
Stochastic Frank-Wolfe Methods for Nonconvex Optimization

Sashank J. Reddi

Carnegie Mellon University
sjakkamr@cs.cmu.edu

Suvrit Sra

Massachusetts Institute of Technology
suvrit@mit.edu

Barnabás Póczos

Carnegie Mellon University
bapoczso@cs.cmu.edu

Alexander J. Smola

Carnegie Mellon University
alex@smola.org

Abstract

We study Frank-Wolfe methods for *nonconvex* stochastic and finite-sum optimization problems. Frank-Wolfe methods (in the convex case) have gained tremendous recent interest in machine learning and optimization communities due to their projection-free property and their ability to exploit structured constraints. However, our understanding of these algorithms in the nonconvex setting is fairly limited. In this paper, we propose nonconvex stochastic Frank-Wolfe methods and analyze their convergence properties. Furthermore, for objective functions that decompose into a finite-sum, we leverage ideas from variance reduction for convex optimization to obtain new variance reduced nonconvex Frank-Wolfe methods that have provably faster convergence than the classical Frank-Wolfe method.

1 INTRODUCTION

We study optimization problems of the form:

$$\min_{x \in \Omega} F(x) := \begin{cases} \mathbb{E}_z[f(x, z)], & \text{(stochastic)} \\ \frac{1}{n} \sum_{i=1}^n f_i(x), & \text{(finite-sum)}. \end{cases} \quad (1)$$

We assume that F , f , and f_i ($i \in \{1, \dots, n\} \triangleq [n]$) are all differentiable, but possibly *nonconvex*; the domain Ω is *convex* and *compact*.

Problems of this form are at the heart of machine learning and statistics; for instance, the finite-sum problem arises under the name empirical loss minimization and M-estimation. Examples of such problems include multiclass classification, matrix learning, recommendation systems [13, 11, 12].

Within convex optimization, problem (1) is relatively well-studied. Two particularly popular approaches for solving it are: (i) Projected stochastic gradient descent (SGD); and (b) the Frank-Wolfe (FW) method. At each iteration, SGD takes a step in a direction opposite to a stochastic approximation of the gradient ∇F and uses projection onto Ω to ensure feasibility. While computing a stochastic approximation to ∇F is usually inexpensive, in many real settings, the cost projecting onto Ω can be very high (e.g., projecting onto the trace-norm ball, onto base polytopes in submodular minimization [6]); and in extreme cases projection can even be computationally intractable [3].

In such cases, projection based methods like SGD become impractical. This difficulty underlies the recent surge of interest in Frank-Wolfe methods [5, 13] (also known as conditional gradient), due to their projection-free property. In particular, FW methods avoid the expensive projection operation and requires just a *linear oracle* that solves problems of the form $\min_{x \in \Omega} \langle x, g \rangle$ at each iteration.

Algorithm	SFO/IFO Complexity	LO Complexity
Frank-Wolfe	$O(n/\epsilon^2)$	$O(1/\epsilon^2)$
SFW	$O(1/\epsilon^4)$	$O(1/\epsilon^2)$
SVFW	$O(n + n^{2/3}/\epsilon^2)$	$O(1/\epsilon^2)$
SAGAFW	$O(n + n^{1/3}/\epsilon^2)$	$O(1/\epsilon^2)$

Figure 1: Table comparing the *best* SFO/IFO and LO complexity of algorithms discussed in the paper. The complexity is measured by the number of oracle calls required to achieve an ϵ -accurate solution (see Section 2 for definitions of SFO/IFO and LO complexity). The results marked in red are contributions of this paper. For clarity, we hide the dependence of SFO/IFO and LO complexity on the initial point and few parameters related to the function F and domain Ω .

Despite the remarkable success of FW approaches in the convex setting, including stochastic problems [12], their applicability and *non-asymptotic convergence* for *nonconvex* optimization is largely unstudied. Even for SGD, it is only recently that non-asymptotic convergence analysis for nonconvex optimization was obtained [8, 9]. More recently, [19, 20] obtained variance reduced stochastic methods that converge faster than SGD in the *nonconvex* finite-sum setting.

Similar fast variants of FW for nonconvex problems are not known. Given the vast importance of nonconvex models in machine learning (e.g., in deep learning) and the need to incorporate non-trivial constraints in such models, it is imperative to develop scalable, projection-free methods. This paper presents new FW methods towards this goal. Our main contributions are summarized below, while the key complexity results are listed in Figure 1.

Main Contributions. For the nonconvex stochastic setting in (1) we propose a stochastic Frank-Wolfe algorithm (SFW), and provide its convergence analysis. For the nonconvex finite-sum setting, we propose two variance reduced (VR) algorithms: SVFW and SAGAFW, based on the popular VR algorithms SVRG and SAGA, respectively. By carefully selecting the parameters of these algorithms, we prove that SVFW and SAGAFW are faster than deterministic FW by a factor of $n^{1/3}$ and $n^{2/3}$ respectively. To our knowledge, our work presents the first theoretical improvement for stochastic variants of Frank-Wolfe in the context of nonconvex optimization.

We refer the readers to the Appendix for the related work.

2 PRELIMINARIES

As stated above, we study two different problem settings: (1) *stochastic*, where $F(x) = \mathbb{E}_z[f(x, z)]$ and z is random variable whose distribution \mathcal{P} is supported on $\Xi \subset \mathbb{R}^p$; and (2) *finite-sums*, where $F(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$. For the stochastic setting, we assume that F is L -smooth, i.e., its gradient is Lipschitz continuous with constant L , so $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|$, $\forall x, y \in \Omega$. Here $\|\cdot\|$ denotes the ℓ^2 -norm. Furthermore, for the stochastic setting, we also assume $f(x, z)$ is G -Lipschitz for all $z \in \Xi$. This assumption can be equivalently written as $\|\nabla f(x, z)\| \leq G$ for all $z \in \Xi$.

For the finite-sum setting, we assume that the individual functions f_i ($i \in [n]$) are L -smooth i.e., $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, $\forall x, y \in \Omega$. Note that this implies that the function F is also L -smooth. The domain $\Omega \in \mathbb{R}^d$ is assumed to be convex and compact with diameter D ; i.e., $\|x - y\| \leq D$ for all $x, y \in \Omega$. Such an assumption is common to all Frank-Wolfe methods.

