochastic (convex/nonconvex) optimization is a special case of problem (1). We focus on providing efficient solvers for problem (1) in terms of the sample complexity of $\mathbf{z}$ (a.k.a. calls to the stochastic oracle), where F is (non)concave or continuous submodular and the feasible set $\mathcal{C}$ is a bounded convex body. Note that maximizing a nonconcave function F is equivalent to minimizing a nonconvex function $-F$. However, in order to unify

*Received by the editors December 10, 2019; accepted for publication (in revised form) September 4, 2020; published electronically December 14, 2020. The authors are listed in alphabetical order.
https://doi.org/10.1137/19M1304271
†Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (hassani@seas.upenn.edu, zebang@seas.upenn.edu).
‡Department of Electrical Engineering and Computer Science, Yale University, New Haven, CT 06511 USA (amin.karbasi@yale.edu).
§Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (mokhtari@austin.utexas.edu).

the language between the nonconvex minimization and continuous submodular maximization, we resort to formulation (1). In the following, we discuss three concrete instances of nonoblivious stochastic optimization, namely, multilinear extension of a discrete submodular function, maximum a posteriori (MAP) inference in determinantal point processes, and policy gradient in reinforcement learning. In all of these problems the stochasticity of the objective function crucially depends on the decision variable $\mathbf{x}$ at which we evaluate the function.

Our contributions. In this paper, we develop a new variance reduction technique for nonoblivious stochastic optimization problem (1). Note that the success of the variance reduction technique in the standard oblivious stochastic setting relies on the property that the difference of gradients at two points can be unbiasedly estimated using a single sample. However, for the more general nonoblivious case, this crucial property is missing, which invalidates the applicability of the previous variance reduction techniques in the problems of interest here (we elaborate on this at the end of section 2). The key algorithmic contribution of this paper is a way to estimate the difference of gradients without introducing extra bias, as discussed in detail in section 3.1.1. This draws a clear distinction of our work from the literature on stochastic first order algorithms. In particular, we show the following results for problem (1).

- For maximizing a general nonconcave function F (or minimizing a nonconvex function), we develop `Stochastic Frank--Wolfe`++ (SFW++), which converges to an ϵ-first order stationary point (ϵ-FOSP) using $O(1/\epsilon^3)$ stochastic gradients in total. Our result improves upon the previously best convergence rate of $O(1/\epsilon^4)$ by [50] in the oblivious stochastic setting. Moreover, as a byproduct, SFW++ provides the first trajectory complexity of $O(1/\epsilon^3)$ for policy gradient methods in reinforcement learning with convex constraints.
- When the function F is concave, SFW++ achieves an ϵ-approximate optimum while using $O(1/\epsilon^2)$ stochastic gradients, thus achieving the optimum rate for this instance. This result improves upon the previously best convergence rate of $O(1/\epsilon^3)$ in [42]. In the oblivious stochastic setting, the convergence rate of SFW++ is on par with the stochastic gradient sliding [39].
- For maximizing a monotone DR-continuous function F, the `Stochastic Continuous Greedy`++ (SCG++) method is introduced, the first algorithm that achieves the tight $[(1-1/e)\text{OPT}-\epsilon]$ solution by using $\mathcal{O}(1/\epsilon^2)$ stochastic gradients in total. Through an information theoretic argument, we also show that no first order algorithm can achieve a converegnce rate better than $\mathcal{O}(1/\epsilon^2)$. This result improves upon the previously best convergence rate of $O(1/\epsilon^3)$ in [41]. Moreover, SCG++ leads to the fastest method for maximizing a multilinear extension of a monotone submodular set function.
- For maximizing a nonmonotone DR-continuous function F, subject to a down-closed convex body $\mathcal{C}$, we develop `Stochastic Measured Continuous Greedy`++ (SMCG++), which achieves a $[(1/e)\text{OPT}-\epsilon]$ solution by using at most $\mathcal{O}(1/\epsilon^2)$ stochastic gradients. This result improves upon the previously best convergence rate of $O(1/\epsilon^3)$ in [42]. Moreover, SMCG++ (along with lossless rounding schemes such as contention resolution, randomized pipage rounding, etc.) provides a rigorous $1/e$ approximation guarantee for MAP estimation of a determinantal point process, improving upon the semiheuristic $1/4$ approximation guarantee in [27].

1.1. Examples. In this subsection, we briefly mention some instances of the nonoblivious optimization problem in (1).

Multilinear extension of a discrete submodular set function. One canonical example of the stochastic optimization problem in (1) is the multilinear extension of a discrete submodular function. Specifically, consider a discrete submodular set function $f : 2^V \to \mathbb{R}_+$ defined over the set V. The aim is to solve $\max_{S\in\mathcal{I}} f(S)$ where $\mathcal{I}$ encodes a matroid constraint. For this case, the greedy algorithm leads to a 1/2 approximation guarantee, but one can achieve the optimal approximation guarantee of $(1 - 1/e)$ by maximizing its multilinear extension $F : [0, 1]^V \to \mathbb{R}_+$, defined as

$$F(\mathbf{x}) := \mathbb{E}_{\mathbf{z}\sim\mathbf{x}}[f(\mathbf{z}(\mathbf{x}))] := \sum_{S\subseteq V} f(S) \prod_{i\in S} x_i \prod_{j\notin S}(1 - x_j). \tag{2}$$

Here, each element e of the random set $\mathbf{z}(\mathbf{x})$ is sampled with probability x_e. This problem is an instance of (1) if we define $\tilde{F}(\mathbf{x}, \mathbf{z})$ as $f(\mathbf{z}(\mathbf{x}))$, the joint probability $p(\mathbf{x}, \mathbf{z})$ as the distribution of the random set $\mathbf{z}(\mathbf{x})$ (i.e., each coordinate z_e is generated according to a Bernoulli distribution with parameter x_e), and the set $\mathcal{C}$ as the convex hull of $\mathcal{I}$. Later, we show how constraint submodular maximization can be solved efficiently, providing a fast method for maximizing a multilinear extension function.

MAP inference in determinantal point processes (DPPs). DPPs are a class of discrete probabilistic models that were introduced in statistical physics and random matrix theory. Due to their ability to model repulsion and negative correlations, they have been shown to be key concepts for many applications in machine learning [35]. Formally, given a positive definite matrix A of size n, a DPP assigns to any subset $S \subseteq [n] \triangleq \{1, 2, \dots, n\}$, a probability value $\Pr(S) = \det(A_S)/\det(A)$. MAP inference in such a model consists of maximizing the value $\det(A_S)$, or equivalently the log-likelihood $\log\det(A_S)$, over all the subsets $S \subseteq [n]$. Indeed, the set function $f(S) = \log\det(A_S)$ is submodular but generally nonmonotone. As a result, MAP inference in DPPs is an instance of a nonmonotone submodular maximization problem. Moreover, MAP inference may be restricted to subsets that satisfy some given constraints. For instance, for the cardinality constraint, the problem is to find a subset with a size of at most k, which has the largest probability. There are in general two approaches to maximizing the log-likelihood function in DPPs subject to feasibility constraints, both of which rely on appropriate continuous extensions. The first approach is to form the multilinear extension, defined in (2), and solve the resulting constrained nonmonotone submodular optimization problem. Since the multilinear extension involves summing over exponentially many terms, it was generally believed that the optimization will be computationally expensive and convergence issues may arise. This paper overcomes these challenges completely by providing the first $((1/e)\mathrm{OPT} - \epsilon)$ solution in $O(1/\epsilon^2)$ stochastic iterations whenever the sampled sets form a matroid and the function $\log\det(A_S)$ is nonnegative for all S. Another approach, proposed by [27], is to form the so-called *softmax* extension $G : [0, 1]^n \to \mathbb{R}$, defined as

$$G(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim\mathbf{x}}\left[\exp(f(\mathbf{z}(\mathbf{x})))\right] = \log\det\left(\mathrm{diag}(\mathbf{x})(A - I) + I\right).$$

The softmax extension is a deterministic function that can be maximized within a 1/4 approximation to the optimal value OPT. Unlike the multilinear extension, no provable rounding scheme is known for the softmax extension. Therefore, our approach not only improves the approximation ratio of the MAP estimator but also enjoys a rigorous end-to-end guarantee for this problem.

Reinforcement learning. Consider a discrete time index $h \geq 0$ and a Markov system with states $s_h \in \mathcal{S}$ and actions $a_h \in \mathcal{A}$. The probability distribution of the

Table 1.1
Convergence guarantees of conditional gradient (FW) methods for convex minimization.

Ref.	Setting	Assumptions	Batch	Rate/iter	Complexity	Nonobl.
[31]	det.	smooth	—	$\mathcal{O}(1/t)$	—	✗
[29]	stoch.	smooth, bounded grad.	$\mathcal{O}(t)$	$\mathcal{O}(1/t^{1/2})$	$\mathcal{O}(1/\epsilon^4)$	✗
[30]	stoch.	smooth, bounded grad.	$\mathcal{O}(t^2)$	$\mathcal{O}(1/t)$	$\mathcal{O}(1/\epsilon^3)$	✗
[42]	stoch.	smooth, bounded var.	$\mathcal{O}(1)$	$O(1/t^{1/3})$	$\mathcal{O}(1/\epsilon^3)$	✗
This paper	stoch.	smooth, bounded var.	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✓

initial state is $\rho(s_0)$, and the conditional probability distribution of transitioning into s_{h+1} given that we are in state s_h and take action a_h is $\mathcal{P}(s_{h+1}|s_h, a_h)$. Actions are chosen based on a random policy π in which $\pi(a_h|s_h)$ is the distribution for taking action a_h when observing state s_h. We assume that policies are parametrized by a vector $\theta \in \mathbb{R}^d$ and use π_θ as shorthand for the conditional distribution $\pi(a_h|s_h;\theta)$ associated to θ. For a given time horizon H we define the trajectory $\tau := (s_1, a_1, \dots, s_H, a_H)$ as the collection of state-action pairs experienced up until time $h = H$. Given the initial distribution $\rho(s_0)$, the transition kernel $\mathcal{P}(s_{h+1}|s_h, a_h)$, and the Markov property of the system, it follows that the probability distribution over trajectories τ is

$$p(\tau;\pi_\theta) \stackrel{\text{def}}{=} \rho(s_0) \prod_{h=1}^{H} \mathcal{P}(s_{h+1}|s_h, a_h)\, \pi(a_h|s_h). \tag{3}$$

Associated with a state-action pair we have a reward function $r(s_h, a_h)$. When following a trajectory $\tau = (s_1, a_1, \dots, s_H, a_H)$, we consider the accumulated reward discounted by a geometric factor γ $\mathcal{R}(\tau) = \sum_{h=1}^{H} \gamma^h r(s_h, a_h)$. Our goal in reinforcement learning is to find the policy parameter θ that maximizes the expected reward

$$\max J(\theta) \stackrel{\text{def}}{=} \mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\mathcal{R}(\tau)] = \int \mathcal{R}(\tau) p(\tau;\pi_\theta)\mathbf{d}\tau. \tag{4}$$

Here, the underlying distribution p depends on the variable θ, and therefore this problem can be considered as an instance of the nonoblivious formulation in (1).

To find an ϵ-FOSP of problem (4), the trajectory complexities of classic stochastic gradient descent (SGD) based methods like REINFORCE are $\mathcal{O}(1/\epsilon^4)$ [58]. While a direct application of the recent variance reduced gradient estimation leads to a biased gradient estimator of $\mathcal{J}(\cdot)$, due to the inherent difficulty of the nonoblivious optimization of (4), a recent work [56] proposed a policy Hessian method to aid the estimation of the policy gradient. They improve the trajectory complexity from $\mathcal{O}(1/\epsilon^4)$ to $\mathcal{O}(1/\epsilon^3)$ in the unconstrained setting, i.e., $\mathcal{C} = \mathbb{R}^d$. In this paper, we show how to find an ϵ-FOSP for the more general constrained problem (4) with the same improved trajectory complexity in a projection-free manner. Moreover, we emphasize that the reinforcement learning (RL) problem (4) is strictly a special case of the more general objective (1): The reward function $\mathcal{R}(\tau)$ in (4) depends only on the random variable τ and is independent of the variable θ; in (1), the function $\tilde{F}(\mathbf{x};\mathbf{z})$ depends on both the variable $\mathbf{x}$ and the random variable $\mathbf{z}$, which is hence more general.

1.2. Related work. Our work on the conditional gradient method in the nonoblivious stochastic setting has consequences for convex, continuous submodular, and nonconvex cases. In the following, we review some of the most relevant work and our results with respect to them. The comparisons are summarized in Tables 1.1, 1.2, and 1.3 for the convex, continuous submodular, and nonconvex cases, respectively.

We would like to emphasize that all the other related work only provide guarantees for the oblivious setting.

Convex minimization. The problem of minimizing a stochastic convex function subject to a convex constraint using stochastic projected gradient descent–type methods has been studied extensively in the past [45, 46, 52]. Although stochastic gradient computation is inexpensive, the cost of the projection step can be prohibitive [23] or intractable [16]. In such cases, the projection-free methods, a.k.a. Frank–Wolfe or conditional gradient, are the method of choice [21, 31]. In the stochastic setting, the online Frank–Wolfe algorithm proposed in [29] requires $\mathcal{O}(1/\epsilon^4)$ stochastic gradient evaluations to reach an ϵ-approximate optimum, i.e., $F(\mathbf{x}) - OPT \leq \epsilon$, under the assumption that the objective function is convex and has bounded gradients. The stochastic variant of Frank–Wolfe studied in [30] uses an increasing batch size of $b = \mathcal{O}(t^2)$ (at iteration t) to obtain an improved stochastic oracle complexity of $\mathcal{O}(1/\epsilon^3)$ under the assumptions that the expected objective function is smooth and Lipschitz continuous. Recently, [42] proposed a momentum gradient estimator which achieves a similar $\mathcal{O}(1/\epsilon^3)$ stochastic gradient evaluation while fixing the batch size to 1. Reference [39] proposed a stochastic conditional gradient sliding method which finds an ϵ-approximate solution after $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon)$ calls to a linear minimization oracle. The main idea in gradient sliding algorithms is to simulate the projected gradient descent step by solving a sequence of properly chosen linear minimization problems [6, 37, 38, 39]. Our proposed method SFW++ also requires $\mathcal{O}(1/\epsilon^2)$ calls to a stochastic gradient oracle (for oblivious and nonoblivious settings) and $\mathcal{O}(1/\epsilon)$ calls to a linear minimization oracle. However, unlike gradient sliding, we do not resort to simulating the projection step and more closely follow the recipe of the Frank–Wolfe method. In this sense, SFW++ might be considered the first variant of Frank–Wolfe that achieves the optimum convergence rate in the convex setting.

Submodular maximization. Submodular set functions [44] capture the intuitive notion of diminishing returns and have become increasingly important in various fields. The celebrated result of [44] shows that for a monotone submodular function and subject to a cardinality constraint, a simple greedy algorithm achieves the tight $(1-1/e)$ approximation guarantee. However, the vanilla greedy method does not provide the tightest guarantees for many classes of feasibility constraints. To circumvent this issue, the continuous relaxation of submodular functions, through the multilinear extension, have been extensively studied [9, 15, 20, 26, 59, 60]. In particular, it is known that the continuous greedy algorithm achieves the tight $(1 - 1/e)$ approximation guarantee for monotone submodular functions under a general matroid constraint [9]. In the nonmonotone setting, a slight variant of continuous greedy, called measured continuous greedy, achieves a $1/e$ approximation guarantee [20]. In the absence of constraints, two recent works [48, 53] are able to achieve the tight 1/2 approximation guarantee for the online unconstrained nonmonotone submodular maximization problem by exploiting the offline bi-greedy algorithm proposed in [7]. The continuous relaxation of submodular functions has also been used to robustify submodular optimization in the stochastic settings [28, 34, 41].

Continuous DR-submodular functions, an important subclass of nonconvex functions, generalize the notion of diminishing returns to the continuous domains [3, 61]. It has been recently shown that monotone continuous DR-submodular functions can be (approximately) maximized over convex bodies using first order methods [5, 28, 41]. When exact gradient information is available, [5] showed that the continuous greedy algorithm, which itself is a variant of the conditional gradient method, achieves

Table 1.2
Convergence guarantees for continuous DR-submodular function maximization.

