

Stochastic L-BFGS: Improved Convergence Rates and Practical Acceleration Strategies

Renbo Zhao, *Student Member*, William B. Haskell, and Vincent Y. F. Tan, *Senior Member*

Abstract—We revisit the stochastic limited-memory BFGS (L-BFGS) algorithm. By proposing a new coordinate transformation framework for the convergence analysis, we prove improved convergence rates and computational complexities of the stochastic L-BFGS algorithms compared to previous works. In addition, we propose several practical acceleration strategies to speed up the empirical performance of such algorithms. We also provide theoretical analyses for most of the strategies. Experiments on large-scale logistic and ridge regression problems demonstrate that our proposed strategies yield significant improvements vis-à-vis competing state-of-the-art algorithms.

Index Terms—Stochastic optimization, L-BFGS algorithm, Large-scale data, Linear Convergence, Acceleration strategies

I. INTRODUCTION

We are interested in the following (unconstrained) convex finite-sum minimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left[f(\mathbf{x}) \triangleq \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) \right], \quad (1)$$

where d and n denote the ambient dimension of the decision vector and the number of component functions respectively. Problems in the form of (1) play important roles in machine learning and signal processing. One important class of such problems is the *empirical risk minimization* (ERM) problem, where each f_i assumes the form

$$f_i(\mathbf{x}) \triangleq \ell(\mathbf{a}_i^T \mathbf{x}, b_i) + \lambda R(\mathbf{x}). \quad (2)$$

In (2), $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$ denotes a smooth loss function, $R : \mathbb{R}^d \rightarrow \mathbb{R}_+$ a smooth convex regularizer (e.g., Tikhonov), $\lambda \geq 0$ the regularization weight and $\{(\mathbf{a}_i, b_i)\}_{i=1}^n \subseteq \mathbb{R}^{d+1}$ the set of feature-response pairs. Depending on the form of ℓ and R , many important machine learning problems—such as logistic regression, ridge regression and soft-margin support vector machines—are special cases of ERM.

We focus on the case where both n and d are large, and f is ill-conditioned (i.e., the condition number of f is large).¹ In the context of ERM, this means the dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^n$ that defines (1) is large and the feature vectors \mathbf{a}_i have high ambient dimension. However, the points typically belong to a

low-dimensional manifold. Such a setting is particularly relevant in the big-data era, due to unprecedented data acquisition abilities.

When n is large, the computational costs incurred by the batch optimization methods (both first- and second-order) are prohibitive, since in such methods the gradients of all the component functions $\{f_i\}_{i=1}^n$ need to be computed at each iteration. Therefore, stochastic (randomized) optimization methods have become very popular. At each iteration, only a subset of component functions, rather than all of them, are processed. In this way, for a given time budget, much more progress can be made towards global optima compared to a single-step taken for batch methods. When d is large, Newton- or quasi-Newton-based methods (both batch and stochastic) incur both high computational and storage complexities. Consequently only first-order and limited-memory quasi-Newton methods (e.g., L-BFGS [2]) are practical in this setting.

A. Related Works

When both n and d are large, as in our setting, most of research efforts have been devoted to *stochastic first-order methods*, which include stochastic gradient descent (SGD) [3], [4] and its variance-reduced modifications [5]–[8]. However, these methods do not make use of the curvature information. This limits their abilities to find highly accurate solutions for ill-conditioned problems. In order to incorporate the curvature information in the limited-memory setting, recently much progress have been made toward developing *stochastic L-BFGS* algorithm. A partial list of such works includes [9]–[17]. In particular, the first convergent algorithm was proposed by Mokhtari and Ribeiro [12]. Later, the algorithm in [15] makes use of the subsampled Hessian-vector products to form the correction pairs (as opposed to using difference of stochastic gradients) and achieves better results than previous methods. However, the convergence rate is sublinear (in the strongly-convex case), similar to that of SGD. Later, Moritz *et al.* [16] combines this method with *stochastic variance-reduced gradient* (SVRG) and proves linear convergence of the resulting algorithm. The algorithm in [17] maintains the structure of this algorithm but incorporates the *block BFGS update* to collect more curvature information in the optimization process. Although the convergence rate of this new method is similar to that in [16], experimental results demonstrate practical speedups introduced by the block BFGS update. Finally, there also exist a large volume of works on decentralized second-order methods [18]–[24] that aim to coordinate multiple distributed agents (with computational and storage abilities) in the optimization task. Since we are not concerned with

An extended abstract of this paper was accepted by UAI 2017 [1]. R. Zhao is with the Department of Electrical and Computer Engineering (ECE), the Department of Industrial and Systems Engineering (ISE) and the Department of Mathematics (Math), National University of Singapore (NUS). W. B. Haskell is with the Department of ISE, NUS. V. Y. F. Tan is with the Department ECE and the Department of Math, NUS. R. Zhao and V. Y. F. Tan are supported in part by the NUS Young Investigator Award (grant number R-263-000-B37-133) and an MoE AcRF Tier 1 Grant (R-263-000-C12-112). W. B. Haskell is supported by an MOE Tier I Grant (R-266-000-104-112).

¹In this work, the condition number of a (strongly) convex function refers to that of its Hessian.

decentralized optimization algorithms in this paper, we do not discuss these works in details here.

B. Motivations and Main Contributions

Our work can be motivated from both *theory* and *practice*. In terms of theory, although linear convergence (in expectation) has been shown for both algorithms in [16] and [17], the convergence rates (and hence computational complexities) therein can be potentially further improved. (The analysis method in [17] mainly follows that in [16], so we treat the analyses in both works in a unified manner.) In addition, these results may be strengthened in a probabilistic sense, e.g., from convergence in expectation to convergence in probability or almost surely. In terms of practice, in addition to block BFGS update, there may exist several other practical strategies that can potentially further accelerate² the algorithm in [16]. Based on these two aspects, our work consists of the following main contributions.

1) We propose a *coordinate transformation framework* to analyze stochastic L-BFGS-type algorithms in [16] and [17]. Our analysis framework yields a much improved (linear) convergence rate (both in expectation and almost surely) and computational complexity. The essential idea of our method is to unify the analysis of stochastic first-order and second-order methods; as a result, it *opens new avenues of designing and analyzing* other variants of stochastic second-order algorithms based on their first-order counterparts.

2) We conduct a *computational complexity analysis* for the stochastic L-BFGS algorithms, which is the first of its kind.

3) We propose *several practical acceleration strategies* to speed up the convergence of the stochastic L-BFGS algorithm in [16]. We demonstrate the efficacy of our strategies through numerical experiments on logistic and ridge regression problems. We also prove linear convergence for most of these strategies.

II. PRELIMINARIES

A. Notations

We use lowercase, bold lowercase and bold uppercase letters to denote scalars, vectors and matrices respectively. For a matrix $\mathbf{U} \in \mathbb{R}^{m_1 \times m_0}$, we denote its (p, q) -th entry as u_{pq} . For a function $f: \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}$, define the function $f \circ \mathbf{U}$ as the composition $(f \circ \mathbf{U})(\mathbf{z}) \triangleq f(\mathbf{U}\mathbf{z})$, for any $\mathbf{z} \in \mathbb{R}^{m_0}$. A continuously differentiable function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ is L -smooth ($L > 0$) if and only if ∇g is L -Lipschitz on \mathbb{R}^d . We use \mathbb{N} to denote the set of natural numbers. For any $n \in \mathbb{N}$, we define $[n] \triangleq \{1, \dots, n\}$ and $\{n\} \triangleq \{0, 1, \dots, n\}$. Accordingly, for a sequence of sets $\{\mathcal{A}_n\}_{n \geq 0}$, define $\mathcal{A}_{[n]} \triangleq \{\mathcal{A}_0, \dots, \mathcal{A}_n\}$, for any $n \in \mathbb{N}$. As usual, $\liminf_{n \rightarrow \infty} \mathcal{A}_n \triangleq \bigcup_{n \geq 0} \bigcap_{j \geq n} \mathcal{A}_j$ and $\limsup_{n \rightarrow \infty} \mathcal{A}_n \triangleq \bigcap_{n \geq 0} \bigcup_{j \geq n} \mathcal{A}_j$. For a set \mathcal{A} , denote its complement as \mathcal{A}^c . For any sequence $\{x_i\}_{i \geq 0}$, we define $\sum_{i=p}^q x_i \triangleq 0$ if $p > q$. We use $\|\cdot\|$ to denote both the ℓ_2 norm of a vector and the spectral norm of a matrix. We use \mathbf{B} and \mathbf{H} (with subscripts and superscripts) to denote

the approximate Hessian and approximate inverse Hessian in L-BFGS algorithms respectively, following the convention in [25]. \mathbf{H} is also known as the *metric matrix* [26]. In this work, technical lemmas (whose indices begin with ‘T’) will appear in Appendix E.

B. Assumptions on Component Functions f_i

Assumption 1. For each $i \in [n]$, f_i is convex and twice differentiable on \mathbb{R}^d . For ERM problems (2), we assume these two properties are satisfied by the loss function ℓ in its first argument on \mathbb{R} and by the regularizer R on \mathbb{R}^d .

Assumption 2. For each $i \in [n]$, f_i is μ_i -strongly convex and L_i -smooth on \mathbb{R}^d , where $0 < \mu_i \leq L_i$.

Remark 1. Assumptions 1 and 2 are standard in the analysis of both deterministic and stochastic second-order optimization methods. The strong convexity of f_i in Assumption 2 ensures positive curvature at any point in \mathbb{R}^d , which in turn guarantees the well-definedness of the BFGS update. As a common practice in the literature [15], [16], this condition can typically be enforced by adding a strongly convex regularizer (e.g., Tikhonov) to f_i . Due to the strong convexity, (1) has a unique solution, denoted as \mathbf{x}^* .

III. ALGORITHM

We will provide a refined analysis of the optimization algorithm (with some modifications) suggested in [16] and so we recapitulate it in Algorithm 1. This algorithm can be regarded as a judicious combination of SVRG and L-BFGS algorithms. We use s and t to denote the outer and inner iteration indices respectively and r to denote the index of metric matrices $\{\mathbf{H}_r\}_{r \geq 0}$. We also use $\mathbf{x}_{s,t}$ and \mathbf{x}^s to denote an inner iterate and outer iterate respectively.

Each outer iteration s consists of m inner iterations. Before the inner iterations, we first compute a *full gradient* $\mathbf{g}_s \triangleq \nabla f(\mathbf{x}^s)$. In each inner iteration (s, t) , the only modification that we make with respect to (w.r.t.) the original algorithm in [16] is that in computing the stochastic gradient $\mathbf{v}_{s,t}$, the index set $\mathcal{B}_{s,t} \subseteq [n]$ of size b is sampled with replacement *nonuniformly* [27], [28]. Specifically, the elements in $\mathcal{B}_{s,t}$ are sampled i.i.d. from a discrete distribution $P \triangleq (p_1, \dots, p_n)$, such that for any $i \in [n]$, $p_i = L_i / \sum_{j=1}^n L_j$. As will be seen in Lemma 3, compared to uniform sampling, nonuniform sampling leads to a better variance bound on the stochastic gradient $\mathbf{v}_{s,t}$. Using $\mathcal{B}_{s,t}$ and $\nabla f(\mathbf{x}^s)$, we then compute $\mathbf{v}_{s,t}$ according to (7) in Algorithm 1, where

$$\nabla f_{\mathcal{B}_{s,t}}(\mathbf{x}) \triangleq \frac{1}{b} \sum_{i \in \mathcal{B}_{s,t}} \frac{1}{np_i} \nabla f_i(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^d. \quad (3)$$

This specific way to construct $\mathbf{v}_{s,t}$ reduces the variance of $\mathbf{v}_{s,t}$ to zero as $s \rightarrow \infty$ (see Lemma 3), and serves as a crucial step in the SVRG framework.

Then we compute the search direction $\mathbf{H}_r \mathbf{v}_{s,t}$. The metric matrix \mathbf{H}_r serves as an approximate of the inverse Hessian matrix and therefore contains the local curvature information at the recent iterates. Consequently, $\mathbf{H}_r \mathbf{v}_{s,t}$ may act as a better descent direction than $\mathbf{v}_{s,t}$. Since storing \mathbf{H}_r may incur high

²In this work, we refer “acceleration” to general strategies that speed up the algorithm, not necessarily the ones based on momentum methods.

storage cost (indeed, $\Theta(d^2)$ space) for large d , (stochastic) L-BFGS-type methods compute $\mathbf{H}_r \mathbf{v}_{s,t}$ each time from a set of recent *correction pairs* \mathcal{H}_r (that only occupies $\Theta(d)$ space) and $\mathbf{v}_{s,t}$. In this way, the limitation on memory can be overcome.

Denote $M \in \mathbb{N}$ as the memory parameter. We next describe the construction of the set of recent correction pairs $\mathcal{H}_r \triangleq \{(\mathbf{s}_j, \mathbf{y}_j)\}_{j=r-M'+1}^r$, where $M' \triangleq \min\{r, M\}$. Together with $\mathbf{v}_{s,t}$, this set will be used to compute the matrix-vector product $\mathbf{H}_r \mathbf{v}_{s,t}$. Before doing so, in line 13, we first compute the averaged past iterates $\{\bar{\mathbf{x}}_r\}_{r \geq 0}$ from $\{\mathbf{x}_{s,t}\}_{s \geq 0, t \in [m]}$ for every Υ inner iterates, where $\Upsilon \in [m]$. Based on $\bar{\mathbf{x}}_{r-1}$ and $\bar{\mathbf{x}}_r$, we compute the most recent correction pair $(\mathbf{s}_r, \mathbf{y}_r)$ in line 15. Following [15], in computing \mathbf{y}_r , we first sample an index set $\mathcal{T}_r \subseteq [n]$ of size b_H uniformly without replacement, and then let $\mathbf{y}_r = \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_r$, where

$$\nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \triangleq \frac{1}{b_H} \sum_{i \in \mathcal{T}_r} \nabla^2 f_i(\bar{\mathbf{x}}_r), \quad (4)$$

denotes the sub-sampled Hessian at $\bar{\mathbf{x}}_r$. Finally, we update \mathcal{H}_{r-1} to \mathcal{H}_r by inserting $(\mathbf{s}_r, \mathbf{y}_r)$ into \mathcal{H}_{r-1} and deleting $(\mathbf{s}_{r-M'}, \mathbf{y}_{r-M'})$ from it.

