Journal of Machine Learning Research 21 (2020) 1-48 Submitted 3/19; Revised 2/20; Published 5/20

ProxSARAH: An Efficient Algorithmic Framework for Stochastic Composite Nonconvex Optimization

Nhan H. Pham[†] NHANPH@LIVE.UNC.EDU
Lam M. Nguyen[‡] LAMNGUYEN.MLTD@IBM.COM
Dzung T. Phan[‡] PHANDU@US.IBM.COM
[‡]*IBM Research, Thomas J. Watson Research Center*
Yorktown Heights, NY10598, USA
Quoc Tran-Dinh[†] QUOCTD@EMAIL.UNC.EDU
[†]*Department of Statistics and Operations Research*
The University of North Carolina at Chapel Hill, Chapel Hill, NC27599, USA.

Editor: Zaid Harchaoui

Abstract

We propose a new stochastic first-order algorithmic framework to solve stochastic composite nonconvex optimization problems that covers both finite-sum and expectation settings. Our algorithms rely on the SARAH estimator introduced in Nguyen et al. (2017a) and consist of two steps: a proximal gradient and an averaging step making them different from existing nonconvex proximal-type algorithms. The algorithms only require an average smoothness assumption of the nonconvex objective term and additional bounded variance assumption if applied to expectation problems. They work with both constant and dynamic step-sizes, while allowing single sample and mini-batches. In all these cases, we prove that our algorithms can achieve the best-known complexity bounds in terms of stochastic first-order oracle. One key step of our methods is the new constant and dynamic step-sizes resulting in the desired complexity bounds while improving practical performance. Our constant step-size is much larger than existing methods including proximal SVRG scheme in the single sample case. We also specify our framework to the non-composite case that covers existing state-of-the-arts in terms of oracle complexity bounds. Our update also allows one to trade-off between step-sizes and mini-batch sizes to improve performance. We test the proposed algorithms on two composite nonconvex problems and neural networks using several well-known data sets.

Keywords: Stochastic proximal gradient descent; variance reduction; composite nonconvex optimization; finite-sum minimization; expectation minimization.

1. Introduction

In this paper, we consider the following stochastic composite, nonconvex, and possibly nonsmooth optimization problem:

$$\min_{w\in\mathbb{R}^d}\Big\{F(w) := f(w) + \psi(w) \equiv \mathbb{E}\left[\mathbf{f}(w;\xi)\right] + \psi(w)\Big\}, \tag{1}$$

where $f(w) := \mathbb{E}\left[\mathbf{f}(w;\xi)\right]$ is the expectation of a stochastic function $\mathbf{f}(w;\xi)$ depending on a random vector ξ in a given probability space $(\Omega, \mathbb{P})$, and $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed, and convex function.

As a special case of (1), if ξ is a random vector defined on a finite support set $\Omega := \{\xi_1, \xi_2, \cdots, \xi_n\}$ with a probability distribution p, then by defining $f_i(w) := np_i\mathbf{f}(w;\xi_i)$, (1) can be written into the following composite finite-sum minimization problem:

$$\min_{w\in\mathbb{R}^d}\Big\{F(w) := f(w) + \psi(w) \equiv \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \psi(w)\Big\}. \tag{2}$$

We can also obtain (2) from (1) through a sample average approximation (SAA) (Nemirovski et al., 2009). Note that problem (2) is often referred to as a regularized empirical risk minimization in machine learning and finance.

1.1 Motivation

Problems (1) and (2) cover a broad range of applications in machine learning and statistics, especially in neural networks (see Bottou, 1998, 2010; Bottou et al., 2018; Goodfellow et al., 2016; Sra et al., 2012). Hitherto, state-of-the-art numerical optimization methods for solving these problems rely on stochastic approaches (see Johnson and Zhang, 2013; Schmidt et al., 2017; Shapiro et al., 2009; Defazio et al., 2014; Frostig et al., 2015; Lei and Jordan, 2017; Lin et al., 2015). In the convex case, both non-composite and composite settings of (1) and (2) have been intensively studied with different schemes such as standard stochastic gradient (Robbins and Monro, 1951), proximal stochastic gradient (Ghadimi and Lan, 2013; Nemirovski et al., 2009), stochastic dual coordinate descent (Shalev-Shwartz and Zhang, 2013), variance reduction methods (Allen-Zhu, 2017; Defazio et al., 2014; Johnson and Zhang, 2013; Nitanda, 2014; Schmidt et al., 2017; Shalev-Shwartz and Zhang, 2014; Xiao and Zhang, 2014), stochastic conditional gradient (Frank-Wolfe) methods (Reddi et al., 2016a), and stochastic primal-dual methods (Chambolle et al., 2018). The most popular variance reduction methods in the literature are perhaps SAGA and SVRG. While SAGA (fast incremental gradient algorithm) is a successor of SAG (**S**tochastic **A**verage **G**radient) (Schmidt et al., 2017) and aims at solving finite-sum problems, SVRG (**S**tochastic **V**ariance **R**educed **G**radient) (Johnson and Zhang, 2013) can solve both finite-sum and expectation problems. Thanks to variance reduction techniques, several efficient methods with constant step-sizes have been developed for convex settings that match the lower-bound worst-case complexity (Agarwal et al., 2010). However, variance reduction methods for nonconvex settings are still limited and heavily focus on the non-composite form of (1) and (2), i.e., $\psi = 0$, and the SVRG estimator.

Theory and stochastic methods for nonconvex problems are still in progress and require substantial effort to obtain efficient algorithms with rigorous convergence guarantees. It is shown in Fang et al. (2018); Zhou and Gu (2019) that there is still a gap between the upper-bound complexity in state-of-the-art methods and the lower-bound worst-case complexity for the nonconvex problem (2) under standard smoothness assumption. Motivated by this fact, we attempt to develop a new algorithmic framework that can reduce and at least nearly close this gap in the composite finite-sum setting (2). In addition to the best-known complexity bounds, we expect to design practical algorithms advancing beyond existing

methods by providing a dynamic rule to update step-sizes with rigorous complexity analysis. Our algorithms rely on a recent biased stochastic estimator for the objective gradient, called SARAH (**S**toch**A**stic **R**ecursive gr**A**dient algorit**H**m), introduced in Nguyen et al. (2017a) for convex problems.

1.2 Related Work

In the nonconvex case, both problems (1) and (2) have been intensively studied in recent years with a vast number of research papers. While numerical algorithms for solving the non-composite setting, i.e., $\psi = 0$, are well-developed and have received considerable attention (see Allen-Zhu, 2018; Allen-Zhu and Li, 2018; Allen-Zhu and Yuan, 2016; Fang et al., 2018; Lihua et al., 2017; Nguyen et al., 2017b, 2018b, 2019; Reddi et al., 2016b; Zhou et al., 2018), methods for composite setting remain limited (Reddi et al., 2016b; Wang et al., 2019). In terms of algorithms, Reddi et al. (2016b) study a non-composite finite-sum problem as a special case of (2) using SVRG estimator from Johnson and Zhang (2013). Additionally, they extend their method to the composite setting by simply applying the proximal operator of ψ as in the well-known forward-backward scheme. Another related work using SVRG estimator can be found in Li and Li (2018). These algorithms have some limitation as will be discussed later. The same technique is applied in Wang et al. (2019) to develop other variants for both (1) and (2), but using the SARAH estimator from Nguyen et al. (2017a). The authors derive a large constant step-size, but at the same time control mini-batch size to achieve desired complexity bounds. Consequently, it has an essential limitation as will also be discussed in Subsection 3.4. Both algorithms achieve the best-known complexity bounds for solving (1) and (2). In addition, Reddi et al. (2016a) propose a stochastic Frank-Wolfe method that can handle constraints as special cases of (2). Recently, a stochastic variance reduction method with momentum was studied in Zhou et al. (2019) for solving (2) which can be viewed as a modification of SpiderBoost in Wang et al. (2019).

Our algorithm remains a variance reduction stochastic method, but it is different from these works at two major points: an additional averaging step and two different step-sizes (*cf.* Algorithm 1). Having two step-sizes allows us to flexibly trade-off them and develop a dynamic update rule. Note that our averaging step looks similar to the robust stochastic gradient method in Nemirovski et al. (2009), but is fundamentally different since it evaluates the proximal step at the averaging point. In fact, it is closely related to averaged fixed-point schemes in the literature (see Bauschke and Combettes, 2017). While we only focus on stochastic gradient-type methods in this paper, some recent techniques such as Nesterov's momentum, catalyst, and nonlinear acceleration (see Paquette et al., 2018) could also be interesting to investigate for developing new variants of our methods.

In terms of theory, many researchers have focused on theoretical aspects of existing algorithms. For example, Ghadimi and Lan (2013) appears to be one of the first pioneering works studying convergence rates of stochastic gradient descent-type methods for nonconvex and non-composite finite-sum problems. They later extend it to the composite setting in Ghadimi et al. (2016). Wang et al. (2019) also investigate the gradient dominance case, and Karimi et al. (2016) consider both finite-sum and composite finite-sum under different assumptions, including Polyak-Łojasiewicz condition.

Whereas many researchers have been trying to improve complexity upper bounds of stochastic first-order methods using different techniques (Allen-Zhu, 2018; Allen-Zhu and Li, 2018; Allen-Zhu and Yuan, 2016; Fang et al., 2018), other researchers attempt to construct examples for lower-bound complexity estimates. In the convex case, there exist numerous research papers including Agarwal et al. (2010); Nemirovskii and Yudin (1983); Nesterov (2004). In Fang et al. (2018); Zhou and Gu (2019), the authors have constructed a lower-bound complexity for nonconvex finite-sum problem covered by (2). They showed that the lower-bound complexity for any stochastic gradient method using only smoothness assumption to achieve an ε-stationary point in expectation is $\Omega\left(n^{1/2}\varepsilon^{-2}\right)$ given that the number of objective components n does not exceed $\mathcal{O}\left(\varepsilon^{-4}\right)$, where ε is a desired accuracy.

For the expectation problem (1), the best-known complexity bound to achieve an ε-stationary point in expectation is $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$ as shown in Fang et al. (2018); Wang et al. (2019), where $\sigma > 0$ is an upper bound of the variance (see Assumption 2.3). This complexity matches the lower bound recently developed in Arjevani et al. (2019) up to a given constant under the same assumptions for the non-composite setting of (1).

1.3 Our Approach and Contributions

We exploit the SARAH estimator, a biased stochastic recursive gradient estimator, in Nguyen et al. (2017a), to design new proximal variance reduction stochastic gradient algorithms to solve both composite expectation and finite-sum problems (1) and (2). The SARAH algorithm is simply a double-loop stochastic gradient method with a flavor of SVRG (Johnson and Zhang, 2013), but using a novel biased estimator that is different from SVRG. SARAH is a recursive method as SAGA (Defazio et al., 2014), but can avoid the major issue of storing gradients as in SAGA. Our method will rely on the SARAH estimator as in SPIDER and SpiderBoost combining with an averaging proximal-gradient scheme to solve both (1) and (2).

The ultimate goal of this paper is to develop a new stochastic gradient-based algorithmic framework that covers different variants with constant and dynamic step-sizes, single sample and mini-batch, and achieves best-known theoretical oracle complexity bounds. More specifically, our main contributions can be summarized as follows:

(a) ***Novel algorithms:*** We propose a new and general stochastic variance reduction framework (Algorithm 1) relying on the SARAH estimator to solve both expectation and finite-sum problems (1) and (2) in composite settings. As usual, the algorithm has double loops, where the outer loop can either take full gradient or mini-batch to reduce computational burden in large-scale and expectation settings. The inner loop can work with single sample or a broad range of mini-batch sizes. This framework has two different step-sizes as opposed to existing methods. We also derive different variants of Algorithm 1 for using constant or dynamic step-sizes and for non-composite settings of (1) and (2) (i.e., $\psi = 0$)

(b) ***Best-known complexity guarantees under constant step-sizes:*** We analyze our framework and its variants to design appropriate constant step-sizes instead of diminishing step-sizes as in standard **S**tochastic **G**radient **D**escent (SGD) methods. In the finite-sum setting (2), our methods achieve $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ complexity bound to attain an ε-stationary point in expectation under only the smoothness of f_i. This

complexity matches the lower-bound worst-case complexity in Fang et al. (2018); Zhou and Gu (2019) up to a constant factor when $n \leq \mathcal{O}\left(\varepsilon^{-4}\right)$. In the expectation setting (1), our algorithms require $\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-3}\right)$ stochastic first-order oracle calls of $\mathbf{f}$ to achieve an ε-stationary point in expectation under only the smoothness of $\mathbf{f}$ and bounded variance $\sigma^2 > 0$. To the best of our knowledge, this is the best-known complexity so far for (1) under standard assumptions in both the single sample and mini-batch cases. This complexity also matches the lower bound recently studied in Arjevani et al. (2019) up to a constant.

(c) ***Best-known complexity guarantees under dynamic step-sizes:*** Apart from constant step-size algorithms, we also analyze variants of Algorithm 1 using dynamic step-sizes for both composite and non-composite settings in both single sample and mini-batch cases. Our dynamic step-sizes are increasing along the inner iterations rather than diminishing as usually used in standard SGDs.

Our result covers the non-composite setting in the finite-sum case (Nguyen et al., 2019), and matches the best-known complexity in Fang et al. (2018); Wang et al. (2019) for both problems (1) and (2). Since the composite setting covers a broader class of nonconvex problems including convex constraints, we believe that our method has better chance to handle new applications than non-composite methods. It also allows one to deal with composite problems under different type of regularizers such as sparsity or constraints on weights as in neural network training applications.