Ref.	Setting	Function	Const.	Utility	Complexity
[14]	det.	mon.smooth sub.	poly.	$(1-1/e)\text{OPT}-\epsilon$	$O(1/\epsilon^2)$
[5]	det.	mon. DR-sub.	cvx-down	$(1-1/e)\text{OPT}-\epsilon$	$O(1/\epsilon)$
[4]	det.	non-mon. DR-sub.	cvx-down	$(1/e)\text{OPT}-\epsilon$	$O(1/\epsilon)$
[28]	det.	mon. DR-sub.	convex	$(1/2)\text{OPT}-\epsilon$	$O(1/\epsilon)$
[28]	stoch.	mon. DR-sub.	convex	$(1/2)\text{OPT}-\epsilon$	$O(1/\epsilon^2)$
[42]	stoch.	mon. DR-sub.	convex	$(1-1/e)\text{OPT}-\epsilon$	$O(1/\epsilon^3)$
[42]	stoch.	non-mon. DR-sub.	convex	$(1/e)\text{OPT}-\epsilon$	$O(1/\epsilon^3)$
This paper	stoch.	mon. DR-sub.	convex	$(1-1/e)\text{OPT}-\epsilon$	$\mathcal{O}(1/\epsilon^2)$
This paper	stoch.	non-mon. DR-sub.	convex	$(1/e)\text{OPT}-\epsilon$	$\mathcal{O}(1/\epsilon^2)$

Table 1.3
Convergence guarantees of conditional gradient (FW) methods for nonconvex minimization.

Ref.	Setting	Assumptions	Batch	#iter	Complexity	Nonobl.
[36]	det.	smooth	—	$\mathcal{O}(1/\epsilon^2)$	—	✗
[30]	stoch.	smooth, bounded var.	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	✗
[30]	stoch.	smooth, bounded var.	$\mathcal{O}(1/\epsilon^{4/3})$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^{10/3})$	✗
[55, 62]	stoch.	smooth, bounded var.	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	✗
[64]	stoch.	smooth, bounded var.	$\mathcal{O}(1)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	✓
This paper	stoch.	smooth, bounded var.	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	✓

$[(1-1/e)\text{OPT}-\epsilon]$ with $O(1/\epsilon)$ gradient evaluations. However, the problem becomes considerably more challenging when we only have access to a *stochastic* first order oracle. In particular, [28] showed that the stochastic gradient ascent achieves $[(1/2)\text{OPT}-\epsilon]$ by using $O(1/\epsilon^2)$ stochastic gradients. In contrast, [41] proposed the stochastic variant of the continuous greedy algorithm that achieves $[(1-1/e)\text{OPT}-\epsilon]$ by using $O(1/\epsilon^3)$ stochastic gradients. In this paper, we show that SCG++ achieves $[(1-1/e)\text{OPT}-\epsilon]$ by $O(1/\epsilon^2)$ stochastic gradient evaluations. We also show that the convergence rate of $O(1/\epsilon^2)$ is optimal. We further generalize our result to the nonmonotone DR-continuous submodular setting by proposing the stochastic variant of measured continuous greedy [20]. Specifically, SMCG++ achieves a $[(1/e)\text{OPT}-\epsilon]$ solution by using $\mathcal{O}(1/\epsilon^2)$ stochastic gradients. Note that for the nonmonotone DR-submodular maximization (in contrast to the monotone case), one needs the extra assumption that the set $\mathcal{C}$ is down-closed, or otherwise no constant factor approximation in polynomial time is possible.

Nonconvex minimization. The focus of this paper is on constrained optimization in the nonoblivious stochastic setting. Nevertheless, convergence to FOSPs for nonconvex functions has been widely studied in the unconstrained case for oblivious problems [2, 40, 50, 51]. Recently, the finite-time analysis for convergence to an FOSP of *constrained* problems has also received a lot of attention. In particular, in the deterministic setting, [36] showed that the sequence of iterates generated by the update of Frank–Wolfe converges to an ϵ-FOSP after $\mathcal{O}(1/\epsilon^2)$ iterations. In contrast, [25] considered the norm of gradient mapping as a measure of nonstationarity and showed that the projected gradient method has the same complexity of $\mathcal{O}(1/\epsilon^2)$. Similar results for the accelerated projected gradient method were shown in [24]. Note that ϵ-FOSPs and the norm of gradient mapping are not directly related to one another. Adaptive cubic regularization methods in [11, 12, 13] improved these results by using second order information in order to obtain an ϵ-FOSP after $\mathcal{O}(1/\epsilon^{3/2})$ iterations.

Later, [43] showed that projected gradient descent reaches an ϵ-FOSP after $\mathcal{O}(1/\epsilon^2)$ iterations in a deterministic setting in terms of the Frank–Wolfe gap. In the oblivious stochastic setting, [50] introduced a stochastic variant of Frank–Wolfe which finds an ϵ-FOSP after $\mathcal{O}(1/\epsilon^4)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon^2)$ calls to a linear minimization oracle. The authors of [49] introduced a variant of the gradient sliding method that finds an ϵ-FOSP after $\mathcal{O}(1/\epsilon^4)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon^2)$ calls to a linear minimization oracle when we measure first order optimality in terms of proximal gradient mapping. Again, this result cannot be compared with ours, as we measure first order optimality based on the Frank–Wolfe gap.

Concurrent work. In this part, we briefly discuss some recent results from concurrent works that appeared after we made the first version of this paper publicly available on arXiv. In particular, [55, 62] considered the oblivious stochastic setting (a special case of problem (1)) and introduced a variance reduced version of the Frank–Wolfe method based on the Stochastic Path-Integrated Differential Estimator (SPIDER) approach [18]. Their proposed methods find an ϵ-FOSP in nonconvex minimization after $\mathcal{O}(1/\epsilon^3)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon^2)$ calls to a linear minimization oracle. The authors of [62] also noted that in the oblivious stochastic convex minimization, the same method achieves an ϵ-approximate solution after $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon)$ calls to a linear minimization oracle. Finally, [63] proposed a quantized Frank–Wolfe algorithm, by relying on SPIDER, to develop a communication-efficient distributed method.

2. Preliminaries.

In this section, we state some of the required definitions and then review variance reduced methods for stochastic optimization.

DEFINITION 2.1. *A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if it has L-Lipschitz continuous gradients on $\mathbb{R}^n$; i.e., for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^n$, we have $||\nabla\phi(\mathbf{x}) - \nabla\phi(\hat{\mathbf{x}})|| \leq L||\mathbf{x} - \hat{\mathbf{x}}||$.*

DEFINITION 2.2. *A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is convex on $\mathbb{R}^n$ if we have $\phi(\hat{\mathbf{x}}) \geq \phi(\mathbf{x}) + \nabla\phi(\mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})$ for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^n$. Further, $\phi(\mathbf{x})$ is concave if $-\phi(\mathbf{x})$ is convex.*

Submodularity. A set function $f : 2^V \to \mathbb{R}_+$, defined on the ground set V, is submodular if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all subsets $A, B \subseteq V$. Even though submodularity is mostly considered on discrete domains, the notion can be naturally extended to arbitrary lattices [22]. To this aim, let us consider a subset of $\mathbb{R}^d_+$ of the form $\mathcal{X} = \prod_{i=1}^d \mathcal{X}_i$, where each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}_+$. A function $F : \mathcal{X} \to \mathbb{R}_+$ is *continuous submodular* if, for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X}$, we have $F(\mathbf{x}) + F(\mathbf{y}) \geq F(\mathbf{x} \vee \mathbf{y}) + F(x \wedge \mathbf{y})$, where $\mathbf{x} \vee \mathbf{y} \doteq \max(\mathbf{x}, \mathbf{y})$ (componentwise) and $\mathbf{x} \wedge \mathbf{y} \doteq \min(\mathbf{x}, \mathbf{y})$ (componentwise). A submodular function is monotone if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$, we have $F(\mathbf{x}) \leq F(\mathbf{y})$ (here, by $\mathbf{x} \leq \mathbf{y}$ we mean that every coordinate of $\mathbf{x}$ is less than the corresponding coordinate of $\mathbf{y}$). When twice differentiable, F is submodular if and only if all cross-second-derivatives are nonpositive [3], i.e., $\forall i \neq j, \forall \mathbf{x} \in \mathcal{X}, \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j} \leq 0$. This expression makes it clear that continuous submodular functions are neither convex nor concave in general, as concavity (convexity) implies $\nabla^2 F \preceq 0$ ($\nabla^2 F \succeq 0$). A proper subclass of submodular functions is called *DR-submodular* [5, 57] if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$ and any standard basis vector $\mathbf{e}_i \in \mathbb{R}^n$ and a nonnegative number $z \in \mathbb{R}_+$ such that $z\mathbf{e}_i + \mathbf{x} \in \mathcal{X}$ and $z\mathbf{e}_i + \mathbf{y} \in \mathcal{X}$, then $F(z\mathbf{e}_i + \mathbf{x}) - F(\mathbf{x}) \geq F(z\mathbf{e}_i + \mathbf{y}) - F(\mathbf{y})$. One can easily verify that for a differentiable DR-submodular function the gradient is an antitone mapping, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$ we have $\nabla F(\mathbf{x}) \geq \nabla F(\mathbf{y})$ [5]. A crucial example of a DR-submodular function is the multilinear extension [9] that we study in section 5.

Variance reduction. Beyond the vanilla stochastic gradient, variance reduced algorithms [1, 17, 32, 47, 50, 54] have been successful in reducing stochastic first order oracle complexity in the *oblivious* stochastic optimization

$$\max_{\mathbf{x}\in\mathcal{C}} F(\mathbf{x}) \;:=\; \max_{\mathbf{x}\in\mathcal{C}}\; \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\tilde{F}(\mathbf{x};\mathbf{z})], \tag{5}$$

where each function $\tilde{F}(\cdot;\mathbf{z})$ is L-*smooth*. In contrast to (1), the underlying distribution p of (5) is invariant to the variable $\mathbf{x}$ and is hence called oblivious. We will now explain a recent variance reduction technique for solving (5) using stochastic gradient information. Consider the following *unbiased* estimate of the gradient at $\mathbf{x}^t$:

$$\mathbf{g}^t \stackrel{\text{def}}{=} \mathbf{g}^{t-1} + \nabla\tilde{F}(\mathbf{x}^t;\mathcal{M}) - \nabla\tilde{F}(\mathbf{x}^{t-1};\mathcal{M}), \tag{6}$$

where $\nabla\tilde{F}(\mathbf{y};\mathcal{M}) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{M}|}\sum_{\mathbf{z}\in\mathcal{M}}\nabla\tilde{F}(\mathbf{y};\mathbf{z})$ for some $\mathbf{y}\in\mathbb{R}^d$, $\mathbf{g}^{t-1}$ is an unbiased gradient estimator at $\mathbf{x}^{t-1}$, and $\mathcal{M}$ is a minibatch of random samples drawn from $p(\mathbf{z})$. The authors of [18] showed that, with the gradient estimator (6), $\mathcal{O}(1/\epsilon^3)$ stochastic gradient evaluations are sufficient to find an ϵ-FOSP of problem (5), improving upon the $\mathcal{O}(1/\epsilon^4)$ complexity of SGD. A crucial property leading to the success of the variance reduction method given in (6) is that $\nabla\tilde{F}(\mathbf{x}^t;\mathcal{M})$ and $\nabla\tilde{F}(\mathbf{x}^{t-1};\mathcal{M})$ use *the same* minibatch sample $\mathcal{M}$ in order to exploit the L-smoothness of component functions $\tilde{F}(\cdot;\mathbf{z})$. Such a construction is only possible in the oblivious setting where $p(\mathbf{z})$ is independent of the choice of $\mathbf{x}$; it would introduce bias in the more general nonoblivious case (1): To see this point, let $\mathcal{M}$ be the minibatch of random variable $\mathbf{z}$ sampled according to distribution $p(\mathbf{z};\mathbf{x}^t)$. We have $\mathbb{E}[\nabla\tilde{F}(\mathbf{x}^t;\mathcal{M})] = \nabla F(\mathbf{x}^t)$ but $\mathbb{E}[\nabla\tilde{F}(\mathbf{x}^{t-1};\mathcal{M})] \neq \nabla F(\mathbf{x}^{t-1})$ since the distribution $p(\mathbf{z};\mathbf{x}^{t-1})$ is not the same as $p(\mathbf{z};\mathbf{x}^t)$. The same argument renders all the existing variance reduction techniques inapplicable to the nonoblivious setting of problem (1).

3. Stochastic (non)convex minimization.

In this section, we focus on two specific cases of problem (1): the case where the objective function F is a concave function and the case where it is a general nonconcave function. As stated earlier, maximizing a (non)concave function can be written as minimizing a (non)convex function. We hence rewrite (1) as

$$\min_{\mathbf{x}\in\mathcal{C}} F(\mathbf{x}) \;:=\; \min_{\mathbf{x}\in\mathcal{C}}\; \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{F}(\mathbf{x};\mathbf{z})], \tag{7}$$

where we assume that the expected function F is either convex or a general nonconvex function. In this section we study a nonoblivious stochastic optimization problem, but our results trivially hold for the oblivious stochastic problem as a special case.

3.1. Stochastic Frank–Wolfe++.

Now we introduce the `Stochastic Frank-Wolfe++` method (SFW++) to solve the nonoblivious minimization problem (7). Recall that the Frank–Wolfe method requires access to the gradient of the objective function ∇F. However, evaluation of ∇F may not be possible in many settings as either the probability distribution p is not available or evaluating the expectation in (7) is computationally prohibitive. Our goal is to design an unbiased estimator $\mathbf{g}$ for approximating the exact gradient ∇F that is computationally affordable and has a low variance. Once the gradient approximation $\mathbf{g}^t$ for step t is evaluated, we can find the descent direction $\mathbf{v}^t$ by solving the linear optimization program

$$\mathbf{v}^t := \operatorname*{argmin}_{\mathbf{v}\in\mathcal{C}}\{\mathbf{v}^\top\mathbf{g}^t\}, \tag{8}$$

Algorithm 3.1 `Stochastic Frank--Wolfe++` (SFW++)

Input: Number of iteration T, minibatch size $|\mathcal{M}_0^t|$ and $|\mathcal{M}_h^t|$, step size η_t

for $t = 0$ **to** T **do**
  if $\mod(t, q) = 0$ (nonconcave) or $\log_2 t \in \mathbb{Z}$ (concave) **then**
    Sample $\mathcal{M}_0^t$ of $\mathbf{z}$ with distribution $p(\mathbf{z}; \mathbf{x}^t)$ to find $\mathbf{g}^t \stackrel{\text{def}}{=} \nabla F(\mathbf{x}^t; \mathcal{M}_0^t)$;
  else
    Sample a minibatch $\mathcal{M}_h^t$ of $(a, \mathbf{z}(a))$ to calculate $\tilde{\nabla}_t^2$ using (14)
    Compute $\mathbf{g}^t := \mathbf{g}^{t-1} + \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})$; $\diamond$ Or use (22) instead.
  end if
  $\mathbf{v}^t := \operatorname{argmax}_{\mathbf{v}\in\mathcal{C}}\{\mathbf{v}^\top \mathbf{g}^t\}$;
  $\mathbf{x}^{t+1} := \mathbf{x}^t + \eta_t \cdot (\mathbf{v}^t - \mathbf{x}^t)$;
end for

Output: $\mathbf{x}^{\bar{t}}$ with $\bar{t}$ uniformly sampled from $[T]$ (nonconcave case); $\mathbf{x}^T$ (concave case).

and then compute the updated variable $\mathbf{x}^{t+1}$ by performing the following update:

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

where η_t is a properly chosen stepsize. The steps of the SFW++ are summarized in Algorithm 3.1. In SFW++ we restart the gradient estimation $\mathbf{g}$ after an episode of iterates with a certain length. This step is necessary to ensure that the noise of gradient approximation stays bounded by a proper constant. The details for computing the gradient approximation $\mathbf{g}^t$ are provided in the following section.

3.1.1. Stochastic gradient approximation. Given a sequence of iterates, $\{\mathbf{x}^s\}_{s=0}^t$, the gradient of F at $\mathbf{x}^t$ can be written in a path-integral form as follows:

$$\nabla F(\mathbf{x}^t) = \nabla F(\mathbf{x}^0) + \sum_{s=1}^{t} \left\{ \Delta^s \stackrel{\text{def}}{=} \nabla F(\mathbf{x}^s) - \nabla F(\mathbf{x}^{s-1}) \right\}. \tag{10}$$

By obtaining an unbiased estimate of $\Delta^t = \nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1})$ and reusing the previous unbiased estimates for $s < t$, we obtain recursively an unbiased estimator of $\nabla F(\mathbf{x}^t)$ which has a reduced variance. Estimating $\nabla F(\mathbf{x}^s)$ and $\nabla F(\mathbf{x}^{s-1})$ separately as suggested in (6) would cause the bias issue in the the nonoblivious case (see the discussion at the end of section 2). Therefore, we propose an approach for *directly estimating the difference* $\Delta^t = \nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1})$ in an unbiased manner.