Convergence criteria. We use the following criterion, typically referred to the *Frank-Wolfe gap*, for our convergence analysis :

$$\mathcal{G}(x) = \max_{v \in \Omega} \langle v - x, -\nabla F(x) \rangle. \quad (2)$$

For convex functions, the FW gap provides an upper bound on the suboptimality. For nonconvex functions, the gap $\mathcal{G}(x) = 0$ if and only if x is a stationary point. To state our convergence results we will also need the following bound: $\beta \geq 2(F(x_0) - F(x^*)) / LD^2$, given some (unspecified) initial point $x_0 \in \Omega$.

Algorithm 1 Nonconvex SFW $(x_0, T, \{\gamma_i\}_{i=0}^{T-1}, \{b_i\}_{i=0}^{T-1})$

- 1: **Input:** $x_0 \in \Omega$, number of iterations T , $\{\gamma_i\}_{i=0}^{T-1}$ where $\gamma_i \in [0, 1]$ for all $i \in \{0, \dots, T-1\}$, minibatch size $\{b_i\}_{i=0}^{T-1}$
 - 2: **for** $t = 0$ **to** $T - 1$ **do**
 - 3: Uniformly randomly pick i.i.d samples $\{z_1^t, \dots, z_{b_t}^t\}$ according to the distribution \mathcal{P} .
 - 4: Compute $v_t = \arg \max_{v \in \Omega} \langle v, -\frac{1}{b_t} \sum_{i=1}^{b_t} \nabla f(x_t, z_i) \rangle$
 - 5: $x_{t+1} = x_t + \gamma_t(v_t - x_t)$
 - 6: **end for**
 - 7: **Output:** Iterate x_a chosen uniformly random from $\{x_t\}_{t=0}^{T-1}$.
-

Oracle model. To compare convergence speed of different algorithms, we use the following black-box oracles: (i) *Stochastic First-Order Oracle* (SFO): For a function $F(\cdot) = \mathbb{E}_z[f(\cdot, z)]$ where $z \sim \mathcal{P}$, an SFO takes a point x and returns the pair $(f(x, z'), \nabla f(x, z'))$ where z' is a sample drawn i.i.d. from \mathcal{P} [17], (ii) *Incremental First-Order Oracle* (IFO): For a function $F(\cdot) = \frac{1}{n} \sum_i f_i(\cdot)$, an IFO takes an index $i \in [n]$ and a point $x \in \mathbb{R}^d$, and returns the pair $(f_i(x), \nabla f_i(x))$ [1] and (iii) *Linear Optimization Oracle* (LO): For a set Ω , an LO takes a direction d and returns $\arg \max_{v \in \Omega} \langle v, d \rangle$.

Throughout the paper, by SFO, IFO and LO complexity of an algorithm, we mean the total number of SFO, IFO and LO calls made by the algorithm to obtain an ϵ -accurate solution, i.e., a solution for which $\mathbb{E}[\mathcal{G}(x)] \leq \epsilon$; the expectation is over any randomization as part of the algorithm. For clarity of presentation, we hide the dependence of these complexities on the initial point $F(x_0) - F(x^*)$, Lipschitz constant G , and the smoothness constant L .

Classical Fw. To place our results in perspective, we begin by recalling the classical Frank-Wolfe (Fw) algorithm [5]. Each iteration of Fw entails calculation of the gradient ∇F and moving towards a minimizer of a linearized objective. Notice that calculation of ∇F may not be possible in the stochastic setting of (1). Furthermore, even in the finite-sum setting, computing ∇F requires n IFO calls, rendering the approach useless in large-scale problems, where n is large. For the nonconvex finite-sum setting, the following key result was proved recently [15].

Theorem 1 [15]. *Under appropriate selection of step sizes γ_t , the IFO and LO complexity of Fw to achieve an ϵ -accurate solution in the finite-sum setting are $O(n/\epsilon^2)$ and $O(1/\epsilon^2)$ respectively.*

3 ALGORITHMS

In this section, we describe Fw algorithms for solving (1). In particular, we explore stochastic and VR versions of the classical Fw method, for the stochastic and finite-sum settings, respectively.

3.1 STOCHASTIC SETTING

We first investigate the convergence of Fw in the stochastic setting. As mentioned earlier, the classical Fw method requires calculation of the full gradient $\nabla F(x)$, which is typically impossible to compute in the stochastic setting. For convex problems, [12] tackle this issue by using the popular Robbins-Monro approximation [22] to the gradient. We use a variant of the algorithm for our nonconvex stochastic setting, which we call SFW.

The pseudocode of SFW is listed in Algorithm 1. Note that the samples $\{z_i\}$ are chosen independently according to the distribution \mathcal{P} . Thus, $\mathbb{E}_{z_i}[\nabla f(x, z_i)] = \nabla F(x)$, i.e., we obtain an unbiased estimate of the gradient. Also, note that the output in Algorithm 1 is randomly selected from all the iterates of the algorithm. For our analysis, we assume that the function $f(x, z)$ is G -Lipschitz for all $z \in \Xi$, i.e., we have $\max_{z \in \Xi} \|\nabla f(x, z)\| \leq G$. This bound on the gradient is crucial for our convergence analysis. We prove the following key result for nonconvex SFW.