Algorithms	Finite-sum	Expectation	Composite	Step-size	D-step-size	Mb-range
GD (Nesterov, 2004)	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	NA	☑	$\mathcal{O}\left(\frac{1}{L}\right)$	☑	☒
SGD (Ghadimi and Lan, 2013)	NA	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^4}\right)$	☑	$\mathcal{O}\left(\frac{1}{L}\right)$	☑	☑
SVRG/SAGA (Reddi et al., 2016b)	$\mathcal{O}\left(n + \frac{n^{2/3}}{\varepsilon^2}\right)$	NA	☑	$\mathcal{O}\left(\frac{1}{nL}\right) \to \mathcal{O}\left(\frac{1}{L}\right)$	☒	☒
SVRG+ (Li and Li, 2018)	$\mathcal{O}\left(n + \frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^{10/3}}\right)$	☑	$\mathcal{O}\left(\frac{1}{nL}\right) \to \mathcal{O}\left(\frac{1}{L}\right)$	☒	☒
SCSG (Lihua et al., 2017)	$\mathcal{O}\left(n + \frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2} + \frac{\sigma}{\varepsilon^{10/3}}\right)$	☒	$\mathcal{O}\left(\frac{1}{L}(\frac{1}{n^{2/3}} \wedge \varepsilon^{4/3})\right)$	☑	☒
SNVRG (Zhou et al., 2018)	$\mathcal{O}\left((n + \frac{n^{1/2}}{\varepsilon^2})\log(n)\right)$	$\mathcal{O}\left((\frac{\sigma^2}{\varepsilon^2} + \frac{\sigma}{\varepsilon^3})\log(\frac{1}{\varepsilon})\right)$	☒	$\mathcal{O}\left(\frac{1}{L}\right)$	☒	☒
SPIDER (Fang et al., 2018)	$\mathcal{O}\left(n + \frac{n^{1/2}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2} + \frac{\sigma}{\varepsilon^3}\right)$	☒	$\mathcal{O}\left(\frac{\varepsilon}{L}\right)$	☑	☑
SpiderBoost (Wang et al., 2019)	$\mathcal{O}\left(n + \frac{n^{1/2}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2} + \frac{\sigma}{\varepsilon^3}\right)$	☑	$\mathcal{O}\left(\frac{1}{L}\right)$	☒	☒
ProxSARAH (This work)	$\mathcal{O}\left(n + \frac{n^{1/2}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2} + \frac{\sigma}{\varepsilon^3}\right)$	☑	$\mathcal{O}\left(\frac{1}{L\sqrt{m}}\right) \to \mathcal{O}\left(\frac{1}{L}\right)$	☑	☑

Table 1: Comparison on SFO (Stochastic First-order Oracle) complexity for nonsmooth nonconvex optimization (both non-composite and composite case). Here, "D-step-size" stands for "using dynamic step-sizes" and "Mb-range" means that the algorithm can obtain the best-known complexity bound with a large range of mini-batch sizes instead of specific values. In addition, m is the number of inner iterations (epoch length) and $\sigma > 0$ is the variance in Assumption 2.3. Note that all the complexity bounds here must depend on the Lipschitz constant L of the smooth components and $F(\widetilde{w}^0) - F^\star$, the difference between the initial objective value $F(\widetilde{w}^0)$ and the lower-bound $F^\star$. For the sake of presentation, we assume that $L = \mathcal{O}(1)$ and ignore these quantities in the complexity bounds.

1.4 Comparison Between Our Methods and Existing Work

Hitherto, we have found three different variance reduction algorithms of the stochastic proximal gradient method for nonconvex problems that are most related to our work: proximal SVRG (called ProxSVRG) in Reddi et al. (2016b), ProxSVRG+ in Li and Li (2018), and ProxSpiderBoost in Wang et al. (2019). Other methods such as proximal stochastic gradient descent (ProxSGD) scheme (Ghadimi et al., 2016), ProxSAGA in Reddi et al. (2016b), and Natasha variants in Allen-Zhu (2018) are quite different and already intensively compared in previous works (see Li and Li, 2018; Reddi et al., 2016b; Wang et al., 2019), and we do not include them here.

In terms of theory, Table 1 compares different methods for solving (1) and (2) regarding the stochastic first-order oracle calls (SFO), the applicability to finite-sum and/or expectation and composite settings, step-sizes, and the use of dynamic step-sizes and mini-batches.

Now, let us compare in detail our algorithms and four methods: ProxSVRG, ProxSVRG+, SPIDER, and ProxSpiderBoost for solving (1) and (2), or their special cases.

(a) ***Assumptions:*** In the finite-sum setting (2), ProxSVRG, ProxSVRG+, and ProxSpiderBoost all use the smoothness of each component f_i in (2), which is stronger than the average smoothness in Assumption 2.2 stated below. They did not consider (2) under Assumption 2.2. However, Assumption 2.2 is often stronger than the L-smoothness of f.

(b) ***Single sample for the finite-sum case:*** The performance of gradient descent-type algorithms crucially depends on the step-size (or learning rate). Let us make a comparison between different methods in terms of step-size for single sample case, and the corresponding complexity bound.

- As shown in Reddi et al. (2016b, Theorem 1), in the single sample case, i.e., the mini-batch size of the inner loop $\hat{b} = 1$, ProxSVRG for solving (2) has a small step-size $\eta = \frac{1}{3Ln}$, and its corresponding complexity is $\mathcal{O}\left(n\varepsilon^{-2}\right)$ (see Reddi et al., 2016b, Corollary 1), which is the same as in standard proximal gradient methods.
- ProxSVRG+ in Li and Li (2018, Theorem 3) is a variant of ProxSVRG, and in the single sample case, it uses a different step-size $\eta = \min\left\{\frac{1}{6L}, \frac{1}{6mL}\right\}$. This step-size is only better than that of ProxSVRG if $2m < n$. With this step-size, the complexity of ProxSVRG+ remains $\mathcal{O}\left(n^{2/3}\varepsilon^{-2}\right)$ as in ProxSVRG.
- In the non-composite case, SPIDER (Fang et al., 2018) relies on a dynamic step-size $\eta_t := \min\left\{\frac{\varepsilon}{L\|v_t\|\sqrt{n}}, \frac{1}{2L\sqrt{n}}\right\}$, where v_t is the SARAH stochastic estimator. Clearly, this step-size is very small if the target accuracy ε is small, and/or $\|v_t\|$ is large. However, SPIDER achieves $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ complexity bound, which is nearly optimal. Note that this step-size is problem-dependent since it depends on v_t. We also emphasize that SPIDER did not consider the composite problems.
- In our constant step-size ProxSARAH variants, we use two step-sizes: averaging step-size $\gamma = \frac{\sqrt{2}}{\sqrt{3m}L}$ and proximal-gradient step-size $\eta = \frac{2\sqrt{3m}}{4\sqrt{3m}+\sqrt{2}}$, and their product presents a combined step-size, which is $\hat{\eta} := \gamma\eta = \frac{2}{L(4\sqrt{3m}+\sqrt{2})}$ (see (25) for our definition of step-size). Clearly, our step-size $\hat{\eta}$ is much larger than that of both ProxSVRG and ProxSVRG+. It can be larger than that of SPIDER if ε is small and $\|v_t\|$ is large. With these step-sizes, our complexity bound is $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$, and if $\varepsilon \leq \mathcal{O}\left(n^{-1/4}\right)$, then it reduces to $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$, which is also nearly optimal.

- As we can observe from Algorithm 1 in the sequel, the number of proximal operator calls in our method remains the same as in ProxSVRG and ProxSVRG+.

(c) ***Mini-batch for the finite-sum case:*** Now, we consider the mini-batch case.

- As indicated in Reddi et al. (2016b, Theorem 2), if we choose the batch-size $\hat{b} = \lfloor n^{2/3} \rfloor$ and $m = \lfloor n^{1/3} \rfloor$, then the step-size η can be chosen as $\eta = \frac{1}{3L}$, and its complexity is improved up to $\mathcal{O}\left(n + n^{2/3}\varepsilon^{-2}\right)$ for ProxSVRG. However, the mini-batch size $n^{2/3}$ is close to the full data set n. Reddi et al. (2016b) do not consider a full range of $\hat{b}$.
- For ProxSVRG+ (Li and Li, 2018), based on Theorem 1, we need to set $\hat{b} = \lfloor n^{2/3} \rfloor$ and $m = \lfloor \sqrt{\hat{b}} \rfloor = \lfloor n^{1/3} \rfloor$ to obtain the best complexity bound for this method, which is $\mathcal{O}\left(n + n^{2/3}\varepsilon^{-2}\right)$. Nevertheless, its step-size is $\eta = \frac{1}{6L}$, which is twice smaller than that of ProxSVRG. In addition, ProxSVRG requires the bounded variance assumption for (2), which can be avoided in our methods by using full batch for snapshot points.
- For SPIDER, again in the non-composite setting, if we choose the batch-size $\hat{b} = \lfloor n^{1/2} \rfloor$, then its step-size is $\eta_t := \min\left\{\frac{\varepsilon}{L\|v_t\|}, \frac{1}{2L}\right\}$. In addition, SPIDER limits the batch-size $\hat{b}$ in the range of $[1, n^{1/2}]$, and did not consider larger mini-batch sizes.
- For SpiderBoost (Wang et al., 2019), it requires to properly set mini-batch size to achieve $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ complexity for solving (2). More precisely, from Wang et al. (2019, Theorem 1), we can see that one needs to set $m = \lfloor \sqrt{n} \rfloor$ and $\hat{b} = \lfloor \sqrt{n} \rfloor$ to achieve such a complexity. This mini-batch size can be large if n is large, and less flexible to adjust the performance of the algorithm. Unfortunately, ProxSpiderBoost does not have theoretical guarantee for the single sample case.
- In our methods, it is flexible to choose the epoch length m and the batch-size $\hat{b}$ such that we can obtain different step-sizes and complexity bounds. Our batch-size $\hat{b}$ can be any value in $[1, n-1]$ for (2). Given $\hat{b} \in [1, \sqrt{n}]$, we can properly choose $m = \mathcal{O}\left(n/\hat{b}\right)$ to obtain the best-known complexity bound $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ when $n > \mathcal{O}\left(\varepsilon^{-4}\right)$ and $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$, otherwise. More details can be found in Subsection 3.4.

(d) ***Online or expectation problems:*** For online or expectation problems, a mini-batch is required to evaluate snapshot gradient estimators for the outer loop.

- In the online or expectation case (1), SPIDER (Fang et al., 2018, Theorem 1) achieves an $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$ complexity. In the single sample case, SPIDER's step-size becomes $\eta_t := \min\left\{\frac{\varepsilon^2}{2\sigma L\|v_t\|}, \frac{\varepsilon}{4\sigma L}\right\}$, which can be very small, and depends on v_t and σ. Note that σ is often unknown or hard to estimate. Moreover, in early iterations, $\|v_t\|$ is often large potentially making this method slow.
- ProxSpiderBoost (Wang et al., 2019) achieves the same complexity bound as SPIDER for the composite problem (1), but requires to set the mini-batch for both outer and inner loops. The size of these mini-batches has to be fixed a priori in order to use a constant step-size, which is certainly less flexible. The total complexity of this method is $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$.
- As shown in Theorem 9, our complexity is $\mathcal{O}\left(\sigma\varepsilon^{-3}\right)$ given that $\sigma \leq \mathcal{O}\left(\varepsilon^{-1}\right)$. Otherwise, it is $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$, which is the same as in ProxSpiderBoost. Note that our complexity can be achieved for both single sample and a wide range of mini-batch sizes as opposed to a predefined mini-batch size of ProxSpiderBoost.

From an algorithmic point of view, our method, Algorithm 1, is fundamentally different from existing methods due to its averaging step and large step-sizes in the composite settings. Moreover, our methods have more chance to improve the performance due to the use of dynamic step-sizes and an additional damped step-size γ_t, and the flexibility to choose the epoch length m, the inner mini-batch size $\hat{b}$, and the snapshot batch-size b_s.

1.5 Paper Organization

The rest of this paper is organized as follows. Section 2 discusses the fundamental assumptions and optimality conditions. Section 3 presents the main algorithmic framework and its convergence results for two settings. Section 4 considers extensions and special cases of our algorithms. Section 5 provides some numerical examples to verify our methods and compare them with existing state-of-the-arts.

2. Mathematical Tools and Preliminary Results

Firstly, we recall some basic notation and concepts in optimization, which can be found in Bauschke and Combettes (2017); Nesterov (2004). Next, we state our blanket assumptions and discuss the optimality condition of (1) and (2). Finally, we provide preliminary results needed in the sequel.

2.1 Basic Notation and Concepts

We work with finite dimensional spaces, $\mathbb{R}^d$, equipped with standard inner product $\langle\cdot,\cdot\rangle$ and Euclidean norm $\|\cdot\|$. Given a function $f : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$, we use $\mathrm{dom}(f) := \big\{w \in \mathbb{R}^d \mid f(w) < +\infty\big\}$ to denote its (effective) domain. If f is proper, closed, and convex, $\partial f(w) := \big\{v \in \mathbb{R}^d \mid f(z) \geq f(w) + \langle v, z - w\rangle, \quad \forall z \in \mathrm{dom}(f)\big\}$ denotes its subdifferential at w, and $\mathrm{prox}_f(w) := \arg\min_z \big\{f(z) + (1/2)\|z - w\|^2\big\}$ denotes its proximal operator. Note that if f is the indicator of a nonempty, closed, and convex set $\mathcal{X}$, i.e., $f(w) = \delta_{\mathcal{X}}(w)$, then $\mathrm{prox}_f(\cdot) = \mathrm{proj}_{\mathcal{X}}(\cdot)$, the projection of w onto $\mathcal{X}$. Any element $\nabla f(w)$ of $\partial f(w)$ is called a subgradient of f at w. If f is differentiable at w, then $\partial f(w) = \{\nabla f(w)\}$, the gradient of f at w. A continuous differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L_f-smooth if ∇f is Lipschitz continuous on its domain, i.e., $\|\nabla f(w) - \nabla f(z)\| \leq L_f\|w - z\|$ for $w, z \in \mathrm{dom}(f)$. We use $\mathbf{U}_p(S)$ to denote a finite set $S := \{s_1, s_2, \cdots, s_n\}$ equipped with a probability distribution p over S. If p is uniform, then we simply use $\mathbf{U}(S)$. For any real number a, $\lfloor a\rfloor$ denotes the largest integer less than or equal to a. We use $[n]$ to denote the set $\{1, 2, \cdots, n\}$. Finally, for the sake of clarity, Table 2 provides some notations commonly used in this paper.