Based on \mathcal{H}_r , a direct approach to compute $\mathbf{H}_r \mathbf{v}_{s,t}$ would be computing \mathbf{H}_r first and then forming the product with $\mathbf{v}_{s,t}$. Computing \mathbf{H}_r involves applying M' BFGS updates to

$$\mathbf{H}_r^{(r-M')} \triangleq \frac{\mathbf{s}_r^T \mathbf{y}_r}{\|\mathbf{y}_r\|^2} \mathbf{I} \quad (5)$$

using $\{(\mathbf{s}_j, \mathbf{y}_j)\}_{j=r-M'+1}^r$. For each $k \in \{r-M'+1, \dots, r\}$, the update is

$$\mathbf{H}_r^{(k)} = \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \right) \mathbf{H}_r^{(k-1)} \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \quad (6)$$

Finally we set $\mathbf{H}_r = \mathbf{H}_r^{(r)}$. Instead of using this direct approach, we adopt the two-loop recursion algorithm [25, Algorithm 7.4] to compute $\mathbf{H}_r \mathbf{v}_{s,t}$ as a whole. This method serves the same purpose as the direct one, but, as we shall see, with much reduced computation.

At the end of each outer iteration s , the starting point of next outer iteration, \mathbf{x}^{s+1} is either uniformly sampled (option I) or averaged (option II) from all the past inner iterates $\{\mathbf{x}_{s,t}\}_{t \in [m]}$. As shown in Theorem 1, these two options can be analyzed in a unified manner.

Remark 2. Under many scenarios (e.g., the ridge and logistic regression problems in Section VII), the smoothness parameters $\{L_i\}_{i=1}^n$ can be accurately estimated. (For ERM problems, these parameters are typically data-dependent.) If in some cases, accurate estimates of these parameters are not available, we can simply employ uniform sampling, which is a special case of our weighted sampling technique.

IV. CONVERGENCE ANALYSIS

A. Definitions

Let $\{\bar{i}_j\}_{j \in [n]}$ and $\{\bar{i}_j\}_{j \in [n]}$ be permutations of $[n]$ such that $\mu_{\min} \triangleq \mu_{\bar{i}_1} \leq \dots \leq \mu_{\bar{i}_n}$ and $L_{\bar{i}_1} \leq \dots \leq L_{\bar{i}_n} \triangleq L_{\max}$. Given

Algorithm 1 Stochastic L-BFGS Algorithm with Nonuniform Mini-batch Sampling

```

1: Input: Initial decision vector  $\mathbf{x}^0$ , mini-batch sizes  $b$  and  $b_H$ , parameters  $m, M$  and  $\Upsilon$ , step-size  $\eta$ , termination threshold  $\epsilon$ 
2: Initialize  $s := 0, r := 0, \bar{\mathbf{x}}_0 = \mathbf{0}, \mathbf{H}_0 := \mathbf{I}, \mathcal{H}_0 := \emptyset$ 
3: Repeat
4:   Compute a full gradient  $\mathbf{g}_s \triangleq \nabla f(\mathbf{x}^s)$ 
5:    $\mathbf{x}_{s,0} := \mathbf{x}^s$ 
6:   for  $t = 0, 1, \dots, m-1$ 
7:     Sample a set  $\mathcal{B}_{s,t}$  with size  $b$ 
8:     Compute a variance-reduced gradient
           
$$\mathbf{v}_{s,t} := \nabla f_{\mathcal{B}_{s,t}}(\mathbf{x}_{s,t}) - \nabla f_{\mathcal{B}_{s,t}}(\mathbf{x}^s) + \mathbf{g}_s \quad (7)$$

9:     Compute  $\mathbf{H}_r \mathbf{v}_{s,t}$  from  $\mathcal{H}_r$  and  $\mathbf{v}_{s,t}$ 
10:     $\mathbf{x}_{s,t+1} := \mathbf{x}_{s,t} - \eta \mathbf{H}_r \mathbf{v}_{s,t}$ 
11:    if  $sm + t > 0$  and  $(sm + t) \equiv 0 \pmod{\Upsilon}$ 
12:       $r := r + 1$ 
13:       $\bar{\mathbf{x}}_r := \frac{1}{\Upsilon} \left( \sum_{l=\max\{0, t-\Upsilon+1\}}^t \mathbf{x}_{s,l} + \sum_{l=\min\{m+1, t-\Upsilon+m+2\}}^m \mathbf{x}_{s-1,l} \right)$ 
14:      Sample a set  $\mathcal{T}_r$  with size  $b_H$ 
15:       $\mathbf{s}_r := \bar{\mathbf{x}}_r - \bar{\mathbf{x}}_{r-1}, \mathbf{y}_r := \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_r$ 
16:      Update  $\mathcal{H}_r := \{(\mathbf{s}_j, \mathbf{y}_j)\}_{j=r-M'+1}^r$ 
17:    end if
18:  end for
19:  Option I: Sample  $\tau_s$  uniformly randomly from  $[m]$  and set  $\mathbf{x}^{s+1} := \mathbf{x}_{s,\tau_s}$ 
20:  Option II:  $\mathbf{x}^{s+1} := \frac{1}{m} \sum_{t=1}^m \mathbf{x}_{s,t}$ 
21:   $s := s + 1$ 
22: Until  $|f(\mathbf{x}^s) - f(\mathbf{x}^{s-1})| < \epsilon$ 
23: Output:  $\mathbf{x}^s$ 

```

any $\tilde{n} \in [n]$, define

$$\bar{\mu}_{\tilde{n}} \triangleq \frac{1}{\tilde{n}} \sum_{j=1}^{\tilde{n}} \mu_{\bar{i}_j} \quad \text{and} \quad \bar{L}_{\tilde{n}} \triangleq \frac{1}{\tilde{n}} \sum_{j=n-\tilde{n}+1}^n L_{\bar{i}_j}. \quad (8)$$

Accordingly, define

$$\kappa_{\max} \triangleq \frac{L_{\max}}{\mu_{\min}} \quad \text{and} \quad \kappa_{\tilde{n}} \triangleq \frac{\bar{L}_{\tilde{n}}}{\bar{\mu}_{\tilde{n}}}. \quad (9)$$

In particular, define $\bar{\mu} \triangleq \bar{\mu}_n, \bar{L} \triangleq \bar{L}_n$ and $\kappa \triangleq \bar{L}/\bar{\mu}$. Denote the probability space on which the sequence of (random) iterates $\{\mathbf{x}_{s,t}\}_{s \geq 0, t \in [m]}$ in Algorithm 1 is defined as $(\Omega, \Sigma, \mathbb{P})$, where Σ is the Borel σ -algebra of Ω . We also define a filtration $\{\mathcal{F}_{s,t}\}_{s \geq 0, t \in [m-1]}$ such that $\mathcal{F}_{s,t}$ contains all the information up to the time (s, t) . Formally, $\mathcal{F}_{s,t} \triangleq \sigma(\{\tau_j\}_{j=0}^{s-1} \cup \{\mathcal{B}_{i,j}\}_{i \in [s-1], j \in [m-1]} \cup \{\mathcal{B}_{s,j}\}_{j=0}^{t-1} \cup \{\mathcal{T}_j\}_{j=0}^{\lfloor (sm+t)/L \rfloor})$, where $\sigma(\{x_j\}_{j=1}^n)$ denotes the σ -algebra generated by random variables $\{x_j\}_{j=1}^n$. Define $\mathcal{F}_s \triangleq \mathcal{F}_{s,0}$.

To introduce our coordinate transformation framework, we define some transforms of variables appearing in Algorithm 1. Specifically, for any $s, t, r \geq 0$, define $\tilde{\mathbf{x}}_{s,t,r} \triangleq \mathbf{H}_r^{-1/2} \mathbf{x}_{s,t}$,

$\tilde{\mathbf{x}}^{s,r} \triangleq \mathbf{H}_r^{-1/2} \mathbf{x}^s$, $\tilde{\mathbf{x}}_r^* \triangleq \mathbf{H}_r^{-1/2} \mathbf{x}^*$ and $\tilde{\mathbf{v}}_{s,t,r} \triangleq \mathbf{H}_r^{1/2} \mathbf{v}_{s,t}$.³ We also define transformed functions $\tilde{f}_{i,r} \triangleq f_i \circ \mathbf{H}_r^{1/2}$ and $\tilde{f}_r \triangleq \frac{1}{n} \sum_{i=1}^n \tilde{f}_{i,r}$, for any $i \in [n]$ and $r \geq 0$.

To state our convergence results, we define the notions of linear convergence and R-linear convergence.

Definition 1. A sequence $\{\mathbf{x}_n\}_{n \geq 0} \subseteq \mathbb{R}^d$ is said to converge to $\bar{\mathbf{x}} \in \mathbb{R}^d$ linearly (or more precisely, Q -linearly) with rate $\iota \in (0, 1)$ if

$$\limsup_{n \rightarrow \infty} \frac{\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|}{\|\mathbf{x}_n - \bar{\mathbf{x}}\|} \leq \iota. \quad (10)$$

We say $\mathbf{x}_n \rightarrow \bar{\mathbf{x}}$ R-linearly with rate $\iota' \in (0, 1)$ if there exists a nonnegative sequence $\{\varepsilon_n\}_{n \geq 0}$ such that $\|\mathbf{x}_n - \bar{\mathbf{x}}\| \leq \varepsilon_n$ for sufficiently large n and $\varepsilon_n \rightarrow 0$ linearly with rate ι' .

B. Preliminary Lemmas

From the definitions of transformed variables and functions in Section IV-A, we immediately have the following lemmas.

Lemma 1. For any $s, t, r \geq 0$ and $i \in [n]$, we have

$$\tilde{f}_{i,r}(\tilde{\mathbf{x}}_{s,t,r}) = f_i(\mathbf{x}_{s,t}), \quad (11)$$

$$\nabla \tilde{f}_{i,r}(\tilde{\mathbf{x}}_{s,t,r}) = \mathbf{H}_r^{1/2} \nabla f_i(\mathbf{x}_{s,t}), \quad (12)$$

$$\nabla^2 \tilde{f}_{i,r}(\tilde{\mathbf{x}}_{s,t,r}) = \mathbf{H}_r^{1/2} \nabla^2 f_i(\mathbf{x}_{s,t}) \mathbf{H}_r^{1/2}. \quad (13)$$

Lemma 2. If there exist $0 < \gamma' \leq \Gamma'$ such that $\gamma' \mathbf{I} \preceq \mathbf{H}_r \preceq \Gamma' \mathbf{I}$ for all $r \geq 0$, then for any $i \in [n]$ and $r \geq 0$, $f_{i,r}$ is twice differentiable, $(\mu_i \gamma')$ -strongly convex and $(L_i \Gamma')$ -smooth on \mathbb{R}^d . Consequently, f_r is twice differentiable, $(\gamma' \bar{\mu})$ -strongly convex and $(\Gamma' \bar{L})$ -smooth on \mathbb{R}^d .

Next we derive two other lemmas that will not only be used in the analysis later, but have the potential to be applied to more general problem settings. Specifically, Lemma 3 can be applied to any stochastic optimization algorithms based on SVRG and Lemma 4 can be applied to any finite-sum minimization algorithms based on L-BFGS methods (not necessarily stochastic in nature). The proofs of Lemmas 3 and 4 are deferred to Appendices A and B respectively.

Lemma 3 (Variance bound of $\mathbf{v}_{s,t}$). In Algorithm 1, we have $\mathbb{E}_{\mathcal{B}_{s,t}} [\mathbf{v}_{s,t} | \mathcal{F}_{s,t}] = \nabla f(\mathbf{x}_{s,t})$ and

$$\begin{aligned} & \mathbb{E}_{\mathcal{B}_{s,t}} \left[\|\mathbf{v}_{s,t} - \nabla f(\mathbf{x}_{s,t})\|^2 | \mathcal{F}_{s,t} \right] \\ & \leq \frac{4\bar{L}}{b} (f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*) + f(\mathbf{x}^s) - f(\mathbf{x}^*)). \end{aligned} \quad (14)$$

Remark 3. In previous works [16], [17], a uniform mini-batch sampling of $\mathcal{B}_{s,t}$ was employed, and different variance bounds of $\mathbf{v}_{s,t}$ were derived. In [16, Lemma 6], the bound was

$$4L_{\max} (f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*) + f(\mathbf{x}^s) - f(\mathbf{x}^*)). \quad (15)$$

In [17, Lemma 2], this bound was slightly improved to

$$\begin{aligned} & 4L_{\max} ((f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*)) \\ & + (1 - 1/\kappa_{\max}) (f(\mathbf{x}^s) - f(\mathbf{x}^*))). \end{aligned} \quad (16)$$

³As will be shown in Lemma 4, for any $r \geq 0$, $\mathbf{H}_r \succeq \gamma \mathbf{I}$ for some $\gamma > 0$. Therefore, $\mathbf{H}_r^{1/2}$ (and $\mathbf{H}_r^{-1/2}$) are well-defined.

However, both of these bounds fail to capture the dependence on the mini-batch size b . In contrast, in this work we consider a nonuniform mini-batch sampling (with replacement). Due to division by b and $\bar{L} \leq L_{\max}$ (indeed in many cases, $\bar{L} \ll L_{\max}$), our bound in (14) is superior to (15) and (16). As will be seen in Theorem 1, our better bound (14) leads to a faster (linear) convergence rate of Algorithm 1.