Theorem 2. *Consider the stochastic setting of (1) where $f(x, z)$ is G -Lipschitz for all $z \in \Xi$ and F is L -smooth. Then, the output x_a of Algorithm 1 with parameters $\gamma_t = \gamma = \sqrt{\frac{2(F(x_0) - F(x^*))}{TL D^2 \beta}}$, $b_t = b = T$ for all $t \in \{0, \dots, T-1\}$, satisfies the following bound:*

$$\mathbb{E}[\mathcal{G}(x_a)] \leq \frac{D}{\sqrt{T}} \left(G + \sqrt{\frac{2L(F(x_0) - F(x^*))}{\beta}} (1 + \beta) \right),$$

Algorithm 2 SVFW ($x_0, T, m, \{\gamma_i\}_{i=0}^{m-1}, \{b_i\}_{i=0}^{m-1}$)

- 1: **Input:** $x_m^0 = x_0 \in \Omega$, epoch size m , number of epochs $S = \lceil T/m \rceil$, $\{\gamma_i\}_{i=0}^{m-1}$ where $\gamma_i \in [0, 1]$ for all $i \in \{0, \dots, m-1\}$, minibatch size $\{b_i\}_{i=0}^{m-1}$
 - 2: **for** $s = 0$ **to** $S - 1$ **do**
 - 3: Let $\tilde{x}^s = x_m^s$
 - 4: Compute $\tilde{g}^s = \nabla F(\tilde{x}^s) = \frac{1}{n} \sum_{i=1}^n f(\tilde{x}^s)$
 - 5: **for** $t = 0$ **to** $m - 1$ **do**
 - 6: Uniformly randomly (with replacement) select subset $I_t = \{i_1, \dots, i_{b_t}\}$ from $[n]$.
 - 7: Compute $v_t^{s+1} = \arg \max_{v \in \Omega} \langle v, -\frac{1}{b_t} (\sum_{i \in I_t} \nabla f_i(x_t^{s+1}) - f_i(\tilde{x}^s) + \tilde{g}^s) \rangle$
 - 8: Compute update direction $d_t^{s+1} = v_t^{s+1} - x_t^{s+1}$
 - 9: $x_{t+1}^{s+1} = x_t^{s+1} + \gamma_t d_t^{s+1}$
 - 10: **end for**
 - 11: **end for**
 - 12: **Output:** Iterate x_a chosen uniformly random from $\{\{x_t^{s+1}\}_{t=0}^{T-1}\}_{s=0}^{S-1}$.
-

where x^* is an optimal solution to (stochastic) problem (1).

An immediate consequence of Theorem 2 is the following complexity result for SFW.

Corollary 1. *Under the setting of Theorem 2, the SFO complexity and LO complexity of Algorithm 1 are $O(1/\epsilon^4)$ and $O(1/\epsilon^2)$, respectively.*

3.2 FINITE-SUM SETTING

In this section, we consider the finite-sum setting of (1). We show that by building on ideas from variance reduction for SGD, one can significantly improve the convergence rates. The key idea is to use a variance reduced approximation of the gradient [14, 4].

SVFW Algorithm

Pseudocode of our first method (SVFW) is presented in Algorithm 2. Similar to [14] and [12], nonconvex SVFW is also epoch-based. At the end of each epoch, the full gradient is computed at the current iterate. This gradient is used for controlling the variance of the stochastic gradients in the inner loop. For epoch size $m = 1$, SVFW reduces to the classic FW algorithm. In general, the epoch size m is chosen such that the total number of IFO calls per epoch is $\Theta(n)$. This ensures that the cost of computing the full gradient at the end of each epoch is amortized. To enable a fair comparison with SFW, we assume that the total number of inner iterations across all epochs in Algorithm 2 is T .

We prove the following key result for Algorithm 2. For ease of exposition, we assume that the total number of inner iterations T is a multiple of m .

Theorem 3. *Consider the finite-sum setting of (1) where the functions $\{f_i\}_{i=1}^n$ are L -smooth. Then, the output x_a of Algorithm 2 with parameters $\gamma_t = \gamma = \sqrt{\frac{F(x_0) - F(x^*)}{TLD^2\beta}}$ and $b_t = b = m^2$ for all $t \in \{0, \dots, m-1\}$, satisfies*

$$\mathbb{E}[\mathcal{G}(x_a)] \leq \frac{2D}{\sqrt{T}\beta} \sqrt{L(F(x_0) - F(x^*))}(1 + \beta),$$

where x^* is an optimal solution of (1) and x_a is the output of Algorithm 3.

The analysis suggests that the value of m should be set appropriately in Theorem 3 to obtain good convergence rates. If m is small, the IFO complexity of Algorithm 2 is dominated by the step involving calculation of the full gradient at the end of each epoch. On the other hand, if m is large, a large minibatch is used in each step of the algorithm (since $b = m^2$), which increases the IFO complexity. With this intuition, we present following important corollary.

Corollary 2. *Under the setting of Theorem 3 and with $m = \lceil n^{1/3} \rceil$, the IFO complexity and LO complexity of Algorithm 1 are $O(n + n^{2/3}/\epsilon^2)$ and $O(1/\epsilon^2)$, respectively.*

A stronger result can be obtained by using SAGA variant of Frank-Wolfe (see Section 3.3 in the Appendix).

References

- [1] A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. *arXiv:1410.0723*, 2014.
- [2] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.
- [3] Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *JMLR*, 9:1775–1822, 2008.
- [4] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS 27*, pages 1646–1654. 2014.
- [5] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, March 1956.
- [6] Satoru Fujishige and Shiguo Isotani. A submodular function minimization algorithm based on the minimum-norm base. *Pacific Journal of Optimization*, 7(1):3–17, 2011.
- [7] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 541–549, 2015.
- [8] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [9] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, December 2014.
- [10] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, apr 2014.
- [11] Elad Hazan and Satyen Kale. Projection-free online learning. In *ICML*. icml.cc / Omnipress, 2012.
- [12] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. *CoRR*, abs/1602.02101, 2016.
- [13] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML’13*, pages 427–435, 2013.
- [14] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS 26*, pages 315–323. 2013.
- [15] Simon Lacoste-Julien. Convergence Rate of Frank-Wolfe for Non-Convex Objectives. abs/1607.00345, 2016.
- [16] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems 28*, pages 496–504, 2015.
- [17] Arkadi Nemirovski and D Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, 1983.
- [18] Yurii Nesterov. *Introductory Lectures On Convex Optimization: A Basic Course*. Springer, 2003.
- [19] Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. *arxiv:1603.06160*, 2016.
- [20] Sashank J. Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Fast incremental method for nonconvex optimization. *arxiv:1603.06159*, 2016.
- [21] Sashank J. Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv:1605.06900*, 2016.
- [22] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, sep 1951.
- [23] Mark W. Schmidt, Nicolas Le Roux, and Francis R. Bach. Minimizing Finite Sums with the Stochastic Average Gradient. *arXiv:1309.2388*, 2013.
- [24] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.
- [25] Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans. Generalized conditional gradient for sparse estimation. abs/arXiv:1410.4828, 2014.