2.2 Fundamental Assumptions

To develop numerical methods for solving (1) and (2), we rely on some basic assumptions usually used in stochastic optimization methods.

Assumption 2.1 (Bounded from below) *Both problems* (1) *and* (2) *are bounded from below. That is* $F^\star := \inf_{w\in\mathbb{R}^d} F(w) > -\infty$. *Moreover,* $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(\psi) \neq \emptyset$.

Notation	Meaning	Type and range
ε	The target accuracy for stochastic gradient mapping	positive real
m	The epoch length (i.e., the number of iterations of the inner loop t)	positive integer
$\mathcal{B}_s$	The mini-batch of the snapshot point $\widetilde{w}_{s-1}$	finite set of realizations
b_s	The size of the mini-batch $\mathcal{B}_s$ of the snapshot point $\widetilde{w}_{s-1}$	positive integer
$\hat{\mathcal{B}}_t^{(s)}$	The mini-batch for evaluating SARAH estimator in the inner loop t	finite set of realizations
$\hat{b}_t^{(s)}$	The size of the mini-batch $\hat{\mathcal{B}}_t^{(s)}$	positive integer

Table 2: Common quantities used in this paper.

This assumption usually holds in practice since f often represents a loss function which is nonnegative or bounded from below. In addition, the regularizer ψ is also nonnegative or bounded from below, and its domain intersects $\mathrm{dom}(f)$.

Our next assumption is the smoothness of f with respect to the argument w.

Assumption 2.2 (L-average smoothness) *In the expectation setting* (1)*, for any realization of $\xi \in \Omega$, $\mathbf{f}(\cdot;\xi)$ is L-smooth (on average), i.e., $\mathbf{f}(\cdot;\xi)$ is continuously differentiable and its gradient $\nabla_w \mathbf{f}(\cdot;\xi)$ is Lipschitz continuous with the same Lipschitz constant $L \in (0,+\infty)$, i.e.:*

$$\mathbb{E}_\xi\left[\|\nabla_w \mathbf{f}(w;\xi) - \nabla_w \mathbf{f}(\hat{w};\xi)\|^2\right] \le L^2\|w-\hat{w}\|^2, \quad w,\hat{w}\in \mathrm{dom}(f). \tag{3}$$

In the finite-sum setting (2)*, the condition* (3) *reduces to*

$$\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(w) - \nabla f_i(\hat{w})\|^2 \le L^2\|w-\hat{w}\|^2, \quad w,\hat{w}\in \mathrm{dom}(f). \tag{4}$$

We can write (4) as $\mathbb{E}_i\left[\|\nabla f_i(w) - \nabla f_i(\hat{w})\|^2\right] \le L^2\|w-\hat{w}\|^2$. Note that (4) is weaker than assuming that each component f_i is L_i-smooth, i.e., $\|\nabla f_i(w) - \nabla f_i(\hat{w})\| \le L_i\|w-\hat{w}\|$ for all $w,\hat{w}\in\mathrm{dom}(f)$ and $i\in[n]$. Indeed, the individual L_i-smoothness implies (4) with $L^2 := \frac{1}{n}\sum_{i=1}^n L_i^2$. Conversely, if (4) holds, then $\|\nabla f_i(w) - \nabla f_i(\hat{w})\|^2 \le \sum_{i=1}\|\nabla f_i(w) - \nabla f_i(\hat{w})\|^2 \le nL^2\|w-\hat{w}\|^2$ for $i\in[n]$. Therefore, each component f_i is $\sqrt{n}L$-smooth, which is larger than (4) within a factor of $\sqrt{n}$ in the worst-case. We emphasize that ProxSVRG, ProxSVRG+, and ProxSpiderBoost all require the L-smoothness of each component f_i in (2). However, the condition (3) is stronger than the L-smoothness of the expected function f (i.e., $\|\nabla f(w) - \nabla f(\hat{w})\| \le L_f\|w-\hat{w}\|$ for $w,\hat{w}\in\mathrm{dom}(f)$) as used in standard SGD algorithms (Ghadimi and Lan, 2013).

It is well-known that the L-smooth condition leads to the following bound

$$\mathbb{E}_\xi\left[\mathbf{f}(\hat{w};\xi)\right] \le \mathbb{E}_\xi\left[\mathbf{f}(w;\xi)\right] + \mathbb{E}_\xi\left[\langle\nabla_w\mathbf{f}(w;\xi),\hat{w}-w\rangle\right] + \frac{L}{2}\|\hat{w}-w\|^2, \quad w,\hat{w}\in\mathrm{dom}(f). \tag{5}$$

Indeed, from (3), we have

$$\begin{aligned}\|\nabla f(w) - \nabla f(\hat{w})\|^2 &= \|\mathbb{E}_\xi\left[\nabla_w\mathbf{f}(w;\xi) - \nabla_w\mathbf{f}(\hat{w};\xi)\right]\|^2\\ &\le \mathbb{E}_\xi\left[\|\nabla_w\mathbf{f}(w;\xi) - \nabla_w\mathbf{f}(\hat{w};\xi)\|^2\right]\\ &\le L^2\|w-\hat{w}\|^2,\end{aligned}$$

which shows that $\|\nabla f(w) - \nabla f(\hat{w})\| \le L\|w - \hat{w}\|$. Hence, using either (3) or (4), we get

$$f(\hat{w}) \le f(w) + \langle \nabla f(w), \hat{w} - w\rangle + \frac{L}{2}\|\hat{w} - w\|^2, \quad w, \hat{w} \in \mathrm{dom}(f). \tag{6}$$

The L-smooth condition also leads to the *L-almost convexity* of f (see Zhou and Gu, 2019) since $f(\cdot) + \frac{L}{2}\|\cdot\|^2$ is convex.

In the expectation setting (1), we need the following bounded variance condition:

Assumption 2.3 (Bounded variance) *For the expectation problem* (1)*, there exists a uniform constant $\sigma \in (0, +\infty)$ such that*

$$\mathbb{E}_\xi\left[\|\nabla_w \mathbf{f}(w;\xi) - \nabla f(w)\|^2\right] \le \sigma^2, \quad \forall w \in \mathrm{dom}(f). \tag{7}$$

For the finite-sum problem (2)*, there exists a uniform constant $\sigma \in (0, +\infty)$ such that*

$$\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(w) - \nabla f(w)\|^2 \le \sigma^2, \quad \forall w \in \mathrm{dom}(f). \tag{8}$$

This assumption is standard in stochastic optimization and often required in almost any solution method for solving (1) (see Ghadimi and Lan, 2013). For problem (2), if n is extremely large, passing over n data points is exhaustive or impossible. We refer to this case as the online case mentioned in Fang et al. (2018), and can be cast into Assumption 2.3. Therefore, we do not consider this case separately. However, our theory and algorithms developed in this paper do apply to such a setting. In addition, for the finite-sum problem (2), if we define $\sigma_n^2(w) := \frac{1}{n}\sum_{i=1}^n \left[\|\nabla f_i(w)\|^2 - \|\nabla f(w)\|^2\right]$, then (8) becomes $\sigma_n^2(w) \le \sigma^2$ for all $w \in \mathrm{dom}(f)$, which is consistent to (7). We only use the condition (8) in Remark 7.

2.3 Optimality Conditions

Under Assumption 2.1, we have $\mathrm{dom}(f) \cap \mathrm{dom}(\psi) \neq \emptyset$. When $\mathbf{f}(\cdot;\xi)$ is nonconvex in w, the first order optimality condition of (1) can be stated as

$$0 \in \partial F(w^\star) \equiv \nabla f(w^\star) + \partial\psi(w^\star) \equiv \mathbb{E}_\xi\left[\nabla_w \mathbf{f}(w^\star;\xi)\right] + \partial\psi(w^\star). \tag{9}$$

Here, $w^\star$ is called a stationary point of F. We denote $\mathcal{S}^\star$ the set of all stationary points. The condition (9) is called the first-order optimality condition, and also holds for (2).

Since ψ is proper, closed, and convex, its proximal operator $\mathrm{prox}_{\eta\psi}$ satisfies the nonexpansiveness, i.e., $\|\mathrm{prox}_{\eta\psi}(w) - \mathrm{prox}_{\eta\psi}(z)\| \le \|w - z\|$ for all $w, z \in \mathbb{R}^d$.

Now, for any fixed $\eta > 0$, we define the following quantity

$$G_\eta(w) := \frac{1}{\eta}\Big(w - \mathrm{prox}_{\eta\psi}(w - \eta\nabla f(w))\Big). \tag{10}$$

This quantity is called the *gradient mapping* of F (Nesterov, 2004). Indeed, if $\psi \equiv 0$, then $G_\eta(w) \equiv \nabla f(w)$, which is exactly the gradient of f. By using $G_\eta(\cdot)$, the optimality condition (9) can be equivalently written as

$$\|G_\eta(w^\star)\|^2 = 0. \tag{11}$$

If we apply gradient-type methods to solve (1) or (2), then we can only aim at finding an ε-approximate stationary point $\widetilde{w}_T$ to $w^\star$ in (11) after at most T iterations within a given accuracy $\varepsilon > 0$, i.e.:

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \le \varepsilon^2. \tag{12}$$

The condition (12) is standard in stochastic nonconvex optimization methods. Stronger results such as approximate second-order optimality or strictly local minimum require additional assumptions and more sophisticated optimization methods such as cubic regularized Newton-type schemes (see Nesterov and Polyak, 2006).

2.4 Stochastic Gradient Estimators

One key step to design a stochastic gradient method for (1) or (2) is to query an estimator for the gradient $\nabla f(w)$ at any w. Let us recall some existing stochastic estimators.

2.4.1 Single sample estimators

A simple estimator of $\nabla f(w)$ can be computed as follows:

$$\widetilde{\nabla} f(w_t) := \nabla_w \mathbf{f}(w_t; \xi_t), \tag{13}$$

where ξ_t is a realization of ξ. This estimator is unbiased, i.e., $\mathbb{E}\left[\widetilde{\nabla} f(w_t) \mid \mathcal{F}_t\right] = \nabla f(w_t)$, but its variance is fixed for any w_t, where $\mathcal{F}_t$ is the history of randomness collected up to the t-th iteration, i.e.:

$$\mathcal{F}_t := \sigma\big(w_0, w_1, \cdots, w_t\big). \tag{14}$$

This is a σ-field generated by random variables $\{w_0, w_1, \cdots, w_t\}$. In the finite-sum setting (2), we have $\widetilde{\nabla} f(w_t) := \nabla f_{i_t}(w_t)$, where $i_t \sim \mathbf{U}([n])$ with $[n] := \{1, 2, \cdots, n\}$.

In recent years, there has been huge interest in designing stochastic estimators with variance reduction properties. The first variance reduction method was perhaps proposed in Schmidt et al. (2017) since 2013, and then in Defazio et al. (2014) for convex optimization. However, the most well-known method is SVRG introduced by Johnson and Zhang (2013) that works for both convex and nonconvex problems. The SVRG estimator for ∇f in (2) is given as

$$\widetilde{\nabla} f(w_t) := \nabla f(\widetilde{w}) + \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}), \tag{15}$$

where $\nabla f(\widetilde{w})$ is the full gradient of f at a snapshot point $\widetilde{w}$, and i_t is a uniformly random index in $[n]$. It is clear that $\mathbb{E}\left[\widetilde{\nabla} f(w_t) \mid \mathcal{F}_t\right] = \nabla f(w_t)$, which shows that $\widetilde{\nabla} f(w_t)$ is an unbiased estimator of $\nabla f(w_t)$. Moreover, its variance is reduced along the snapshots.

Our methods rely on the SARAH estimator introduced in Nguyen et al. (2017a) for the non-composite convex problem instances of (2). We instead consider it in a more general setting to cover both (2) and (1), which is defined as follows:

$$v_t := v_{t-1} + \nabla_w \mathbf{f}(w_t; \xi_t) - \nabla_w \mathbf{f}(w_{t-1}; \xi_t), \tag{16}$$

for a given realization ξ_t of ξ where v_0 is a snapshot gradient estimator whose definition is presented in Section 3.2 and 3.5. Each evaluation of v_t requires two gradient evaluations. Clearly, the SARAH estimator is biased, since $\mathbb{E}\left[v_t \mid \mathcal{F}_t\right] = v_{t-1} + \nabla f(w_t) - \nabla f(w_{t-1}) \neq \nabla f(w_t)$. However, it has a variance reduced property.