We construct an unbiased estimator $\mathbf{g}^t$ of the gradient $\nabla F(\mathbf{x}^t)$ by adding an unbiased estimate $\tilde{\Delta}^t$ of the gradient difference $\Delta^t = \nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1})$ to $\mathbf{g}^{t-1}$, where $\mathbf{g}^{t-1}$ is an unbiased estimator of $\nabla F(\mathbf{x}^{t-1})$. Note that Δ^t can be written as

$$\Delta^t = \int_0^1 \nabla^2 F(\mathbf{x}(a))(\mathbf{x}^t - \mathbf{x}^{t-1})\mathbf{d}a = \left[\int_0^1 \nabla^2 F(\mathbf{x}(a))\mathbf{d}a\right](\mathbf{x}^t - \mathbf{x}^{t-1}), \tag{11}$$

where $\mathbf{x}(a) \stackrel{\text{def}}{=} a \cdot \mathbf{x}^t + (1-a)\cdot \mathbf{x}^{t-1}$ for $a \in [0, 1]$. Hence, if we sample the parameter a uniformly at random from the interval $[0, 1]$, it can be easily verified that $\tilde{\Delta}^t := \nabla^2 F(\mathbf{x}(a))(\mathbf{x}^t - \mathbf{x}^{t-1})$ is an unbiased estimator of the gradient difference Δ^t since

$$\mathbb{E}_a[\nabla^2 F(\mathbf{x}(a))(\mathbf{x}^t - \mathbf{x}^{t-1})] = \nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}). \tag{12}$$

Hence, all we need is an unbiased estimator of the Hessian-vector product $\nabla^2 F(\mathbf{y})(\mathbf{x}^t - \mathbf{x}^{t-1})$ for the nonoblivious objective F at an arbitrary $\mathbf{y} \in \mathcal{C}$. Next, we present an unbiased estimator of $\nabla^2 F(\mathbf{y})$ for any $\mathbf{y} \in \mathcal{C}$ that can be evaluated efficiently.

LEMMA 3.1. *Consider $\mathbf{y} \in \mathcal{C}$, and $\mathbf{z}$ with distribution $p(\mathbf{z};\mathbf{y})$, and define*

$$\begin{aligned}\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z}) \stackrel{\text{def}}{=} &\tilde{F}(\mathbf{y};\mathbf{z})[\nabla \log p(\mathbf{z};\mathbf{y})][\nabla \log p(\mathbf{z};\mathbf{y})]^\top + [\nabla \tilde{F}(\mathbf{x};\mathbf{z})][\nabla \log p(\mathbf{z};\mathbf{y})]^\top \\ &+ [\nabla \log p(\mathbf{z};\mathbf{y})][\nabla \tilde{F}(\mathbf{y};\mathbf{z})]^\top + \nabla^2 \tilde{F}(\mathbf{y};\mathbf{z}) + \tilde{F}(\mathbf{y};\mathbf{z})\nabla^2 \log p(\mathbf{z};\mathbf{y}).\end{aligned} \tag{13}$$

Then, $\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})$ is an unbiased estimator of $\nabla^2 F(\mathbf{y})$.

The result in Lemma 3.1 shows how to evaluate an unbiased estimator of the Hessian $\nabla^2 F(\mathbf{y})$. If we consider a as a random variable with a uniform distribution over the interval $[0,1]$, then we can define the random variable $\mathbf{z}(a)$ with the probability distribution $p(\mathbf{z}(a);\mathbf{x}(a))$, where $\mathbf{x}(a)$ is defined as $\mathbf{x}(a) := a \cdot \mathbf{x}^t + (1-a)\cdot \mathbf{x}^{t-1}$. Considering these two random variables and the result of Lemma 3.1, we can construct an unbiased estimator of the integral $\int_0^1 \nabla^2 F(\mathbf{x}(a))\mathbf{d}a$ in (11) by

$$\tilde{\nabla}_t^2 \stackrel{\text{def}}{=} \frac{1}{|\mathcal{M}|} \sum_{(a,\mathbf{z}(a))\in\mathcal{M}} \tilde{\nabla}^2 F(\mathbf{x}(a);\mathbf{z}(a)), \tag{14}$$

where $\mathcal{M}$ is a minibatch containing $|\mathcal{M}|$ samples of random tuple $(a,\mathbf{z}(a))$. Once we construct $\tilde{\nabla}_t^2$, the gradient difference Δ^t can be approximated by

$$\tilde{\Delta}^t := \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}). \tag{15}$$

Note that for the general objective $F(\cdot)$, the matrix-vector product $\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})$ requires $\mathcal{O}(d^2)$ computation and memory. To resolve this issue, in section 3.1.2, we provide an implementation of (15) using only first order information which reduces the computational and memory complexity to $\mathcal{O}(d)$. Using $\tilde{\Delta}^t$ as an unbiased estimator for the gradient difference Δ^t, we can define our gradient estimator as

$$\mathbf{g}^t = \nabla \tilde{F}(\mathbf{x}^0;\mathcal{M}_0) + \sum_{i=1}^{t} \tilde{\Delta}^t. \tag{16}$$

Indeed, this update can also be rewritten in a recursive way as

$$\mathbf{g}^t = \mathbf{g}^{t-1} + \tilde{\Delta}^t \tag{17}$$

once we set $\mathbf{g}^0 = \nabla \tilde{F}(\mathbf{x}^0;\mathcal{M}_0)$. Note that the proposed approach for gradient approximation in (16) has a variance reduction mechanism which leads to the optimal computational complexity of SFW++ in terms of number of calls to the stochastic oracle. We further highlight this point in the convergence analysis of SFW++.

3.1.2. Implementation of the Hessian-vector product. In this section, we focus on the computation of the gradient difference approximation $\tilde{\Delta}^t$ introduced in (15). We aim to come up with a scheme that avoids explicitly computing the matrix estimator $\tilde{\nabla}_t^2$ which has a complexity of $\mathcal{O}(d^2)$, and present an approach that directly approximates $\tilde{\Delta}^t$ while only using the finite differences of gradients with a complexity of $\mathcal{O}(d)$. Recall the definition of the Hessian approximation $\tilde{\nabla}_t^2$ in (14). Computing $\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})$ is equivalent to computing $|\mathcal{M}|$ instances of $\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})(\mathbf{x}^t - \mathbf{x}^{t-1})$ for some $\mathbf{y} \in \mathcal{C}$ and $\mathbf{z} \in \mathcal{Z}$. Denote $\mathbf{d} = \mathbf{x}^t - \mathbf{x}^{t-1}$ and use the expression in (13) to write

$$\begin{aligned}\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})\mathbf{d} = &\tilde{F}(\mathbf{y};\mathbf{z})[\nabla \log p(\mathbf{z};\mathbf{y})^\top \mathbf{d}]\nabla \log p(\mathbf{z};\mathbf{y}) + [\nabla \log p(\mathbf{z};\mathbf{y})^\top \mathbf{d}]\nabla \tilde{F}(\mathbf{x};\mathbf{z}) \\ &+ [\nabla \tilde{F}(\mathbf{y};\mathbf{z})^\top \mathbf{d}][\nabla \log p(\mathbf{z};\mathbf{y})] + \nabla^2 \tilde{F}(\mathbf{y};\mathbf{z})\mathbf{d} + \tilde{F}(\mathbf{y};\mathbf{z})\nabla^2 \log p(\mathbf{z};\mathbf{y})\mathbf{d}.\end{aligned} \tag{18}$$

The first three terms can be computed[1] in time $\mathcal{O}(d)$ and only the last two terms involve $\mathcal{O}(d^2)$ operations, which can be approximated by the following finite gradient

[1] Also, note that pairs $(\mathbf{x},\mathbf{y})$ with $p(\mathbf{x},\mathbf{y}) = 0$ will never be sampled (i.e., they are of measure zero) and hence do not cause any computational issues.

difference scheme. For any twice differentiable function $\psi : \mathbb{R}^d \to \mathbb{R}$ and any $\mathbf{d} \in \mathbb{R}^d$ with bounded norm $\|\mathbf{d}\| \le D$, we compute, for some small $\delta > 0$,

$$\phi(\delta;\psi) \stackrel{\text{def}}{=} \frac{\nabla\psi(\mathbf{y}+\delta\cdot\mathbf{d}) - \nabla\psi(\mathbf{y}-\delta\cdot\mathbf{d})}{2\delta} \simeq \nabla^2\psi(\mathbf{y})\mathbf{d}. \tag{19}$$

By considering the second order smoothness of the function $\psi(\cdot)$ with constant L_2 we can show that for arbitrary $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$ it holds that $\|\nabla^2\psi(\mathbf{x}) - \nabla^2\psi(\mathbf{y})\| \le L_2\|\mathbf{x}-\mathbf{y}\|$. Therefore, the error of the above approximation can be bounded by

$$\|\nabla^2\psi(\mathbf{y})\mathbf{d} - \phi(\delta;\psi)\| = \|\nabla^2\psi(\mathbf{y})\mathbf{d} - \nabla^2\psi(\tilde{\mathbf{x}})\mathbf{d}\| \le D^2 L_2 \delta, \tag{20}$$

where $\tilde{\mathbf{x}}$ is obtained from the mean value theorem. This quantity can be made arbitrarily small by decreasing δ (up to machine accuracy). By applying (19) to functions $\psi(\mathbf{y}) = \tilde{F}(\mathbf{y};\mathbf{z})$ and $\psi(\mathbf{y}) = \log p(\mathbf{z};\mathbf{y})$, we can approximate (18) in $\mathcal{O}(d)$ as

$$\begin{aligned} \xi_\delta(\mathbf{y};\mathbf{z}) \stackrel{\text{def}}{=} & \tilde{F}(\mathbf{y};\mathbf{z})[\nabla\log p(\mathbf{z};\mathbf{y})^\top\mathbf{d}]\nabla\log p(\mathbf{z};\mathbf{y}) + [\nabla\log p(\mathbf{z};\mathbf{y})^\top\mathbf{d}]\nabla\tilde{F}(\mathbf{x};\mathbf{z}) \\ & + [\nabla\tilde{F}(\mathbf{y};\mathbf{z})^\top\mathbf{d}][\nabla\log p(\mathbf{z};\mathbf{y})] + \phi(\delta;\tilde{F}(\mathbf{y};\mathbf{z})) + \phi(\delta;\log p(\mathbf{z};\mathbf{y})). \end{aligned} \tag{21}$$

We further can define a minibatch version of this implementation as

$$\xi_\delta(\mathbf{x};\mathcal{M}) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{M}|}\sum_{(a,\mathbf{z}(a))\in\mathcal{M}} \xi_\delta(\mathbf{x}(a);\mathbf{z}(a)), \tag{22}$$

which is used in Option II of Step 8 in Algorithm 3.1. Note that $\lim_{\delta\to 0}\xi_\delta(\mathbf{x};\mathcal{M}) = \tilde{\Delta}^t$ and hence (15) is a special case of (22) by taking $\delta \to 0$. Additionally, we show in later sections that setting $\delta = \mathcal{O}(\epsilon^2)$ is sufficient, where ϵ is the target accuracy.

3.2. Convergence analysis of SFW++: Nonconvex setting. In this section, we focus on solving problem (7) when F is smooth but nonconvex. In this case, our goal is to find an ϵ-FOSP, formally defined as

$$V_{\mathcal{C}}(\mathbf{x};F) = \max_{\mathbf{u}\in\mathcal{C}}\langle\nabla F(\mathbf{x}), \mathbf{u}-\mathbf{x}\rangle \le \epsilon, \tag{23}$$

where the parameter $V_{\mathcal{C}}(\mathbf{x};F)$ captures distance to an FOSP and is 0 when $\mathbf{x}$ is an FOSP. The parameter $V_{\mathcal{C}}(\mathbf{x};F)$ is also known as the Frank–Wolfe gap [36].

Before stating our main theorem for the general nonconvex case, we first formally state the required assumption for proving our results.

Assumption 1. *The stochastic function* $\tilde{F}(\mathbf{x};\mathbf{z})$ *has bounded value for all* $\mathbf{z}\in\mathcal{Z}$ *and* $\mathbf{x}\in\mathcal{C}$, *i.e.,* $\exists B$ *s.t.* $\max_{\mathbf{z}\in\mathcal{Z},\mathbf{x}\in\mathcal{C}}\tilde{F}(\mathbf{x};\mathbf{z}) \le B$.

Assumption 2. *The set* $\mathcal{C}$ *is compact with diameter* D.

Assumption 3. *Stochastic gradient* $\nabla\tilde{F}$ *has bounded norm:* $\forall\mathbf{z}\in\mathcal{Z}, \|\nabla\tilde{F}(\mathbf{x};\mathbf{z})\| \le G_{\tilde{F}}$, *and the norm of the gradient of* $\log p$ *has bounded fourth order moment, i.e.,* $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{x};\mathbf{z})}\|\nabla\log p(\mathbf{z};\mathbf{x})\|^4 \le G_p^4$. *Further, we define* $G = \max\{G_{\tilde{F}}, G_p\}$.

Assumption 4. *For all* $\mathbf{x}\in\mathcal{C}$, *the stochastic Hessian of* $\nabla^2\tilde{F}$ *has bounded spectral norm* $\forall\mathbf{z}\in\mathcal{Z}, \|\nabla^2\tilde{F}(\mathbf{x};\mathbf{z})\| \le L_{\tilde{F}}$, *and the spectral norm of the Hessian of the log-probability function has bounded second order moment:* $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\|\nabla^2\log p(\mathbf{z};\mathbf{x})\|^2] \le L_p^2$. *Further, we define* $L = \max\{L_{\tilde{F}}, L_p\}$.

Assumption 5. *The stochastic Hessian is* $L_{2,f}$*-Lipschitz continuous, i.e.,* $\forall\mathbf{x},\mathbf{y}\in\mathcal{C}$ *and all* $\mathbf{z}\in\mathcal{Z}$, $\|\nabla^2\tilde{F}(\mathbf{x};\mathbf{z}) - \nabla^2\tilde{F}(\mathbf{y};\mathbf{z})\| \le L_{2,\tilde{F}}\|\mathbf{x}-\mathbf{y}\|$. *The Hessian of the log probability* $\log p(\mathbf{x};\mathbf{z})$ *is* $L_{2,p}$*-Lipschitz continuous:* $\forall\mathbf{x},\mathbf{y}\in\mathcal{C}$ *and all* $\mathbf{z}\in\mathcal{Z}$, *i.e.,* $\|\nabla^2\log p(\mathbf{x};\mathbf{z}) - \nabla^2\log p(\mathbf{y};\mathbf{z})\| \le L_{2,p}\|\mathbf{x}-\mathbf{y}\|$. *Also, we define* $L_2 = \max\{L_{2,\tilde{F}}, L_{2,p}\}$.

Remark 3.2. We note that high order smoothness leads to faster convergence rates for gradient descent type algorithms. Concretely, for *unconstrained* nonconvex problems, it is known that by assuming higher order smoothness (e.g., the Lipschitz continuity of the Hessian), one can obtain a faster convergence rate by exploiting only the first order gradient algorithmically, e.g., [10] for the deterministic case and [19] for the SGD case (whether SPIDER has a better rate with Lipschitz continuous Hessian is not known). However, in these results, the convergence rate *explicitly* depends on the higher order smoothness parameter. In the constrained case, whether higher order smoothness helps remains unknown. On the other hand, in our results, the higher order smoothness parameters $L_{2,\tilde{F}}$ and $L_{2,p}$ *do not* enter the convergence results and are only required to evaluate the Hessian vector product when we have finite machine accuracy. In fact, if we have the exact expressions of $\tilde{F}$ and $\log p$, Assumption 5 can be avoided by using the autodifferential mechanism of Pytorch.

Next, we formally bound the variance of gradient approximation for SFW++.