APPENDIX

3.3 RELATED WORK

The classical Frank-Wolfe method [5] using line-search was analyzed for smooth convex functions F and polyhedral domains Ω . Here, a convergence rate of $O(1/\epsilon)$ to ensure $F(x) - F^* \leq \epsilon$ was proved without additional conditions [5, 13]. There have been several recent works on improving the convergence rates under additional assumptions [7, 16]. More recently, [12] proposed stochastic variants of FW for convex problems of form (1), and showed theoretical improvements over the classical Frank-Wolfe method.

Algorithm 3 SAGAFW ($x_0, T, \{\gamma_i\}_{i=0}^{T-1}, \{b_i\}_{i=0}^{T-1}$)

- 1: **Input:** $\alpha_0^i = x_0 \in \Omega$ for all $i \in [n]$, number of iterations T , $\{\gamma_i\}_{i=0}^{T-1}$ where $\gamma_i \in [0, 1]$ for all $i \in \{0, \dots, T-1\}$, minibatch size $\{b_i\}_{i=0}^{T-1}$
 - 2: Compute $g_0 = \frac{1}{n} \sum_{i=1}^n \nabla F(\alpha_0^i)$
 - 3: **for** $t = 0$ **to** $T - 1$ **do**
 - 4: Uniformly randomly (with replacement) select subsets I_t, J_t from $[n]$ of size b_t .
 - 5: Compute $v_t = \arg \max_{v \in \Omega} \langle v, -\frac{1}{b_t} (\sum_{i \in I_t} \nabla f_i(x_t) - f_i(\alpha_t^i) + g_t) \rangle$
 - 6: Compute update direction $d_t = v_t - x_t$
 - 7: $x_{t+1} = x_t + \gamma_t d_t$
 - 8: $\alpha_{t+1}^j = x_t$ for $j \in J_t$ and $\alpha_{t+1}^j = \alpha_t^j$ for $j \notin J_t$
 - 9: $g_{t+1} = g_t - \frac{1}{n} \sum_{j \in J_t} (\nabla f_j(\alpha_t^j) - \nabla f_j(\alpha_{t+1}^j))$
 - 10: **end for**
 - 11: **Output:** Iterate x_a chosen uniformly random from $\{x_t\}_{t=0}^{T-1}$.
-

The literature on nonconvex Frank-Wolfe is relatively small. The work [2] proves asymptotic convergence of FW to a stationary point; though, no convergence rates are provided. To the best of our knowledge, [25] is the first to provide convergence rates for FW-type algorithm in the nonconvex setting. Very recently, [15] provided a (non-asymptotic) convergence rate of $O(1/\epsilon^2)$ for nonconvex FW with adaptive step sizes. However, as we shall see later, implementation of classical FW for (1) is expensive (or impossible in the pure stochastic case) since it requires calculation of the gradient ∇F at each iteration. We show that our stochastic variants are provably faster than the existing FW methods. In the nonconvex setting, most of the work on stochastic methods focuses on SGD [8, 9] and analyzes convergence to stationary points. For the finite-sum setting, we build on recent variance reduction techniques [14, 4, 23], which were first proposed for solving unconstrained convex problems of form (1). Projected variants to handle constraints were studied in [4, 24]. More recently, [19, 20, 21] provided nonconvex variants of these methods that converge provably faster than both SGD and its deterministic counterpart.

SAGAFW Algorithm

SVFW is a semi-stochastic algorithm since it requires calculation of the full gradient at the end of each epoch. Below we propose a purely incremental method (SAGAFW) based on the SAGA algorithm of [4]. The pseudocode for SAGAFW is presented in Algorithm 3.

A key feature of SAGAFW is that it entirely avoids calculation of full gradients. Instead, it updates the average gradient vector g_t at each iteration. This update requires maintaining additional vectors α^i ($i \in [n]$), and in the worst case such a strategy incurs additional storage cost of $O(nd)$. However, this cost can be reduced to $O(n)$ in several practical cases (refer to [4, 20]).

For SAGAFW, we prove the following key result.

Theorem 4. *Consider the finite-sum setting of (1) where functions $\{f_i\}_{i=1}^n$ are L -smooth. Define $\theta(b, n, T) = 1/2 + (2n^{3/2}/Tb^{3/2})$. Then the output x_a of Algorithm 3 with parameters $\gamma_t = \gamma = \sqrt{\frac{F(x_0) - F(x^*)}{TL D^2 \theta(b, n, T) \beta}}$ and $b_t = b \leq n$ for all $t \in \{0, \dots, T-1\}$, satisfies the following:*

$$\mathbb{E}[\mathcal{G}(x_a)] \leq \frac{2D}{\sqrt{T\beta}} \sqrt{L\theta(b, n, T)(F(x_0) - F(x^*))}(1 + \beta),$$

where x^* is an optimal solution of problem (1) and x_a is the output of Algorithm 3.

Corollary 3. *Assume $T \geq n$. Under the settings of Theorem 4 and with $b = \lceil n^{1/3} \rceil$, the IFO and LO complexity of Algorithm 1 are $O(n + n^{1/3}/\epsilon^2)$ and $O(1/\epsilon^2)$, respectively.*

Proof of Theorem 2

Proof. First observe the following upper bound:

$$\begin{aligned}
F(x_{t+1}) &\leq F(x_t) + \langle \nabla F(x_t), x_{t+1} - x_t \rangle + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
&= F(x_t) + \langle \nabla F(x_t), \gamma(v_t - x_t) \rangle + \frac{L}{2} \|\gamma(v_t - x_t)\|^2 \\
&\leq F(x_t) + \gamma \langle \nabla F(x_t), v_t - x_t \rangle + \frac{LD^2\gamma^2}{2}.
\end{aligned} \tag{3}$$

The first inequality follows since F is L -smooth (see Lemma 1). The equality is due to the fact that $x_{t+1} - x_t = \gamma(v_t - x_t)$. The second inequality holds because $v_t, x_t \in \Omega$ and because the diameter of Ω is D .