2.4.2 Mini-batch estimators

We consider a mini-batch estimator of the gradient ∇f in (13) and of the SARAH estimator (16) respectively as follows:

$$\begin{aligned} \widetilde{\nabla} f_{\mathcal{B}_t}(w_t) &:= \frac{1}{b_t}\sum_{\xi_i\in\mathcal{B}_t}\nabla_w \mathbf{f}(w_t;\xi_i), \\ \text{and} \quad v_t &:= v_{t-1} + \frac{1}{b_t}\sum_{\xi_i\in\mathcal{B}_t}\left(\nabla_w \mathbf{f}(w_t;\xi_i) - \nabla_w \mathbf{f}(w_{t-1};\xi_i)\right), \end{aligned} \tag{17}$$

where $\mathcal{B}_t$ is a mini-batch of the size $b_t := |\mathcal{B}_t| \geq 1$. For the finite-sum problem (2), we replace $\mathbf{f}(\cdot;\xi_i)$ by $f_i(\cdot)$. In this case, $\mathcal{B}_t$ is a uniformly random subset of $[n]$. Clearly, if $b_t = n$, then we take the full gradient ∇f as the exact estimator.

2.5 Basic Properties of Stochastic and SARAH Estimators

We recall some basic properties of the standard stochastic and SARAH estimators for (1) and (2). The following result was proved in Nguyen et al. (2017a).

Lemma 1 *Let $\{v_t\}_{t\geq 0}$ be defined by (16) and $\mathcal{F}_t$ be defined by (14). Then*

$$\begin{aligned} \mathbb{E}\left[v_t \mid \mathcal{F}_t\right] &= \nabla f(w_t) + \epsilon_t \quad \neq \quad \nabla f(w_t), \quad \textit{where} \quad \epsilon_t := v_{t-1} - \nabla f(w_{t-1}). \\ \mathbb{E}\left[\|v_t - \nabla f(w_t)\|^2 \mid \mathcal{F}_t\right] &= \|v_{t-1} - \nabla f(w_{t-1})\|^2 + \mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] \\ &\quad - \|\nabla f(w_t) - \nabla f(w_{t-1})\|^2. \end{aligned} \tag{18}$$

Consequently, for any $t \geq 0$, we have

$$\begin{aligned} \mathbb{E}\left[\|v_t - \nabla f(w_t)\|^2\right] &= \mathbb{E}\left[\|v_0 - \nabla f(w_0)\|^2\right] + \textstyle\sum_{j=1}^t \mathbb{E}\left[\|v_j - v_{j-1}\|^2\right] \\ &\quad - \textstyle\sum_{j=1}^t \mathbb{E}\left[\|\nabla f(w_j) - \nabla f(w_{j-1})\|^2\right]. \end{aligned} \tag{19}$$

Our next result is some properties of the mini-batch estimators in (17). Most of the proof have been presented in Harikandeh et al. (2015); Lohr (2009); Nguyen et al. (2017b, 2018a), and we only provide the missing proof of (23) and (24) in Appendix A.

Lemma 2 *If $\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)$ is generated by (17), then, under Assumption 2.3, we have*

$$\begin{aligned} &\mathbb{E}\left[\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) \mid \mathcal{F}_t\right] = \nabla f(w_t) \\ \textit{and} \quad &\mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) - \nabla f(w_t)\|^2 \mid \mathcal{F}_t\right] = \frac{1}{b_t}\mathbb{E}\left[\|\nabla_w \mathbf{f}(w_t;\xi) - \nabla f(w_t)\|^2 \mid \mathcal{F}_t\right] \leq \frac{\sigma^2}{b_t}. \end{aligned} \tag{20}$$

If $\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)$ is generated by (17) for the finite-sum problem (2), then

$$\begin{aligned} &\mathbb{E}\left[\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) \mid \mathcal{F}_t\right] = \nabla f(w_t) \\ \textit{and} \quad &\mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) - \nabla f(w_t)\|^2 \mid \mathcal{F}_t\right] \leq \tfrac{1}{b_t}\left(\tfrac{n-b_t}{n-1}\right)\sigma_n^2(w_t), \end{aligned} \tag{21}$$

where $\sigma_n^2(w)$ *is defined as*

$$\sigma_n^2(w) := \frac{1}{n}\sum_{i=1}^{n}\left[\|\nabla f_i(w)\|^2 - \|\nabla f(w)\|^2\right]. \tag{22}$$

If v_t *is generated by* (17) *for the finite-sum problem* (2)*, then*

$$\begin{aligned}\mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] &= \tfrac{n(b_t-1)}{b_t(n-1)}\|\nabla f(w_t) - \nabla f(w_{t-1})\|^2 \\ &\quad + \tfrac{(n-b_t)}{b_t(n-1)}\cdot\tfrac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(w_t) - \nabla f_i(w_{t-1})\|^2.\end{aligned} \tag{23}$$

If v_t *is generated by* (17) *for the expectation problem* (1)*, then*

$$\begin{aligned}\mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] &= \left(1 - \tfrac{1}{b_t}\right)\|\nabla f(w_t) - \nabla f(w_{t-1})\|^2 \\ &\quad + \tfrac{1}{b_t}\mathbb{E}\left[\|\nabla_w \mathbf{f}(w_t;\xi) - \nabla_w \mathbf{f}(w_{t-1};\xi)\|^2 \mid \mathcal{F}_t\right].\end{aligned} \tag{24}$$

Note that if $b_t = n$, i.e., we take a full gradient estimate, then the second estimate of (21) is vanished and independent of $\sigma_n(\cdot)$. The second term of (23) is also vanished.

3. ProxSARAH Framework and Convergence Analysis

We describe our unified algorithmic framework and then specify it to solve different instances of (1) and (2) under appropriate structures. The general algorithm is described in Algorithm 1, which is abbreviated by ProxSARAH.

Algorithm 1 (Proximal SARAH with stochastic recursive gradient estimators)

1: **Initialization:** An initial point $\widetilde{w}_0$ and necessary parameters $\eta_t > 0$ and $\gamma_t \in (0, 1]$ (will be specified in the sequel).
2: **Outer Loop: For** $s := 1, 2, \cdots, S$ **do**
3: Generate a snapshot $v_0^{(s)}$ at $w_0^{(s)} := \widetilde{w}_{s-1}$ using (37) for (1) and (29) for (2).
4: Update $\widehat{w}_1^{(s)} := \mathrm{prox}_{\eta_0\psi}(w_0^{(s)} - \eta_0 v_0^{(s)})$ and $w_1^{(s)} := (1-\gamma_0)w_0^{(s)} + \gamma_0\widehat{w}_1^{(0)}$.
5: **Inner Loop: For** $t := 1, \cdots, m$ **do**
6: Generate a proper single random sample or mini-batch $\hat{\mathcal{B}}_t^{(s)}$.
7: Evaluate $v_t^{(s)} := v_{t-1}^{(s)} + \frac{1}{|\hat{\mathcal{B}}_t^{(s)}|}\sum_{\xi_t^{(s)}\in\hat{\mathcal{B}}_t^{(s)}}\left[\nabla_w \mathbf{f}(w_t^{(s)};\xi_t^{(s)}) - \nabla_w \mathbf{f}(w_{t-1}^{(s)};\xi_t^{(s)})\right]$.
8: Update $\widehat{w}_{t+1}^{(s)} := \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})$ and $w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t\widehat{w}_{t+1}^{(s)}$.
9: **End For**
10: Set $\widetilde{w}_s := w_{m+1}^{(s)}$
11: **End For**

In terms of algorithm, ProxSARAH is different from SARAH where it has one proximal step followed by an additional averaging step, Step 8. However, using an approximation $\widetilde{G}_\eta$ of the gradient mapping G_η defined by (10), we can view Step 8 as:

$$w_{t+1}^{(s)} := w_t^{(s)} - \eta_t\gamma_t\widetilde{G}_{\eta_t}(w_t^{(s)}), \tag{25}$$

where $\widetilde{G}_{\eta_t}(w_t^{(s)}) := \frac{1}{\eta_t}\big(w_t^{(s)} - \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})\big)$ can be considered as an approximation of $G_{\eta_t}(w_t^{(s)})$ and $\hat{\eta}_t := \eta_t\gamma_t$ can be viewed as a combined step-size. Hence, the update (25) is similar to the gradient step applying to the approximate gradient mapping $\widetilde{G}_{\eta_t}(w_t^{(s)})$ of F. In particular, if we set $\gamma_t = 1$, then we obtain a vanilla proximal SARAH variant which is similar to ProxSVRG, ProxSVRG+, and ProxSpiderBoost discussed above. ProxSVRG, ProxSVRG+, and ProxSpiderBoost are simply vanilla proximal gradient-type methods in stochastic setttings. If $\psi = 0$, then $\widetilde{G}_{\eta_t}(w_t^{(s)}) \equiv v_t^{(s)}$ and ProxSARAH is reduced to SARAH in Nguyen et al. (2017a,b, 2018b) with a step-size $\hat{\eta}_t := \gamma_t\eta_t$. Note that Step 8 can be represented as a weighted averaging step with given weights $\{\tau_j^{(s)}\}_{j=0}^m$:

$$w_{t+1}^{(s)} := \frac{1}{\Sigma_t^{(s)}}\sum_{j=0}^{t}\tau_j^{(s)}\widehat{w}_{j+1}^{(s)}, \quad \text{where } \Sigma_t^{(s)} := \sum_{j=0}^{t}\tau_j^{(s)} \text{ and } \gamma_j^{(s)} := \frac{\tau_j^{(s)}}{\Sigma_t^{(s)}}.$$

Compared to Ghadimi and Lan (2012); Nemirovski et al. (2009), ProxSARAH evaluates v_t at the averaged point $w_t^{(s)}$ instead of $\widehat{w}_t^{(s)}$. Therefore, it can be written as

$$w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)}),$$

which is similar to averaged fixed-point schemes (e.g., the Krasnosel'skii—Mann scheme) in the literature (see Bauschke and Combettes, 2017).

In addition, we will show in our analysis a key difference in terms of step-sizes η_t and γ_t, mini-batch, and epoch length between ProxSARAH and existing methods, including SPIDER (Fang et al., 2018) and SpiderBoost (Wang et al., 2019).

3.1 Analysis of The Inner-Loop: Key Estimates

This subsection proves two key estimates of the inner loop for $t = 1$ to m. We break our analysis into two different lemmas, which provide key estimates for our convergence analysis. We assume that the mini-batch size $\hat{b} := |\hat{\mathcal{B}}_t^{(s)}|$ in the inner loop is fixed.

Lemma 3 *Let $\{(w_t, \widehat{w}_t)\}$ be generated by the inner-loop of Algorithm 1 with $|\hat{\mathcal{B}}_t^{(s)}| = \hat{b} \in [n-1]$ fixed. Then, under Assumption 2.2, we have*

$$\begin{aligned}\mathbb{E}\Big[F(w_{m+1}^{(s)})\Big] \;\le\; & \mathbb{E}\Big[F(w_0^{(s)})\Big] + \frac{\rho L^2}{2}\sum_{t=0}^{m}\gamma_t\big(1+2\eta_t^2\big)\sum_{j=1}^{t}\gamma_{j-1}^2\mathbb{E}\Big[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\Big] \\ & - \frac{1}{2}\sum_{t=0}^{m}\gamma_t\left(\frac{2}{\eta_t} - L\gamma_t - 3\right)\mathbb{E}\Big[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\Big] \\ & + \frac{1}{2}\bar{\sigma}^{(s)}\Big(\sum_{t=0}^{m}\beta_t\Big) - \sum_{t=0}^{m}\frac{\gamma_t\eta_t^2}{2}\mathbb{E}\Big[\|G_{\eta_t}(w_t^{(s)})\|^2\Big],\end{aligned} \tag{26}$$

where $\bar{\sigma}^{(s)} := \mathbb{E}\Big[\|v_0^{(s)} - \nabla f(w_0^{(s)})\|^2\Big] \ge 0$, $\rho := \frac{1}{\hat{b}}$ if Algorithm 1 solves (1), and $\rho := \frac{(n-\hat{b})}{\hat{b}(n-1)}$ if Algorithm 1 solves (2).

The proof of Lemma 3 is deferred to Appendix B.1. The next lemma shows how to choose constant step-sizes γ and η by fixing other parameters in Lemma 3 to obtain a descent property. The proof of this lemma is given in Appendix B.2.

Lemma 4 *Under Assumption 2.2 and $\hat{b} := |\hat{\mathcal{B}}_t^{(s)}| \in [n-1]$, let us choose $\eta_t = \eta > 0$ and $\gamma_t = \gamma > 0$ in Algorithm 1 such that*

$$\gamma_t = \gamma := \frac{1}{L\sqrt{\omega m}} \quad \text{and} \quad \eta_t = \eta := \frac{2\sqrt{\omega m}}{4\sqrt{\omega m}+1}, \tag{27}$$

where $\omega := \frac{3}{2\hat{b}}$ if Algorithm 1 solves (1) and $\omega := \frac{3(n-\hat{b})}{2\hat{b}(n-1)}$ if Algorithm 1 solves (2). Then

$$\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \le \mathbb{E}\left[F(w_0^{(s)})\right] - \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] + \frac{\gamma\theta}{2}(m+1)\bar{\sigma}^{(s)}, \tag{28}$$

where $\theta := 1 + 2\eta^2 \le \frac{3}{2}$.