LEMMA 3.3. *Consider the SFW++ method outlined in Algorithm* 3.1 *and assume that in Step* 8 *we follow the update in* (22) *to construct the gradient difference approximation* $\tilde{\Delta}^t$ *(Option* II*). If Assumptions* 1*,* 2*,* 3*,* 4*, and* 5 *hold and we set the minibatch sizes to* $|\mathcal{M}_0| = (G^2/(\bar{L}^2D^2\epsilon^2))$ *and* $|\mathcal{M}| = 2/\epsilon$*, and the error of Hessian-vector product approximation* δ *is* $\mathcal{O}(\epsilon^2)$ *as in* (41)*, then*

$$\mathbb{E}\left[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2\right] \le (1+\epsilon t)\bar{L}^2D^2\epsilon^2 \quad \forall t \in \{0,\dots,T-1\}, \tag{24}$$

where $\bar{L}$ *is a constant defined as* $\bar{L}^2 \stackrel{\text{def}}{=} 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2$.

The result in Lemma 3.3 shows that by $|\mathcal{M}| = \mathcal{O}(\epsilon^{-1})$ calls to the stochastic oracle at each iteration, the variance of gradient approximation in SFW++ after t iterations is on the order of $\mathcal{O}((1+\epsilon t)\epsilon)$. In the following theorem, we use this result to characterize the convergence properties of our proposed SFW++ method for solving stochastic nonconvex minimization problems. For simplicity, we analyze the convergence of gradient-difference estimator (15). However, similar results can be obtained for the Hessian-vector product estimator (22) by setting $\delta = \mathcal{O}(\epsilon^2)$.

THEOREM 3.4. *Consider problem* (7) *when* F *is a general nonconvex function. Further, recall the SFW++ method outlined in Algorithm* 3.1*. Suppose the conditions in Assumptions* 1*,* 2*,* 3*,* 4*, and* 5 *are satisfied. Further, let* $\bar{L}^2 \stackrel{\text{def}}{=} 4B^2G^4+16G^4+4L^2+4B^2L^2$*. If we set SFW++ parameters to* $\eta_t = \epsilon/(\bar{L}D)$*,* $|\mathcal{M}_h^t| = 2G/\epsilon$*,* $q = \lceil G/(16\epsilon)\rceil$*, and* $|\mathcal{M}_0^t| = G^2/(8\epsilon^2)$*, then the iterates generated by SFW++ satisfy the condition* $\mathbb{E}\left[V_{\mathcal{C}}(\mathbf{x}^t;F)\right] \le 5\epsilon D$*, where the total number of iterations is* $T = \bar{L}(F(\mathbf{x}^*)-F(\mathbf{x}^0))/\epsilon^2$.

This theorem shows that after at most $\mathcal{O}(1/\epsilon^2)$ iterations, SFW++ reaches an ϵ-FOSP. To characterize the overall complexity, we take into account the number of stochastic gradient evaluations per iteration in the following corollary.

COROLLARY 3.5 (oracle complexity for nonconcave case). *Assume that the target accuracy* ϵ *satisfies* mod$(T,q)=0$*. The overall stochastic complexity is*

$$\sum_{i=0}^{T}|\mathcal{M}_h^i| + \sum_{k=0}^{T/q}|\mathcal{M}_0^{qk}| = \mathcal{O}\left(\bar{L}G(F(\mathbf{x}^*)-F(\mathbf{x}^0))/\epsilon^3\right). \tag{25}$$

According to Corollary 3.5, SFW++ finds an ϵ-FOSP for general stochastic nonconcave problems after at most $\mathcal{O}(1/\epsilon^3)$ stochastic gradient evaluations.

Remark 3.6. The results in Theorem 3.4 and Corollary 3.5 hold for the general nonoblivious problem (1). Indeed, such complexity bounds also hold for the oblivious setting, and the proof follows similarly (requiring fewer assumptions). More precisely, to prove the same theoretical guarantees for the oblivious case, we only require Assumptions 2 and 4 and the boundedness of variance $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\|\tilde{F}(\mathbf{x};\mathbf{z}) - F(\mathbf{x})\|^2 \leq G^2$.

3.3. Convergence analysis of SFW++: Convex setting. In this section, we establish the complexity of SFW++ for finding an ϵ-approximate solution when the function F in (7) is convex or equivalently the function F in (1) is concave.

THEOREM 3.7. *Consider problem* (1) *when F is a concave function. Further, recall the SFW++ method outlined in Algorithm* 3.1. *Suppose the conditions in Assumptions* 1, 2, 3, 4, *and* 5 *are satisfied. Further, let* $\bar{L}^2 \overset{\text{def}}{=} 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2$. *If we set SFW++ parameters to* $\eta_t = 2/(t+2)$, $|\mathcal{M}_h^t| = 16(t+2)$, *and* $|\mathcal{M}_0^t| = (G^2(t+1)^2)/(\bar{L}^2D^2)$, *then the iterates generated by SFW++ satisfy*

$$F(\mathbf{x}^*) - \mathbb{E}[F(\mathbf{x}^t)] \leq \frac{28\bar{L}D^2 + (F(\mathbf{x}^*) - F(\mathbf{x}^0))}{t+2}.$$

Theorem 3.7 shows that after at most $\mathcal{O}(1/\epsilon)$ iterations and $\mathcal{O}(1/\epsilon)$ calls to a linear minimization oracle SFW++ reaches an ϵ-approximate solution. Next we characterize the overall complexity of SFW++ in terms of stochastic gradient evaluations.

COROLLARY 3.8. *Assume that ϵ satisfies* $t = (28LD^2 + (F(\mathbf{x}^*) - F(\mathbf{x}^0)))/\epsilon = 2^K$ *for some* $K \in \mathbb{N}$. *Then, the overall stochastic complexity is*

$$\sum_{i=0}^{t} |\mathcal{M}_h^i| + \sum_{k=0}^{K} |\mathcal{M}_0^{2^k}| = \mathcal{O}\left(\frac{\bar{L}^2D^4}{\epsilon^2} + \frac{G^2D^2}{\epsilon^2} + \frac{G^2(F(\mathbf{x}^*) - F(\mathbf{x}^0))^2}{\bar{L}^2D^2\epsilon^2}\right). \tag{26}$$

According to Corollary 3.8, SFW++ finds an ϵ-approximate solution for stochastic concave maximization (equivalently convex minimization) after at most computing $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations.

Remark 3.9. The results in Theorem 3.7 and Corollary 3.8 hold for the general nonoblivious problem in (1) when the objective function F is concave. Indeed, such complexity bounds also hold for the oblivious setting, and the proof follows similarly (requiring fewer assumptions). More precisely, the same theoretical guarantees as in Remark 3.6 hold for the oblivious case under Assumptions 2 and 4 and the bounded variance assumption $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\|\tilde{F}(\mathbf{x};\mathbf{z}) - F(\mathbf{x})\|^2 \leq G^2$.

4. Stochastic continuous DR-submodular maximization. In this section, we focus on a special case of the nonoblivious problem in (1) when the function F is continuous DR-submodular. We study both monotone and nonmonotone settings, and for each of them we present a new stochastic variant of the continuous greedy method [9] that can be interpreted as a conditional gradient method. We then extend our results to the problem of maximizing discrete submodular set functions when the objective is defined as an expectation of a collection of random set functions.

4.1. Stochastic continuous Greedy++: Monotone setting. We present `Stochastic Continuous Greedy`++ (SCG++), which is the first method to obtain a $[(1-1/e)\mathrm{OPT} - \epsilon]$ solution with $O(1/\epsilon^2)$ stochastic oracle complexity for maximizing monotone but stochastic DR-submodular functions over a compact convex body. SCG++ essentially operates in a conditional gradient manner. To be more precise,

Algorithm 4.1 `Stochastic (Measured) Continuous Greedy++`.

Input: Minibatch size $|\mathcal{M}_0|$ and $|\mathcal{M}|$, and total number of rounds T

Initialize $\mathbf{x}^0 = 0$;
for $t = 1$ **to** T **do**
  if $t = 1$ **then**
    Sample a minibatch $\mathcal{M}_0$ of $\mathbf{z}$ based on $p(\mathbf{z}; \mathbf{x}^0)$ and find $\mathbf{g}^0 \overset{\text{def}}{=} \nabla \tilde{F}(\mathbf{x}^0; \mathcal{M}_0)$;
  else
    Sample a minibatch $\mathcal{M}$ of $\mathbf{z}$ according to $p(\mathbf{z}; \mathbf{x}(a))$ where a is chosen uniformly at random from $[0, 1]$ and $\mathbf{x}(a) := a \cdot \mathbf{x}^t + (1 - a) \cdot \mathbf{x}^{t-1}$;
    Compute the Hessian approximation $\tilde{\nabla}_t^2$ corresponding to $\mathcal{M}$ based on (14);
    Construct $\tilde{\Delta}^t$ based on (15) (Option I) or (22) (Option II);
    Update the stochastic gradient approximation $\mathbf{g}^t := \mathbf{g}^{t-1} + \tilde{\Delta}^t$;
  end if
  Set feasible set $\mathcal{C}^t = \mathcal{C}$ (SCG++) or $\mathcal{C}^t = \{\mathbf{v} \in \mathcal{C} | \mathbf{v} \le \bar{\mathbf{u}} - \mathbf{x}^t\}$ (SMCG++);
  Compute the ascent direction $\mathbf{v}^t := \operatorname{argmax}_{\mathbf{v} \in \mathcal{C}^t} \{\mathbf{v}^\top \mathbf{g}^t\}$;
  Update the variable $\mathbf{x}^{t+1} := \mathbf{x}^t + 1/T \cdot \mathbf{v}^t$;
end for

at each iteration t, given a gradient estimator $\mathbf{g}^t$, SCG++ solves the subproblem

$$\mathbf{v}^t = \operatorname*{argmax}_{\mathbf{v} \in \mathcal{C}} \langle \mathbf{v}, \mathbf{g}^t \rangle \tag{27}$$

to obtain $\mathbf{v}^t$ in $\mathcal{C}$ as ascent direction, which is then added to the iterate $\mathbf{x}^{t+1}$ with a scaling factor $1/T$, i.e., the new iterate $\mathbf{x}^{t+1}$ is computed by following the update

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \frac{1}{T} \mathbf{v}^t, \tag{28}$$

where T is the total number of iterations of the algorithm. Note the difference between (28) and (9). The iterates are assumed to be initialized at the origin, which may not belong to the feasible set $\mathcal{C}$. Though each iterate $\mathbf{x}^t$ may not necessarily be in $\mathcal{C}$, the feasibility of the final iterate $\mathbf{x}^T$ is guaranteed by the convexity of $\mathcal{C}$. Note that the iterate sequence $\{\mathbf{x}^s\}_{s=0}^T$ can be regarded as a path from the origin (as we manually force $\mathbf{x}^0 = 0$) to some feasible point in $\mathcal{C}$. The key idea in SCG++ is to exploit the high correlation between the consecutive iterates originated from the $\mathcal{O}(1/T)$-sized increments to maintain a highly accurate estimate $\mathbf{g}^t$, which is evaluated based on the gradient estimation scheme presented in section 3.1. Note that by replacing the gradient approximation vector $\mathbf{g}^t$ in the update of SCG++ by the exact gradient of the objective function, we recover the update of continuous greedy [5, 9].

We proceed to analyze the convergence property of Algorithm 4.1 using (22) as the gradient-difference estimation. Similar results can be obtained by using (15). We first specify the extra assumptions required for the analysis of the SCG++ method.

Assumption 6. *The function F satisfies $F(\mathbf{0}) \ge 0$.*

Assumption 7. *F is DR-submodular.*

Assumption 8. *F is monotone.*

Next, we incorporate the bound on the noise of gradient approximation presented in Lemma 3.3 to characterize the convergence guarantee of SCG++. Note that the following result appears as Theorem 1 in [33].

THEOREM 4.1. *Consider SCG++ outlined in Algorithm* 4.1 *and assume that in step* 8 *we use the update in* (22) *to find the gradient difference approximation* $\tilde{\Delta}^t$. *If Assumptions* 1–8 *hold, then the output of SCG++ denoted by* $\mathbf{x}^T$ *satisfies*

$$\mathbb{E}\left[F(\mathbf{x}^T)\right] \geq (1-1/e)F(\mathbf{x}^*) - 2L\epsilon D^2,$$

by setting $|\mathcal{M}_0| = \frac{G^2}{2\bar{L}^2 D^2 \epsilon^2}$, $|\mathcal{M}| = \frac{1}{2\epsilon}$, $T = \frac{1}{\epsilon}$, *and* $\delta = \mathcal{O}(\epsilon^2)$. *Here* $\bar{L}$ *is a constant defined as* $\bar{L}^2 \overset{\text{def}}{=} 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2$.

The result in Theorem 4.1 shows that after at most $T = 1/\epsilon$ iterations the objective function value for the output of SCG++ is at least $(1-1/e)\mathrm{OPT} - \mathcal{O}(\epsilon)$. As the number of calls to the stochastic oracle per iteration is of $\mathcal{O}(1/\epsilon)$, to reach a $(1-1/e)\mathrm{OPT} - \mathcal{O}(\epsilon)$ approximation guarantee the SCG++ method has an overall stochastic first order oracle complexity of $\mathcal{O}(1/\epsilon^2)$. We formally characterize this result in the following corollary.

COROLLARY 4.2. *To find a* $[(1-1/e)OPT - \epsilon]$ *solution to problem* (1) *using Algorithm* 4.1 *with Option* II, *the overall stochastic first order oracle complexity is* $(2G^2D^2 + 4\bar{L}^2D^4)/\epsilon^2$ *and the overall linear optimization oracle complexity is* $2\bar{L}D^2/\epsilon$.

4.2. Stochastic continuous Greedy++: Nonmonotone setting. In this section, we consider the maximization of a nonmonotone stochastic DR-submodular function. To present our method for solving this class of problems we first need to specify the domain $\mathcal{X}$ of the expected function $F : \mathcal{X} \to \mathbb{R}_+$ which is given by $\mathcal{X} = \prod_{i=1}^d \mathcal{X}_i$ where each $\mathcal{X}_i = [\underline{u}_i, \bar{u}_i]$ is a bounded interval. To simplify the notation we define the vectors $\underline{\mathbf{u}} = [\underline{u}_1, \ldots, \underline{u}_d]$ and $\bar{\mathbf{u}} = [\bar{u}_1, \ldots, \bar{u}_d]$. In this section, we further assume that the constraint set $\mathcal{C}$ is down-closed, i.e., $\mathbf{0} \in \mathcal{C}$. It is known that without this assumption, no constant factor approximation guarantee is possible [60].

We present `Stochastic Measured Continuous Greedy`++ (SMCG++) for solving stochastic nonmonotone DR-submodular functions in Algorithm 4.1. Note that many of the steps of SMCG++ are similar to the ones for SCG++, except feasible set selection in step 11, which leads to a different update of ascent direction $\mathbf{v}^t$. In particular, we compute the ascent direction in SMCG++ by solving the problem

$$\mathbf{v}^t := \underset{\{\mathbf{v}\in\mathcal{C} | \mathbf{v} \leq \bar{\mathbf{u}} - \mathbf{x}^t\}}{\operatorname{argmax}} \{\mathbf{v}^\top \mathbf{g}^t\}, \tag{29}$$

where $\mathbf{g}^t$ is the stochastic gradient approximation at step t. This update differs from the update in (27) for the monotone setting by having the extra constraint $\mathbf{v} \leq \bar{\mathbf{u}} - \mathbf{x}^t$. This extra constraint is required to ensure that the outcome of this linear optimization does not grow aggressively, as suggested previously in [4, 20]. In the following theorem, we show that SMCG++ obtains a $1/e$-guarantee.

THEOREM 4.3. *Consider the SMCG++ method outlined in Algorithm* 4.1 *and assume that in step* 8 *we follow the update in* (22) *to construct the gradient difference approximation* $\tilde{\Delta}^t$ *(Option* II*). If Assumptions* 1–7 *hold, then the output of SMCG++ denoted by* $\mathbf{x}^T$ *satisfies*

$$\mathbb{E}\left[F(\mathbf{x}^T)\right] \geq (1/e)F(\mathbf{x}^*) - (4\sqrt{2}+1)/2 \cdot \bar{L}D^2\epsilon,$$

by setting $|\mathcal{M}_0| = \frac{G^2}{2\bar{L}^2 D^2 \epsilon^2}$, $|\mathcal{M}| = \frac{1}{2\epsilon}$, $T = \frac{1}{\epsilon}$, *and* $\delta = \mathcal{O}(\epsilon^2)$ *as in* (41). *Here* $\bar{L}$ *is a constant defined by* $\bar{L}^2 \overset{\text{def}}{=} 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2$.