Next, we introduce the following quantity:

$$\hat{v}_t := \arg \max_{v \in \Omega} \langle v, -\nabla F(x_t) \rangle, \tag{4}$$

which is used purely for our analysis and is *not* part of the algorithm. For brevity, we use ∇_t to denote $\frac{1}{b} \sum_{i=1}^b f(x_t, z_i^t)$.

Rewriting inequality (3) using this quantity, we see that

$$\begin{aligned}
F(x_{t+1}) &\leq F(x_t) + \gamma \langle \nabla_t, v_t - x_t \rangle \\
&\quad + \gamma \langle \nabla F(x_t) - \nabla_t, v_t - x_t \rangle + \frac{LD^2\gamma^2}{2} \\
&\leq F(x_t) + \gamma \langle \nabla_t, \hat{v}_t - x_t \rangle \\
&\quad + \gamma \langle \nabla F(x_t) - \nabla_t, v_t - x_t \rangle + \frac{LD^2\gamma^2}{2} \\
&\leq F(x_t) - \gamma \mathcal{G}(x_t) + \gamma \langle \nabla F(x_t) - \nabla_t, v_t - \hat{v}_t \rangle + \frac{LD^2\gamma^2}{2} \\
&\leq F(x_t) - \gamma \mathcal{G}(x_t) + D\gamma \|\nabla F(x_t) - \nabla_t\| + \frac{LD^2\gamma^2}{2}.
\end{aligned}$$

The second inequality follows from the optimality of v_t in Algorithm 1, while the third inequality follows from the definition of $\mathcal{G}(x_t)$ and the optimality of \hat{v}_t in (4). The last inequality follows from Cauchy-Schwarz and the fact that the diameter of the feasible set Ω is bounded by D .

Taking expectations and using Lemma 2 in the above bound, we obtain the following important bound:

$$\mathbb{E}[F(x_{t+1})] \leq \mathbb{E}[F(x_t)] - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{GD\gamma}{\sqrt{b}} + \frac{LD^2\gamma^2}{2}$$

Summing over t and telescoping, we then obtain

$$\begin{aligned}
\gamma \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}(x_t)] &\leq F(x_0) - \mathbb{E}[F(x_T)] + \frac{TGD\gamma}{\sqrt{b}} + \frac{TLD^2\gamma^2}{2} \\
&\leq F(x_0) - F(x^*) + \frac{TGD\gamma}{\sqrt{b}} + \frac{TLD^2\gamma^2}{2}.
\end{aligned}$$

The latter inequality follows from the optimality of x^* . Using the definition of the output x_a of Algorithm 1 and the parameters specified in the theorem statement, we get

$$\begin{aligned}
\mathbb{E}[\mathcal{G}(x_a)] &\leq \frac{F(x_0) - F(x^*)}{T\gamma} + \frac{GD}{\sqrt{b}} + \frac{LD^2\gamma}{2} \\
&\leq \frac{D}{\sqrt{T}} \left(G + \sqrt{\frac{2L(F(x_0) - F(x^*))}{\beta}} (1 + \beta) \right),
\end{aligned} \tag{5}$$

which concludes the proof of the theorem. \square

Proof of Theorem 3

Proof. We first analyze the convergence properties of iterates within an epoch. Suppose that the current epoch is $s + 1$. For brevity, we drop the symbol s from x_t^{s+1} , \tilde{x}^s and \tilde{g}^s , whenever it safe to do so given the context. The first part of the proof is similar to that of Theorem 2. We again use the quantity $\hat{v}_t = \arg \max_{v \in \Omega} \langle v, -\nabla F(x_t) \rangle$, as before, purely for our analysis.

For the t^{th} iteration within the epoch s , we have

$$\begin{aligned} F(x_{t+1}) &\leq F(x_t) + \langle \nabla F(x_t), \gamma(v_t - x_t) \rangle + \frac{L}{2} \|\gamma(v_t - x_t)\|^2 \\ &\leq F(x_t) + \gamma \langle \nabla F(x_t), v_t - x_t \rangle + \frac{LD^2\gamma^2}{2}. \end{aligned} \quad (6)$$

This is due to Lemma 1 and definition of x_{t+1} in Algorithm 2. We use $\tilde{\nabla}_t$ to denote the following quantity: $\frac{1}{b_t} (\sum_{i \in I_t} \nabla f_i(x_t) - f_i(\tilde{x}) + \tilde{g})$. Rewriting, we then obtain

$$\begin{aligned} F(x_{t+1}) &\leq F(x_t) + \gamma \langle \tilde{\nabla}_t, v_t - x_t \rangle \\ &\quad + \gamma \langle \nabla F(x_t) - \tilde{\nabla}_t, v_t - x_t \rangle + \frac{LD^2\gamma^2}{2} \\ &\leq F(x_t) + \gamma \langle \tilde{\nabla}_t, \hat{v}_t - x_t \rangle \\ &\quad + \gamma \langle \nabla F(x_t) - \tilde{\nabla}_t, v_t - x_t \rangle + \frac{LD^2\gamma^2}{2} \\ &\leq F(x_t) + \gamma \langle \nabla F(x_t), \hat{v}_t - x_t \rangle \\ &\quad + \gamma \langle \nabla F(x_t) - \tilde{\nabla}_t, v_t - \hat{v}_t \rangle + \frac{LD^2\gamma^2}{2} \\ &\leq F(x_t) - \gamma \mathcal{G}(x_t) + D\gamma \|\nabla F(x_t) - \tilde{\nabla}_t\| + \frac{LD^2\gamma^2}{2}. \end{aligned} \quad (7)$$

The second inequality is due to the optimality of v_t in Algorithm 2. The last inequality is due to the definition of $\mathcal{G}(x_t)$, the diameter of set Ω , and an application of Cauchy-Schwarz.