Remark 5 *As mentioned in (25), the main update at Step 8 of Algorithm 1 can be written as $w_{t+1}^{(s)} := w_t^{(s)} - \eta_t\gamma_t\widetilde{G}_{\eta_t}(w_t^{(s)})$, where $\hat{\eta}_t := \eta_t\gamma_t$ can be viewed as a combined step-size. Using (27), we have $\hat{\eta}_t = \frac{2}{L(4\sqrt{\omega m}+1)} = \mathcal{O}\left(\frac{1}{L}\right)$. This step-size is proportional to $\frac{1}{L}$ as commonly seen in gradient-based methods (Nesterov, 2004).*

3.2 Convergence Analysis for The Composite Finite-Sum Problem (2)

In this subsection, we specify Algorithm 1 to solve the composite finite-sum problem (2). We replace $v_0^{(s)}$ at **Step 3** and $v_t^{(s)}$ at **Step 7** of Algorithm 1 by the following ones:

$$v_0^{(s)} := \frac{1}{b_s}\sum_{j\in\mathcal{B}_s}\nabla f_j(w_0^{(s)}), \quad \text{and} \quad v_t^{(s)} := v_{t-1}^{(s)} + \frac{1}{\hat{b}_t^{(s)}}\sum_{i\in\hat{\mathcal{B}}_t^{(s)}}\Big(\nabla f_i(w_t^{(s)}) - \nabla f_i(w_{t-1}^{(s)})\Big), \tag{29}$$

where $\mathcal{B}_s$ is an outer mini-batch of a fixed size $b_s := |\mathcal{B}_s| = b$, and $\hat{\mathcal{B}}_t^{(s)}$ is an inner mini-batch of a fixed size $\hat{b}_t^{(s)} := |\hat{\mathcal{B}}_t^{(s)}| = \hat{b}$. Moreover, $\mathcal{B}_s$ is independent of $\mathcal{B}_t^{(s)}$.

We consider two separate cases of this algorithmic variant: dynamic[1] step-sizes and constant step-sizes, but with fixed inner mini-batch size $\hat{b} \in [n-1]$. The following theorem proves the convergence of the dynamic step-size variant, whose proof is in Appendix B.3.

Theorem 6 *Assume that we apply Algorithm 1 to solve (2), where the estimators $v_0^{(s)}$ and $v_t^{(s)}$ are defined by (29) such that $b_s = b \in [n]$ and $\hat{b}_t^{(s)} = \hat{b} \in [n-1]$, respectively. Let $\eta_t := \eta \in (0, \frac{2}{3})$ be fixed, $\omega_\eta := \frac{(1+2\eta^2)(n-\hat{b})}{\hat{b}(n-1)}$, and $\delta := \frac{2}{\eta} - 3 > 0$. Let $\{\gamma_t\}_{t=0}^m$ be the sequence of step-sizes updated in a backward mode as*

$$\gamma_m := \frac{\delta}{L}, \quad \text{and} \quad \gamma_t := \frac{\delta}{L\big[\eta + \omega_\eta L\sum_{j=t+1}^m\gamma_j\big]}, \quad t = 0, \cdots, m-1, \tag{30}$$

Then, the following statements hold:

1. We call γ_t defined by (30) a dynamic step-size since γ_t is computed based on its previously computed candidates $\gamma_{t+1}, \gamma_{t+2}, \cdots, \gamma_m$.

(a) *The sequence of step-sizes* $\{\gamma_t\}_{t=0}^m$ *satisfies*

$$\begin{aligned} \frac{\delta}{L(1+\delta\omega_\eta m)} &\leq \gamma_0 < \gamma_1 < \cdots < \gamma_m, \\ \text{and} \quad \Sigma_m := \sum_{t=0}^{m}\gamma_t &\geq \tfrac{2\delta(m+1)}{L(\sqrt{2\delta\omega_\eta m+1}+1)}. \end{aligned} \tag{31}$$

(b) *Under Assumptions 2.1 and 2.2, and* $\sigma_n^2(w)$ *defined by* (22) ($\sigma_n^2(w)$ *can be unbounded), the following bound holds:*

$$\begin{aligned} \frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m}\gamma_t\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] &\leq \frac{2}{\eta^2 S\Sigma_m}\left[F(\widetilde{w}_0)-F^\star\right] \\ &\quad + \frac{3}{2\eta^2 S}\sum_{s=1}^{S}\frac{(n-b_s)\sigma_n^2(\widetilde{w}_{s-1})}{nb_s}. \end{aligned} \tag{32}$$

(c) *Under Assumptions 2.1 and 2.2, if we choose* $\eta := \frac{1}{2}$, $m := \left\lfloor \frac{n}{\hat{b}} \right\rfloor$, $b_s := n$, *and* $\hat{b} \in [1, \sqrt{n}]$, *then for* $\widetilde{w}_T \sim \mathbf{U}_p\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ *such that*

$$\mathbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t} := \frac{\gamma_t}{S\Sigma_m},$$

we have

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \frac{4\sqrt{6}L\left[F(\widetilde{w}_0)-F^\star\right]}{S\sqrt{n}}. \tag{33}$$

Consequently, the number of outer iterations S *needed to obtain an output* $\widetilde{w}_T$ *of Algorithm 1 such that* $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ *is at most* $S := \frac{4\sqrt{6}L[F(\widetilde{w}_0)-F^\star]}{\sqrt{n}\varepsilon^2}$. *Moreover, if* $1 \leq n \leq \frac{96L^2[F(\widetilde{w}_0)-F^\star]^2}{\varepsilon^4}$, *then* $S \geq 1$.
The number of individual stochastic gradient evaluations ∇f_i *does not exceed*

$$\mathcal{T}_{\text{grad}} := \frac{20\sqrt{6}L\sqrt{n}\left[F(\widetilde{w}_0)-F^\star\right]}{\varepsilon^2} = \mathcal{O}\left(\frac{L\sqrt{n}}{\varepsilon^2}\left[F(\widetilde{w}_0)-F^\star\right]\right).$$

The number of $\text{prox}_{\eta\psi}$ *operations does not exceed* $\mathcal{T}_{\text{prox}} := \frac{4\sqrt{6}(\sqrt{n}+1)L[F(\widetilde{w}_0)-F^\star]}{\hat{b}\varepsilon^2}$.

Remark 7 *When* n *is sufficiently large, if we choose* $b_s < n$, *then to guarantee convergence of Algorithm 1 for solving* (2), *we need to impose Assumption 2.3 and choose* $b_s := \mathcal{O}\left(n \wedge \varepsilon^{-2}\right)$. *Then we can derive similar conclusions as in Theorem 6*(c).

Alternatively, Theorem 8 below shows the convergence of Algorithm 1 for the constant step-size case, whose proof is given in Appendix B.4.

Theorem 8 *Assume that we apply Algorithm 1 to solve* (2), *where the estimators* $v_0^{(s)}$ *and* $v_t^{(s)}$ *are defined by* (29) *such that* $b_s = b \in [n]$ *and* $\hat{b}_t^{(s)} = \hat{b} \in [n-1]$.

Let us choose constant step-sizes $\gamma_t = \gamma$ *and* $\eta_t = \eta$ *as*

$$\gamma := \frac{1}{L\sqrt{\omega m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\omega m}}{4\sqrt{\omega m}+1}, \quad \text{where} \quad \omega := \frac{3(n-\hat{b})}{2\hat{b}(n-1)} \quad \text{and} \quad \hat{b} \in [1, \sqrt{n}]. \tag{34}$$

Then, under Assumptions 2.1 and 2.2, if we choose $m := \lfloor \frac{n}{\hat{b}} \rfloor$, $b_s := n$, *and* $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$, *then the number of outer iterations* S *to achieve* $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \le \varepsilon^2$ *does not exceed*

$$S := \frac{16\sqrt{3}L}{\sqrt{2n}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big].$$

Moreover, if $n \le \frac{384L^2}{\varepsilon^4}\big[F(\widetilde{w}_0) - F^\star\big]^2$, *then* $S \ge 1$.

Consequently, the number of stochastic gradient evaluations $\mathcal{T}_{\rm grad}$ *does not exceed*

$$\mathcal{T}_{\rm grad} := \frac{16\sqrt{3}L\sqrt{n}}{\sqrt{2}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big] = \mathcal{O}\left(\frac{L\sqrt{n}}{\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big]\right).$$

The number of $\mathrm{prox}_{\eta\psi}$ *operations does not exceed* $\mathcal{T}_{\rm prox} := \frac{16\sqrt{3}L(\sqrt{n}+1)}{\hat{b}\sqrt{2}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big]$.

Note that the condition $n \le \mathcal{O}\left(\varepsilon^{-4}\right)$ is to guarantee that $S \ge 1$ in Theorems 6 and 8. In this case, our complexity bound is $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$. Otherwise, when $n > \mathcal{O}\left(\varepsilon^{-4}\right)$, then our complexity becomes $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ due to the full gradient snapshots. In the non-composite setting, this complexity is the same as SPIDER (Fang et al., 2018), and the range of our mini-batch size $\hat{b} \in [1, \sqrt{n}]$, which is the same as in SPIDER, instead of fixed $\hat{b} = \lfloor\sqrt{n}\rfloor$ as in SpiderBoost (Wang et al., 2019). We can extend our mini-batch size $\hat{b}$ such that $\sqrt{n} < \hat{b} \le n-1$, but our complexity bound is no longer the best-known one.

The step-size η in (34) can be bounded by $\eta \in [\frac{2}{5}, \frac{1}{2}]$ for any batch-size $\hat{b}$ and m instead of fixing at $\eta = \frac{1}{2}$. Nevertheless, this interval can be enlarged by slightly modifying the proof of Lemma 3. For example, we can show that η can go up to $\frac{2}{3}$ by appropriately manipulating the parameters in the proof of Lemma 3. The step-size $\gamma \in (0, 1]$ can change from a small to a large value close to 1 as the batch-size $\hat{b}$ and the epoch length m change as we will discuss in Subsection 3.4.

3.3 Lower-Bound Complexity for The Finite-Sum Problem (2)

Let us analyze a special case of (2) with $\psi = 0$. We consider any stochastic first-order methods to generate an iterate sequence $\{w_t\}$ as follows:

$$[w_t, i_t] := \mathcal{A}^{t-1}\left(\omega, \nabla f_{i_0}(w^0), \nabla f_{i_1}(w^1), \cdots, \nabla f_{i_{t-1}}(w^{t-1})\right), \quad t \ge 1, \tag{35}$$

where $\mathcal{A}^{t-1}$ are measure mapping into $\mathbb{R}^{d+1}$, f_{i_t} is an individual function chosen by $\mathcal{A}^{t-1}$ at iteration t, $\omega \sim \mathbf{U}([0,1])$ is a random vector, and $[w^0, i_0] := \mathcal{A}^0(\omega)$. Clearly, Algorithm 1 can be cast as a special case of (35). As shown in Fang et al. (2018, Theorem 3) and later in Zhou and Gu (2019, Theorem 4.5.), under Assumptions 2.1 and 2.2, for any $L > 0$ and $2 \le n \le \mathcal{O}\left(L^2\left[F(w^0) - F^\star\right]^2\varepsilon^{-4}\right)$, there exists a dimension $d = \widetilde{\mathcal{O}}(L^2\left[F(w^0) - F^\star\right]^2 n^2\varepsilon^{-4})$ such that the lower-bound complexity of Algorithm 1 to produce an output $\widetilde{w}_T$ such that $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \le \varepsilon^2$ is $\Omega\left(\frac{L\left[F(w^0)-F^\star\right]\sqrt{n}}{\varepsilon^2}\right)$. This lower-bound clearly matches the upper bound $\mathcal{T}_{\rm grad}$ in Theorems 6 and 8 up to a given constant factor.

3.4 Mini-Batch Size and Learning Rate Trade-offs

Although our step-size defined by (34) in the single sample case is much larger than that of ProxSVRG (Reddi et al., 2016b, Theorem 1), it still depends on $\sqrt{m}$, where m is the epoch length. To obtain larger step-sizes, we can choose m and the mini-batch size $\hat{b}$ using the same trick as in Reddi et al. (2016b, Theorem 2). Let us first fix $\gamma := \bar{\gamma} \in (0, 1]$. From (34), we have $\omega m = \frac{1}{L^2\bar{\gamma}^2}$. It makes sense to choose $\bar{\gamma}$ close to 1 in order to use new information from $\widehat{w}_{t+1}^{(s)}$ instead of the old one in $w_t^{(s)}$.

Our goal is to choose m and $\hat{b}$ such that $\omega m = \frac{3(n-\hat{b})m}{2\hat{b}(n-1)} = \frac{1}{L^2\bar{\gamma}^2}$. If we define $C := \frac{2}{3L^2\bar{\gamma}^2}$, then the last condition implies that $\hat{b} := \frac{mn}{Cn+m-C} \leq \frac{m}{C}$ provided that $m \geq C$. Our suggestion is to choose

$$\gamma := \bar{\gamma} \in (0, 1], \quad \hat{b} := \Big\lfloor \frac{mn}{Cn + m - C} \Big\rfloor, \quad \text{and} \quad \eta := \frac{2}{4 + L\bar{\gamma}}. \tag{36}$$

If we choose $m = \lfloor n^{1/3} \rfloor$, then $\hat{b} = \mathcal{O}\left(n^{1/3}\right) \leq \frac{n^{1/3}}{C}$. This mini-batch size is much smaller than $\lfloor n^{2/3} \rfloor$ in ProxSVRG. Note that, in ProxSVRG, they set $\gamma := 1$ and $\eta := \frac{1}{3L}$.

In ProxSpiderBoost (Wang et al., 2019), m and the mini-batch size $\hat{b}$ were chosen as $m = \hat{b} = \lfloor n^{1/2} \rfloor$ so that they can use constant step-sizes $\gamma = 1$ and $\eta = \frac{1}{2L}$. In our case, if $\gamma = 1$, then $\eta = \frac{2}{4+L}$. Hence, if $L = 1$, then $\eta_{\text{ProxSpiderBoost}} = \frac{1}{2} > \eta_{\text{ProxSARAH}} = \frac{2}{5} > \eta_{\text{ProxSVRG}} = \frac{1}{3}$. But if $L > 4$, then our step-size $\eta_{\text{ProxSARAH}}$ dominates $\eta_{\text{ProxSpiderBoost}}$. However, by manipulating some parameters in the proof of Lemma 3, we can obtain $\eta_{\text{ProxSARAH}} = \frac{2}{3}$, which shows that $\eta_{\text{ProxSARAH}} > \eta_{\text{ProxSpiderBoost}} = \frac{1}{2}$ when $L = 1$.