Lemma 4 (Uniform Spectral Bound of $\{\mathbf{H}_r\}_{r \geq 0}$). The spectra of $\{\mathbf{H}_r\}_{r \geq 0}$ are uniformly bounded, i.e., for each $r \geq 0$, $\gamma \mathbf{I} \preceq \mathbf{H}_r \preceq \Gamma \mathbf{I}$, where⁴

$$\gamma \triangleq \frac{1}{(M+1)\bar{L}_{b_H}} \quad \text{and} \quad \Gamma \triangleq \frac{\kappa_{b_H}^{M+1}}{\bar{\mu}_{b_H}(\kappa_{b_H} - 1)}. \quad (17)$$

Remark 4. In [13], [15], [16], the authors make use of a classical technique in [2] to derive a different uniform spectral bound of $\{\mathbf{H}_r\}_{r \geq 0}$. Their technique involves applying $\text{tr}(\cdot)$ and $\det(\cdot)$ recursively to the BFGS update rule

$$\mathbf{B}_r^{(k)} = \mathbf{B}_r^{(k-1)} - \frac{\mathbf{B}_r^{(k-1)} \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_r^{(k-1)}}{\mathbf{s}_k^T \mathbf{B}_r^{(k-1)} \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}, \quad (18)$$

where $\mathbf{B}_r^{(k)} \triangleq (\mathbf{H}_r^{(k)})^{-1}$ denotes the approximate Hessian matrix at step k in the reconstruction of $\mathbf{B}_r \triangleq (\mathbf{H}_r)^{-1}$. The lower and upper bounds derived by this technique are

$$\tilde{\gamma} = \frac{1}{(d+M)L_{\max}} \quad \text{and} \quad \tilde{\Gamma} = (d+M)^{d+M-1} \frac{\kappa_{\max}^{d+M-1}}{\mu_{\min}}$$

respectively. As will be seen in Proposition 1, the overall computational complexity of Algorithm 1 heavily depends on the estimated (uniform) condition number of $\{\mathbf{H}_r\}_{r \geq 0}$. Therefore, it is instructive to compute this quantity for both (γ, Γ) and $(\tilde{\gamma}, \tilde{\Gamma})$ as

$$\kappa_H \triangleq \frac{\Gamma}{\gamma} = (M+1) \frac{\kappa_{b_H}^{M+2}}{\kappa_{b_H} - 1} \approx (M+1) \kappa_{b_H}^{M+1}, \quad (19)$$

$$\tilde{\kappa}_H \triangleq \frac{\tilde{\Gamma}}{\tilde{\gamma}} = (M+d)^{M+d} \kappa_{\max}^{M+d}, \quad (20)$$

where the approximation in (19) follows from $\kappa_{b_H} \gg 1$ (see Footnote 4). By comparing (19) and (20), we notice our estimate for the condition number of $\{\mathbf{H}_r\}_{r \geq 0}$, namely κ_H , is smaller than those in [15] and [16], namely $\tilde{\kappa}_H$, in several aspects. First, κ_H does not grow (exponentially) with the data dimension d . Second, κ_H depends on κ_{b_H} , which is usually much smaller than κ_{\max} . Third, even if we set $d = 1$ in (20), the factor $M+1$ in (19) is much smaller than the factor $(M+1)^{M+1}$ in (20). As a result, our improved estimate of the condition number of $\{\mathbf{H}_r\}_{r \geq 0}$ will lead to a much better computational complexity estimate (see Proposition 1).

C. Main Results

Our main convergence results consist of Theorem 1 and Corollary 1, which provide linear convergence guarantees of $f(\mathbf{x}^s)$ to $f(\mathbf{x}^*)$ in expectation and almost surely, respectively.

⁴We assume $\kappa_{b_H} > 1$ for any $b_H \in [n]$ since we focus on the setting where f is ill-conditioned, i.e., $\kappa_{b_H} \geq \kappa \gg 1$. If $\kappa_{b_H} = 1$ for some $b_H \in [n]$, then $\Gamma = (\kappa_{b_H}^M + M)/\bar{\mu}_{b_H}$ and γ remains the same. The proof for this case can be straightforwardly adapted from that in Section B.

Theorem 1. In Algorithm 1, choose $\eta < \min\{b/12, 1\}/(\Gamma\bar{L})$ and m sufficiently large. With either option I or II, we have

$$\mathbb{E}[f(\mathbf{x}^s) - f(\mathbf{x}^*)] \leq \rho^s (f(\mathbf{x}^0) - f(\mathbf{x}^*)), \text{ where} \quad (21)$$

$$\rho = \frac{b}{\gamma\bar{\mu}m\eta(b - 4\eta\Gamma\bar{L})} + \frac{4\eta\Gamma\bar{L}}{b - 4\eta\Gamma\bar{L}} \left(1 + \frac{1}{m}\right) < 1. \quad (22)$$

Proof. Fix an outer iteration s and consider an inner iteration t . Define $r \triangleq \lfloor (sm + t)/L \rfloor$. For brevity, we omit the dependence of r on s and t . The iteration in line 10 of Algorithm 1 becomes

$$\tilde{\mathbf{x}}_{s,t+1,r} = \tilde{\mathbf{x}}_{s,t,r} - \eta \tilde{\mathbf{v}}_{s,t,r}. \quad (23)$$

Define $\tilde{\delta}_{s,t,r} \triangleq \tilde{\mathbf{v}}_{s,t,r} - \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r})$. From Lemmas 1 and 3,

$$\mathbb{E}_{\mathcal{B}_{s,t}}[\tilde{\mathbf{v}}_{s,t,r} | \mathcal{F}_{s,t}] = \mathbf{H}_r^{1/2} \nabla f(\mathbf{x}_{s,t}) = \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) \text{ and} \quad (24)$$

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_{s,t}}[\|\tilde{\delta}_{s,t,r}\|^2 | \mathcal{F}_{s,t}] &\leq \left\|\mathbf{H}^{1/2}\right\|^2 \frac{4\bar{L}}{b} (f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*) \\ &\quad + f(\mathbf{x}^s) - f(\mathbf{x}^*)) \\ &\leq \frac{4\Gamma\bar{L}}{b} \left(\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) + \tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*) \right), \end{aligned} \quad (25)$$

where $r' \triangleq \lfloor sm/L \rfloor$. Using (23), we can express the distance between $\tilde{\mathbf{x}}_{s,t+1,r}$ and $\tilde{\mathbf{x}}_r^*$ as

$$\begin{aligned} \|\tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^*\|^2 &= \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \\ &\quad + 2\eta \left(\frac{\eta}{2} \|\tilde{\mathbf{v}}_{s,t,r}\|^2 - \langle \tilde{\mathbf{v}}_{s,t,r}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle \right). \end{aligned} \quad (26)$$

We can show

$$\begin{aligned} \frac{\eta}{2} \|\tilde{\mathbf{v}}_{s,t,r}\|^2 - \langle \tilde{\mathbf{v}}_{s,t,r}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle &\leq - \left(\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \right) \\ &\quad - \left\langle \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^* \right\rangle - \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \end{aligned} \quad (27)$$

from steps (28) to (32) on the next page. In (30), we use the condition $\eta \leq 1/(\Gamma\bar{L})$. In (32), we use the $(\Gamma\bar{L})$ -smoothness of \tilde{f}_r and the $(\gamma\bar{\mu})$ -strong convexity of \tilde{f}_r in Lemma 2 respectively.

Now, substituting (27) into (26), we have

$$\begin{aligned} \|\tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^*\|^2 &\leq (1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \\ &\quad - 2\eta \left(\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \right) - 2\eta \left\langle \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^* \right\rangle \\ &= (1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 - 2\eta \left(\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \right) \\ &\quad + 2\eta^2 \|\tilde{\delta}_{s,t,r}\|^2 - 2\eta \left\langle \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \right\rangle. \end{aligned} \quad (33)$$

Taking expectation w.r.t. $\mathcal{B}_{s,t}$ and using (24) and (25), we have

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_{s,t}}[\|\tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^*\|^2 + 2\eta(\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*)) | \mathcal{F}_{s,t}] \\ \leq (1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 + \frac{8}{b} \Gamma\bar{L}\eta^2 (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \\ + \tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)). \end{aligned} \quad (34)$$

By bounding the factor $1 - \eta\gamma\bar{\mu}$ by 1, we can telescope (34) over $t = 0, \dots, m-1$ and obtain

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_{s,(m-1)}}[\|\tilde{\mathbf{x}}_{s,m,r} - \tilde{\mathbf{x}}_r^*\|^2 | \mathcal{F}_s] &+ 2m\eta \left(1 - \frac{4}{b} \Gamma\bar{L}\eta\right) \\ &\times \left\{ \frac{1}{m} \sum_{t=1}^m \mathbb{E}_{\mathcal{B}_{s,(t-1)}}[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) | \mathcal{F}_{s,t-1}] \right\} \end{aligned}$$

$$\leq \|\tilde{\mathbf{x}}^{s,r'} - \tilde{\mathbf{x}}_{r'}^*\|^2 + \frac{8}{b} \Gamma\bar{L}\eta^2 (1+m) \left(\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*) \right). \quad (35)$$

If we use option II to choose \mathbf{x}^{s+1} (in line 20), we have $\tilde{\mathbf{x}}^{s+1,r''} = 1/m \sum_{i=1}^m \tilde{\mathbf{x}}_{s,t,r''}$, where $r'' \triangleq \lfloor (s+1)m/L \rfloor$. Using (11) and Jensen's inequality, we have

$$\begin{aligned} \frac{1}{m} \sum_{t=1}^m \mathbb{E}_{\mathcal{B}_{s,(t-1)}}[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) | \mathcal{F}_{s,t-1}] \\ \geq \mathbb{E}_{\mathcal{B}_{s,(m-1)}}[\tilde{f}_{r''}(\tilde{\mathbf{x}}^{s+1,r''}) - \tilde{f}_{r''}(\tilde{\mathbf{x}}_{r''}^*) | \mathcal{F}_s]. \end{aligned} \quad (36)$$

Alternatively, if we use option I to determine \mathbf{x}^{s+1} (in line 19), we still have (36) (with inequality replaced by equality). If we further use Lemma T-1 to upper bound the term $\|\tilde{\mathbf{x}}^{s,r'} - \tilde{\mathbf{x}}_{r'}^*\|^2$ in (35), we have

$$\begin{aligned} 2m\eta \left(1 - \frac{4}{b} \Gamma\bar{L}\eta\right) \mathbb{E}_{\mathcal{B}_{s,(m-1)}}[\tilde{f}_{r''}(\tilde{\mathbf{x}}^{s+1,r''}) - \tilde{f}_{r''}(\tilde{\mathbf{x}}_{r''}^*) | \mathcal{F}_s] \\ \leq \left(\frac{8}{b} \Gamma\bar{L}\eta^2 (1+m) + \frac{2}{\gamma\bar{\mu}} \right) \left(\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*) \right). \end{aligned} \quad (37)$$

Using (11) again and rearranging, we have

$$\mathbb{E}[f(\mathbf{x}^{s+1}) - f(\mathbf{x}^*) | \mathcal{F}_s] \leq \rho(f(\mathbf{x}^s) - f(\mathbf{x}^*)). \quad (38)$$

We take expectation on both sides to complete the proof. \square

Remark 5. We compare our linear convergence rate ρ in (22) with those in [16] and [17]. Since the convergence rates in these two works are almost the same, we use the rate in [16] for comparison. The linear rate $\tilde{\rho}$ in [16] equals

$$\frac{1}{2\tilde{\gamma}\mu_{\min}m\eta(1 - \eta\tilde{\Gamma}L_{\max}\kappa_{\max}\tilde{\kappa}_H)} + \frac{\eta\tilde{\Gamma}L_{\max}\kappa_{\max}\tilde{\kappa}_H}{1 - \eta\tilde{\Gamma}L_{\max}\kappa_{\max}\tilde{\kappa}_H}.$$

For simplicity, if we let $b = 1$, $\mu_{\min} = \bar{\mu}$, $L_{\max} = \bar{L}$, $\tilde{\gamma} = \gamma$, $\tilde{\Gamma} = \Gamma$ and ignore other constant factors,⁵ we notice that there is an additional multiplicative factor $\kappa\kappa_H$ associated with $\eta\Gamma\bar{L}$ in $\tilde{\rho}$. As a result $\tilde{\rho} > \rho$. A more direct way to observe the detrimental effects of this additional $\kappa\kappa_H$ is to compare the computational complexities resulting from ρ and $\tilde{\rho}$. See Remark 9 for details. The reason that we manage to avoid this factor in our rate ρ is precisely because we adopt the *coordinate transformation framework* in our analysis (see proof above). Specifically, by absorbing the sequence of metric matrices $\{\mathbf{H}_r\}_{r \geq 0}$ into decision vectors and functions, we are able to proceed through bounding the (expected squared Euclidean) distance between $\tilde{\mathbf{x}}_{s,t+1,r}$ and $\tilde{\mathbf{x}}_r^*$, instead of directly bounding $f(\mathbf{x}_{s,t+1})$ via the smoothness property of f (cf. proof of Theorem 7 in [16]). Thus in our analysis, we avoid the additional appearance of \bar{L} and Γ (which leads to the additional factor $\kappa\kappa_H$).

Corollary 1. In Algorithm 1, $\{f(\mathbf{x}^s)\}_{s \geq 0}$ converges to $f(\mathbf{x}^*)$ R -linearly almost surely with rate ρ .

Proof. Our proof is inspired by [29, Corollary 2] and mainly leverages the Borel-Cantelli lemma [30]. For any $\epsilon' > 0$ and

⁵However, bear in mind that by doing all these substitutions, $\tilde{\rho}$ has already been much improved.

$$\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) + \frac{\eta}{2} \|\tilde{\mathbf{v}}_{s,t,r}\|^2 - \langle \tilde{\mathbf{v}}_{s,t,r}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle + \left\langle \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^* \right\rangle + \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \quad (28)$$

$$= \tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \left\langle \tilde{\mathbf{v}}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} + \frac{\eta}{2} \tilde{\mathbf{v}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \right\rangle + \left\langle \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^* \right\rangle + \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \quad (29)$$

$$\leq \tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \frac{\Gamma\bar{L}}{2} \eta^2 \|\tilde{\mathbf{v}}_{s,t,r}\|^2 - \left\langle \tilde{\mathbf{v}}_{s,t,r} - \tilde{\delta}_{s,t,r}, \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^* \right\rangle + \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \quad (30)$$

$$= \tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \left\langle \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}), \tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_{s,t,r} \right\rangle - \frac{\Gamma\bar{L}}{2} \|\eta \tilde{\mathbf{v}}_{s,t,r}\|^2 - \left\langle \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}), \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \right\rangle + \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \quad (31)$$

$$\leq \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) + \left\langle \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}), \tilde{\mathbf{x}}_r^* - \tilde{\mathbf{x}}_{s,t,r} \right\rangle + \frac{\gamma\bar{\mu}}{2} \|\tilde{\mathbf{x}}_r^* - \tilde{\mathbf{x}}_{s,t,r}\|^2 \leq \tilde{f}_r(\tilde{\mathbf{x}}_r^*). \quad (32)$$

$0 < \delta < 1 - \rho$, consider the sequence of events $\{\mathcal{E}_s\}_{s \in \mathbb{N}}$ such that

$$\mathcal{E}_s \triangleq \left\{ \omega \in \Omega : \frac{f(\mathbf{x}^s(\omega)) - f(\mathbf{x}^*)}{(\rho + 1/\sqrt{s})^s} > \epsilon' \right\}, \forall s \in \mathbb{N}. \quad (39)$$