Taking expectations and using Lemma 3 in inequality (7) we obtain the bound

$$\begin{aligned} \mathbb{E}[F(x_{t+1})] &\leq \mathbb{E}[F(x_t)] - \gamma \mathbb{E}[\mathcal{G}(x_t)] \\ &\quad + \frac{LD\gamma}{\sqrt{b}} \mathbb{E}[\|x_t - \tilde{x}\|] + \frac{LD^2\gamma^2}{2}. \end{aligned} \quad (8)$$

To aid further analysis of this bound, we introduce the following Lyapunov function:

$$R_t = \mathbb{E}[F(x_t) + c_t \|x_t - \tilde{x}\|],$$

where $c_m = 0$ and $c_t = c_{t+1} + (LD\gamma)/\sqrt{b}$ for all $t \in \{0, \dots, m-1\}$. Using the relationship in (8), we see that

$$\begin{aligned} R_{t+1} &= \mathbb{E}[F(x_{t+1}) + c_{t+1} \|x_{t+1} - \tilde{x}\|] \\ &\leq \mathbb{E}[F(x_t)] - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{LD\gamma}{\sqrt{b}} \mathbb{E}[\|x_t - \tilde{x}\|] \\ &\quad + \frac{LD^2\gamma^2}{2} + c_{t+1} \mathbb{E}[\|x_{t+1} - \tilde{x}\|] \\ &\leq \mathbb{E}[F(x_t)] - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{LD\gamma}{\sqrt{b}} \mathbb{E}[\|x_t - \tilde{x}\|] \\ &\quad + \frac{LD^2\gamma^2}{2} + c_{t+1} \mathbb{E}[\|x_{t+1} - x_t\| + \|x_t - \tilde{x}\|] \\ &\leq R_t - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{LD^2\gamma^2}{2} + c_{t+1} D\gamma. \end{aligned} \quad (9)$$

The second inequality follows from the triangle inequality, while the last inequality holds because: (a) $c_t = c_{t+1} + (LD\gamma)/\sqrt{b}$, and (b) $\|x_{t+1} - x_t\| = \gamma \|v_t - x_t\| \leq D\gamma$ (recall the definition of

diameter of Ω). Telescoping over all the iterations within an epoch, we obtain

$$\begin{aligned} R_m &\leq R_0 - \gamma \sum_{t=0}^{m-1} \mathbb{E}[\mathcal{G}(x_t)] + \frac{LmD^2\gamma^2}{2} + D\gamma \sum_{t=1}^m c_t \\ &= R_0 - \gamma \sum_{t=0}^{m-1} \mathbb{E}[\mathcal{G}(x_t)] + \frac{LmD^2\gamma^2}{2} + \frac{L(m-1)mD^2\gamma^2}{2\sqrt{b}}. \end{aligned} \quad (10)$$

The equality follows from the relationship $c_t = c_{t+1} + (LD\gamma)/\sqrt{b}$. Since $c_m = 0$ and $x_0^{s+1} = \tilde{x}^s = x_m^s$ (in Algorithm 2), from (10) we obtain

$$\begin{aligned} \mathbb{E}[F(x_m^{s+1})] &\leq \mathbb{E}[F(x_m^s)] - \gamma \sum_{t=0}^{m-1} \mathbb{E}[\mathcal{G}(x_t^{s+1})] \\ &\quad + \frac{LmD^2\gamma^2}{2} + \frac{L(m-1)mD^2\gamma^2}{2\sqrt{b}}. \end{aligned}$$

Now telescoping over all epochs, we obtain

$$\begin{aligned} \mathbb{E}[F(x_m^S)] &\leq F(x_0) - \gamma \sum_{s=0}^{S-1} \sum_{t=0}^{m-1} \mathbb{E}[\mathcal{G}(x_t^{s+1})] \\ &\quad + \frac{TL D^2 \gamma^2}{2} + \frac{TL(m-1)D^2\gamma^2}{2\sqrt{b}}. \end{aligned}$$

Rearranging this inequality and using the definition of the output in Algorithm 2, we finally obtain

$$\begin{aligned} \mathbb{E}[\mathcal{G}(x_a)] &\leq \frac{F(x_0) - \mathbb{E}[F(x_m^S)]}{T\gamma} + \frac{LD^2\gamma}{2} + \frac{L(m-1)D^2\gamma}{2\sqrt{b}} \\ &\leq \frac{F(x_0) - F(x^*)}{T\gamma} + LD^2\gamma \\ &\leq 2\sqrt{\frac{LD^2(F(x_0) - F(x^*))}{T\beta}}(1 + \beta). \end{aligned}$$

The second inequality follows from the optimality of x^* and because $b = m^2$. The last inequality follows from the choice of γ stated in the theorem. This concludes the proof. \square

Proof of Theorem 4

Proof. We use the following quantities in our analysis:

$$\begin{aligned} \check{\nabla}_t &= \frac{1}{b_t} \sum_{i \in I_t} (\nabla f_i(x_t) - f_i(\alpha_t^i) + g_t) \\ \hat{v}_t &= \arg \max_{v \in \Omega} \langle v, -\nabla F(x_t) \rangle. \end{aligned}$$

The first part of our proof is similar to that of Theorem 3. Using essentially the same argument until (7), we have

$$\begin{aligned} &\mathbb{E}[F(x_{t+1})] \\ &\leq F(x_t) - \gamma \mathcal{G}(x_t) + D\gamma \|\nabla F(x_t) - \check{\nabla}_t\| + \frac{LD^2\gamma^2}{2} \\ &\leq F(x_t) - \gamma \mathcal{G}(x_t) + \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \frac{1}{n} \sum_{i=1}^n \mathbb{E}\|x_t - \alpha_t^i\| + \frac{LD^2\gamma^2}{2}. \end{aligned} \quad (11)$$