The result in Theorem 4.3 implies that after at most $T = 1/\epsilon$ iterations the objective function value for the output of SMCG++ is at least $(1/e)\mathrm{OPT} - \mathcal{O}(\epsilon)$. As the number of calls to the stochastic oracle per iteration is of $\mathcal{O}(1/\epsilon)$, to reach a $(1/e)\mathrm{OPT} - \mathcal{O}(\epsilon)$ approximation guarantee SMCG++ has an overall stochastic first order oracle complexity of $\mathcal{O}(1/\epsilon^2)$ with $\mathcal{O}(1/\epsilon)$ calls to a linear optimization oracle.

5. Stochastic discrete submodular maximization. In this section, we focus on extending our result to the case where F is the multilinear extension of a discrete submodular function f. This is also an instance of the nonoblivious stochastic optimization (1). Indeed, once such a result is achieved, with a proper rounding scheme such as pipage rounding [8] or contention resolution method [15], we can obtain discrete solutions. Let V denote a finite set of d elements, i.e., $V = \{1, \dots, d\}$. Consider a discrete submodular function $f : 2^V \to \mathbb{R}_+$, which is defined as an *expectation* over a set of functions $f_\gamma : 2^V \to \mathbb{R}_+$. Our goal is to maximize f subject to some constraint $\mathcal{I}$, where the $\mathcal{I}$ is a collection of the subsets of V; i.e., we aim to solve the following discrete and stochastic submodular function maximization problem:

$$\max_{S\in\mathcal{I}} f(S) := \max_{S\in\mathcal{I}} \mathbb{E}_{\gamma\sim p(\gamma)}[f_\gamma(S)], \tag{30}$$

where $p(\gamma)$ is an arbitrary distribution. In particular, we assume the pair $M = \{V, \mathcal{I}\}$ forms a matroid with rank r. The prototypical example is maximization under the cardinality constraint; i.e., for a given integer r, find $S \subseteq V$, $|S| \le r$, which maximizes f. The challenge here is to find a solution with near-optimal quality for the problem in (30) without computing the expectation in (30). That is, we assume access to an oracle that, given a set S, outputs an independently chosen sample $f_\gamma(S)$ where $\gamma \sim p(\gamma)$. The focus of this section is on extending our results into the discrete domain and showing that SCG++ can be applied for maximizing a stochastic submodular set function f, namely problem (30), through the multilinear extension of the function f. Specifically, in lieu of solving (30) we can solve its continuous extension

$$\max_{\mathbf{x}\in\mathcal{C}} \; F(\mathbf{x}), \tag{31}$$

where $F : [0,1]^V \to \mathbb{R}_+$ is the multilinear extension of f and is defined as

$$F(\mathbf{x}) \;:=\; \sum_{S\subseteq V} f(S) \prod_{i\in S} x_i \prod_{j\notin S}(1-x_j) \;=\; \sum_{S\subseteq V} \mathbb{E}_{\gamma\sim p(\gamma)}[f_\gamma(S)] \prod_{i\in S} x_i \prod_{j\notin S}(1-x_j), \tag{32}$$

and the convex set $\mathcal{C} = \mathrm{conv}\{1_I : I \in \mathcal{I}\}$ is the matroid polytope [8]. Note that here x_i denotes the ith component of $\mathbf{x}$. In other words, $F(\mathbf{x})$ is the expected value of f over sets wherein each element i is included with probability x_i independently.

To solve the multilinear extension problem in (32) using SCG++ (for the monotone case) and SMCG++ (for the nonmonotone case), we need access to unbiased estimators of the gradient and the Hessian. In the following lemma, we first study the structure of the Hessian of the objective function (32).

LEMMA 5.1 (see [8]). *Recall the definition of F in* (32) *as the multilinear extension of the set function f in* (30). *Then, for $i = j$ we have $[\nabla^2 F(\mathbf{y})]_{i,j} = 0$, and for $i \neq j$*

$$\begin{aligned}[\nabla^2 F(\mathbf{y})]_{i,j} =& F(\mathbf{y}; \mathbf{y}_i \leftarrow 1, \mathbf{y}_j \leftarrow 1) - F(\mathbf{y}; \mathbf{y}_i \leftarrow 1, \mathbf{y}_j \leftarrow 0)\\ &- F(\mathbf{y}; \mathbf{y}_i \leftarrow 0, \mathbf{y}_j \leftarrow 1) + F(\mathbf{y}; \mathbf{y}_i \leftarrow 0, \mathbf{y}_j \leftarrow 0),\end{aligned} \tag{33}$$

where the vector $\mathbf{y}; \mathbf{y}_i \leftarrow c_i, \mathbf{y}_j \leftarrow c_j$ is defined as a vector at which the ith and jth entries of $\mathbf{y}$ are set to c_i and c_j, respectively.

Note that each term in (33) is an expectation which can be estimated in a bias-free manner by direct sampling. We will now construct the Hessian approximation $\tilde{\nabla}_t^2$ using Lemma 5.1. Let a be a uniform random variable between $[0,1]$, and let $\mathbf{e} = (e_1, \ldots, e_d)$ be a random vector in which e_i's are generated i.i.d. according to the uniform distribution over the unit interval $[0,1]$. In each iteration, a minibatch $\mathcal{M}$ of $|\mathcal{M}|$ samples of $\{a, \mathbf{e}, \gamma\}$ (recall that γ is the random variable that parametrizes the component function f_γ), i.e., $\mathcal{M} = \{a_k, \mathbf{e}_k, \gamma_k\}_{k=1}^{|\mathcal{M}|}$, is generated. Then for all $k \in [|\mathcal{M}|]$, we let $\mathbf{x}_{a_k} = a_k \mathbf{x}^t + (1-a_k)\mathbf{x}^{t-1}$ and construct the random set $S(\mathbf{x}_{a_k}, \mathbf{e}_k)$ using $\mathbf{x}_{a_k}$ and $\mathbf{e}_k$ in the following way: $s \in S(\mathbf{x}_{a_k}, \mathbf{e}_k)$ if and only if $[\mathbf{e}_k]_s \le [\mathbf{x}_{a_k}]_s$ for $s \in [d]$. Having $S(\mathbf{x}_{a_k}, \mathbf{e}_k)$ and γ_k, each entry of the Hessian estimator $\tilde{\nabla}_t^2 \in \mathbb{R}^{d\times d}$ is

$$
\begin{aligned}
(34) \qquad \left[\tilde{\nabla}_t^2\right]_{i,j} = \frac{1}{|\mathcal{M}|} \sum_{k\in[|\mathcal{M}|]} & f_{\gamma_k}(S(\mathbf{x}_{a_k}, \mathbf{e}_k) \cup \{i,j\}) - f_{\gamma_k}(S(\mathbf{x}_{a_k}, \mathbf{e}_k) \cup \{i\} \setminus \{j\}) \\
& - f_{\gamma_k}(S(\mathbf{x}_{a_k}, \mathbf{e}_k) \cup \{j\} \setminus \{i\}) + f_{\gamma_k}(S(\mathbf{x}_{a_k}, \mathbf{e}_k) \setminus \{i,j\}),
\end{aligned}
$$

where $i \neq j$, and if $i = j$, then $[\tilde{\nabla}_t^2]_{i,j} = 0$. As linear optimization over the rank-r matroid polytope returns $\mathbf{v}^t$ with at most r nonzero entries, the complexity of computing (34) is $\mathcal{O}(rd)$. Now we use the above approximation of the Hessian to solve the multilinear extension as a special case of problem (1) using SCG++ and SMCG++. To do so, we first introduce the following definitions.

Definition 5.2. *Let D_γ denote the maximum marginal value of f_γ, i.e., $D_\gamma = \max_{i\in V} f_\gamma(i)$, and further define $D_f = (\mathbb{E}_\gamma[D_\gamma^2])^{1/2}$.*

Based on Definition 5.2, the Hessian estimator $\tilde{\nabla}_t^2$ has a bounded $\|\cdot\|_{2,\infty}$ norm: $\mathbb{E}[\|\tilde{\nabla}_t^2\|_{2,\infty}^2] = \mathbb{E}[\max_{i\in[d]} \|\tilde{\nabla}_t^2(:,i)\|^2] \le 4d \cdot \mathbb{E}_\gamma D_\gamma^2 = 4d \cdot D_f^2$.

5.1. Convergence results. We first analyze the convergence of SCG++ for solving problem (31) when f is monotone. Compared to Theorem 4.1, Theorem 5.3 has a dependency on the problem dimension d and exploits the sparsity of $\mathbf{v}^t$. Note that this result is presented as Theorem 2 in [33].

Theorem 5.3. *Consider the multilinear extension of a monotone stochastic submodular set function, and recall the definition of D_f. By using the minibatch size $|\mathcal{M}| = \mathcal{O}(\sqrt{r^3 d} D_f/\epsilon)$ and $|\mathcal{M}_0| = \mathcal{O}(\sqrt{d} D_f/\sqrt{r}\epsilon^2)$, SCG++ finds a $[(1-1/e)OPT - 6\epsilon]$ approximation of the multilinear extension problem at most $(\sqrt{r^3 d} D_f/\epsilon)$ iterations. Moreover, the overall stochastic oracle cost is $\mathcal{O}(r^3 d D_f^2/\epsilon^2)$.*

Since the cost of a single stochastic gradient computation is $\mathcal{O}(d)$, Theorem 5.3 shows that the overall computation complexity of Algorithm 4.1 is $\mathcal{O}(d^2/\epsilon^2)$. Note that, in the multilinear extension case, the smoothness Assumption 4 required for the results in section 4.1 is absent, and that is why we need to develop a more sophisticated gradient-difference estimator to achieve a similar theoretical guarantee.

Remark 5.4 (optimality of oracle complexities). In order to achieve the tight $1 - 1/e - \epsilon$ approximation, the stochastic oracle complexity $\mathcal{O}(1/\epsilon^2)$, obtained in Theorem 5.3, is optimal in terms of its dependency on ϵ. A lower bound on the stochastic oracle complexity is given in the following section.

We proceed to derive our result for stochastic and nonmonotone discrete submodular function maximization.

Theorem 5.5. *Consider the multilinear extension of a nonmonotone stochastic submodular set function, and recall the definition of D_f. By using the minibatch size*

$|\mathcal{M}| = \mathcal{O}(\sqrt{r^3 d}D_f/\epsilon)$ and $|\mathcal{M}_0| = \mathcal{O}(\sqrt{d}D_f/\sqrt{r}\epsilon^2)$, SMCG++ finds a $[(1/e)OPT - \epsilon]$ approximation of the multilinear extension problem at most $(\sqrt{r^3 d}D_f/\epsilon)$ iterations. Moreover, the overall stochastic oracle cost is $\mathcal{O}(r^3 d D_f^2/\epsilon^2)$.

5.2. Lower bound. In this section, we show that reaching a $(1-1/e-\epsilon)$-optimal solution of problem (1) when F is a monotone DR-submodular function requires at least $\mathcal{O}(1/\epsilon^2)$ calls to an oracle that provides stochastic first order information. To do so, we first construct a stochastic submodular set function f, defined as $f(S) = \mathbb{E}_{\gamma\sim p(\gamma)}[f_\gamma(S)]$, with the following property: Obtaining a $(1-1/e-\epsilon)$-optimal solution for maximization of f under a cardinality constraint requires at least $\mathcal{O}(1/\epsilon^2)$ samples of the form $f_\gamma(\cdot)$ where γ is generated i.i.d. from distribution p. Such a lower bound on sample complexity can be directly extended to problem (1) with a stochastic first order oracle, by considering the multilinear extension of the function f, denoted by F, and noting that (i) problems (30) and (31) have the same optimal values, and (ii) one can construct an unbiased estimator of the gradient of the multilinear extension using d independent samples from the underlying stochastic set function f. Hence, any method for maximizing (31) is also a method for maximizing (30) with the same guarantees on the quality of the solution and with sample complexities that differ at most by a factor of d. Now formalize the above argument. We note that this result is presented as Theorem 3 in [33].

THEOREM 5.6. *There exist a distribution $p(\gamma)$ and a monotone submodular function $f : 2^V \to \mathbb{R}_+$, given as $f(S) = \mathbb{E}_{\gamma\sim p(\gamma)}[f_\gamma(S)]$, such that the following holds: In order to find a $(1 - 1/e - \epsilon)$-optimal solution for (30) with k-cardinality constraint, any algorithm requires at least $\min\{\exp(\alpha k), \beta/\epsilon^2\}$ stochastic samples $f_\gamma(\cdot)$.*

COROLLARY 5.7. *There exist a monotone DR-submodular function F, a convex constraint $\mathcal{C}$, and a stochastic first order oracle $\mathcal{O}_{first}$ such that any method for maximizing F subject to $\mathcal{C}$ requires at least $\min\{\exp(\alpha n), \beta/\epsilon^2\}$ queries from $\mathcal{O}_{first}$.*

6. Conclusion. In this paper we studied a class of stochastic conditional gradient methods for solving nonoblivious convex and nonconvex minimization problems as well as continuous DR-submodular maximization problems. In particular, (i) we proposed a stochastic variant of the Frank–Wolfe method called SFW++ for minimizing a smooth *nonconvex* stochastic function subject to a convex body constraint. We showed that SFW++ finds an ϵ-FOSP after at most $O(1/\epsilon^3)$ stochastic gradient evaluations; (ii) we further studied the convergence rate of SFW++ when we face a constrained *convex* minimization problem and showed that SFW++ achieves an ϵ-approximate optimum while using $O(1/\epsilon^2)$ stochastic gradients; (iii) we also extended the idea of our proposed variance reduced stochastic condition gradient method to the *submodular* setting and developed SCG++, the first efficient variant of continuous greedy for maximizing a stochastic, continuous, monotone DR-submodular function subject to a convex constraint. We showed that SCG++ achieves a tight $[(1 - 1/e)\mathrm{OPT} - \epsilon]$ solution while using $O(1/\epsilon^2)$ stochastic gradients. We further derived a tight lower bound on the number of calls to the first-order stochastic oracle for achieving a $[(1 - 1/e)\mathrm{OPT} - \epsilon]$ approximate solution. This result showed that SCG++ has the optimal sample complexity for finding an optimal $(1 - 1/e)$ approximation guarantee for monotone but stochastic DR-submodular functions. Finally, for maximizing a nonmonotone continuous DR-submodular function, SCG++ achieves a $[(1/e)\mathrm{OPT} - \epsilon]$ solution after computing $O(1/\epsilon^2)$ stochastic gradients.

Appendix A. Proof of Lemma 3.3. We first present a lemma which bounds the second moment of the spectral norm of the Hessian estimator $\nabla^2\tilde{F}(\mathbf{y}; \mathbf{z})$.