Therefore,

$$\begin{aligned} \sum_{s=1}^{\infty} \mathbb{P}(\mathcal{E}_s) &\stackrel{(a)}{\leq} \sum_{s=1}^{\infty} \frac{\mathbb{E}[f(\mathbf{x}^s) - f(\mathbf{x}^*)]}{\epsilon'(\rho + 1/\sqrt{s})^s} \\ &\stackrel{(b)}{\leq} \frac{f(\mathbf{x}^0) - f(\mathbf{x}^*)}{\epsilon'} \sum_{s=1}^{\infty} \left(\frac{\rho}{\rho + 1/\sqrt{s}} \right)^s \\ &\stackrel{(c)}{\leq} \frac{f(\mathbf{x}^0) - f(\mathbf{x}^*)}{\epsilon'} \sum_{s=1}^{\infty} \exp\left(-\frac{\sqrt{s}}{\rho + 1/\sqrt{s}}\right) \\ &\stackrel{(d)}{\leq} \frac{f(\mathbf{x}^0) - f(\mathbf{x}^*)}{\epsilon'} \sum_{s=1}^{\infty} \exp\left(-\frac{\sqrt{s}}{\rho + 1}\right) < \infty, \end{aligned} \quad (40)$$

where in (a) we use Markov's inequality, in (b) we use (21) in Theorem 1, in (c) we use

$$(1+x)^s \leq e^{sx}, \forall s \in \mathbb{N} \cup \{0\}, \forall x > -1, \quad (41)$$

and in (d) we use the fact that $s \geq 1$. Thus by the Borel-Cantelli lemma, $\mathbb{P}(\limsup_{s \rightarrow \infty} \mathcal{E}_s) = 0$, or equivalently,

$$\mathbb{P}\left(\liminf_{s \rightarrow \infty} \mathcal{E}_s^c\right) = 1. \quad (42)$$

The definition of \mathcal{E}_s in (39) implies that

$$\liminf_{s \rightarrow \infty} \mathcal{E}_s^c = \left\{ \omega \in \Omega : \limsup_{s \rightarrow \infty} \frac{f(\mathbf{x}^s(\omega)) - f(\mathbf{x}^*)}{(\rho + 1/\sqrt{s})^s} \leq \epsilon' \right\}.$$

Since $\epsilon' > 0$ is arbitrary, we have

$$\mathbb{P}\left(\lim_{s \rightarrow \infty} \frac{f(\mathbf{x}^s) - f(\mathbf{x}^*)}{(\rho + 1/\sqrt{s})^s} = 0\right) = 1 \quad (43)$$

or equivalently, $f(\mathbf{x}^s) - f(\mathbf{x}^*) = o((\rho + 1/\sqrt{s})^s)$ almost surely, for any $s \geq 0$. For convenience, define a sequence $\{\varpi_s\}_{s \geq 0}$ such that $\varpi_s \triangleq (\rho + 1/\sqrt{s})^s$, for any $s \geq 0$. By applying (41) to ϖ_s , we have $\varpi_s = \rho^{s-\Theta(\sqrt{s})}$ (the implied constant in the $\Theta(\cdot)$ notation is positive). Since $\rho^{-\Theta(\sqrt{s+1})}/\rho^{-\Theta(\sqrt{s})} \rightarrow 1$ as $s \rightarrow \infty$, $\lim_{s \rightarrow \infty} \varpi_{s+1}/\varpi_s = \rho$ as desired. \square

Remark 6. Note that the analysis techniques in Corollary 1 can be applied to any stochastic algorithm with linear convergence in expectation, therefore are of independent interest. Although a similar result was proved in [29, Corollary 2], it is weaker than Corollary 1. Specifically, the almost sure linear conver-

gence rate therein is *strictly* worse than the corresponding linear rate in expectation. In contrast, by leveraging refined analysis techniques, we show that such a degradation can indeed be avoided.

Remark 7. By the $\bar{\mu}$ -strong convexity of f (see Assumption 2), we have $\|\mathbf{x} - \mathbf{x}^*\|^2 \leq (2/\bar{\mu})(f(\mathbf{x}) - f(\mathbf{x}^*))$, for any $\mathbf{x} \in \mathbb{R}^d$. Therefore, the linear convergence of $\{f(\mathbf{x}^s)\}_{s \geq 0}$ to $f(\mathbf{x}^*)$ in expectation (in Theorem 1) also implies the R-linear convergences of $\{\mathbf{x}^s\}_{s \geq 0}$ to \mathbf{x}^* in expectation. Similarly, we can also derive the almost sure R-linear convergence of $\{\mathbf{x}^s\}_{s \geq 0}$ to \mathbf{x}^* from Corollary 1.

V. COMPLEXITY ANALYSIS

In this section we provide a framework for analyzing the computational complexity of the stochastic L-BFGS algorithm in Algorithm 1. Our framework can be easily generalized to other stochastic second-order algorithms, e.g., SQN algorithm in [15]. To begin with, we make two additional assumptions.

Assumption 3. For any $\mathbf{x} \in \mathbb{R}^d$ and $i \in [n]$, the gradient $\nabla f_i(\mathbf{x})$ can be computed in $O(d)$ operations.⁶

Assumption 4. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $i \in [n]$, the Hessian-vector product $\nabla^2 f_i(\mathbf{x})\mathbf{y}$ can be computed in $O(d)$ operations.

Remark 8. These two assumptions are naturally satisfied for ERM problems (2) with Tikhonov regularization. For these problems, $R(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$ and

$$\nabla \ell(\mathbf{a}_i^T \mathbf{x}, b_i) = \ell'(\mathbf{a}_i^T \mathbf{x}, b_i) \mathbf{a}_i + \lambda \mathbf{x}, \quad (44)$$

$$\nabla^2 \ell(\mathbf{a}_i^T \mathbf{x}, b_i) \mathbf{y} = \ell''(\mathbf{a}_i^T \mathbf{x}, b_i) (\mathbf{a}_i^T \mathbf{y}) \mathbf{a}_i + \lambda \mathbf{y}, \quad (45)$$

where $\ell'(\cdot, \cdot)$ and $\ell''(\cdot, \cdot)$ are first and second derivatives of $\ell(\cdot, \cdot)$ w.r.t. the first argument. We easily see that the right-hand sides of both (44) and (45) can be computed in $O(d)$ operations.

From Algorithm 1, we observe that its total computational cost C can be split into three parts. The first part C_1 involves computing the variance-reduced gradient $\mathbf{v}_{s,t}$ in (7), the second part C_2 involves computing $\mathbf{H}_r \mathbf{v}_{s,t}$ (via two-loop recursion) in line 9, and the third part C_3 involves computing the correction pair $(\mathbf{s}_r, \mathbf{y}_r)$ in line 15.

⁶An operation refers to evaluation of an elementary function, such as addition, multiplication and logarithm.

Proposition 1. *Let Assumptions 1 to 4 hold. In Algorithm 1,*

$$C_1 = O((n + \kappa\kappa_H)d \log(1/\epsilon)), \quad (46)$$

$$C_2 = O(\kappa\kappa_H d \log(1/\epsilon)), \quad (47)$$

$$C_3 = O(d \log(1/\epsilon)). \quad (48)$$

Thus the total computational cost $C \triangleq \sum_{i=1}^3 C_i$ equals

$$O((n + \kappa\kappa_H)d \log(1/\epsilon)). \quad (49)$$

Proof. We leverage techniques that have become standard in the SVRG literature (e.g., [27]). In (22), if we choose $\eta = \theta b/(\Gamma\bar{L})$ for some $0 < \theta < 1/12$, $m = \theta'\kappa\kappa_H/b$ for some large enough positive constant θ' , and use $1 + 1/m \leq 2$, then

$$\rho = \frac{1}{\theta'\theta(1-4\theta)} + \frac{8\theta}{1-4\theta} < 1. \quad (50)$$

As a result, the required number of outer iterations to achieve ϵ -suboptimality is $O(\log(1/\epsilon))$. Thus (46) follows from Assumption 3 and that $2mb$ gradients (of component functions) are computed in each inner iteration. If we further choose $M = \Theta(b)$, then (47) follows from the fact that two-loop recursion can be done in $O(Md)$ time [25, Chapter 7]. Lastly, if we choose $b_H = \Theta(\Upsilon)$, then we obtain (48) using Assumption 4. \square

Remark 9. Following a similar argument, we can deduce the total complexity estimate \tilde{C} based on the linear rate $\tilde{\rho}$ (see Remark 5) derived in [16] as

$$\tilde{C} = O((n + b(\kappa_{\max}\tilde{\kappa}_H)^2)d \log(1/\epsilon)). \quad (51)$$

Compared with \tilde{C} , we observe that our complexity estimate C in (49) is much better, in several aspects. First, the dependence of C on the condition number $\kappa\kappa_H$ is linear, rather than quadratic. The quadratic dependence of $\kappa_{\max}\tilde{\kappa}_H$ in \tilde{C} is precisely caused by the additional $\kappa_{\max}\tilde{\kappa}_H$ in $\tilde{\rho}$ (see Remark 5). Second, C is independent of the mini-batch size b . The appearance of b in \tilde{C} is a result of the loose bound on variance of $\mathbf{v}_{s,t}$ (cf. (15) and (16)). Third, the condition number $\kappa\kappa_H$ in C is much more smaller than $\kappa_{\max}\tilde{\kappa}_H$ in \tilde{C} for ill-conditioned problems. This is a result of the non-uniform sampling of $\mathcal{B}_{s,t}$ and our improved bound on the spectra of $\{\mathbf{H}_r\}_{r \geq 0}$.

Remark 10. As our coordinate transformation framework unifies the design and analysis of stochastic first- and second-order algorithms, we believe that momentum-based acceleration techniques for stochastic first-order methods [31], [32] can be applied to Algorithm 1 as well. (Details are left to future work.) In this case, the dependence on $\kappa\kappa_H$ in C may be further improved to $\sqrt{\kappa\kappa_H}$ [32].

VI. ACCELERATION STRATEGIES

In this section, we propose three practical acceleration strategies. We follow the notations in Section III and Algorithm 1. As will be shown in Section VII-B, all of these strategies result in faster convergence in practice. For the first and second strategies, we also provide their theoretical analyses in Propositions 2 and 3, respectively. See Appendices C and D for the proofs of these two propositions.

A. Geometric Sampling/Averaging Scheme

Instead of choosing \mathbf{x}^{s+1} according to option I or II in Algorithm 1, inspired by [33], we can introduce a “forgetting” effect by considering two other schemes:

option III: Sample τ_s randomly from $[m]$ from the distribution $Q \triangleq (\beta^{m-1}/c, \beta^{m-2}/c, \dots, 1/c)$ and set $\mathbf{x}^{s+1} := \mathbf{x}_{s,\tau_s}$,

option IV: $\mathbf{x}^{s+1} := \frac{1}{c} \sum_{t=1}^m \beta^{m-t} \mathbf{x}_{s,t}$,

where $0 < \beta \leq 1 - \eta\gamma\bar{\mu} < 1$ and the normalization constant $c \triangleq \sum_{t=1}^m \beta^{m-t}$. Since $\beta \in (0, 1)$, we observe that in both options III and IV, more recent iterates (i.e., iterates $\mathbf{x}_{s,t}$ with larger indices t) will have larger contributions to \mathbf{x}^{s+1} . Theoretically, these two schemes can be analyzed in a unified manner, as shown in the following proposition.

Proposition 2. *In Algorithm 1, choose $\eta < \min\{b/12, 1\}/(\Gamma\bar{L})$ and m sufficiently large. With either option III or IV, we have*

$$\mathbb{E}[f(\mathbf{x}^s) - f(\mathbf{x}^*)] \leq \bar{\rho}^s (f(\mathbf{x}^0) - f(\mathbf{x}^*)), \text{ where} \quad (52)$$

$$\bar{\rho} \triangleq \frac{b}{\gamma\bar{\mu}c'\eta(b - 4\eta\Gamma\bar{L}/(1 - \eta\gamma\bar{\mu}))} + \frac{4\eta\Gamma\bar{L}}{b - 4\eta\Gamma\bar{L}/(1 - \eta\gamma\bar{\mu})} \left(1 + \frac{1}{c'}\right) < 1 \quad (53)$$

and $c' \triangleq c/(1 - \eta\gamma\bar{\mu})^m$.

Remark 11. In the literature [16], [17], usually option I or II (in Algorithm 1) is analyzed to prove that the stochastic L-BFGS algorithms therein converge linearly. However, for faster convergence in practice, \mathbf{x}^{s+1} is chosen to be the last inner iterate $\mathbf{x}_{s,m}$. However, the latter strategy is not amenable to linear convergence analysis. This *gap between theory and practice* is filled in by our geometric sampling or averaging scheme, i.e., option III or IV. Specifically, as shown in Figure 2, our scheme not only yields linear convergence in theory, but also performs as well as the “last inner iterate” scheme in practice.

B. Subsampled Gradient Stabilization

In Algorithm 1, at the beginning of each outer iteration indexed by s , we compute a full gradient \mathbf{g}_s to stabilize the subsequent (inner) iterations. Inspired by [7], we propose a strategy that only computes a subsampled gradient $\tilde{\mathbf{g}}_s$ at the start of each outer iteration s . Specifically, we uniformly sample a subset $\tilde{\mathcal{B}}_s$ of $[n]$ with size \tilde{b}_s *without replacement* and then form $\tilde{\mathbf{g}}_s \triangleq (1/\tilde{b}_s) \sum_{i \in \tilde{\mathcal{B}}_s} \nabla f_i(\mathbf{x}^s)$. The size of $\tilde{\mathcal{B}}_s$, namely \tilde{b}_s , increases with the index s until it reaches n . By judiciously choosing \tilde{b}_s , we can show that the resulting algorithm still enjoys linear convergence with rate $\bar{\rho}$, when integrated with the geometric sampling/averaging scheme in Section VI-A. Before we formally state this result in Proposition 3, we first make an assumption in addition to Assumptions 1 and 2.