The second inequality is due to Lemma 4. Next, we define the following Lyapunov function:

$$R_t = \mathbb{E}[F(x_t)] + c_{t+1} \frac{1}{n} \sum_{i=1}^n \mathbb{E}\|x_t - \alpha_t^i\|,$$

where $c_T = 0$ and $c_t = (1 - \rho)c_{t+1} + (LD\gamma\sqrt{n})/\sqrt{b}$ for all $t \in \{0, \dots, T-1\}$, where ρ is the probability $1 - (1 - 1/n)^b$ of an index i being in J_t . We can bound ρ from below as

$$\rho = 1 - \left(1 - \frac{1}{n}\right)^b \geq 1 - \frac{1}{1+(b/n)} = \frac{b/n}{1+b/n} \geq \frac{b}{2n}, \quad (12)$$

where the first inequality follows from $(1-y)^r \leq 1/(1+ry)$ (which holds for $y \in [0, 1]$ and $r \geq 1$), while the second inequality holds because $b \leq n$. Now observe the following:

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n \mathbb{E} \|x_{t+1} - \alpha_{t+1}^i\| \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{E} [\rho \|x_{t+1} - x_t\| + (1-\rho) \|x_{t+1} - \alpha_{t+1}^i\|] \\ &\leq \frac{1}{n} \sum_{i=1}^n \mathbb{E} \left[\rho \|x_{t+1} - x_t\| \right. \\ &\quad \left. + (1-\rho) (\|x_{t+1} - x_t\| + \|x_t - \alpha_{t+1}^i\|) \right] \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{E} [\|x_{t+1} - x_t\| + (1-\rho) \|x_t - \alpha_{t+1}^i\|] \end{aligned} \quad (13)$$

The first equality follows from the definition of α_{t+1}^i in Algorithm 3, while the inequality is just the triangle inequality. Using the above relationship and the bound in (11), we obtain

$$\begin{aligned} R_{t+1} &\leq \mathbb{E}[F(x_t)] - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{LD^2\gamma^2}{2} \\ &\quad + \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|x_t - \alpha_t^i\|] + c_{t+1} \mathbb{E}[\|x_{t+1} - x_t\|] \\ &\quad + c_{t+1}(1-\rho) \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|x_t - \alpha_t^i\|] \\ &\leq R_t - \gamma \mathbb{E}[\mathcal{G}(x_t)] + \frac{LD^2\gamma^2}{2} + c_{t+1}D\gamma. \end{aligned} \quad (14)$$

The second inequality holds because: (a) $c_t = (1 - \rho)c_{t+1} + (LD\gamma\sqrt{n})/\sqrt{b}$, and (b) $\|x_{t+1} - x_t\| = \gamma \|v_t - x_t\| \leq D\gamma$ (due to our bound on the diameter of the set Ω). Telescoping over all the iterations, we see that

$$\begin{aligned} R_T &\leq R_0 - \gamma \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}(x_t)] + \frac{TLD^2\gamma^2}{2} + D\gamma \sum_{t=1}^T c_t \\ &\leq R_0 - \gamma \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}(x_t)] + \frac{TLD^2\gamma^2}{2} + \frac{LD^2\gamma^2\sqrt{n}}{\rho\sqrt{b}} \\ &\leq R_0 - \gamma \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}(x_t)] + \frac{TLD^2\gamma^2}{2} + \frac{2LD^2\gamma^2n^{3/2}}{b^{3/2}}. \end{aligned}$$

The second inequality follows from the fact that $\sum_{t=1}^T c_t \leq LD\gamma\sqrt{n}/(\rho\sqrt{b})$. This can, in turn, be obtained from the recursion $c_t = (1 - \rho)c_{t+1} + (LD\gamma\sqrt{n})/\sqrt{b}$ and $c_T = 0$. The third inequality is due to the bound on ρ in (12). Rearranging the above inequality and using the definition of x_a from Algorithm 3, we finally obtain the bound

$$\begin{aligned} \mathbb{E}[\mathcal{G}(x_a)] &\leq \frac{F(x_0) - \mathbb{E}[F(x_T)]}{T\gamma} + \frac{LD^2\gamma}{2} + \frac{2LD^2\gamma n^{3/2}}{Tb^{3/2}} \\ &\leq \frac{F(x_0) - F(x^*)}{T\gamma} + LD^2\gamma\theta(b, n, T). \end{aligned}$$

The first inequality uses the fact that $c_T = 0$ and $\alpha_0^i = x_0$ (in Algorithm 3). The second inequality uses the optimality of x^* and the definition of $\theta(b, n, T)$. Using the setting of γ in the theorem statement, we obtain the desired result. \square

The following bound on the value of functions with Lipschitz continuous gradients is classical (see e.g., [18]).

Lemma 1. *If $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is L -smooth, then*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2,$$

for all $x, y \in \mathbb{R}^d$.

The following lemma is useful for bounding the variance of the gradient estimate used in the stochastic setting.

Lemma 2. *Suppose the function $F(x) = \mathbb{E}_z[f(x, z)]$ where z is a random variable with distribution \mathcal{P} and support Ξ , and $\max_{z \in \Xi} \|\nabla f(x, z)\| \leq G$. Also, let $\bar{\nabla} = \frac{1}{b} \sum_{i \in I_t} \nabla f(x, z_i)$ where $\{z_i\}_{i=1}^b$ are i.i.d. samples from the distribution \mathcal{P} . Then, the following holds:*

$$\mathbb{E}[\|\bar{\nabla} - \nabla F(x)\|] \leq \frac{G}{\sqrt{b}}.$$

Proof. The proof follows from a simple application of Jensen's inequality. \square

The following result is useful for bounding the variance of the updates of SVFW and follows from a slight modification of a result in [19]. We give the proof here for completeness.