LEMMA A.1. *Recall the definition of the Hessian estimator $\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})$ in* (13). *Under Assumptions* 1, 3, *and* 4, *we bound for any* $\mathbf{y} \in \mathcal{C}$

$$\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{y})}[\|\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})\|^2] \leq 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2 \stackrel{\text{def}}{=} \bar{L}^2. \tag{35}$$

Proof. From the definition of $\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})$, we have from Assumptions 1 and 3

$$\|\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})\| \leq B\|\nabla \log p(\mathbf{z};\mathbf{y})\|^2 + 2G\|\nabla \log p(\mathbf{z};\mathbf{y})\| + L + B\|\nabla^2 \log p(\mathbf{z};\mathbf{y})\|. \tag{36}$$

Further, taking expectation on both sides, we use Assumption 4 to bound

$$\mathbb{E}[\|\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z})\|^2] \leq 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2. \tag{37}$$ □

Proof of Lemma 3.3. We prove via induction. When $t = 0$, using the unbiasedness of $\nabla\tilde{F}(\mathbf{x}^0;\mathbf{z})$ and Assumption 3, we bound

$$\begin{aligned}\mathbb{E}_{\mathcal{M}_0}[\|F(\mathbf{x}^0) - \mathbf{g}^0\|^2] =& \frac{1}{|\mathcal{M}_0|}\mathbb{E}[\|F(\mathbf{x}^0) - \nabla\tilde{F}(\mathbf{x}^0;\mathbf{z})\|^2] \\ \leq& \frac{1}{|\mathcal{M}_0|}\mathbb{E}[\|\nabla\tilde{F}(\mathbf{x}^0;\mathbf{z})\|^2] \leq \frac{G^2}{|\mathcal{M}_0|} \leq \bar{L}^2D^2\epsilon^2.\end{aligned}$$

Now assume that we have the result for $t = \bar{t}$. When $t = \bar{t} + 1$, we have

$$\begin{aligned}\mathbf{g}^t - \nabla F(\mathbf{x}^t) = \left[\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\right] + \left[\xi_\delta(\mathbf{x};\mathcal{M}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\right] \\ + \left[\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\right].\end{aligned}$$

Expand $\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2$ to obtain

$$\begin{aligned}\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2 =& \|\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\|^2 + \|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2 \\ &+ 2\langle\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}), \mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\rangle \\ &+ 2\langle\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M}), \nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\rangle \\ &+ 2\langle\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M}), \mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\rangle \\ &+ \|\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M})\|^2.\end{aligned} \tag{38}$$

Using the unbiasedness of $\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})$, we have

$$\mathbb{E}[\langle\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}), \mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\rangle] = 0. \tag{39}$$

Additionally, from the unbiasedness of $\tilde{\Delta}^t$, we have

$$\mathbb{E}[\|\tilde{\Delta}^t - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\|^2] \leq \frac{\epsilon^2D^2}{|\mathcal{M}|}\mathbb{E}[\|\nabla^2 F(\mathbf{x}(a_1);\mathbf{z}_1(a_1))\|^2] \leq \frac{\epsilon^2\bar{L}^2D^2}{|\mathcal{M}|}, \tag{40}$$

where we use Lemma A.1. Taking expectation on both sides of (38), we have

$$
\begin{aligned}
&\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] \\
&= \mathbb{E}[\|\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
&+ 2\mathbb{E}[\|\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M})\|\|\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\|] \\
&+ 2\mathbb{E}[\|\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M})\|\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|] + \mathbb{E}[\|\tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1}) - \xi_\delta(\mathbf{x};\mathcal{M})\|^2] \\
&\le \mathbb{E}[\|\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] + 4D^4L_2^2\delta^2 \\
&+ 4D^2L_2\delta\|\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}) - \tilde{\nabla}_t^2(\mathbf{x}^t - \mathbf{x}^{t-1})\| + 4D^2L_2\delta\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\| \\
&\le \frac{\bar{L}^2D^2\epsilon^2}{|\mathcal{M}|} + (1+\epsilon(t-1))\bar{L}^2D^2\epsilon^2 + 4\delta\left[\frac{D^2L_2\bar{L}D\epsilon}{\sqrt{|\mathcal{M}|}} + D^2L_2\sqrt{(1+\epsilon(t-1))}\bar{L}D\epsilon + D^4L_2^2\delta\right].
\end{aligned}
$$

By taking δ sufficiently small such that

$$
4\delta\left((1/\sqrt{|\mathcal{M}|})D^2L_2\bar{L}D\epsilon + D^2L_2\sqrt{(1+\epsilon(t-1))}\bar{L}D\epsilon + D^4L_2^2\delta\right) \le \bar{L}^2D^2\epsilon^3/2, \tag{41}
$$

we have shown that the induction holds for $t = \bar{t} + 1$. ∎

Appendix B. Proof of Theorem 3.4.

First we prove the following lemma.

LEMMA B.1. *Recall the definition in* (13). *Under Assumptions* 1, 3, *and* 4, *and by taking* $q = G/(16\epsilon)$, $|\mathcal{M}_0^t| = G^2/(8\epsilon^2)$, $|\mathcal{M}_h^t| = 2G/\epsilon$, *and* $\eta_t = \epsilon/(\bar{L}D)$, *we bound*

$$
\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \le \epsilon^2/4, \tag{42}
$$

where $\bar{L}$ *is defined as in Lemma* 3.3.

Proof. For t such that $mod(t,p) \neq 0$,

$$
\begin{aligned}
&\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] = \mathbb{E}[\|\tilde{\nabla}_t^2[\mathbf{x}^t - \mathbf{x}^{t-1}] + \mathbf{g}^{t-1} - \nabla F(\mathbf{x}^t)\|^2] \\
&= \mathbb{E}[\|\tilde{\nabla}_t^2[\mathbf{x}^t - \mathbf{x}^{t-1}] - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
&= \frac{1}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2[\mathbf{x}^t - \mathbf{x}^{t-1}] - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
&\le \frac{1}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2[\mathbf{x}^t - \mathbf{x}^{t-1}]\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2].
\end{aligned}
$$

Observe that $\mathbf{x}^{t+1} - \mathbf{x}^t = \eta_t(\mathbf{v}^t - \mathbf{x}^t)$ and therefore

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] &\le \frac{4\eta_t^2D^2}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
&\le \frac{4\eta_t^2D^2\bar{L}^2}{|\mathcal{M}_h^t|} + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2],
\end{aligned}
$$

where we use Lemma A.1 in the second inequality. Denote $k_0 = q \times \lfloor t/q \rfloor$ and $k = mod(t,q) \le q$. Repeat the above recursion k times to obtain

$$
\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \le \frac{4k\eta_t^2D^2\bar{L}^2}{|\mathcal{M}_h^t|} + \mathbb{E}[\|\mathbf{g}^{k_0} - \nabla F(\mathbf{x}^{k_0})\|^2] = \frac{4q\eta_t^2D^2\bar{L}^2}{|\mathcal{M}_h^t|} + \frac{G^2}{|\mathcal{M}_0^t|}.
$$

By setting $q = G/(16\epsilon)$, $|\mathcal{M}_0^t| = G^2/(8\epsilon^2)$, $|\mathcal{M}_h^t| = 2G/\epsilon$, and $\eta_t = \epsilon/\bar{L}D$, we have

$$
\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \le \epsilon^2/4. \tag{43}
$$

∎

Now we are ready to prove the claim in Theorem 3.4. From Lemma A.1, we have

$$\|\nabla^2 F(\mathbf{x})\|^2 \le \|\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\nabla^2 \tilde{F}(\mathbf{x};\mathbf{z})]\|^2 \le \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\|\nabla^2 \tilde{F}(\mathbf{x};\mathbf{z})\|^2] \le \bar{L}^2. \tag{44}$$

Hence, F is $\bar{L}$-smooth. From the smoothness of F

$$\begin{aligned}
F(\mathbf{x}^{t+1}) &\ge F(\mathbf{x}^t) + \langle \nabla F(\mathbf{x}^t), \mathbf{x}^{t+1}-\mathbf{x}^t\rangle - \frac{\bar{L}}{2}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 \\
&= F(\mathbf{x}^t) + \langle \mathbf{g}^t, \mathbf{x}^{t+1}-\mathbf{x}^t\rangle - \frac{\bar{L}}{2}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 + \langle \nabla F(\mathbf{x}^t)-\mathbf{g}^t, \mathbf{x}^{t+1}-\mathbf{x}^t\rangle \\
&= F(\mathbf{x}^t) + \eta\langle \mathbf{g}^t, \mathbf{v}^t-\mathbf{x}^t\rangle - \frac{\bar{L}\eta^2}{2}\|\mathbf{u}^t-\mathbf{x}^t\|^2 + \eta\langle \nabla F(\mathbf{x}^t)-\mathbf{g}^t, \mathbf{v}^t-\mathbf{x}^t\rangle \\
&\ge F(\mathbf{x}^t) + \eta\langle \mathbf{g}^t, \mathbf{v}^t-\mathbf{x}^t\rangle - 2\bar{L}\eta^2 D^2 - 2\eta D\|\nabla F(\mathbf{x}^t)-\mathbf{g}^t\|.
\end{aligned}$$

Denoting $\mathbf{v}^+ = \operatorname{argmax}_{\mathbf{v}\in C}\langle \nabla F(\mathbf{x}^t), \mathbf{v}-\mathbf{x}^t\rangle$, we have

$$V_{\mathcal{C}}(\mathbf{x}^t;F) = \langle \nabla F(\mathbf{x}^t)-\mathbf{g}^t, \mathbf{v}^+-\mathbf{x}^t\rangle + \langle \mathbf{g}^t, \mathbf{v}^+-\mathbf{x}^t\rangle \le 2D\|\nabla F(\mathbf{x}^t)-\mathbf{g}^t\| + \langle \mathbf{g}^t, \mathbf{v}^t-\mathbf{x}^t\rangle.$$

These two bounds together give

$$\eta V_{\mathcal{C}}(\mathbf{x}^t;F) \le F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) + 4\eta D\|\nabla F(\mathbf{x}^t)-\mathbf{g}^t\| + 2\bar{L}\eta^2 D^2.$$

As $\eta = \epsilon/\bar{L}D$ and $\mathbb{E}[\|\nabla F(\mathbf{x}^t)-\mathbf{g}^t\|^2] \le \epsilon^2/4$ in Lemma B.1, we have for $t \ge 1$

$$\begin{aligned}
\frac{\epsilon}{\bar{L}D}\mathbb{E}[V_{\mathcal{C}}(\mathbf{x}^t;F)] \le\ & \mathbb{E}[F(\mathbf{x}^{t+1})] - \mathbb{E}[F(\mathbf{x}^t)] + 2\epsilon^2/\bar{L} + 4\epsilon/\bar{L}\cdot\mathbb{E}[\|\nabla F(\mathbf{x}^t)-\mathbf{g}^t\|] \\
\le\ & \mathbb{E}[F(\mathbf{x}^{t+1})] - \mathbb{E}[F(\mathbf{x}^t)] + 4\epsilon^2/\bar{L}.
\end{aligned}$$

Sum the above inequalities from $t=1$ to T and multiply both sides by $\bar{L}D$ to obtain

$$\sum_{t=1}^{T} \epsilon\mathbb{E}[V_{\mathcal{C}}(\mathbf{x}^t;f)] \le \bar{L}D(F(\mathbf{x}^*) - F(\mathbf{x}^1)) + T\cdot 4D\epsilon^2.$$

Hence, by sampling t_0 from $[T]$ uniformly at random, we have

$$\mathbb{E}[V_{\mathcal{C}}(\mathbf{x}^t;F)] \le \frac{D\bar{L}(F(\mathbf{x}^*) - F(\mathbf{x}^1))}{T\epsilon} + 4\epsilon D,$$

and thus when $T = \bar{L}(F(\mathbf{x}^*) - F(\mathbf{x}^1))/\epsilon^2$, we have $\mathbb{E}[V_{\mathcal{C}}(\mathbf{x}^t;F)] \le 5\epsilon D$.

Appendix C. Proof of Theorem 3.7. We first prove the following lemma.

LEMMA C.1. *Recall the definition of the Hessian estimator in* (13). *Under Assumptions* 1, 3, *and* 4, *by taking* $|\mathcal{M}_h^t| = 16(t+2)$ *and* $|\mathcal{M}_0^t| = \frac{G^2(t+1)^2}{\bar{L}^2D^2}$, *we bound*

$$\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \le 2\bar{L}^2D^2\eta_t^2, \tag{45}$$

where $\bar{L}$ *is defined as in Lemma* 3.3.

Proof. Assume iteration t is in the kth epoch, i.e., $2^k \le t < 2^{k+1}$. For $t \ne 2^k$,

$$\begin{aligned}
&\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] = \mathbb{E}[\|\tilde{\nabla}_t^2[\mathbf{x}^t - \mathbf{x}^{t-1}] + \mathbf{g}^{t-1} - \nabla F(\mathbf{x}^t)\|^2] \\
=&\mathbb{E}[\|\tilde{\nabla}_t^2[\mathbf{x}^t - \mathbf{x}^{t-1}] - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
=&\frac{1}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2[\mathbf{x}^t - \mathbf{x}^{t-1}] - (\nabla F(\mathbf{x}^t) - \nabla F(\mathbf{x}^{t-1}))\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\
\le&\frac{1}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2[\mathbf{x}^t - \mathbf{x}^{t-1}]\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2].
\end{aligned}$$

Observe that $\mathbf{x}^{t+1} - \mathbf{x}^t = \eta_t(\mathbf{v}^t - \mathbf{x}^t)$, and therefore

$$\begin{aligned}\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] &\leq \frac{4\eta_t^2 D^2}{|\mathcal{M}_h^t|}\mathbb{E}[\|\tilde{\nabla}_1^2\|^2] + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2] \\ &\leq \frac{4\eta_t^2 D^2 \bar{L}^2}{|\mathcal{M}_h^t|} + \mathbb{E}[\|\mathbf{g}^{t-1} - \nabla F(\mathbf{x}^{t-1})\|^2],\end{aligned}$$

where we use Lemma A.1 in the second inequality. Repeating the above recursion $t - 2^k < 2^k$ times (since $t < 2^{k+1}$), we obtain

$$\begin{aligned}&\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \leq \mathbb{E}[\|\mathbf{g}^{2^k} - \nabla F(\mathbf{x}^{2^k})\|^2] + \sum_{i=2^k}^{t} 4D^2\bar{L}^2 \cdot \frac{\eta_i^2}{|\mathcal{M}_h^i|} \\ &\leq \frac{G^2}{|\mathcal{M}_0^{2^k}|} + \sum_{i=2^k}^{t} \frac{D^2\bar{L}^2}{(i+2)^3} \leq \bar{L}^2 D^2 \eta_{2^{k+1}}^2 + \frac{D^2\bar{L}^2}{2}\sum_{i=2^k}^{t}\left[\frac{1}{(i+1)(i+2)} - \frac{1}{(i+2)(i+3)}\right] \\ &\leq \bar{L}^2 D^2 \eta_{2^{k+1}}^2 + D^2\bar{L}^2 \frac{1}{(2^k+1)(2^{k-1}+1)} \leq 2\bar{L}^2 D^2 \eta_{2^{k+1}}^2 \leq 2\bar{L}^2 D^2 \eta_t^2. \qquad \square\end{aligned}$$

Now we are ready to prove the claim in Theorem 3.7. From Lemma A.1, we have

$$\|\nabla^2 F(\mathbf{x})\|^2 \leq \|\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{\nabla}^2 F(\mathbf{x};\mathbf{z})]\|^2 \leq \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\|\tilde{\nabla}^2 F(\mathbf{x};\mathbf{z})\|^2] \leq \bar{L}^2. \tag{46}$$

The boundedness of the Hessian $\nabla^2 F$ is equivalent to the smoothness of F. Let $\mathbf{x}^*$ be a global maximizer within the constraint set $\mathcal{C}$. By the smoothness of F, we have

$$\begin{aligned}F(\mathbf{x}^{t+1}) &\geq F(\mathbf{x}^t) + \langle \nabla F(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t\rangle - (\bar{L}/2)\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \\ &= F(\mathbf{x}^t) + \eta_t\langle \nabla F(\mathbf{x}^t), \mathbf{v}^t - \mathbf{x}^t\rangle - (\bar{L}\eta_t^2/2)\|\mathbf{v}^t - \mathbf{x}^t\|^2 \\ &= F(\mathbf{x}^t) + \eta_t\langle \mathbf{g}^t, \mathbf{v}^t - \mathbf{x}^t\rangle + \eta_t\langle \nabla F(\mathbf{x}^t) - \mathbf{g}^t, \mathbf{v}^t - \mathbf{x}^t\rangle - 2\bar{L}\eta_t^2 D^2 \\ &\geq F(\mathbf{x}^t) + \eta_t\langle \mathbf{g}^t, \mathbf{x}^* - \mathbf{x}^t\rangle + \eta_t\langle \nabla F(\mathbf{x}^t) - \mathbf{g}^t, \mathbf{v}^t - \mathbf{x}^t\rangle - 2\bar{L}\eta_t^2 D^2,\end{aligned} \tag{47}$$

where we use the optimality and boundedness of $\mathbf{v}^t$ in the last inequality. Take expectation, and use the unbiasedness of $\mathbf{g}_{vr}^t$ and Young's inequality to obtain

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \mathbb{E}[F(\mathbf{x}^t)] + \eta_t\langle \nabla F(\mathbf{x}^t), \mathbf{x}^* - \mathbf{x}^t\rangle - \frac{1}{2\bar{L}}\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] - 6\bar{L}\eta_t^2 D^2.$$

From the convexity of F, we have $\langle \nabla F(\mathbf{x}^t), \mathbf{x}^* - \mathbf{x}^t\rangle \geq F(\mathbf{x}^*) - F(\mathbf{x}^t)$, and thus

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \mathbb{E}[F(\mathbf{x}^t)] + \eta_t\mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^t)] - \frac{1}{2\bar{L}}\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] - 6\bar{L}\eta_t^2 D^2. \tag{48}$$

By using Lemma 3.3 with $|\mathcal{M}_0| = \frac{G^2}{\bar{L}^2 D^2 \eta_t^2}$ and $|\mathcal{M}_h| = \frac{1}{\eta_t}$, we have

$$\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] \leq 2\bar{L}^2 D^2 \eta_t^2. \tag{49}$$

Let $\delta_t \stackrel{\text{def}}{=} F(\mathbf{x}^*) - F(\mathbf{x}^t)$ and $c \stackrel{\text{def}}{=} \max\{14\bar{L}D^2, \delta_0\}$. Combining (48) and (49) gives

$$\mathbb{E}[\delta_{t+1}] \leq (1 - \eta_t)\mathbb{E}[\delta_t] + c\eta_t^2/2.$$

By taking $\eta_t = \frac{2}{t+2}$ and by induction we obtain $\mathbb{E}\delta_t \leq \frac{2c}{t+2}$: For $t = 0$, it trivially holds. Assume $\mathbb{E}[\delta_{t_0}] \leq \frac{2c}{t_0+2}$ with $t_0 \geq 1$. For $t = t_0 + 1$,

$$\mathbb{E}[\delta_{t_0+1}] \leq \frac{t_0}{t_0+2} \cdot \frac{2c}{t_0+2} + \frac{2c}{(t_0+2)^2} \leq \frac{2c}{t_0+3}. \tag{50}$$

In conclusion, we have $F(\mathbf{x}^*) - \mathbb{E}[F(\mathbf{x}^t)] \leq \frac{28\bar{L}D^2 + (F(\mathbf{x}^*) - F(\mathbf{x}^0))}{t+2}$.