Assumption 5. *The inner iterates $\{\mathbf{x}_{s,t}\}_{s \geq 0, t \in [m]}$ generated by the modified algorithm in Section VI-B are bounded almost surely, i.e., there exists $B < \infty$ such that for any $s \geq 0$ and $t \in [m]$, $\|\mathbf{x}_{s,t} - \mathbf{x}^*\| \leq B$.*

Proposition 3. Let Assumptions 1, 2 and 5 hold. For any $\xi > 0$ and $S \in \mathbb{N}$, and for any $s \in (S]$, if we choose $\tilde{b}_s \geq \underline{b}_s \triangleq nS^2\alpha_s/(S^2\alpha_s + (n-1)\xi^2\bar{\rho}^{2s})$, where $\alpha_s \triangleq 1/n \sum_{i=1}^n \|\nabla f_i(\mathbf{x}^s)\|^2$, we have

$$\mathbb{E}[f(\mathbf{x}^s) - f(\mathbf{x}^*)] \leq \bar{\rho}^s \left\{ (f(\mathbf{x}^0) - f(\mathbf{x}^*)) + \left(1 + \frac{1}{c'}\right) \frac{\xi b}{b - 4\Gamma L\eta/(1 - \eta\Gamma\bar{\mu})} (\kappa_H^{1/2}B + \eta\Gamma\xi) \right\}. \quad (54)$$

Remark 12. Several remarks are in order. First, we remark that assumptions involving almost sure boundedness of iterates (e.g., Assumption 5) are commonplace in the stochastic optimization literature [7], [34], [35] and are always observed to hold in experiments. Second, under this assumption, we can show that $\{\alpha_s\}_{s \geq 0}$ are bounded almost surely using the Lipschitz continuity of ∇f_i , for any $i \in [n]$ in Assumption 2. Consequently, there exists $B' < \infty$ such that $\alpha_s \leq B'$ for any $s \geq 0$ and hence

$$\tilde{b}_s \leq \frac{nS^2B'}{S^2B' + (n-1)\xi^2\bar{\rho}^{2s}} \quad (55)$$

$$\leq \frac{nS^2B'}{(n-1)\xi^2} (\bar{\rho}^{-2})^s. \quad (56)$$

As a sanity check, we observe that (55) increases to n as $s \rightarrow \infty$. By further upper bounding (55) by (56), we obtain a practical rule to select \tilde{b}_s . Namely, it suffices to choose $\tilde{b}_s = \min\{\zeta v^s, n\}$, for some constants $\zeta > 0$ and $v > 1$. As shown in Section VII, this rule works well in practice. Third, for any $\epsilon > 0$, we can choose $S \in \mathbb{N}$ such that our algorithm achieves ϵ -suboptimality, i.e., $\mathbb{E}[f(\mathbf{x}^S) - f(\mathbf{x}^*)] < \epsilon$.

C. Low-dimensional Approximate Hessians

In addition to high dimensionality and large size, *sparsity* is also a typical attribute for modern data, i.e., many feature vectors only have a few nonzero entries. For ERM problems (2) (with Tikhonov regularization), this implies that the Hessian of f in (1),

$$\nabla^2 f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \ell''(\mathbf{a}_i^T \mathbf{x}, b_i)(\mathbf{a}_i \mathbf{a}_i^T) + \lambda \mathbf{I}, \quad (57)$$

tends to be sparse. Based on this observation, we propose a strategy that aims to approximate $\nabla^2 f(\mathbf{x})$ by several smaller Hessian matrices and update them efficiently. For sparse data, collecting curvature information via smaller dense Hessians can be more effective than directly manipulating the high-dimensional sparse Hessian [25]. As a result, the algorithm converges faster in practice (see Figure 4).

Before describing our strategy, we first introduce some notations. We partition $[n]$ into K groups, and denote the set of partitions as $\mathcal{P} \triangleq \{\mathcal{P}_1, \dots, \mathcal{P}_K\}$. For any $i \in [K]$, we define $\mathcal{S}_i \triangleq \cup_{j \in \mathcal{P}_i} \text{supp}(\mathbf{a}_j)$, where $\text{supp}(\mathbf{a}_j)$ denotes the support of the vector \mathbf{a}_j . We define $d_i \triangleq |\mathcal{S}_i|$ and denote the elements in \mathcal{S}_i as $\{s_{i,1}, \dots, s_{i,d_i}\}$. We also define $F_i = \sum_{j \in \mathcal{P}_i} f_j$ so that $f = \frac{1}{n} \sum_{i=1}^K F_i$. We define a *projection matrix* $\mathbf{U}_i \in \mathbb{R}^{d_i \times d}$ such that for any $p \in [d_i]$ and $q \in [d]$, $u_{pq} = 1$ if $q = s_{i,p}$ and 0 otherwise. Accordingly, for any $l \in \mathcal{P}_i$, define a

function $\phi_{i,l} : \mathbb{R}^{d_i} \rightarrow \mathbb{R}$ such that $f_l \triangleq \phi_{i,l} \circ \mathbf{U}_i$. Note that $\phi_{i,l}$ is uniquely defined by the definition \mathbf{U}_i . Also define $\phi_i \triangleq \sum_{l \in \mathcal{P}_i} \phi_{i,l}$. Therefore,

$$\nabla^2 f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^K \mathbf{U}_i^T \nabla^2 \phi_i(\mathbf{U}_i \mathbf{x}) \mathbf{U}_i, \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad (58)$$

We now describe our strategy. In Algorithm 1, for any $i \in [K]$ and any $j \in \{r - M' + 1, \dots, r\}$, define correction pairs $\mathbf{s}_{j,i} \triangleq \mathbf{U}_i(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_{j-1})$ and $\mathbf{y}_{j,i} \triangleq \sum_{l \in \mathcal{T}_{r,i}} \nabla^2 \phi_{i,l}(\mathbf{U}_i \bar{\mathbf{x}}_j) \mathbf{s}_{j,i}$, where $\mathcal{T}_{r,i}$ with size b_H/K is uniformly sampled from \mathcal{P}_i . Accordingly, define $\mathbf{S}_{r,i} \triangleq [\mathbf{s}_{r-M'+1,i}, \dots, \mathbf{s}_{r,i}]$, $\mathbf{Y}_{r,i} \triangleq [\mathbf{y}_{r-M'+1,i}, \dots, \mathbf{y}_{r,i}]$. Instead of storing \mathcal{H}_r , we only store matrices $\{(\mathbf{S}_{r,i}, \mathbf{Y}_{r,i})\}_{i=1}^K$. To reconstruct approximation $\mathbf{B}_{r,i}$ for each $\nabla^2 \phi_i$ at $\mathbf{U}_i \bar{\mathbf{x}}_r$, as usual, we apply M' BFGS updates (18) (using the correction pairs stored in $\mathbf{S}_{r,i}$ and $\mathbf{Y}_{r,i}$) to $\mathbf{B}_{r,i}^{(0)} \triangleq \delta_{r,i} \mathbf{I}$, where $\delta_{r,i} \triangleq \|\mathbf{y}_{r,i}\|^2 / \mathbf{s}_{r,i}^T \mathbf{y}_{r,i}$. This procedure can be implemented *efficiently* via the method of *compact representation* [25], i.e.,

$$\mathbf{B}_{r,i} = \delta_{r,i} \mathbf{I} - \mathbf{W}_{r,i} \mathbf{M}_{r,i}^{-1} \mathbf{W}_{r,i}^T, \quad (59)$$

$$\mathbf{M}_{r,i} \triangleq \begin{bmatrix} \delta_{r,i} \mathbf{S}_{r,i}^T \mathbf{S}_{r,i} & \mathbf{L}_{r,i} \\ \mathbf{L}_{r,i}^T & -\mathbf{D}_{r,i} \end{bmatrix}, \quad (60)$$

where $\mathbf{W}_{r,i} \triangleq [\delta_{r,i}^{(i)} \mathbf{S}_{r,i}^{(i)}, \mathbf{Y}_{r,i}^{(i)}]$ and $\mathbf{L}_{r,i}$ and $\mathbf{D}_{r,i}$ are the lower triangular matrix (excluding diagonal) and diagonal matrix of $\mathbf{S}_{r,i}^T \mathbf{Y}_{r,i}$ respectively. Analogous to (58), we define the approximation of $\nabla^2 f$ at $\mathbf{U}_i \bar{\mathbf{x}}_r$, denoted as \mathbf{B}_r , as

$$\mathbf{B}_r \triangleq \frac{1}{n} \sum_{i=1}^K \mathbf{U}_i^T \mathbf{B}_{r,i} \mathbf{U}_i. \quad (61)$$

We remark that the strong convexity of each function f_i (see Assumption 2), together with the full-row-rank property of \mathbf{U}_i , ensures $\nabla^2 \phi_i \succ 0$ on \mathbb{R}^{d_i} . This implies *positive curvature* $\mathbf{S}_{r,i}^T \mathbf{Y}_{r,i} > 0$ and hence the positive definiteness of $\mathbf{B}_{r,i}$, for any $r \geq 0$ and $i \in [K]$. As a result, $\mathbf{B}_r \succ 0$ on \mathbb{R}^d . This suggests the usage of the *conjugate gradient* (CG) method to compute the search direction at time (s, t) , namely $\mathbf{p}_{s,t} \triangleq -\mathbf{B}_r^{-1} \mathbf{v}_{s,t}$, via solving the positive definite system $\mathbf{B}_r \mathbf{p}_{s,t} = -\mathbf{v}_{s,t}$. In particular, for any $\mathbf{z} \in \mathbb{R}^d$, $\mathbf{B}_r \mathbf{z}$ and $\mathbf{z}^T \mathbf{B}_r \mathbf{z}$ in CG can be computed very efficiently using (59) and (61). For example,

$$\mathbf{z}^T \mathbf{B}_r \mathbf{z} = \frac{1}{n} \sum_{i=1}^K \delta_{r,i} \|\mathbf{z}_i\|^2 - (\mathbf{W}_{r,i}^T \mathbf{z}_i)^T \mathbf{M}_{r,i}^{-1} (\mathbf{W}_{r,i}^T \mathbf{z}_i), \quad (62)$$

where $\mathbf{z}_i \triangleq \mathbf{U}_i \mathbf{z}$. We observe that the total computational cost in (62) is $O(M'(M'^2 + d'))$, where $d' \triangleq \sum_{i=1}^K d_i$. For sparse data, usually $d' = O(d)$, so this cost is still linear in d . In addition, we can compute (62) *in parallel* across $i \in [K]$. (Intuitively, this amounts to *collecting curvature* from each function ϕ_i in parallel.) In this case, the computational time will be greatly reduced to $O(M'(M'^2 + \max_i d_i))$. Since typically $\max_i d_i \ll d$, the computational savings from *parallel curvature collection* can be significant.⁷

⁷The memory parameter M (note that $M' \leq M$) is usually set to a small constant, e.g., 5 or 10. Thus it has less effect on the computational complexity compared to d or $\max_i d_i$.

Remark 13. Note that if we interpret the matrices $\{\mathbf{U}_i^T\}_{i \in [K]}$ as *sketching matrices* [36], then the acceleration technique in Section VI-C can be regarded as a way of performing (approximate) *Hessian sketching*. However, most existing methods in the literature [17], [37]–[39] either use random sketching matrices or the (deterministic) frequent directions approach [40], which is based on the singular value decomposition. A certain amount of information contained in the Hessian is lost or modified in these sketching processes. In contrast, by using the sparse binary matrices $\{\mathbf{U}_i^T\}_{i \in [K]}$, our approach merely (deterministically) compresses the large sparse Hessian matrix into K small dense Hessians, without changing any information contained therein.

Remark 14. In [17], the authors proposed another acceleration strategy called the *block BFGS update* [41], [42]. We remark that this strategy can be straightforwardly combined with all the other acceleration strategies proposed above, and may result in further acceleration of the convergence of Algorithm 1.

VII. NUMERICAL EXPERIMENTS

A. Experimental Setup

We consider two ERM problems, including logistic regression (with Tikhonov regularization) and ridge regression. For logistic regression, $b_i \in \{-1, 1\}$ and

$$f_i^{\log}(\mathbf{x}) \triangleq \log \left(1 + e^{-b_i(\mathbf{a}_i^T \mathbf{x})} \right) + \frac{\lambda}{2} \|\mathbf{x}\|^2, \quad \forall i \in [n]. \quad (63)$$

For ridge regression, $b_i \in \mathbb{R}$ and

$$f_i^{\text{rid}}(\mathbf{x}) \triangleq (\mathbf{a}_i^T \mathbf{x} - b_i)^2 + \frac{\lambda}{2} \|\mathbf{x}\|^2, \quad \forall i \in [n]. \quad (64)$$

Accordingly, define $f^{\log} \triangleq \frac{1}{n} \sum_{i=1}^n f_i^{\log}$ and $f^{\text{rid}} \triangleq \frac{1}{n} \sum_{i=1}^n f_i^{\text{rid}}$. Simple calculations reveal that the smoothness parameters L_i of f_i^{\log} and f_i^{rid} are given by $\|\mathbf{a}_i\|^2/4 + \lambda$ and $2\|\mathbf{a}_i\|^2 + \lambda$, respectively. Define the data matrix $\mathbf{A} \triangleq [\mathbf{a}_1, \dots, \mathbf{a}_n]$. The condition numbers κ of f^{\log} and f^{rid} are given by $(\frac{1}{4n}\sigma_{\max}^2(\mathbf{A}) + \lambda)/\lambda$ and $(2\sigma_{\max}^2(\mathbf{A}) + n\lambda)/(2\sigma_{\min}^2(\mathbf{A}) + n\lambda)$ respectively, where $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ denote the largest and smallest singular values of \mathbf{A} respectively. In both (63) and (64), we choose $\lambda = 1/n$, following the convention in the literature (e.g., [17]).

We tested logistic and ridge regression problems on `rcv1.binary` and `E2006-tfidf` datasets respectively [43]. (In the sequel we abbreviate them as `rcv1` and `E2006-tf`.) From (57), we defined a sparsity estimate of $\nabla^2 f$ at any $\mathbf{x} \in \mathbb{R}^d$ as $\varrho \triangleq |\text{supp}(\mathbf{A}\mathbf{A}^T + \mathbf{I})|/d^2$. The statistics of both datasets, including the (ambient) data dimension d , number of data samples n , sparsity parameter ϱ and condition number κ (of f^{\log} or f^{rid} defined by the datasets), are summarized in Table I.⁸ From it, we observe that both datasets are large-scale and sparse, but with different d -to- n ratios and condition numbers. Through these differences, we are able to infer the reasons for the different performances of some acceleration strategies on different ERM problems (shown in Section VII-B).

⁸The data dimension d for the original `E2006-tf` dataset is 150360. Due to memory issues, we randomly subsampled its features so that $d = 15036$.

TABLE I
STATISTICS OF `rcv1` AND `E2006-tf` DATASETS.

Datasets	d	n	d/n	ϱ	κ
<code>rcv1</code>	47236	20242	2.33	0.0154	113.17
<code>E2006-tf</code>	15036	16087	0.93	0.0404	1.70

For both datasets, the norms of all feature vectors $\{\mathbf{a}_i\}_{i=1}^n$ have been normalized to unit. Since the smoothness parameters L_i for both ERM problems are only dependent on $\|\mathbf{a}_i\|$ and λ , we have $L_i = L_j$ for any $i, j \in [n]$. Therefore the nonuniform distribution P in Section III becomes uniform, and the merit of nonuniform sampling of $\mathcal{B}_{s,t}$ cannot be observed.