If we choose $m = \mathcal{O}\left(n^{1/2}\right)$ and $\hat{b} = \mathcal{O}\left(n^{1/2}\right)$, then we maintain the same complexity bound $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$ as in Theorems 6 and 8. Nevertheless, if we choose $m = \mathcal{O}\left(n^{1/3}\right)$ and $\hat{b} = \mathcal{O}\left(n^{1/3}\right)$, then the complexity bound becomes $\mathcal{O}\left((n^{2/3} + n^{1/3})\varepsilon^{-2}\right)$, which is similar to ProxSVRG. The choice of m in Theorem 6 affects the values of $\{\gamma_t\}_{t=0}^m$. Hence, a reasonably small value of m is recommended in the dynamic step-size case.

3.5 Convergence Analysis for The Composite Expectation Problem (1)

In this subsection, we apply Algorithm 1 to solve the general expectation setting (1). In this case, we generate the snapshot at **Step 3** of Algorithm 1 as follows:

$$v_0^{(s)} := \frac{1}{b_s} \sum_{\zeta_i^{(s)} \in \mathcal{B}_s} \nabla_w \mathbf{f}(w_0^{(s)}; \zeta_i^{(s)}), \tag{37}$$

where $\mathcal{B}_s := \left\{\zeta_1^{(s)}, \cdots, \zeta_{b_s}^{(s)}\right\}$ is a mini-batch of i.i.d. realizations of ξ at the s-th outer iteration and independent of ξ_t from the inner loop, and $b_s := |\mathcal{B}_s| = b \geq 1$ is fixed.

Now, we analyze the convergence of Algorithm 1 for solving (1) using (37) above. For simplicity of discussion, we only consider the constant step-size case. The dynamic step-size variant can be derived similarly as in Theorem 6 and we omit the details. The proof of the following theorem can be found in Appendix B.5.

Theorem 9 *Let us apply Algorithm 1 to solve* (1) *using* (37) *for* $v_0^{(s)}$ *at* ***Step 3*** *of Algorithm 1 with fixed outer loop batch-size* $b_s = b \geq 1$ *and inner loop batch-size* $\hat{b} := |\mathcal{B}_t^{(s)}| \geq 1$.

If we choose fixed step-sizes γ *and* η *as*

$$\gamma := \frac{1}{L\sqrt{\bar{\omega}m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\bar{\omega}m}}{4\sqrt{\bar{\omega}m}+1}, \quad \text{with } \bar{\omega} := \frac{3}{2\hat{b}}, \tag{38}$$

then, under Assumptions 2.1, 2.2, and 2.3, we have the following estimate:

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^\star\big] + \frac{3\sigma^2}{2\eta^2 b}. \tag{39}$$

In particular, if we choose $b := \left\lfloor \frac{75\sigma^2}{\varepsilon^2} \right\rfloor$ *and* $m := \left\lfloor \frac{\sigma^2}{\hat{b}\varepsilon^2} \right\rfloor$ *for* $\hat{b} \leq \frac{\sigma^2}{\varepsilon^2}$, *then after at most*

$$S := \frac{32L[F(\widetilde{w}_0) - F^\star]}{\sigma\varepsilon}$$

outer iterations, we obtain $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, *where* $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$.

Consequently, the number of individual stochastic gradient evaluations $\nabla_w f(w_t^{(s)};\xi_t)$ *and the number of proximal operations* $\mathrm{prox}_{\eta\psi}$, *respectively do not exceed:*

$$\mathcal{T}_{\mathrm{grad}} := \frac{2464\sigma L[F(\widetilde{w}_0) - F^\star]}{\varepsilon^3}, \quad \text{and} \quad \mathcal{T}_{\mathrm{prox}} := \frac{32\sigma L[F(\widetilde{w}_0) - F^\star]}{\hat{b}\varepsilon^2}.$$

If $\sigma = 0$, i.e., no stochasticity involved in our problem (1), then (39) reduces to

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^\star\big],$$

where the expectation is taken over all the randomness generated by the algorithm. From this bound, we can derive the well-known $\mathcal{O}\left(\varepsilon^{-2}\right)$ oracle complexity bound for gradient-based methods in the deterministic case as often seen in the literature.

If $\sigma > 0$, then Theorem 9 achieves the best-known complexity $\mathcal{O}\left(\sigma L\varepsilon^{-3}\right)$ for the composite expectation problem (1) as long as $\sigma \leq \frac{32L[F(\widetilde{w}_0)-F^\star]}{\varepsilon^2}$. Otherwise, our complexity is $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$ due to the snapshot gradient for evaluating $v_0^{(s)}$. This complexity is the same as SPIDER (Fang et al., 2018) in the non-composite setting and ProxSpiderBoost (Wang et al., 2019) in the mini-batch setting. It also matches the lower bound complexity recently studied in Arjevani et al. (2019) up to a constant under the same set of assumptions, but only for the non-composite setting of (1). Hence, our complexity is nearly optimal. Note that our method does not require to perform mini-batch in the inner loop, i.e., it is independent of $\hat{\mathcal{B}}_t^{(s)}$, and the mini-batch is independent of the number of iterations m of the inner loop, while in (Wang et al., 2019), the mini-batch size $|\hat{\mathcal{B}}_t^{(s)}|$ must be proportional to $\sqrt{|\mathcal{B}_s|} = \mathcal{O}\left(\varepsilon^{-1}\right)$, where $\mathcal{B}_s$ is the mini-batch of the outer loop. This is perhaps the reason why ProxSpiderBoost can take a large constant step-size $\eta = \frac{1}{2L}$ as discussed in Subsection 3.4.

Remark 10 *We have not attempted to optimize the constants in the complexity bounds of all theorems above, Theorems 6, 8, and 9. Our analysis can be refined to possibly obtain smaller constants in these complexity bounds by manipulating different parameters.*

4. Dynamic Step-size Variants for Non-Composite Problems

In this section, we consider the non-composite settings of (1) and (2) as special cases of Algorithm 1. Note that if we solely apply Algorithm 1 with constant step-sizes to solve the non-composite case of (1) and (2) when $\psi \equiv 0$, then by using the same step-size as in Theorems 6, 8, and 9, we can obtain the same complexity as stated in Theorems 6, 8, and 9, respectively. However, we will modify our proof of Theorem 6 to take advantage of the extra term $\sum_{t=0}^m \frac{\gamma_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)} - (\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\|^2\right]$ in the proof of Lemma 3. The proof of this theorem is given in Appendix C.

Theorem 11 *Let $\{w_t^{(s)}\}$ be generated by a variant of Algorithm 1 to solve the non-composite instance of (1) or (2) using the following update for both Step 4 and Step 8:*

$$w_{t+1}^{(s)} := w_t^{(s)} - \hat{\eta}_t v_t^{(s)}. \tag{40}$$

Let $\rho := \frac{1}{\hat{b}}$ for (1) and $\rho := \frac{n-\hat{b}}{\hat{b}(n-1)}$ for (2), and $\hat{\eta}_t$ is computed recursively as:

$$\hat{\eta}_m = \frac{1}{L} \quad \text{and} \quad \hat{\eta}_t := \frac{1}{L\big(1 + \rho L\sum_{j=t+1}^m \hat{\eta}_j\big)}, \quad \forall t = 0, \cdots, m-1. \tag{41}$$

Then, we have $\Sigma_m := \sum_{t=0}^m \hat{\eta}_t \geq \frac{2(m+1)}{(\sqrt{2\rho m+1}+1)L}$.
Suppose that Assumptions 2.1 and 2.2 hold. Then, we have

$$\frac{1}{S\Sigma_m}\sum_{s=1}^S\sum_{t=0}^m \hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \leq \frac{(\sqrt{2\nu m+1}+1)L}{S(m+1)}\big[f(\widetilde{w}_0) - f^\star\big] + \frac{1}{S}\sum_{s=1}^S \hat{\sigma}_s, \tag{42}$$

where $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]$.

Let $\widetilde{w}_T \sim \mathbf{U}_p\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ such that $\mathbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t} := \frac{\hat{\eta}_t}{S\Sigma_m}$ for all $s = 1, \cdots, S$ and $t = 0, \cdots, m$, be the output of Algorithm 1. Then:

(a) ***The finite-sum case:*** *If we apply this variant of Algorithm 1 to solve (2) with $\psi = 0$ using $b_s := n$, $m := \lfloor \frac{n}{\hat{b}} \rfloor$, and $\hat{b} \in [1, \sqrt{n}]$, then under Assumptions 2.1 and 2.2:*

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{2L}{S\sqrt{n}}[f(\widetilde{w}_0) - f^\star]. \tag{43}$$

Consequently, the total of outer iterations S to achieve $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ does not exceed $S := \frac{2L[f(\widetilde{w}_0)-f^\star]}{\sqrt{n}\varepsilon^2}$. The number of individual stochastic gradient evaluations ∇f_i does not exceed $\mathcal{T}_{\rm grad} := \frac{10\sqrt{n}L[f(\widetilde{w}_0)-f^\star]}{\varepsilon^2}$.

(b) ***The expectation case:*** *If we apply this variant of Algorithm 1 to solve (1) with $\psi = 0$ using $b_s = b := \frac{2\sigma^2}{\varepsilon^2}$ for the outer-loop, $m := \frac{\sigma^2}{\hat{b}\varepsilon^2}$, and $\hat{b} \leq \frac{\sigma^2}{\varepsilon^2}$, then under Assumptions 2.1, 2.2, and 2.3:*

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{2L}{S\sqrt{\hat{b}m}}\big[f(\widetilde{w}_0) - f^\star\big] + \frac{\sigma^2}{b}. \tag{44}$$

Consequently, the total of outer iterations S to achieve $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ does not exceed $S := \frac{4L[f(\widetilde{w}_0)-f^\star]}{\sigma\varepsilon}$. The number of individual stochastic gradient evaluations does not exceed $\mathcal{T}_{\rm grad} := \frac{16\sigma L[f(\widetilde{w}_0)-f^\star]}{\varepsilon^3}$, provided that $\sigma \leq \frac{8L[f(\widetilde{w}_0)-f^\star]}{\varepsilon}$.

Note that the first statement (a) of Theorem 11 covers the nonconvex case of Nguyen et al. (2019) by fixing step-size $\hat{\eta}_t = \hat{\eta} = \frac{2}{L(1+\sqrt{4m+1})}$. However, this constant step-size is rather small if $m \leq \mathcal{O}(n)$ is large. Hence, it is better to update $\hat{\eta}_t$ dynamically increasing as in (41), where $\hat{\eta}_m = \frac{1}{L}$ is a large step-size. In addition, Nguyen et al. (2019) only study the finite-sum problem, while we also consider the expectation setting (1).

Again, by combining the first statement (a) of Theorem 11 and the lower-bound complexity in Fang et al. (2018), we can conclude that this algorithmic variant still achieves a nearly-optimal complexity $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$ for the non-composite finite-sum problem in (2) to find an ε-stationary point in expectation if $n \leq \mathcal{O}\left(\varepsilon^{-4}\right)$. In Statement (b), if $\sigma > \frac{8L[f(\widetilde{w}_0)-f^\star]}{\varepsilon}$, then the complexity of our method is $\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-3}\right)$ due to the gradient snapshot of the size $b = \mathcal{O}\left(\sigma^2\varepsilon^{-2}\right)$ to evaluate $v_0^{(s)}$. It matches the lower bound in (Arjevani et al., 2019).

5. Numerical Experiments

We present three numerical examples to illustrate our theory and compare our methods with state-of-the-art algorithms in the literature. We implement 8 different variants of our ProxSARAH algorithm:

- ProxSARAH-v1: Single sample and fixed step-sizes $\gamma := \frac{\sqrt{2}}{L\sqrt{3m}}$ and $\eta := \frac{2\sqrt{3m}}{4\sqrt{3m}+\sqrt{2}}$.
- ProxSARAH-v2: $\gamma := 0.95$ and mini-batch size $\hat{b} := \left\lfloor \frac{\sqrt{n}}{C} \right\rfloor$ and $m := \lfloor \sqrt{n} \rfloor$.
- ProxSARAH-v3: $\gamma := 0.99$ and mini-batch size $\hat{b} := \left\lfloor \frac{\sqrt{n}}{C} \right\rfloor$ and $m := \lfloor \sqrt{n} \rfloor$.
- ProxSARAH-v4: $\gamma := 0.95$ and mini-batch size $\hat{b} := \left\lfloor \frac{n^{\frac{1}{3}}}{C} \right\rfloor$ and $m := \lfloor n^{\frac{1}{3}} \rfloor$.
- ProxSARAH-v5: $\gamma := 0.99$ and mini-batch size $\hat{b} := \left\lfloor \frac{n^{\frac{1}{3}}}{C} \right\rfloor$ and $m := \lfloor n^{\frac{1}{3}} \rfloor$.
- ProxSARAH-A-v1: Single sample (i.e., $\hat{b} = 1$), and dynamic step-sizes.
- ProxSARAH-A-v2: $\gamma_m := 0.99$ and mini-batch size $\hat{b} := \lfloor \sqrt{n} \rfloor$ and $m := \lfloor \sqrt{n} \rfloor$.
- ProxSARAH-A-v3: $\gamma_m := 0.99$ and mini-batch size $\hat{b} := \lfloor n^{\frac{1}{3}} \rfloor$ and $m := \lfloor n^{\frac{1}{3}} \rfloor$.