Appendix D. Proof of Theorem 4.1. Theorem 4.1 is identical to Theorem 1 of the conference version of this paper [33], and we refer the reader to the detailed proof therein. Here we only provide a sketch. From Lemma A.1, F can be proved to be $\bar{L}$-smooth like in Theorem 3.4. By using Lemma 3.3, for all $t \in \{0, \ldots, T-1\}$ we have $\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] \leq 2\bar{L}^2 D^2 \epsilon^2$. Let $\mathbf{x}^*$ be the global maximizer within the constraint set $\mathcal{C}$. We can prove function value increases for $T = 1/\epsilon$:

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \mathbb{E}[F(\mathbf{x}^t)] + \epsilon \mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^t)] - 2\bar{L}\epsilon^2 D^2,$$

which is equivalent to $\mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^{t+1})] \leq (1-\epsilon)^T \mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^t)] - 2\bar{L}\epsilon D^2$. In conclusion, we have $\mathbb{E}[F(\mathbf{x}^T)] \geq (1 - 1/e)F(\mathbf{x}^*) - 2\bar{L}\epsilon D^2$.

Appendix E. Proof of Theorem 4.3. We note that SMCG++ shares the same structure as the Nonmonotone Stochastic Continuous Greedy (NMSCG) algorithm in [42] except for the gradient estimation. Following the same proof in Appendix H of [42], we arrive at the following inequality ((113) in [42]):

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \geq \frac{1}{T}\left[\left(1 - \frac{1}{T}\right)^t F(\mathbf{x}^*) - F(\mathbf{x}^t)\right] - \frac{2D}{T}\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\| - \frac{\bar{L}D^2}{2T^2}. \tag{51}$$

Recall the variance bound in Lemma 3.3. By taking $\epsilon = 1/T$, we have $\mathbb{E}\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\| \leq \sqrt{\mathbb{E}\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2} \leq \sqrt{2}\bar{L}D/T$. Take expectations on both sides of (51) and plug in the above variance bound to arrive at

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \left(1 - \frac{1}{T}\right)\mathbb{E}[F(\mathbf{x}^t)] + \frac{1}{T}\left(1 - \frac{1}{T})^t F(\mathbf{x}^*\right) - \frac{(4\sqrt{2}+1)\bar{L}D^2}{2T^2}. \tag{52}$$

Multiplying $(1 - \frac{1}{T})^{-(t+1)}$ on both sides of (52), we have

$$\left(1-\frac{1}{T}\right)^{-(t+1)} \mathbb{E}[F(\mathbf{x}^{t+1})] \geq \left(1-\frac{1}{T}\right)^{-t} \mathbb{E}[F(\mathbf{x}^t)] + \frac{1}{T}\left(1-\frac{1}{T}\right)^{-1} F(\mathbf{x}^*) - \frac{(4\sqrt{2}+1)\bar{L}D^2}{2T^2(1-\frac{1}{T})^{(t+1)}}.$$

Sum the above inequality from $t = 0$ to $T-1$ and use $F(\mathbf{x}^0) \geq 0$ to obtain

$$\begin{aligned} &(1 - T^{-1})^{-T}\mathbb{E}[F(\mathbf{x}^T)] \\ &\geq F(\mathbf{x}^0) + (1 - T^{-1})^{-1}F(\mathbf{x}^*) - (2\sqrt{2}+0.5)\bar{L}D^2T^{-2} \cdot [(1-T^{-1})^{-T} - 1](T-1) \\ &\geq (1 - T^{-1})^{-1}F(\mathbf{x}^*) - (2\sqrt{2}+0.5)\bar{L}D^2T^{-2} \cdot [(1-T^{-1})^{-T} - 1](T-1). \end{aligned} \tag{53}$$

Multiply $(1 - T^{-1})^T$ on both sides of (53) to obtain

$$\begin{aligned} \mathbb{E}[F(\mathbf{x}^T)] &\geq (1 - T^{-1})^{T-1}F(\mathbf{x}^*) - (2\sqrt{2}+0.5)\bar{L}D^2T^{-2} \cdot [1 - (1-T^{-1})^T](T-1) \\ &\geq e^{-1}F(\mathbf{x}^*) - (2\sqrt{2}+0.5)\bar{L}D^2T^{-1}. \end{aligned} \tag{54}$$

E.1. Multilinear extension as nonoblivious stochastic optimization. We proceed to show that the problem in (32) is captured by (1). To do so, use $\text{Ber}(b; m)$ with $b \in \{0, 1\}$ and $m \in [0, 1]$ to denote the Bernoulli distribution with parameter m, i.e., $\text{Ber}(b; m) = m^b(1-m)^{1-b}$. Define $p(\mathbf{z}, \gamma; \mathbf{x})$ as

$$p(\mathbf{z}, \gamma; \mathbf{x}) = p(\gamma) \times \prod_{i=1}^{d} \text{Ber}(\mathbf{z}_i; \mathbf{x}_i), \tag{55}$$

Algorithm E.1 (SCG++) for Multilinear Extension.

Input: Minibatch size $|\mathcal{M}_0|$ and $|\mathcal{M}|$, and total number of rounds T

for $t = 1$ **to** T **do**
  if $t = 1$ **then**
    Sample $\mathcal{M}_0$ of $(\gamma, \mathbf{z})$ according to $p(\mathbf{z}, \gamma; \mathbf{x}^0)$ and compute $\mathbf{g}^0$ using (59);
  else
    Compute the Hessian approximation $\tilde{\nabla}^2_{\mathcal{M}} = \frac{1}{|\mathcal{M}|} \sum_{k=1}^{|\mathcal{M}|} \tilde{\nabla}^2_k$ based on (34);
    Construct $\tilde{\Delta}^t$ based on (15);
    Update the stochastic gradient approximation $\mathbf{g}^t := \mathbf{g}^{t-1} + \tilde{\Delta}^t$;
  end if
  Compute the ascent direction $\mathbf{v}^t := \operatorname{argmax}_{\mathbf{v} \in \mathcal{C}} \{\mathbf{v}^\top \mathbf{g}^t\}$;
  Update the variable $\mathbf{x}^{t+1} := \mathbf{x}^t + 1/T \cdot \mathbf{v}^t$;
end for

where $p(\gamma)$ is defined as in (30), $\mathbf{z}_i$ is the ith entry of $\mathbf{z}$, and $\mathbf{x}_i$ is the ith entry of $\mathbf{x}$. Let $N(\mathbf{z})$ be a subset of N such that $i \in N(\mathbf{z})$ if and only if $\mathbf{z}_i = 1$. We then define $\tilde{F}(\mathbf{x}; \mathbf{z}, \gamma)$ as

$$\tilde{F}(\mathbf{x}; \mathbf{z}, \gamma) = f_\gamma(N(\mathbf{z})), \tag{56}$$

where f_γ is defined as in (30). For a fixed $\mathbf{z}$, the stochastic function $\tilde{F}$ does not depend on $\mathbf{x}$, and hence $\nabla \tilde{F}(\mathbf{x}; \mathbf{z}) = 0$. By considering the definition of $\tilde{F}(\mathbf{x}; \mathbf{z}, \gamma)$ in (56), the multilinear extension function F in (32), and the probability distribution $p(\mathbf{z}, \gamma; \mathbf{x})$ in (55), it can be verified that F is the expectation of the random $\tilde{F}(\mathbf{x}; \mathbf{z}, \gamma)$, and, therefore, the problem in (32) can be written as (1).

At first glance, it seems that we can apply SCG++ to maximize the multilinear extension function F. However, the smoothness conditions required for the result in Theorem 4.1 do not hold in the multilinear setting. Following the result in Lemma 3.1, we can derive an unbiased estimator for the second order differential of (32) using

$$\begin{aligned} \tilde{\nabla}^2 F(\mathbf{y}; \mathbf{z}) &= \tilde{F}(\mathbf{y}; \mathbf{z}) \left[[\nabla \log p(\mathbf{z}, \gamma; \mathbf{y})][\nabla \log p(\mathbf{z}, \gamma; \mathbf{y})]^\top + \nabla^2 \log p(\mathbf{z}, \gamma; \mathbf{y}) \right] \\ &= f_\gamma(N(\mathbf{z})) \left[\left[\sum_{i=1}^d \nabla \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i) \right] \left[\sum_{i=1}^d \nabla \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i) \right]^\top + \sum_{i=1}^d \nabla^2 \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i) \right], \end{aligned}$$

where we use $\nabla \tilde{F}(\mathbf{x}; \mathbf{z}) = 0$ in the first equality and use (55) and (56) in the second one. Further, note that $[\nabla \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i)]^2 + \nabla^2 \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i) = 0$ for all $i \in [d]$, and hence, the above estimator can be further simplified to

$$\tilde{\nabla}^2 F(\mathbf{y}; \mathbf{z}, \gamma) = f_\gamma(N(\mathbf{z})) \sum_{i,j=1}^d \mathbb{1}_{i \neq j} [\nabla \log \mathrm{Ber}(\mathbf{z}_i; \mathbf{x}_i)][\nabla \log \mathrm{Ber}(\mathbf{z}_j; \mathbf{x}_j)]^\top. \tag{57}$$

Despite the simple form of (57), the smoothness property in Assumption 4 is absent since every entry in the matrix $\tilde{\nabla}^2 F(\mathbf{y}; \mathbf{z}, \gamma)$ can have unbounded second order moment when $\mathbf{x}_i \to 0$ or $\mathbf{x}_i \to 1$.

E.2. Detailed implementation of SCG++ for multilinear extension. We briefly mentioned the Hessian estimator $\tilde{\nabla}^2_k$ in (34). In this section, we describe SCG++ for minimizing the multilinear extension (31) in Algorithm E.1. In particular, we specify the gradient construction for $\mathbf{x}^0$ by using the following equality:

$$[\nabla F(\mathbf{x})]_i = F(\mathbf{x};\mathbf{x}_i \leftarrow 1) - F(\mathbf{x};\mathbf{x}_i \leftarrow 0). \tag{58}$$

Since both terms in (58) are in expectation, we can sample a minibatch $\mathcal{M}_0$ of $(\gamma,\mathbf{z})$ from (58) to obtain an unbiased estimator of $\nabla F(\mathbf{x})$:

$$[\mathbf{g}^0]_i \stackrel{\text{def}}{=} \frac{1}{|\mathcal{M}_0|}\sum_{k=1}^{|\mathcal{M}_0|} f_{\gamma_k}(N(\mathbf{z}_k)\cup\{i\}) - f_{\gamma_k}(N(\mathbf{z}_k)\setminus\{i\}). \tag{59}$$

Appendix F. Proof of Lemma 5.1. First note that we can write the gradient $\nabla_{\mathbf{x}_i}\log Ber(\mathbf{z}_i;\mathbf{x}_i) = \frac{\mathbf{z}_i}{\mathbf{x}_i} - \frac{1-\mathbf{z}_i}{1-\mathbf{x}_i}$. We use $\mathbf{z}_{\backslash i,j}$ to denote the random vector $\mathbf{z}$ excluding the ith and jth entries, and we denote by $\mathbf{z};\mathbf{z}_i \leftarrow c_i, \mathbf{z}_j \leftarrow c_j$ the vectors obtained by setting the ith and jth entries of $\mathbf{z}$ to the corresponding c_i and c_j. Compute $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\mathbf{x})}[\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z},\gamma)]_{i,j}$ using (57):

$$\begin{aligned}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z},\gamma;\mathbf{x})}[\tilde{\nabla}^2 F(\mathbf{y};\mathbf{z},\gamma)]_{i,j} =& \mathbb{E}_{\mathbf{z}}\left[f(N(\mathbf{z}))[\nabla_{\mathbf{x}_i}\log Ber(\mathbf{z}_i;\mathbf{x}_i)][\nabla_{\mathbf{x}_j}\log Ber(\mathbf{z}_j;\mathbf{x}_j)]\right]\\ =& \sum_{c_i,c_j\in\{0,1\}^2} \mathbb{E}_{\mathbf{z}_{\backslash i,j}} f(N(\mathbf{z};\mathbf{z}_i\leftarrow c_i,\mathbf{z}_j\leftarrow c_j))(-1)^{c_i}(-1)^{c_j}, \end{aligned}\tag{60}$$

where in the first equality we use $\mathbb{E}_\gamma f_\gamma = f$ and in the second one we use

$$\mathbf{x}_i^{c_i}\cdot(1-\mathbf{x}_i)^{1-c_i}\cdot\left[\frac{\mathbf{c}_i}{\mathbf{x}_i} - \frac{1-\mathbf{c}_i}{1-\mathbf{x}_i}\right] = -(-1)^{c_i}. \tag{61}$$

While there are four possible configurations for c_i and c_j in (60), we discuss in detail the configuration of $c_i = c_j = 1$. The other three can be obtained similarly.

$$\mathbb{E}_{\mathbf{z}_{\backslash i,j}} f(N(\mathbf{z};\mathbf{z}_i\leftarrow 1,\mathbf{z}_j\leftarrow 1)) = F(\mathbf{y};\mathbf{y}_i\leftarrow 1,\mathbf{y}_j\leftarrow 1), \tag{62}$$

which recovers the first term in (33).

Appendix G. Proof of Theorem 5.3. Theorem 5.3 is identical to Theorem 2 of the conference version of this paper [33]; hence we refer the reader to the detailed proof therein. Here we only provide a sketch. By exploiting the sparsity of $\mathbf{v}^t$ and the upper bound on the $\|\cdot\|_{2,\infty}$ norm of $\tilde{\nabla}_k^2$, we can obtain the following variance bound on $\mathbf{g}^t$, which has an explicit dependence on the problem dimension d.

LEMMA G.1. *Recall the constructions of the gradient estimator* (17) *and the Hessian estimator* (34). *In the multilinear extension problem* (31), *under Definition* 5.2, *we have the following variance bound:*

$$\mathbb{E}[\|\mathbf{g}^t - \nabla F(\mathbf{x}^t)\|^2] \le \frac{4r^2 d\cdot\epsilon}{|\mathcal{M}|}D_f^2 + \frac{dD_f^2}{|\mathcal{M}_0|}. \tag{63}$$

By choosing $|\mathcal{M}| = \frac{2}{\epsilon}$ and $|\mathcal{M}_0| = \frac{1}{2r^2\epsilon^2}$, we have $\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2] \le 4r^2 d\cdot\epsilon^2 D_f^2$. Further, from Taylor's expansion we can prove

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) - \langle\nabla F(\mathbf{x}^t), \mathbf{x}^{t+1}-\mathbf{x}^t\rangle \ge -\frac{1}{2}\sqrt{rdD_f^2}\cdot\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2. \tag{64}$$

Following the update rule of SCG++, the function value can be proved to decrease in each iteration: for $T = \frac{1}{\epsilon}$

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \ge \mathbb{E}[F(\mathbf{x}^t)] + \epsilon\mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^t)] - 6\sqrt{r^3 dD_f^2}\cdot\epsilon^2,$$

which can be translated to $\mathbb{E}[F(\mathbf{x}^*)-F(\mathbf{x}^{\frac{1}{\epsilon}})] \leq (1-\epsilon)^{\frac{1}{\epsilon}}[F(\mathbf{x}^*)-F(\mathbf{x}^0)]-6\sqrt{r^3 d}\cdot D_f \cdot \epsilon$. In conclusion, we have $\mathbb{E}[F(\mathbf{x}^{\frac{1}{\epsilon}})] \geq (1-1/e)F(\mathbf{x}^*) - 6\sqrt{r^3 d}\cdot D_f \cdot \epsilon$.