To estimate the global optimum \mathbf{x}^* as ground truth, we used batch L-BFGS-B algorithm [44]. We randomly initialized \mathbf{x}^0 according to a scaled standard normal distribution. (The performance of our algorithms were observed to be insensitive to the initialization of \mathbf{x}^0 .) We used the number of data passes (i.e., number of data points accessed divided by n), rather than running time, to measure the convergence rates of all the algorithms under comparison. This has been a well-established convention in the literature on both stochastic first-order [5]–[7] and second-order methods [15]–[17] to make convergence results agnostic to the actual implementation of the algorithms, e.g., programming languages.⁹

Finally we describe the parameter setting. We set the mini-batch size $b = \sqrt{n}$, Hessian update period $\Upsilon = 10$ and the memory parameter $M = 10$. We set $b_H = b\Upsilon$ so that the computation for \mathbf{y}_r can be amortized to each inner iteration. We set the number of inner iterations $m = n/b$, so that each outer iteration will access $2n$ data points. Lastly, we set $\eta = 1 \times 10^{-2}$. From Figure 1, we observe that when η is too large, e.g., $\eta = 0.1$, Algorithm 1 only converges sublinearly; whereas when η is too small, e.g., $\eta = 1 \times 10^{-3}$, Algorithm 1 converges linearly but slowly. This corresponds well to our theoretical analysis in Theorem 1, which indicates that when η falls below a threshold, ρ increases as η decreases. For both ERM problems, we see that $\eta = 1 \times 10^{-2}$ yields fast linear convergence.

B. Performance of Acceleration Strategies

We first examine the performance of Algorithm 1 with different schemes of choosing \mathbf{x}^{s+1} . We consider five schemes in total, including (a) uniform sampling (option I), (b) uniform averaging (option II), (c) geometric sampling (option III), (d) geometric averaging (option IV) and (e) last inner iterate (in Remark 11). For options III and IV, we set $\beta = 1/2$. From Figure 2, we observe that options III and IV perform as well as the “last inner iterate” scheme, on both ERM problems, and outperform the schemes based on uniform sampling/averaging significantly. For all the subsequent experiments, we use option IV to select \mathbf{x}^{s+1} .

⁹Specifically, our algorithm (Algorithm 1) was implemented in Matlab®. However, some benchmarking algorithms (see Section VII-C) were implemented in other languages, e.g., the SVRG algorithm [5] and the stochastic L-BFGS algorithm [16] were implemented in C++ and Julia respectively. The differences in programming language typically have a significant impact on the actual running time of algorithms.

We next compare the performance of Algorithm 1 with and without using the subsampled gradient stabilization strategy in Section VI-B. As suggested by Remark 12, we chose $\tilde{b}_s = \min\{\zeta v^s, n\}$, where $\zeta = n/v^q$, $v = 3$ and $q = 8$. That is, the number of component gradients in $\tilde{\mathbf{g}}_s$ exponentially increases in the first $p = 8$ outer iterations and then remains at n . From Figure 3, we observe that this simple parameter selection method works well on both ERM problems, especially in the initial phase (when s is small). In addition, we also observe when s is large, both algorithms have almost the same (linear) convergence rates. This corroborates our analysis in Proposition 3.

We finally compare the performance of Algorithm 1 with and without using the low-dimensional approximate Hessian strategy in Section VI-C. We set the number of partitions $K = 5$ and partition $[n]$ evenly and randomly. From Figure 4, we observe that our strategy leads to improvements of convergence on both logistic and ridge regression problems, and the improvement on the latter is *very significant*. It could be possible that the smaller condition number of the `E2006-tf` dataset enables more effective curvature collection by low-dimensional Hessians. Nevertheless, for the `rcv1` dataset, which has a large condition number, our strategy is still efficacious. Additionally, we observe that our strategy *preserves the linear convergence* of Algorithm 1 on both problems. (A theoretical analysis of this linear convergence is left to future work.) Figure 4 shows the performance of our strategy in a single-threaded mode; as discussed in Section VI-C, our strategy can be much more efficient under scenarios where parallel computational resources are available.

C. Comparison to Other Algorithms

We combined all of our acceleration strategies in Section VI and compared the resulting algorithm with three benchmarking algorithms, including SVRG in [5] (with mini-batch sampling of $\mathcal{B}_{s,t}$) and two state-of-the-art stochastic L-BFGS algorithms in [16] and [17]. For the algorithm in [17], we focused on its variant (b), since it consistently outperformed other variants in experiments. We tested all the three benchmarking algorithms on both ERM problems with different step sizes $\eta \in \{10^{-4}, 10^{-3}, \dots, 1\}$ and selected the best η for each algorithm. The outer iterate \mathbf{x}^{s+1} in all these algorithms were selected via “the last inner iterate” scheme. From Figure 5, we observe that on both ERM problems, our algorithm yields faster convergence compared to all the benchmarking algorithms. This is due to the incorporation of the acceleration strategies in Sections VI-B and VI-C. The improvement of convergence is particularly significant on the ridge regression problem. This observation is consistent with our observations in Section VII-B. In addition, we indeed observe that with the aid of curvature information, all the stochastic L-BFGS methods outperform the stochastic first-order method SVRG.

VIII. FUTURE WORK AND AN OPEN PROBLEM

We propose to pursue future work in the following two directions. First, we aim to develop and analyze *proximal* and *momentum-based accelerated* stochastic L-BFGS algorithms,

based on our coordinate transformation framework. The proximal variant enables our algorithm to be applied to composite nonsmooth (convex) objective function. The accelerated variant can potentially improve the linear convergence rate of our algorithm, and thus reduce the total computational complexity. Second, we aim to analyze the convergence of the strategy in Section VI-C. In particular, Figure 4 suggests that Algorithm 1 may still converge linearly under this strategy.

Besides future work, there is also an open problem we hope to resolve. Although we have improved the linear convergence rate and computational complexity of Algorithm 1 as compared to those in [16] and [17], it seems our improved complexity in (49) is still inferior to that of SVRG. In SVRG, the complexity is $O((n + \kappa)d \log(1/\epsilon))$ [27], so our complexity (49) has an additional multiplicative factor κ_H . This contradicts the experimental results in [16], [17] and Section VII-C, where stochastic L-BFGS-type algorithms have been repeatedly shown to outperform their first-order counterparts. *Therefore, an interesting problem consists in obtaining a (computational) complexity bound of the stochastic L-BFGS algorithm that is better (or at least as good as) that of SVRG.* Indeed, a careful analysis reveals that the additional κ_H arises from the uniform spectral bound of the metric matrices $\{\mathbf{H}_r\}_{r \geq 0}$. This uniform bound is effectively a worst-case bound, and does not reflect the *local curvature information* contained in recent iterates at any time (s, t) . Since the judicious use of curvature information serves as a very important reason for the fast convergence of the stochastic quasi-Newton algorithms, such information should also be reflected in theoretical analysis as well (possibly in an adaptive spectral bound for $\{\mathbf{H}_r\}_{r \geq 0}$). We believe an effective adaptive bound is critical for improving the complexity result in (49).

Interestingly, an *incremental* quasi-Newton (IQN) method was proposed by Mokhtari *et al.* [14] recently. The proposed algorithm makes use of the aggregated optimization variables, as well as the aggregated gradients and approximate Hessians of all the component functions to reduce the noise of gradient and Hessian approximations. As a result, it achieves the local superlinear convergence rate, but requires $\Theta(nd^2)$ storage space. (Note that most of the stochastic L-BFGS methods, such as [15], [16] and Algorithm 1, only require $\Theta(d)$ memory.) The key idea in [14] is to show the descent direction in the IQN algorithm asymptotically converges to that of the Newton’s method. If this condition holds, then the additional factor κ_H in the complexity estimate (49) may be removed. Therefore, how to design a stochastic quasi-Newton algorithm that satisfies this condition in the *memory-limited* setting would also be an interesting problem to pursue in the future.

APPENDIX A PROOF OF LEMMA 3

The proof of Lemma 3 is shown in (65) to (71) on page 13. In (67), we use the independence of i_j and $i_{j'}$ for $j \neq j'$ and (95) in Lemma T-2. In (68), we use (95) and the fact that $\mathbb{E}[\|\mathbf{a} - \mathbb{E}[\mathbf{a}]\|^2 | \mathcal{G}] \leq \mathbb{E}[\|\mathbf{a}\|^2 | \mathcal{G}]$ almost surely, for any σ -algebra \mathcal{G} . In addition, the inequalities in (70) and (71) follow from $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$ and (96) respectively.

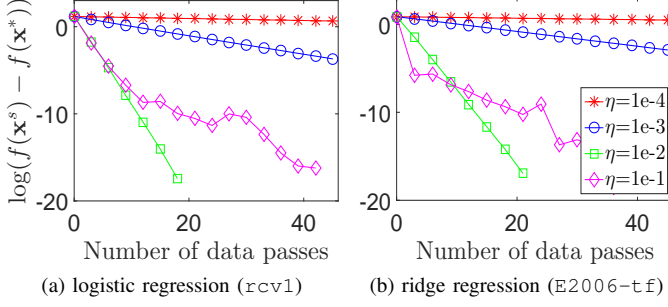


Fig. 1. Log suboptimality versus number of passes through data of Algorithm 1 with different step sizes η .

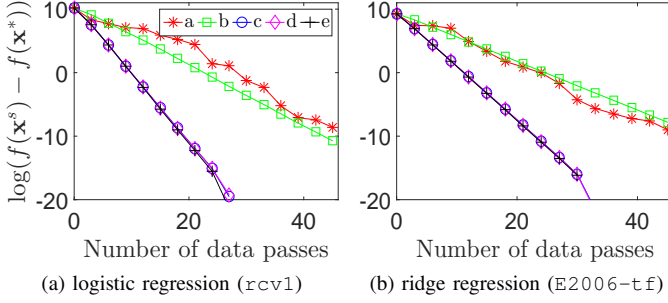


Fig. 2. Comparison of Algorithm 1 with different selection schemes for \mathbf{x}^{s+1} .

APPENDIX B PROOF OF LEMMA 4

Our proof is inspired by [17]. For any $\mathbf{x} \in \mathbb{R}^d$ and $\mathcal{T} \subseteq [n]$ with cardinality b_H , we have

$$\bar{\mu}_{b_H} \mathbf{I} \preceq \nabla^2 f_{\mathcal{T}}(\mathbf{x}) \preceq \bar{L}_{b_H} \mathbf{I}. \quad (72)$$

Define $\mathbf{V}_k \triangleq \mathbf{I} - \mathbf{y}_k \mathbf{s}_k^T / (\mathbf{y}_k^T \mathbf{s}_k)$ and $\mathbf{Q}_k \triangleq \mathbf{s}_k \mathbf{s}_k^T / (\mathbf{y}_k^T \mathbf{s}_k)$, then (6) becomes

$$\mathbf{H}_r^{(k)} = \mathbf{V}_k \mathbf{H}_r^{(k-1)} \mathbf{V}_k^T + \mathbf{Q}_k. \quad (73)$$

Fix any $r \in \mathbb{N}$. Since $\mathbf{y}_k = \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_k$, we have

$$\begin{aligned} \mathbf{V}_k &= \mathbf{I} - \frac{\nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_k} \\ &= \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r)^{1/2} \left(\mathbf{I} - \frac{\tilde{\mathbf{s}}_k \tilde{\mathbf{s}}_k^T}{\tilde{\mathbf{s}}_k^T \tilde{\mathbf{s}}_k} \right) \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r)^{-1/2}, \end{aligned}$$

where $\tilde{\mathbf{s}}_k \triangleq \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r)^{1/2} \mathbf{s}_k$. Hence

$$\begin{aligned} \|\mathbf{V}_k\| &\leq \left\| \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r)^{1/2} \right\| \left\| \mathbf{I} - \frac{\tilde{\mathbf{s}}_k \tilde{\mathbf{s}}_k^T}{\|\tilde{\mathbf{s}}_k\|^2} \right\| \left\| \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r)^{-1/2} \right\| \\ &\leq \bar{L}_{b_H}^{1/2} \bar{\mu}_{b_H}^{-1/2} = \kappa_{b_H}^{1/2}. \end{aligned}$$

Similarly, $\|\mathbf{Q}_k\| = \|\mathbf{s}_k\|^2 / (\mathbf{s}_k^T \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \mathbf{s}_k) \leq 1 / \bar{\mu}_{b_H}$. By (73),

$$\begin{aligned} \|\mathbf{H}_r^{(k)}\| &\leq \|\mathbf{V}_k\|^2 \|\mathbf{H}_r^{(k-1)}\| + \|\mathbf{Q}_k\| \\ &\leq \kappa_{b_H} \|\mathbf{H}_r^{(k-1)}\| + \frac{1}{\bar{\mu}_{b_H}}. \end{aligned} \quad (74)$$

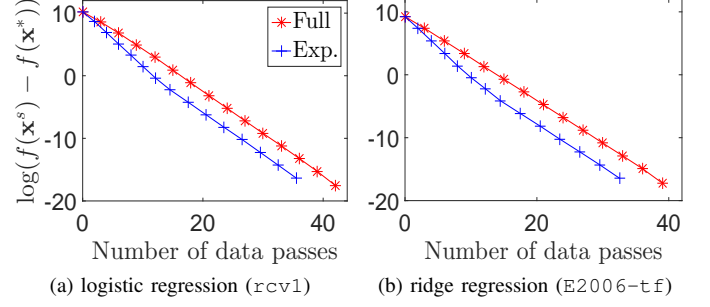


Fig. 3. Comparison of Algorithm 1 without (Full) and with (Exp.) using the partial gradient statblization strategy in Section VI-B.

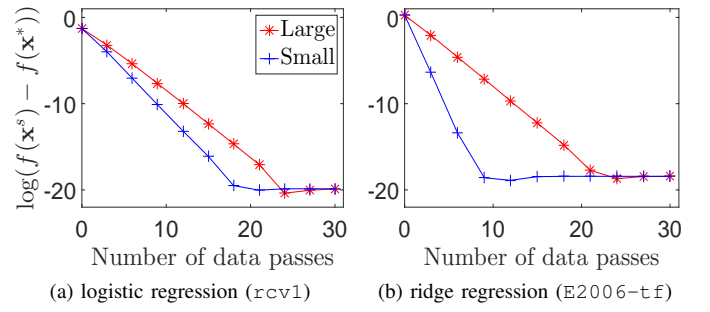


Fig. 4. Comparison of Algorithm 1 without (Large) and with (Small) using the low-dimensional Hessian strategy in Section VI-C.