Here, C is given in Subsection 3.4. We also implement 4 other algorithms:

- ProxSVRG: The proximal SVRG algorithm in Reddi et al. (2016b) for single sample with theoretical step-size $\eta = \frac{1}{3nL}$, and for the mini-batch case with $\hat{b} := \lfloor n^{2/3} \rfloor$, the epoch length $m := \lfloor n^{1/3} \rfloor$, and the step-size $\eta := \frac{1}{3L}$.
- ProxSpiderBoost: The proximal SpiderBoost method in Wang et al. (2019) with $\hat{b} := \lfloor \sqrt{n} \rfloor$, $m := \lfloor \sqrt{n} \rfloor$, and step-size $\eta := \frac{1}{2L}$.
- ProxSGD: Proximal stochastic gradient descent scheme (Ghadimi and Lan, 2013) with step-size $\eta_t := \frac{\eta_0}{1+\tilde{\eta}\lfloor t/n \rfloor}$, where $\eta_0 > 0$ and $\tilde{\eta} \geq 0$ will be given in each example.
- ProxGD: Standard proximal gradient descent algorithm with step-size $\eta := \frac{1}{L}$.

All algorithms are implemented in Python running on a single node of a Linux server (called Longleaf) with configuration: 3.40GHz Intel processors, 30M cache, and 256GB RAM. For the last example, we implement these algorithms in **TensorFlow** (Abadi et al., 2015) running on a GPU system. Our code is available online at

https://github.com/unc-optimization/StochasticProximalMethods.

To be fair for comparison, we compute the norm of gradient mapping $\|G_\eta(w_t^{(s)})\|$ for visualization at the same value $\eta := 0.5$ in all methods. To compute the relative loss residuals $\frac{F(\widetilde{w}_T)-F^*}{|F^*|}$, we use $F^* := \min\{\widetilde{F}_j^* \mid j\}$ as the minimum loss values $\widetilde{F}_j^*$ generated by all algorithms. To increase the readability of figures, we only plot the performance of some representative variants among the 8 instead of reporting them all. We run the first and second examples for 20 and 30 epochs, respectively whereas we increase it up to 150 and 300 epochs in the last example. Several data sets used in this paper are from (Chang and Lin, 2011)[2]. Two other well-known data sets are `mnist`[3] and `fashion_mnist`[4].

5.1 Nonnegative Principal Component Analysis

We reconsider the problem of non-negative principal component analysis (NN-PCA) studied in Reddi et al. (2016b). More precisely, for a given set of samples $\{z_i\}_{i=1}^n$ in $\mathbb{R}^d$, we solve the following constrained nonconvex problem:

$$f^\star := \min_{w\in\mathbb{R}^d}\Big\{f(w) := -\frac{1}{2n}\sum_{i=1}^n w^\top(z_iz_i^\top)w \mid \|w\| \le 1,\ w \ge 0\Big\}. \tag{45}$$

By defining $f_i(w) := -\frac{1}{2}w^\top(z_iz_i^\top)w$ for $i = 1,\cdots,n$, and $\psi(w) := \delta_{\mathcal{X}}(w)$, the indicator of $\mathcal{X} := \{w \in \mathbb{R}^d \mid \|w\| \le 1, w \ge 0\}$, we can formulate (45) into (2). Moreover, since z_i is normalized, the Lipschitz constant of ∇f_i is $L = 1$ for $i = 1,\cdots,n$. Since (45) is nonconvex, it may have different stationary points. For a given algorithm to approximate a good stationary point of (45), it crucially depends on initial point. Following Reddi et al. (2016b), we use ProxSGD to generate an initial point and use it for all algorithms.

(a) ***Small and medium data sets:*** We test all the algorithms on three different well-known data sets: `mnist` ($n = 60000$, $d = 784$), `rcv1-binary` ($n = 20242$, $d = 47236$), and `real-sim` ($n = 72309$, $d = 20958$). In ProxSGD, after manipulating different values, we set $\eta_0 := 0.1$ and $\tilde{\eta} := 1.0$ that allow us to obtain good performance.

Experiment 1 (Single sample comparison): We first verify our theory by running 5 algorithms with single sample (i.e., $\hat{b} = 1$). The relative objective residuals and the absolute norm of gradient mappings of these algorithms after 20 epochs are plotted in Figure 1.

Figure 1 shows that both ProxSARAH-v1 and its dynamic variant work really well and dominate all other methods. ProxSARAH-A-v1 is still better than ProxSARAH-v1. ProxSVRG is slow since its theoretical step-size $\frac{1}{3nL}$ is too small.

Experiment 2 (The effect o mini-batch sizes on ProxSARAH): In this experiment, we evaluate the effect of mini-batch sizes on the performance of ProxSARAH by running ProxSARAH on these data sets with different mini-batch sizes. We choose $\hat{b}$ among 6 values $\{n^{1/2}, 0.75n^{1/2}, 0.5n^{1/2}, 0.25n^{1/2}, 0.1n^{1/2}, 0.05n^{1/2}\}$. The results are shown in Figure 2.

As we can see from Figure 2 that the performance of each particular batch-size varies between data sets. Variants with larger mini-batch sizes work well in the `mnist` data set while

2. Available online at https://www.csie.ntu.edu.tw/∼cjlin/libsvm/
3. Available online at http://yann.lecun.com/exdb/mnist/
4. Available online at https://github.com/zalandoresearch/fashion-mnist

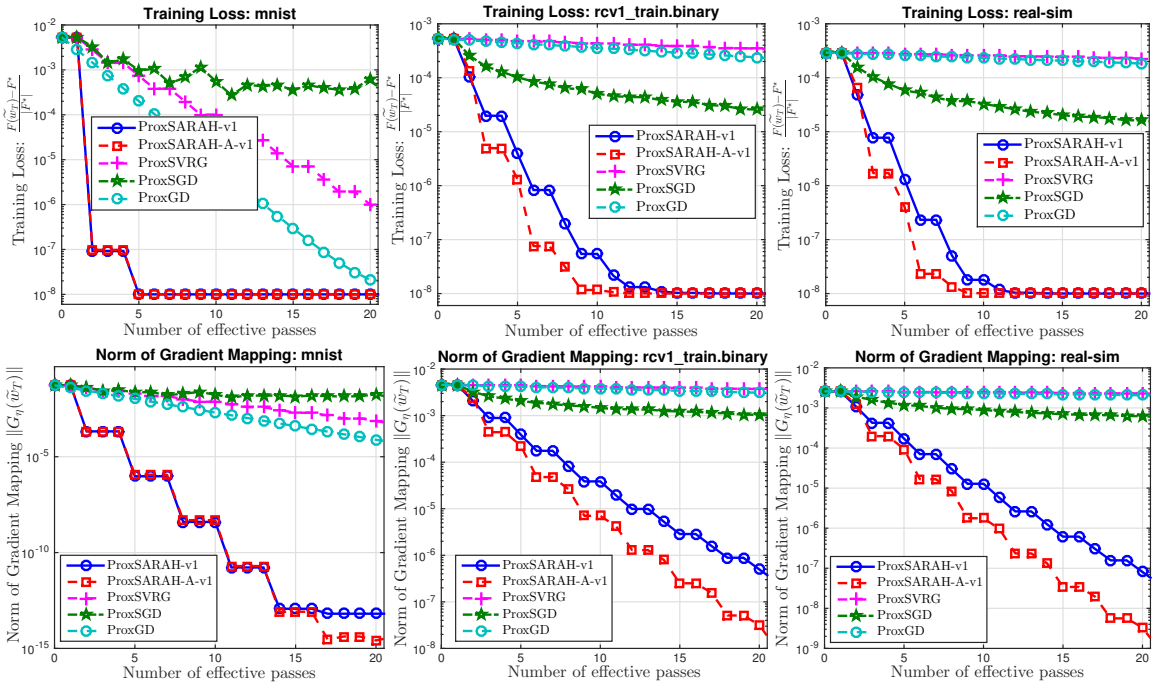

Figure 1: The objective value residuals and gradient mapping norms of (45) on three data sets: `mnist`, `rcv1-binary`, and `real-sim`.

variants with smaller mini-batch sizes are better in `rcv1_train.binary` and `real-sim`. It is unclear for our methods to show that a larger mini-batch size leads to a better performance or vice versa. Therefore, to achieve the best performance, a search over mini-batch size is recommended for each particular data set.

Experiment 3 (*Mini-batch comparison*)*:* Next, we run all the mini-batch variants of the methods described above to solve (45). The relative objective residuals and the norms of gradient mapping are plotted in Figure 3.

From Figure 3, we observe that ProxSpiderBoost works well since it has a large step-size $\eta = \frac{1}{2L}$, and it is comparable with ProxSARAH-A-v2. The variants with $\hat{b} = \mathcal{O}\left(n^{\frac{1}{3}}\right)$ of ProxSARAH and ProxSARAH-A perform well for `mnist` data set while the variants with $\hat{b} = \mathcal{O}\left(n^{\frac{1}{2}}\right)$ are better for the other two data sets. Although ProxSVRG takes $\eta = \frac{1}{3L}$, its choice of batch-size and epoch length also affects the performance resulting in a slower convergence. ProxSGD has good progress at early stage but then its relative objective residual is saturated around 10^{-5} accuracy. Also, its gradient mapping norms do not significantly decrease as in ProxSARAH variants or ProxSpiderBoost. Note that ProxSARAH variants with large step-size γ (e.g., $\gamma = 0.99$) are very similar to ProxSpiderBoost which results in resemblance in their performance.

(b) ***Large data sets:*** Now, we test these algorithms on larger data sets: `url_combined` ($n = 2,396,130; d = 3,231,961$), `news20.binary` ($n = 19,996; d = 1,355,191$), and

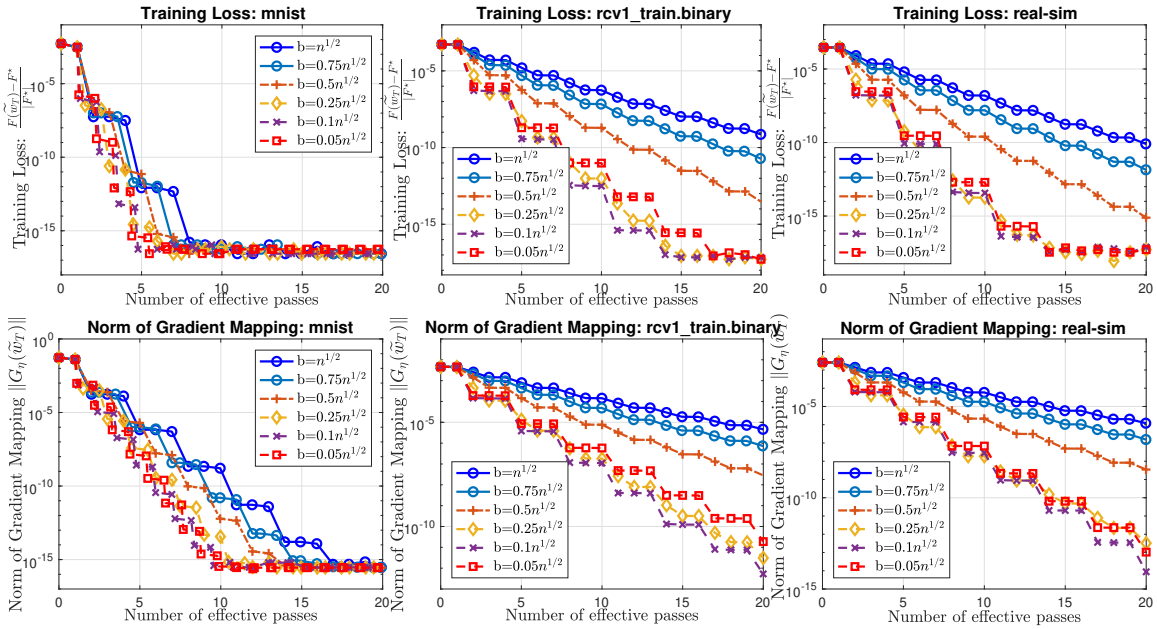

Figure 2: The relative objective residuals and the norms of gradient mappings of Prox-SARAH algorithms with different mini-batch sizes for solving (45) on three data sets: `mnist`, `rcv1-binary`, and `real-sim`.