Appendix H. Proof of Theorem 5.5. From (64) we have

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) - \langle \nabla F(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t \rangle \geq -\frac{1}{2}\sqrt{rdD_f^2} \cdot \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2. \tag{65}$$

By using the above result and following the similar proof in Appendix H of [42], we arrive at the following inequality:

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \geq \frac{1}{T}\left[\left(1-\frac{1}{T}\right)^t F(\mathbf{x}^*) - F(\mathbf{x}^t)\right] - \frac{2r}{T}\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\| - \frac{\sqrt{r^3 dD_f^2}}{2T^2}. \tag{66}$$

By using Lemma G.1 and by choosing $|\mathcal{M}| = \frac{2}{\epsilon}$ and $|\mathcal{M}_0| = \frac{1}{2r^2\epsilon^2}$, we have

$$\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|] \leq \sqrt{\mathbb{E}[\|\nabla F(\mathbf{x}^t) - \mathbf{g}^t\|^2]} \leq 2r\sqrt{d}\epsilon D_f. \tag{67}$$

Take expectation on both sides of (66) and plug in the above variance bound to arrive at

$$\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \left(1-\frac{1}{T}\right)\mathbb{E}[F(\mathbf{x}^t)] + \frac{1}{T}\left(1-\frac{1}{T}\right)^t F(\mathbf{x}^*) - \frac{5}{2T^2}\sqrt{r^3 dD_f^2}. \tag{68}$$

By multiplying $(1-\frac{1}{T})^{-(t+1)}$ on both sides of (68), we have

$$\left(1-\frac{1}{T}\right)^{-(t+1)}\mathbb{E}[F(\mathbf{x}^{t+1})] \geq \left(1-\frac{1}{T}\right)^{-t}\mathbb{E}[F(\mathbf{x}^t)] + \frac{1}{T}\left(1-\frac{1}{T}\right)^{-1}F(\mathbf{x}^*) - \frac{5\sqrt{r^3 dD_f^2}}{2T^2(1-\frac{1}{T})^{(t+1)}}.$$

Sum the above inequality from $t = 0$ to $T-1$ and use $F(\mathbf{x}^0) \geq 0$ to obtain

$$\begin{aligned}
&(1-T^{-1})^{-T}\mathbb{E}[F(\mathbf{x}^T)] \\
&\geq F(\mathbf{x}^0) + (1-T^{-1})^{-1}F(\mathbf{x}^*) - 2.5T^{-2}\sqrt{r^3 dD_f^2}\cdot[(1-T^{-1})^{-T}-1](T-1) \\
&\geq (1-T^{-1})^{-1}F(\mathbf{x}^*) - 2.5T^{-2}\sqrt{r^3 dD_f^2}\cdot[(1-T^{-1})^{-T}-1](T-1). \qquad (69)
\end{aligned}$$

Multiply $(1-T^{-1})^T$ on both sides of (69) to derive

$$\begin{aligned}
\mathbb{E}[F(\mathbf{x}^T)] &\geq (1-T^{-1})^{T-1}F(\mathbf{x}^*) - 2.5T^{-2}\sqrt{r^3 dD_f^2}\cdot[1-(1-T^{-1})^T](T-1) \\
&\geq e^{-1}F(\mathbf{x}^*) - 2.5T^{-1}\sqrt{r^3 dD_f^2}, \qquad (70)
\end{aligned}$$

where we use $(1-T^{-1})^{T-1} \geq e^{-1}$.

Appendix I. Proof of Theorem 5.6. Theorem 5.3 is identical to Theorem 3 of the conference version of this paper [33], and we refer the reader to the detailed proof therein. Here we only provide a sketch. Our goal is to construct a submodular function f, defined through the expectation $f(S) = \mathbb{E}_{\gamma\sim p(\gamma)}[f_\gamma(S)]$, such that obtaining a $(1-1/e-\epsilon)$-optimal solution for maximizing f under the k-cardinality constraint requires at least $\min\{\exp(\alpha(\epsilon)k), O(1/\epsilon^2)\}$ i.i.d. samples $f_\gamma(\cdot)$. For maximizing monotone submodular functions under the k-cardinality constraint, we know that going beyond the approximation factor $(1-1/e)$ is computationally hard. In other words, one can construct a specific monotone submodular set function (call it f_0) such that finding a $(1 - 1/e + \delta)$-optimal solution requires at least $\exp\{\alpha(\delta)k\}$ function queries. The main idea of the proof is to slightly change the value of f_0 by adding Bernoulli random variables whose success probabilities are small—say, of order ϵ—such that the following holds: In order to obtain a $(1-1/e-\epsilon)$-optimal solution for the new function f under the cardinality constraint, one would need to either find a $(1-1/e+\epsilon/4)$-optimal solution for f_0 (which requires exponentially many samples) or accurately estimate the parameters of the added Bernoulli random variables—a task that is known information theoretically to require at least $O(1/\epsilon^2)$ i.i.d. samples from the Bernoulli random variables. The function f is the desired stochastic function of the theorem.

REFERENCES

[1] Z. Allen-Zhu, *Natasha 2: Faster non-convex optimization than SGD*, in Advances in Neural Information Processing Systems 32, Curran Associates, Red Hook, NY, 2018, pp. 2680–2691.

[2] Z. Allen-Zhu and E. Hazan, *Variance reduction for faster non-convex optimization*, in Proceedings of the 33rd International Conference on Machine Learning, New York, NY, 2016, pp. 699–707.

[3] F. Bach, *Submodular functions: From discrete to continuous domains*, Math. Program., 175 (2019), pp. 419–459.

[4] A. Bian, K. Levy, A. Krause, and J. M. Buhmann, *Continuous DR-submodular maximization: Structure and algorithms*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2017, pp. 486–496.

[5] A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause, *Guaranteed non-convex optimization: Submodular maximization over continuous domains*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, 2017.

[6] G. Braun, S. Pokutta, and D. Zink, *Lazifying conditional gradient algorithms*, J. Mach. Learn. Res., 20 (2019), 71.

[7] N. Buchbinder and M. Feldman, *Deterministic algorithms for submodular maximization problems*, ACM Trans. Algorithms, 14 (2018), 32.

[8] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, *Maximizing a submodular set function subject to a matroid constraint (extended abstract)*, in Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization, IPCO '07, Ithaca, NY, 2007, pp. 182–196.

[9] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, *Maximizing a monotone submodular function subject to a matroid constraint*, SIAM J. Comput., 40 (2011), pp. 1740–1766, https://doi.org/10.1137/080733991.

[10] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *"Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions*, in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017.

[11] C. Cartis, N. Gould, and P. Toint, *An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity*, IMA J. Numer. Anal., 32 (2012), pp. 1662–1695.

[12] C. Cartis, N. Gould, and P. Toint, *On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization*, SIAM J. Optim., 23 (2013), pp. 1553–1574, https://doi.org/10.1137/120869687.

[13] C. Cartis, N. Gould, and P. Toint, *On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods*, SIAM J. Numer. Anal., 53 (2015), pp. 836–851, https://doi.org/10.1137/130915546.

[14] C. Chekuri, T. Jayram, and J. Vondrák, *On multiplicative weight updates for concave and submodular function maximization*, in Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS '15, Rehovot, Israel, 2015.

[15] C. Chekuri, J. Vondrák, and R. Zenklusen, *Submodular function maximization via the multilinear relaxation and contention resolution schemes*, SIAM J. Comput., 43 (2014), pp. 1831–1879, https://doi.org/10.1137/110839655.

[16] M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett, *Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks*, J. Mach. Learn. Res., 9 (2008), pp. 1775–1822.
[17] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Advances in Neural Information Processing Systems 27, Curran Associates, Red Hook, NY, 2014.
[18] C. Fang, C. J. Li, Z. Lin, and T. Zhang, *SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2018.
[19] C. Fang, Z. Lin, and T. Zhang, *Sharp analysis for nonconvex SGD escaping from saddle points*, in Proceedings of the Thirty-Second Conference on Learning Theory, Phoenix, AZ, 2019, pp. 1192–1234
[20] M. Feldman, J. S. Naor, and R. Schwartz, *A unified continuous greedy algorithm for submodular maximization*, in Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, FOCS '11, IEEE, Washington, DC, 2011, pp. 570–579.
[21] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logistics Quart., 3 (1956), pp. 95–110.
[22] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed., Ann. Discrete Math. 58, Elsevier, New York, 2005.
[23] S. Fujishige and S. Isotani, *A submodular function minimization algorithm based on the minimum-norm base*, Pac. J. Optim., 7 (2011), pp. 3–17.
[24] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99.
[25] S. Ghadimi, G. Lan, and H. Zhang, *Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization*, Math. Program., 155 (2016), pp. 267–305.
[26] S. O. Gharan and J. Vondrák, *Submodular maximization by simulated annealing*, in Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, ACM, New York, SIAM, Philadelphia, 2011, pp. 1098–1116, https://doi.org/10.1137/1.9781611973082.83.
[27] J. Gillenwater, A. Kulesza, and B. Taskar, *Near-optimal map inference for determinantal point processes*, in Advances in Neural Information Processing Systems 25, Curran Associates, Red Hook, NY, 2012.
[28] H. Hassani, M. Soltanolkotabi, and A. Karbasi, *Gradient methods for submodular maximization*, in Advances in Neural Information Processing Systems 30, Curran Associates, Red Hook, NY, 2017.
[29] E. Hazan and S. Kale, *Projection-free online learning*, in Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, 2012.
[30] E. Hazan and H. Luo, *Variance-reduced and projection-free stochastic optimization*, in Proceedings of the 33rd International Conference on Machine Learning, New York, NY, 2016, pp. 1263–1271.
[31] M. Jaggi, *Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, in Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, 2013, pp. 427–435.
[32] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13, Curran Associates, Red Hook, NY, 2013.
[33] A. Karbasi, H. Hassani, A. Mokhtari, and Z. Shen, *Stochastic continuous greedy++: When upper and lower bounds match*, in Advances in Neural Information Processing Systems, Curran Associates, Red Hook, NY, 2019.
[34] M. Karimi, M. Lucic, H. Hassani, and A. Krause, *Stochastic submodular maximization: The case of coverage functions*, in Advances in Neural Information Processing Systems, Curran Associates, Red Hook, NY, 2017.
[35] A. Kulesza and B. Taskar, *Determinantal Point Processes for Machine Learning*, Now Publishers, Hanover, MA, 2012.
[36] S. Lacoste-Julien, *Convergence Rate of Frank-Wolfe for Non-Convex Objectives*, preprint, https://arxiv.org/abs/1607.00345, 2016.
[37] G. Lan, *Gradient sliding for composite optimization*, Math. Program., 159 (2016), pp. 201–235.
[38] G. Lan, S. Pokutta, Y. Zhou, and D. Zink, *Conditional accelerated lazy stochastic gradient descent*, in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017, pp. 1965–1974.
[39] G. Lan and Y. Zhou, *Conditional gradient sliding for convex optimization*, SIAM J. Optim., 26 (2016), pp. 1379–1409, https://doi.org/10.1137/140992382.

[40] L. Lei, C. Ju, J. Chen, and M. I. Jordan, *Non-convex finite-sum optimization via SCSG methods*, in Advances in Neural Information Processing Systems 30, Curran Associates, Red Hook, NY, 2017.

[41] A. Mokhtari, H. Hassani, and A. Karbasi, *Conditional gradient method for stochastic submodular maximization: Closing the gap*, in Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Playa Blanca, Canary Islands, 2018, pp. 1886–1895.

[42] A. Mokhtari, H. Hassani, and A. Karbasi, *Stochastic conditional gradient methods: From convex minimization to submodular maximization*, J. Mach. Learn. Res., 21 (2020), 105.

[43] A. Mokhtari, A. Ozdaglar, and A. Jadbabaie, *Escaping saddle points in constrained optimization*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2018.

[44] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, *An analysis of approximations for maximizing submodular set functions.* I, Math. Program., 14 (1978), pp. 265–294.

[45] A. Nemirovski and D. B. Yudin, *On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions*, Soviet Math. Dokl., 239 (1978), pp. 1056–1059.

[46] A. Nemirovskii, D. B. Yudin, and E. R. Dawson, *Problem Complexity and Method Efficiency in Optimization*, Wiley, New York, 1983.

[47] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, *SARAH: A novel method for machine learning problems using stochastic recursive gradient*, in Proceedings of the 34th International Conference on Machine Learning - Vol. 70, ICML'17, Sydney, Australia, 2017, pp. 2613–2621.

[48] R. Niazadeh, N. Golrezaei, J. Wang, F. Susan, and A. Badanidiyuru, *Online Learning via Offline Greedy Algoritihms: Applications in Market Design and Optimization*, Chicago Booth Res. paper 20-27; available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3613756#, 2020.

[49] C. Qu, Y. Li, and H. Xu, *Non-convex conditional gradient sliding*, in Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 2018, pp. 4208–4217.

[50] S. J. Reddi, A. Hefny, S. Sra, B. Póczos, and A. Smola, *Stochastic variance reduction for nonconvex optimization*, in Proceedings of the 33rd International Conference on Machine Learning, New York, NY, 2016, pp. 314–323.

[51] S. J. Reddi, S. Sra, B. Póczos, and A. Smola, *Fast incremental method for smooth nonconvex optimization*, in Proceedings of the 55th IEEE Conference on Decision and Control (CDC), IEEE, Washington, DC, 2016, pp. 1971–1977.

[52] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics, 22 (1951), pp. 400–407.

[53] T. Roughgarden and J. R. Wang, *An optimal learning algorithm for online unconstrained submodular maximization*, in Proceedings of the 31st Conference on Learning Theory, Singapore, 2018, pp. 1307–1325.

[54] M. Schmidt, N. Le Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Math. Program., 162 (2017), pp. 83–112.

[55] Z. Shen, C. Fang, P. Zhao, J. Huang, and H. Qian, *Complexities in projection-free stochastic non-convex minimization*, Proc. Mach. Learn. Res., 89 (2019), pp. 2868–2876.

[56] Z. Shen, A. Ribeiro, H. Hassani, H. Qian, and C. Mi, *Hessian aided policy gradient*, in Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, 2019, pp. 5729–5738.

[57] T. Soma and Y. Yoshida, *A generalization of submodular cover via the diminishing return property on the integer lattice*, in Advances in Neural Information Processing Systems 28, Curran Associates, Red Hook, NY, 2015.

[58] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 2018.

[59] M. Sviridenko, J. Vondrák, and J. Ward, *Optimal approximation for submodular and supermodular optimization with bounded curvature*, in Proceedings of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '15, ACM, New York, SIAM, Philadelphia, 2015, pp. 1134–1148, https://doi.org/10.1137/1.9781611973730.76.

[60] J. Vondrák, *Optimal approximation for the submodular welfare problem in the value oracle model*, in Proceedings of ACM Symposium on Theory of Computing, STOC, ACM, New York, 2008, pp. 67–74.

[61] L. A. Wolsey, *An analysis of the greedy algorithm for the submodular set covering problem*, Combinatorica, 2 (1982), pp. 385–393.

[62] A. Yurtsever, S. Sra, and V. Cevher, *Conditional gradient methods via stochastic path-integrated differential estimator*, in Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, 2019, pp. 7282–7291.

[63] M. Zhang, L. Chen, A. Mokhtari, H. Hassani, and A. Karbasi, *Quantized Frank-Wolfe: Faster optimization, lower communication, and projection free*, in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, held online, 2020, pp. 3696–3706.

[64] M. Zhang, Z. Shen, A. Mokhtari, H. Hassani, and A. Karbasi, *One sample stochastic Frank-Wolfe*, in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, held online, 2020, pp. 4012–4023.