Lemma 3 [19]. *Let $\tilde{\nabla}_t = \frac{1}{b_t} (\sum_{i \in I_t} \nabla f_i(x_t^{s+1}) - f_i(\tilde{x}^s) + \tilde{g}^s)$ in Algorithm 2. For the iterates x_t^{s+1} and \tilde{x}^s where $t \in \{0, \dots, m-1\}$ and $s \in \{0, \dots, S-1\}$ in Algorithm 2, the following inequality holds:*

$$\mathbb{E}[\|\nabla F(x_t^{s+1}) - \tilde{\nabla}_t\|] \leq \frac{L}{\sqrt{b_t}} \|x_t^{s+1} - \tilde{x}^s\|.$$

Proof. For the ease of exposition, we first define

$$\zeta_t^{s+1} = \frac{1}{|I_t|} \sum_{i \in I_t} (\nabla f_i(x_t^{s+1}) - \nabla f_i(\tilde{x}^s)).$$

Using this notation, we then obtain the following bound:

$$\begin{aligned} & \mathbb{E}[\|\nabla F(x_t^{s+1}) - \tilde{\nabla}_t\|^2] \\ &= \mathbb{E}[\|\zeta_t^{s+1} + \nabla F(\tilde{x}^s) - \nabla F(x_t^{s+1})\|^2] \\ &= \mathbb{E}[\|\zeta_t^{s+1} - \mathbb{E}[\zeta_t^{s+1}]\|^2] \\ &= \frac{1}{b_t^2} \mathbb{E} \left[\left\| \sum_{i \in I_t} (\nabla f_i(x_t^{s+1}) - \nabla f_i(\tilde{x}^s) - \mathbb{E}[\zeta_t^{s+1}]) \right\|^2 \right]. \end{aligned}$$

The second equality is due to the fact that $\mathbb{E}[\zeta_t^{s+1}] = \nabla F(x_t^{s+1}) - \nabla F(\tilde{x}^s)$. From the above relationship, we get

$$\begin{aligned} & \mathbb{E}[\|\nabla F(x_t^{s+1}) - \tilde{\nabla}_t\|^2] \\ & \leq \frac{1}{b_t} \mathbb{E} \left[\sum_{i \in I_t} \|\nabla f_i(x_t^{s+1}) - \nabla f_i(\tilde{x}^s) - \mathbb{E}[\zeta_t^{s+1}]\|^2 \right] \\ & \leq \frac{1}{b_t} \mathbb{E} \left[\sum_{i \in I_t} \|\nabla f_i(x_t^{s+1}) - \nabla f_i(\tilde{x}^s)\|^2 \right] \\ & \leq \frac{L^2}{b_t} \mathbb{E}[\|x_t^{s+1} - \tilde{x}^s\|^2]. \end{aligned}$$

The first inequality follows from Lemma 5. The second inequality is due to the fact that for a random variable ζ , $\mathbb{E}[\|\zeta - \mathbb{E}[\zeta]\|^2] \leq \mathbb{E}[\|\zeta\|^2]$. The last inequality follows from L -smoothness of f_i . The result follows from a simple application of Jensen's inequality to the inequality above. \square

The following result is important for bounding the variance in SAGAFW. The key difference from Lemma 3 is that the variance term in SAGAFW involves α_t^i . Again, we provide the proof for completeness.

Lemma 4. *Let $\check{\nabla}_t = \frac{1}{b_t}(\sum_{i \in I_t} \nabla f_i(x_t) - f_i(\alpha_t^i) + g_t)$ in Algorithm 3. For the iterates x_t, v_t and $\{\alpha_t^i\}_{i=1}^n$ where $t \in \{0, \dots, T-1\}$ in Algorithm 3, we have the inequality*

$$\mathbb{E}[\|\nabla F(x_t) - \check{\nabla}_t\|] \leq \frac{L}{\sqrt{b_t}} \sum_{i=1}^n \frac{1}{\sqrt{n}} \mathbb{E}\|x_t - \alpha_t^i\|.$$

Proof. As before we first define the quantity

$$\zeta_t = \frac{1}{|I_t|} \sum_{i \in I_t} (\nabla f_i(x_t) - \nabla f_i(\alpha_t^i)).$$

With this notation, we then obtain the equality

$$\begin{aligned} & \mathbb{E}[\|\nabla F(x_t) - \check{\nabla}_t\|^2] \\ &= \mathbb{E} \left[\left\| \zeta_t + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\alpha_t^i) - \nabla F(x_t) \right\|^2 \right] = \mathbb{E}[\|\zeta_t - \mathbb{E}[\zeta_t]\|^2] \\ &= \frac{1}{b_t^2} \mathbb{E} \left[\left\| \sum_{i \in I_t} (\nabla f_i(x_t) - \nabla f_i(\alpha_t^i) - \mathbb{E}[\zeta_t]) \right\|^2 \right]. \end{aligned}$$

The second equality follows from the fact that $\mathbb{E}[\zeta_t] = \nabla F(x_t) - \frac{1}{n} \sum_{i=1}^n \nabla f_i(\alpha_t^i)$. From the above inequality, we get the following bound:

$$\begin{aligned} & \mathbb{E}[\|\nabla F(x_t) - \check{\nabla}_t\|^2] \\ & \leq \frac{1}{b_t} \mathbb{E} \left[\sum_{i \in I_t} \|\nabla f_i(x_t) - \nabla f_i(\alpha_t^i) - \mathbb{E}[\zeta_t]\|^2 \right] \\ & \leq \frac{1}{b_t} \mathbb{E} \left[\sum_{i \in I_t} \|\nabla f_i(x_t) - \nabla f_i(\alpha_t^i)\|^2 \right] \leq \frac{L^2}{nb_t} \sum_{i=1}^n \mathbb{E}[\|x_t - \alpha_t^i\|^2]. \end{aligned}$$

The first inequality is due to Lemma 5, while the second inequality holds because for a random variable ζ , $\mathbb{E}[\|\zeta - \mathbb{E}[\zeta]\|^2] \leq \mathbb{E}[\|\zeta\|^2]$. The last inequality is from L -smoothness of f_i ($i \in [n]$) and uniform randomness of the set I_t . By applying Jensen's inequality, we get the desired result. \square

Lemma 5. *For random variables z_1, \dots, z_r that are independent and have mean 0, we have*

$$\mathbb{E}[\|z_1 + \dots + z_r\|^2] = \mathbb{E}[\|z_1\|^2 + \dots + \|z_r\|^2].$$

Proof. Expanding the left hand side we have

$$\mathbb{E}[\|z_1 + \dots + z_r\|^2] = \sum_{i,j=1}^r \mathbb{E}[z_i z_j] = \mathbb{E} \left[\sum_{i=1}^r \|z_i\|^2 \right];$$

the second equality here follows from the our hypothesis. \square