We apply (74) repeatedly over $k = r - M' + 1, \dots, r$, then

$$\begin{aligned} \|\mathbf{H}_r\| &= \|\mathbf{H}_r^{(r)}\| \leq \kappa_{b_H}^{M'} \|\mathbf{H}_r^{(r-M')}\| + \frac{1}{\bar{\mu}_{b_H}} \sum_{i=0}^{M'-1} \kappa_{b_H}^i \\ &\stackrel{(a)}{\leq} \frac{1}{\bar{\mu}_{b_H}} \left(\kappa_{b_H}^{M'} + \frac{\kappa_{b_H}^{M'} - 1}{\kappa_{b_H} - 1} \right) \\ &\leq \frac{1}{\bar{\mu}_{b_H}} \kappa_{b_H}^{M'} \left(1 + \frac{1}{\kappa_{b_H} - 1} \right) \\ &\stackrel{(b)}{\leq} \frac{\kappa_{b_H}^{M+1}}{\bar{\mu}_{b_H} (\kappa_{b_H} - 1)}, \end{aligned}$$

where (a) follows from the definition of $\mathbf{H}_r^{(r-M')}$ in (5) and

$$\frac{\mathbf{s}_r^T \mathbf{y}_r}{\mathbf{y}_r^T \mathbf{y}_r} = \frac{\tilde{\mathbf{s}}_r^T \tilde{\mathbf{s}}_r}{\tilde{\mathbf{s}}_r^T \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \tilde{\mathbf{s}}_r} \leq \frac{1}{\bar{\mu}_{b_H}}, \quad (75)$$

and (b) follows from $\kappa_{b_H} \geq 1$ and $M' \leq M$. To show $\gamma = 1/(M+1)\bar{L}_{b_H}$, it suffices to show $\|\mathbf{B}_r\| \leq (M+1)\bar{L}_{b_H}$. We derive this bound using (18). Since

$$\begin{aligned} &\left\| \mathbf{B}_r^{(k-1)} - \frac{\mathbf{B}_r^{(k-1)} \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_r^{(k-1)}}{\mathbf{s}_k^T \mathbf{B}_r^{(k-1)} \mathbf{s}_k} \right\| \\ &\leq \left\| \left(\mathbf{B}_r^{(k-1)} \right)^{1/2} \right\|^2 \left\| \mathbf{I} - \frac{\hat{\mathbf{s}}_k \hat{\mathbf{s}}_k^T}{\|\hat{\mathbf{s}}_k\|^2} \right\| = \|\mathbf{B}_r^{(k-1)}\|, \end{aligned}$$

and

$$\left\| \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right\| = \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T \mathbf{y}_k} = \frac{\tilde{\mathbf{s}}_k^T \nabla^2 f_{\mathcal{T}_r}(\bar{\mathbf{x}}_r) \tilde{\mathbf{s}}_k}{\tilde{\mathbf{s}}_k^T \tilde{\mathbf{s}}_k} \leq \bar{L}_{b_H}, \quad (76)$$

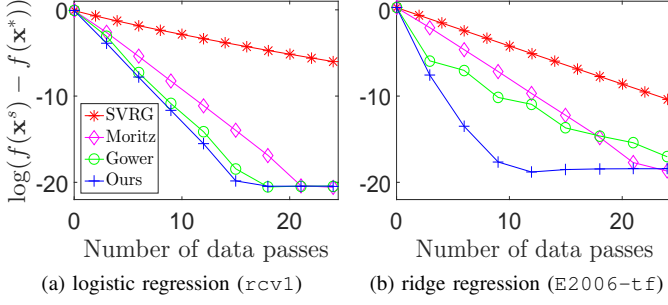


Fig. 5. Comparison of our algorithm (Ours) with benchmarking algorithms (SVRG, Moritz and Gower) on the rcv1 and E2006-tf datasets.

we have from (18) that

$$\|\mathbf{B}_r^{(k)}\| \leq \|\mathbf{B}_r^{(k-1)}\| + \left\| \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right\| \leq \|\mathbf{B}_r^{(k-1)}\| + \bar{L}_{b_H}.$$

Therefore,

$$\|\mathbf{B}_r\| = \|\mathbf{B}_r^{(r)}\| \leq \|\mathbf{B}_r^{(r-M')}\| + M' \bar{L}_{b_H} \stackrel{(a)}{\leq} (M+1) \bar{L}_{b_H},$$

where (a) follows from (76).

APPENDIX C PROOF OF PROPOSITION 2

Our proof is modified from that of Theorem 1. Specifically, our proof leverages a refined telescoping of (34). (The steps up to (34) are unchanged.) For each $t \in [m]$, we multiply both sides of (34) by $(1 - \eta\gamma\bar{\mu})^{m-t}$ and obtain

$$\begin{aligned} & (1 - \eta\gamma\bar{\mu})^{m-t} \left(\mathbb{E}_{\mathcal{B}_{s,t}} \left[\|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 \middle| \mathcal{F}_{s,t} \right] \right. \\ & \quad \left. + 2\eta \mathbb{E}_{\mathcal{B}_{s,t}} \left[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \middle| \mathcal{F}_{s,t} \right] \right) \\ & \leq (1 - \eta\gamma\bar{\mu})^{m-t} \left((1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t-1,r} - \tilde{\mathbf{x}}_r^*\|^2 \right. \\ & \quad \left. + \frac{8}{b} \Gamma \bar{L} \eta^2 (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t-1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) + \tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)) \right). \end{aligned} \quad (77)$$

Telescope (77) over $t = 1, \dots, m$ and we have

$$\begin{aligned} & 2c\eta \left(1 - \frac{4}{b} \frac{\Gamma \bar{L} \eta}{1 - \eta\gamma\bar{\mu}} \right) \\ & \quad \cdot \frac{1}{c} \sum_{t=1}^m (1 - \eta\gamma\bar{\mu})^{m-t} \mathbb{E}_{\mathcal{B}_{s,(t)}} \left[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \middle| \mathcal{F}_s \right] \\ & \leq (1 - \eta\gamma\bar{\mu})^m \|\tilde{\mathbf{x}}^{s,r'} - \tilde{\mathbf{x}}_{r'}^*\|^2 \\ & \quad + \frac{8}{b} \Gamma \bar{L} \eta^2 ((1 - \eta\gamma\bar{\mu})^m + c) (\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)). \end{aligned} \quad (78)$$

Now we consider using option IV to choose \mathbf{x}^{s+1} . Since $0 < \beta \leq 1 - \eta\gamma\bar{\mu}$, using (11) and Jensen's inequality, we have

$$\begin{aligned} & \frac{1}{c} \sum_{t=1}^m (1 - \eta\gamma\bar{\mu})^{m-t} \mathbb{E}_{\mathcal{B}_{s,(t)}} \left[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \middle| \mathcal{F}_s \right] \\ & \geq \mathbb{E}_{\mathcal{B}_{s,(m)}} \left[\tilde{f}_{r''}(\tilde{\mathbf{x}}^{s+1,r''}) - \tilde{f}(\tilde{\mathbf{x}}_{r''}^*) \middle| \mathcal{F}_s \right]. \end{aligned} \quad (79)$$

Alternatively, if \mathbf{x}^{s+1} is determined using option III, the definition of distribution Q still yields (79). Finally, using (93)

in Lemma T-1 to bound $\|\tilde{\mathbf{x}}^{s,r'} - \tilde{\mathbf{x}}_{r'}^*\|^2$ in (78), we have

$$\begin{aligned} & 2c'\eta \left(1 - \frac{4}{b} \frac{\Gamma \bar{L} \eta}{1 - \eta\gamma\bar{\mu}} \right) \mathbb{E}_{\mathcal{B}_{s,(m)}} \left[\tilde{f}_{r''}(\tilde{\mathbf{x}}^{s+1,r''}) - \tilde{f}(\tilde{\mathbf{x}}_{r''}^*) \middle| \mathcal{F}_s \right] \\ & \leq \left(\frac{8}{b} \Gamma \bar{L} \eta^2 (1 + c') + \frac{2}{\gamma\bar{\mu}} \right) (\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)). \end{aligned} \quad (80)$$

Taking expectation on both sides and using (11), we arrive at (52).

APPENDIX D PROOF OF PROPOSITION 3

The subsampled gradient strategy essentially introduces errors in $\{\mathbf{g}_s\}_{s \geq 0}$. Therefore, we explicit model this error by $\mathbf{e}_s \triangleq \tilde{\mathbf{g}}_s - \mathbf{g}_s$. We first bound the second moment of \mathbf{e}_s . Since $\mathbf{g}_s = \nabla f(\mathbf{x}^s)$, by Lemma T-3 and $b_s \geq nS^2\alpha_s/(S^2\alpha_s + (n-1)\xi^2\bar{\rho}^{2s})$, we have for any $s \in [S]$,

$$\mathbb{E}_{\tilde{\mathcal{B}}_s} [\|\mathbf{e}_s\|^2] \leq \frac{n - \tilde{b}_s}{\tilde{b}_s(n-1)} \alpha_s \leq \frac{\xi^2}{S^2} \bar{\rho}^{2s}. \quad (81)$$

As a result,

$$\mathbb{E}_{\tilde{\mathcal{B}}_s} [\|\mathbf{e}_s\|] \leq \sqrt{\mathbb{E}_{\tilde{\mathcal{B}}_s} [\|\mathbf{e}_s\|^2]} \leq \frac{\xi}{S} \bar{\rho}^s. \quad (82)$$

The introduction of random sets $\{\tilde{\mathcal{B}}_s\}_{s \geq 0}$ requires us to redefine the filtration $\{\mathcal{F}_{s,t}\}_{s \geq 0, t \in (m-1]}$ as

$$\begin{aligned} \mathcal{F}_{s,t} \triangleq \sigma \left(\{\tau_j\}_{j=0}^{s-1} \cup \{\tilde{\mathcal{B}}_j\}_{j \in [s]} \cup \{\mathcal{B}_{i,j}\}_{i \in (s-1], j \in (m-1]} \right. \\ \left. \cup \{\mathcal{B}_{s,j}\}_{j=0}^{t-1} \cup \{\mathcal{T}_j\}_{j=0}^{\lfloor (sm+t)/L \rfloor} \right). \end{aligned} \quad (83)$$

As usual, we define $\mathcal{F}_s \triangleq \mathcal{F}_{s,0}$. In the sequel, we follow the naming convention in coordinate transformation framework as in Section IV-A. In particular, we define $\tilde{\mathbf{e}}_{s,r'} \triangleq \mathbf{H}_{r'}^{1/2} \mathbf{e}_s$. Now, define $\tilde{\mathbf{v}}'_{s,t,r} \triangleq \tilde{\mathbf{v}}_{s,t,r} + \tilde{\mathbf{e}}_{s,r'}$, then from (24) and (25), we have

$$\mathbb{E}_{\mathcal{B}_{s,t}} [\tilde{\mathbf{v}}'_{s,t,r} | \mathcal{F}_{s,t}] = \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) + \tilde{\mathbf{e}}_{s,r'}, \quad (84)$$

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_{s,t}} \left[\left\| \tilde{\mathbf{v}}'_{s,t,r} - \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) \right\|^2 \middle| \mathcal{F}_{s,t} \right] & \leq \frac{4\Gamma \bar{L}}{b} (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) \\ & - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) + \tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)) + \|\tilde{\mathbf{e}}_{s,r'}\|^2. \end{aligned} \quad (85)$$

Define $\tilde{\delta}'_{s,t,r} \triangleq \tilde{\mathbf{v}}'_{s,t,r} - \nabla \tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r})$. We can derive an inequality similar to (33) in the proof of Theorem 1, i.e.,

$$\begin{aligned} \|\tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^*\|^2 & \leq (1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 - 2\eta (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) \\ & - \tilde{f}_r(\tilde{\mathbf{x}}_r^*)) + 2\eta^2 \|\tilde{\delta}'_{s,t,r}\|^2 - 2\eta \langle \tilde{\delta}'_{s,t,r}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle, \end{aligned}$$

Taking expectation w.r.t. $\mathcal{B}_{s,t}$ and using (84) and (85), we have

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_{s,t}} \left[\|\tilde{\mathbf{x}}_{s,t+1,r} - \tilde{\mathbf{x}}_r^*\|^2 + 2\eta (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t+1,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*)) \middle| \mathcal{F}_{s,t} \right] \\ \leq (1 - \eta\gamma\bar{\mu}) \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\|^2 + \frac{8}{b} \Gamma \bar{L} \eta^2 (\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \\ + \tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*)) + 2\eta^2 \|\tilde{\mathbf{e}}_{s,r'}\|^2 - 2\eta \langle \tilde{\mathbf{e}}_{s,r'}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle. \end{aligned} \quad (86)$$

Using Cauchy-Schwartz inequality, we have

$$\begin{aligned} -2\eta \langle \tilde{\mathbf{e}}_{s,r'}, \tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^* \rangle & \leq 2\eta \|\tilde{\mathbf{e}}_{s,r'}\| \|\tilde{\mathbf{x}}_{s,t,r} - \tilde{\mathbf{x}}_r^*\| \\ & \leq 2\eta \|\tilde{\mathbf{e}}_{s,r'}\| \|\mathbf{H}_r^{-1/2}\| \|\mathbf{x}_{s,t} - \mathbf{x}^*\| \leq 2\eta B\gamma^{-1/2} \|\tilde{\mathbf{e}}_{s,r'}\|. \end{aligned} \quad (87)$$

$$\mathbb{E}_{\mathcal{B}_{s,t}} \left[\|\mathbf{v}_{s,t} - \nabla f(\mathbf{x}_{s,t})\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (65)$$

$$= \mathbb{E}_{\mathcal{B}_{s,t}} \left[\left\| \frac{1}{b} \sum_{j=1}^b \left(\frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}_{s,t}) - \nabla f_{i_j}(\mathbf{x}^s)) + \nabla f(\mathbf{x}^s) - \nabla f(\mathbf{x}_{s,t}) \right) \right\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (66)$$

$$= \frac{1}{b^2} \sum_{j=1}^b \mathbb{E}_{i_j} \left[\left\| \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}_{s,t}) - \nabla f_{i_j}(\mathbf{x}^s)) + \nabla f(\mathbf{x}^s) - \nabla f(\mathbf{x}_{s,t}) \right\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (67)$$

$$\leq \frac{1}{b^2} \sum_{j=1}^b \mathbb{E}_{i_j} \left[\left\| \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}_{s,t}) - \nabla f_{i_j}(\mathbf{x}^s)) \right\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (68)$$

$$= \frac{1}{b^2} \sum_{j=1}^b \mathbb{E}_{i_j} \left[\left\| \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}_{s,t}) - \nabla f_{i_j}(\mathbf{x}^*)) + \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}^*) - \nabla f_{i_j}(\mathbf{x}^s)) \right\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (69)$$

$$\leq \frac{2}{b^2} \sum_{j=1}^b \mathbb{E}_{i_j} \left[\left\| \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}_{s,t}) - \nabla f_{i_j}(\mathbf{x}^*)) \right\|^2 \middle| \mathcal{F}_{s,t} \right] + \mathbb{E}_{i_j} \left[\left\| \frac{1}{np_{i_j}} (\nabla f_{i_j}(\mathbf{x}^s) - \nabla f_{i_j}(\mathbf{x}^*)) \right\|^2 \middle| \mathcal{F}_{s,t} \right] \quad (70)$$

$$\leq \frac{2}{b^2} \sum_{j=1}^b 2\bar{L}(f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*)) + 2\bar{L}(f(\mathbf{x}^s) - f(\mathbf{x}^*)) = \frac{4\bar{L}}{b} (f(\mathbf{x}_{s,t}) - f(\mathbf{x}^*) + f(\mathbf{x}^s) - f(\mathbf{x}^*)). \quad (71)$$

Substituting (87) into (86), and using the telescoping techniques in Appendix C, we have

$$\begin{aligned} & 2c\eta \left(1 - \frac{4}{b} \frac{\Gamma\bar{L}\eta}{1 - \eta\gamma\bar{\mu}} \right) \frac{1}{c} \sum_{t=1}^m (1 - \eta\gamma\bar{\mu})^{m-t} \\ & \cdot \mathbb{E}_{\mathcal{B}_{s,(t)}} \left[\tilde{f}_r(\tilde{\mathbf{x}}_{s,t,r}) - \tilde{f}_r(\tilde{\mathbf{x}}_r^*) \middle| \mathcal{F}_s \right] \leq (1 - \eta\gamma\bar{\mu})^m \|\tilde{\mathbf{x}}^{s,r'} - \mathbf{x}_{r'}^*\|^2 \\ & + \frac{8}{b} \Gamma\bar{L}\eta^2 ((1 - \eta\gamma\bar{\mu})^m + c) \left(\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*) \right) \\ & + 2\eta ((1 - \eta\gamma\bar{\mu})^m + c) \left(B\gamma^{-1/2} \|\tilde{\mathbf{e}}_{s,r'}\| + \eta \|\tilde{\mathbf{e}}_{s,r'}\|^2 \right). \end{aligned}$$

With either option III or IV and (93) in Lemma T-1, we have

$$\begin{aligned} & 2c'\eta \left(1 - \frac{4}{b} \frac{\Gamma\bar{L}\eta}{1 - \eta\gamma\bar{\mu}} \right) \mathbb{E}_{\mathcal{B}_{s,(m)}} \left[\tilde{f}_{r''}(\tilde{\mathbf{x}}^{s+1,r''}) - \tilde{f}(\tilde{\mathbf{x}}_{r''}^*) \middle| \mathcal{F}_s \right] \\ & \leq \left(\frac{8}{b} \Gamma\bar{L}\eta^2 (1 + c') + \frac{2}{\gamma\bar{\mu}} \right) \left(\tilde{f}_{r'}(\tilde{\mathbf{x}}^{s,r'}) - \tilde{f}_{r'}(\tilde{\mathbf{x}}_{r'}^*) \right) \\ & + 2\eta (1 + c') \left(B\gamma^{-1/2} \|\tilde{\mathbf{e}}_{s,r'}\| + \eta \|\tilde{\mathbf{e}}_{s,r'}\|^2 \right). \quad (88) \end{aligned}$$

Taking expectation on both sides and using (11), we have

$$\begin{aligned} & \mathbb{E} [f(\mathbf{x}^{s+1}) - f(\mathbf{x}^*)] \leq \bar{\rho} \mathbb{E} [f(\mathbf{x}^s) - f(\mathbf{x}^*)] + (1 + 1/c') \\ & \cdot \frac{b}{b - 4\Gamma\bar{L}\eta/(1 - \eta\gamma\bar{\mu})} \left(B\gamma^{-1/2} \mathbb{E} [\|\tilde{\mathbf{e}}_{s,r'}\|] + \eta \mathbb{E} [\|\tilde{\mathbf{e}}_{s,r'}\|^2] \right). \quad (89) \end{aligned}$$

From (81) and (82), we have

$$\mathbb{E} [\|\tilde{\mathbf{e}}_{s,r'}\|] \leq \mathbb{E} [\|\mathbf{H}_{r'}^{1/2}\| \|\mathbf{e}_s\|] \leq \frac{\Gamma^{1/2}\xi}{S} \bar{\rho}^s, \quad (90)$$

$$\mathbb{E} [\|\tilde{\mathbf{e}}_{s,r'}\|^2] \leq \mathbb{E} [\|\mathbf{H}_{r'}^{1/2}\|^2 \|\mathbf{e}_s\|^2] \leq \frac{\Gamma\xi^2}{S^2} \bar{\rho}^{2s}. \quad (91)$$

Substituting (90) and (91) into (89), we have

$$\begin{aligned} & \mathbb{E} [f(\mathbf{x}^{s+1}) - f(\mathbf{x}^*)] \leq \bar{\rho} \mathbb{E} [f(\mathbf{x}^s) - f(\mathbf{x}^*)] \\ & + \left(1 + \frac{1}{c'} \right) \frac{b}{b - 4\Gamma\bar{L}\eta/(1 - \eta\gamma\bar{\mu})} \left(\kappa_H^{1/2} B + \eta\Gamma\xi \right) \frac{\xi}{S} \bar{\rho}^s. \quad (92) \end{aligned}$$

Applying (92) recursively and we reach (54).

APPENDIX E TECHNICAL LEMMAS

Lemmas T-1, T-2, and T-3 can be found in [45, Chapter 9], [27], and [46] respectively.

Lemma T-1. *If a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -strongly convex and L -smooth, then for any $\mathbf{x} \in \mathbb{R}^d$,*

$$\frac{\mu}{2} \|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{1}{2\mu} \|\nabla f(\mathbf{x})\|^2, \quad (93)$$

$$\frac{1}{2L} \|\nabla f(\mathbf{x})\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{x}^*\|^2, \quad (94)$$

where \mathbf{x}^* denotes the unique minimizer of f on \mathbb{R}^d .

Lemma T-2. *Let f be defined as in (1) and satisfies Assumption 2. Define a distribution p with support $[n]$ such that $p_i = L_i/(n\bar{L})$, for any $i \in [n]$. Then for any $\mathbf{x} \in \mathbb{R}^d$,*

$$\mathbb{E}_{i \sim p} \left[\frac{1}{np_i} \nabla f_i(\mathbf{x}) \right] = \nabla f(\mathbf{x}), \quad (95)$$

$$\mathbb{E}_{i \sim p} \left[\left\| \frac{1}{np_i} (\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}^*)) \right\|^2 \right] \leq 2\bar{L}(f(\mathbf{x}) - f(\mathbf{x}^*)), \quad (96)$$

where \mathbf{x}^* denotes the unique minimizer of f on \mathbb{R}^d .

Lemma T-3. *Let $\{\mathbf{z}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ and define $\bar{\mathbf{z}} \triangleq 1/n \sum_{i=1}^n \mathbf{z}_i$. Uniformly sample a random subset \mathcal{S} of $[n]$ with size b without replacement. Then*

$$\mathbb{E}_{\mathcal{S}} \left[\left\| \frac{1}{b} \sum_{i \in \mathcal{S}} \mathbf{z}_i - \bar{\mathbf{z}} \right\|^2 \right] \leq \frac{n-b}{b(n-1)} \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{z}_i\|^2 \right). \quad (97)$$

REFERENCES

- [1] R. Zhao, W. B. Haskell, and V. Y. F. Tan, “Stochastic l-bfgs revisited: Improved convergence rates and practical acceleration strategies,” in *Proc. UAI*, Sydney, Australia, Aug 2017.
- [2] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Math. Program.*, vol. 45, no. 3, Dec. 1989.
- [3] L. Bottou, “Online algorithms and stochastic approximations,” in *Online Learning and Neural Networks*. Cambridge University Press, 1998.
- [4] L. Bottou and Y. LeCun, “Large scale online learning,” in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2004, pp. 1361–1368.
- [5] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Proc. NIPS*, Lake Tahoe, Nevada, USA, 2013, pp. 315–323.
- [6] A. Defazio, F. Bach, and S. Lacoste-Julien, “Saga: A fast incremental gradient method with support for non-strongly convex composite objectives,” in *Proc. NIPS*, Montréal, Québec, Canada, 2014, pp. 1646–1654.
- [7] R. Harikandeh, M. O. Ahmed, A. Virani, M. Schmidt, J. Konečný, and S. Sallinen, “Stopwasting my gradients: Practical svrg,” in *Proc. NIPS*, Montréal, Quebec, Canada, 2015, pp. 2251–2259.
- [8] M. Schmidt, N. Le Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” *Math. Program.*, vol. 162, no. 1, pp. 83–112, 2017.
- [9] N. N. Schraudolph, J. Yu, and S. Günter, “A stochastic quasi-Newton method for online convex optimization,” in *Proc. ATSTATS*, San Juan, Puerto Rico, March 2007.
- [10] A. Bordes, L. Bottou, and P. Gallinari, “SGD-QN: Careful quasi-newton stochastic gradient descent,” *J. Mach. Learn. Res.*, vol. 10, pp. 1737–1754, Dec. 2009.
- [11] J. Sohl-Dickstein, B. Poole, and S. Ganguli, “Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods,” in *Proc. ICML*, Beijing, China, June 2014, pp. 604–612.
- [12] A. Mokhtari and A. Ribeiro, “RES: Regularized stochastic bfgs algorithm,” *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6089–6104, 2014.
- [13] —, “Global convergence of online limited memory BFGS,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3151–3181, Jan 2015.
- [14] A. Mokhtari, M. Eisen, and A. Ribeiro, “IQN: An incremental quasi-newton method with local superlinear convergence rate,” arXiv:1702.00709, 2017.
- [15] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, “A stochastic quasi-newton method for large-scale optimization,” *SIAM J. Optim.*, vol. 26, no. 2, pp. 1008–1031, 2016.
- [16] P. Moritz, R. Nishihara, and M. I. Jordan, “A linearly-convergent stochastic L-BFGS algorithm,” in *Proc. AISTATS*, Cadiz, Spain, May 2016, pp. 249–258.
- [17] R. M. Gower, D. Goldfarb, and P. Richtárik, “Stochastic block BFGS: squeezing more curvature out of data,” in *Proc. ICML*, New York City, NY, USA, June 2016, pp. 1869–1878.
- [18] E. Wei, A. Ozdaglar, and A. Jadbabaie, “A distributed newton method for network utility maximization-i: Algorithm,” *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2162–2175, 2013.
- [19] D. P. Bertsekas, “Centralized and distributed newton methods for network optimization and extensions,” arXiv:1507.00702, 2015.
- [20] A. Mokhtari, A. Koppel, and A. Ribeiro, “A class of parallel doubly stochastic algorithms for large-scale learning,” arXiv:1606.04991, 2016.
- [21] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, “A decentralized second-order method with exact linear convergence rate for consensus optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 507–522, 2016.
- [22] A. Mokhtari, Q. Ling, and A. Ribeiro, “Network newton distributed optimization methods,” *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, 2017.
- [23] M. Eisen, A. Mokhtari, and A. Ribeiro, “A decentralized quasi-newton method for dual formulations of consensus optimization,” in *Proc. CDC*, 2016, pp. 1951–1958.
- [24] —, “Decentralized quasi-newton methods,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613–2628, 2017.
- [25] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.
- [26] D. Goldfarb, “A family of variable metric updates derived by variational means,” *Math. Comput.*, vol. 24, no. 109, pp. 23–26, 1970.
- [27] L. Xiao and T. Zhang, “A proximal stochastic gradient method with progressive variance reduction,” *SIAM J. Optim.*, vol. 24, no. 4, pp. 2057–2075, 2014.
- [28] P. Zhao and T. Zhang, “Stochastic optimization with importance sampling for regularized loss minimization,” in *Proc. ICML*, Lille, France, Jul 2015, pp. 1–9.
- [29] D. Goldfarb, G. Iyengar, and C. Zhou, “Linear Convergence of Stochastic Frank Wolfe Variants,” in *Proc. AISTATS*, Fort Lauderdale, FL, USA, Apr 2017, pp. 1066–1074.
- [30] D. Williams, *Probability with Martingales*. Cambridge University Press, 1991.
- [31] H. Lin, J. Mairal, and Z. Harchaoui, “A universal catalyst for first-order optimization,” in *Proc. NIPS*, 2015, pp. 3384–3392.
- [32] Z. Allen-Zhu, “Katyusha: The first direct acceleration of stochastic gradient methods,” arXiv:1603.05953, 2016.
- [33] J. Konečný and P. Richtárik, “Semi-stochastic gradient descent methods,” arXiv:1312.1666, 2013.
- [34] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Cambridge, 2008.
- [35] C. Hu, W. Pan, and J. T. Kwok, “Accelerated gradient methods for stochastic optimization and online learning,” in *Proc. NIPS*, Vancouver, B.C., Canada, 2009, pp. 781–789.
- [36] D. P. Woodruff, “Sketching as a tool for numerical linear algebra,” *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 1–2, pp. 1–157, 2014.
- [37] M. Pilanci and M. J. Wainwright, “Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares,” *J. Mach. Learn. Res.*, vol. 17, no. 53, pp. 1–38, 2016.
- [38] L. Luo, Z. Chen, Z. Zhang, and W.-J. Li, “A proximal stochastic quasi-newton algorithm,” arXiv:1602.00223, 2016.
- [39] M. Pilanci and M. J. Wainwright, “Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence,” *SIAM J. Optim.*, vol. 27, no. 1, pp. 205–245, 2017.
- [40] M. Ghashami, E. Liberty, J. M. Phillips, and D. P. Woodruff, “Frequent directions: Simple and deterministic matrix sketching,” *SIAM J. Comput.*, vol. 45, no. 5, pp. 1762–1792, 2016.
- [41] R. M. Gower and J. Gondzio, “Action constrained quasi-newton methods,” 2014.
- [42] P. Hennig, “Probabilistic interpretation of linear solvers,” *SIAM J. Optim.*, vol. 25, no. 1, pp. 234–260, 2015.
- [43] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011.
- [44] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Trans. Math. Softw.*, vol. 23, no. 4, Dec 1997.
- [45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [46] A. Nitanda, “Stochastic proximal gradient descent with acceleration techniques,” in *Proc. NIPS*, Montreal, Quebec, Canada, Dec. 2014, pp. 1574–1582.