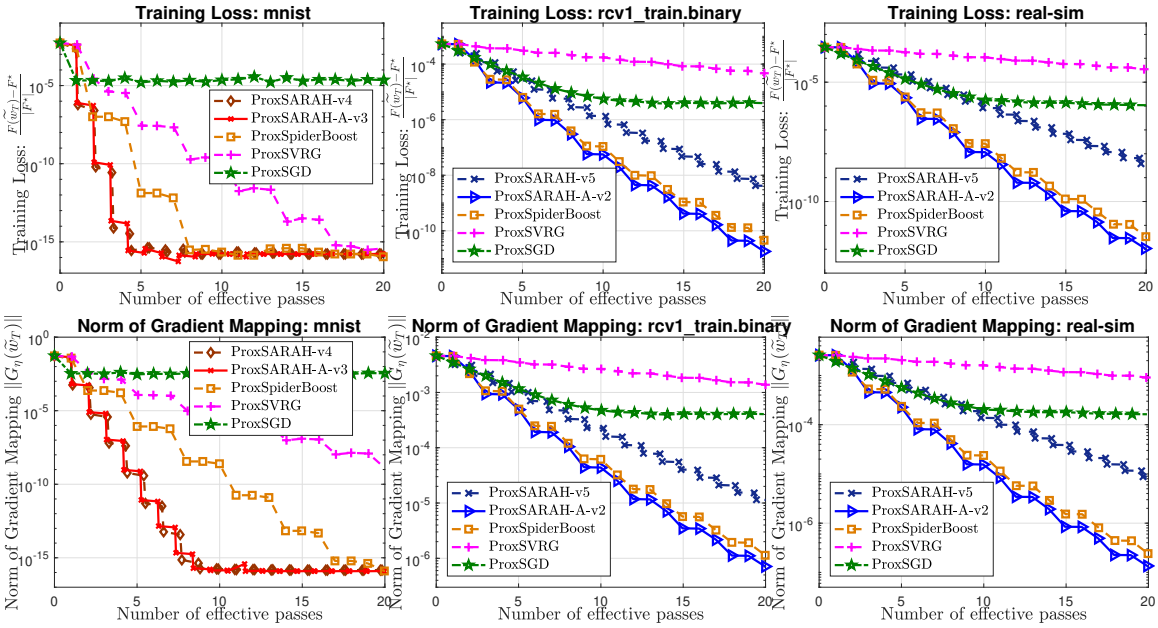

Figure 3: The relative objective residuals and the norms of gradient mappings of 5 algorithms for solving (45) on three data sets: `mnist`, `rcv1-binary`, and `real-sim`.

`avazu-app` ($n = 14,596,137; d = 999,990$). The relative objective residuals and the absolute norms of gradient mapping of this experiment are depicted in Figure 4.

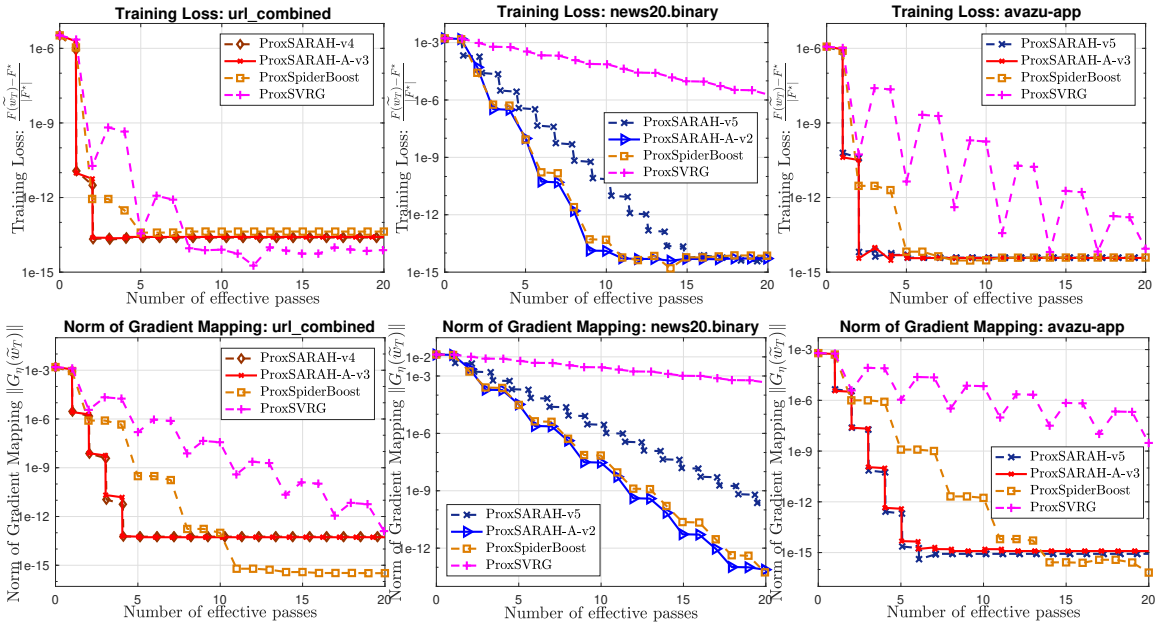

Figure 4: The relative objective residuals and the absolute gradient mapping norms of 4 algorithms for solving (45) on three data sets: `url_combined`, `news20.binary`, and `avazu-app`.

Experiment 4 (*Mini-batch comparison on large data sets*)*:* Figure 4 shows that ProxSARAH variants still work well and depend on the data set in which ProxSARAH-A-v2 or the variants with $\hat{b} = \mathcal{O}(n^{\frac{1}{3}})$ dominates other algorithms. In this experiment, ProxSpiderBoost gives smaller gradient mapping norms for `url_combined` and `avazu-app` in the last epochs than the others. However, these algorithms have achieved up to 10^{-13} accuracy in absolute values, the improvement of ProxSpiderBoost may not be necessary. With the same step-size as in the previous test, ProxSGD performs quite poorly on these three data sets, and we did not report its performance here.

5.2 Sparse Binary Classification with Nonconvex Losses

We consider the following sparse binary classification involving nonconvex loss function:

$$\min_{w\in\mathbb{R}^d}\left\{F(w) := \frac{1}{n}\sum_{i=1}^{n}\ell(a_i^\top w, b_i) + \lambda\|w\|_1\right\}, \tag{46}$$

where $\{(a_i, b_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{-1, 1\}^n$ is a given training data set, $\lambda > 0$ is a regularization parameter, and $\ell(\cdot,\cdot)$ is a given smooth and nonconvex loss function as studied in Zhao et al. (2010). By setting $f_i(w) := \ell(a_i^\top w, b_i)$ and $\psi(w) := \lambda\|w\|_1$ for $i \in [n]$, we obtain (2).

The loss function ℓ is chosen from one of the following three cases (Zhao et al., 2010):

- ***Normalized sigmoid loss:*** $\ell_1(s,\tau) := 1 - \tanh(\omega\tau s)$ for a given $\omega > 0$. Since $\left|\frac{d^2\ell_1(s,\tau)}{ds^2}\right| \le \frac{8(2+\sqrt{3})(1+\sqrt{3})\omega^2\tau^2}{(3+\sqrt{3})^2}$ and $|\tau| = 1$, we can show that $\ell_1(\cdot,\tau)$ is L-smooth with respect to s, where $L := \frac{8(2+\sqrt{3})(1+\sqrt{3})\omega^2}{(3+\sqrt{3})^2} \approx 0.7698\omega^2$.

- ***Nonconvex loss in 2-layer neural networks:*** $\ell_2(s,\tau) := \left(1 - \frac{1}{1+\exp(-\tau s)}\right)^2$. For this function, we have $\left|\frac{d^2\ell_2(s,\tau)}{ds^2}\right| \le 0.15405\tau^2$. If $|\tau| = 1$, then this function is also L-smooth with $L = 0.15405$.
- ***Logistic difference loss:*** $\ell_3(s,\tau) := \ln(1+\exp(-\tau s)) - \ln(1+\exp(-\tau s - \omega))$ for some $\omega > 0$. With $\omega = 1$, we have $|\frac{d^2\ell_3(s,\tau)}{ds^2}| \le 0.092372\tau^2$. Therefore, if $|\tau| = 1$, then this function is also L-smooth with $L = 0.092372$.

We set the regularization parameter $\lambda := \frac{1}{n}$ in all the tests, which gives us relatively sparse solutions. We test the above algorithms on different scenarios ranging from small to large data sets, where we use 6 different data sets from LIBSVM.

(a) ***Small and medium data sets:*** We consider three small to medium data sets: `rcv1.binary` ($n = 20,242$, $d = 47,236$), `real-sim` ($n = 72,309$, $d = 20,958$), and `epsilon` ($n = 400,000$, $d = 2,000$).

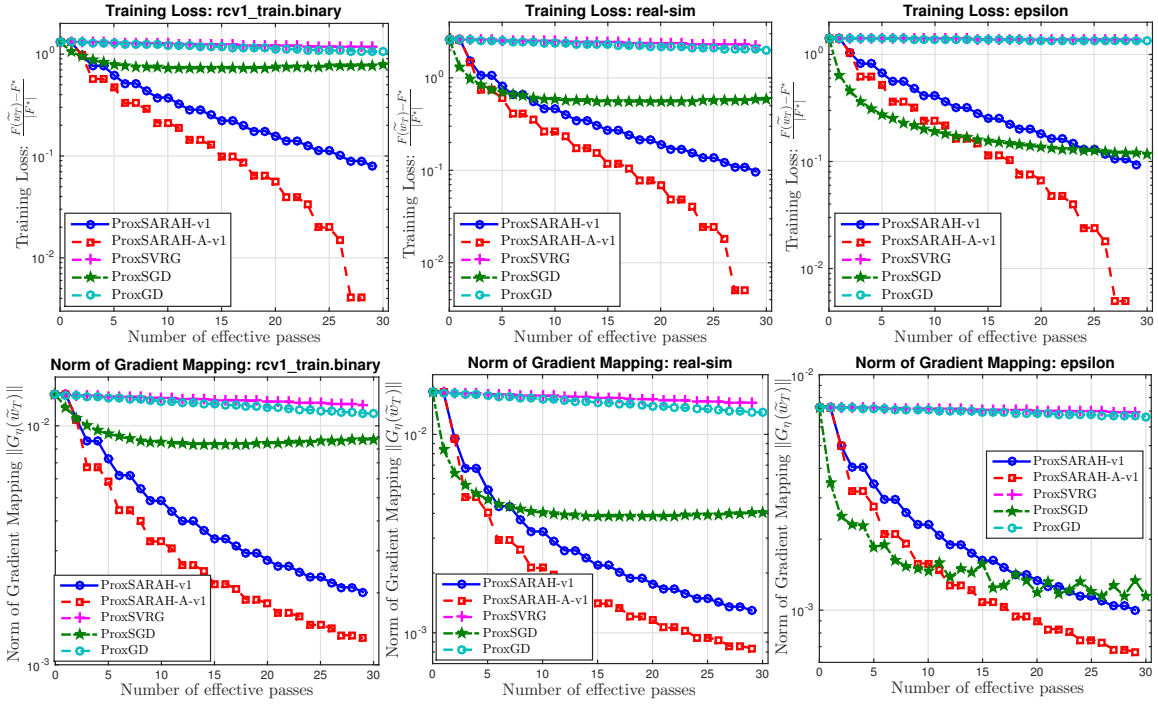

Figure 5: The relative objective residuals and gradient mapping norms of (46) on three data sets using the loss $\ell_2(s,\tau)$ - The single sample case.

Experiment 5 (*Singe sample comparison on* (46))*:* Figure 5 shows the relative objective residuals and the gradient mapping norms on these three data sets for the loss function $\ell_2(\cdot)$ in the single sample case. Similar to the first example, ProxSARAH-v1 and its dynamic variant work well, whereas ProxSARAH-A-v1 is better. ProxSVRG is still slow due to small step-size. ProxSGD appears to be better than ProxSVRG and ProxGD within 30 epochs.

Now, we test the loss function $\ell_2(\cdot)$ with the mini-batch variants using the same three data sets. Figure 6 shows the results on these data sets.

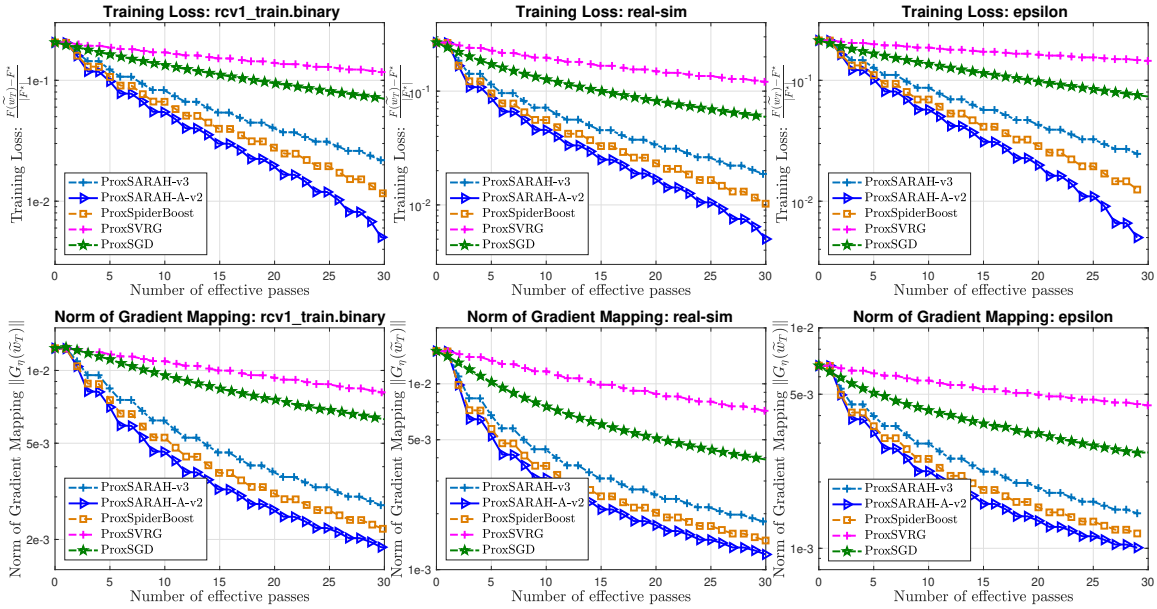

Figure 6: The relative objective residuals and gradient mapping norms of (46) on three data sets using the loss $\ell_2(s,\tau)$ - The mini-batch case.

We can see that ProxSARAH-A-v2 is the most effective algorithm whereas ProxSpiderBoost also performs well due to large step-size as discussed. ProxSVRG remains slow in this test. Notice that ProxSARAH dynamic variants normally perform better than their corresponding fixed step-size variants in this experiment. Additionally, ProxSARAH-A-v2 still preserves the best-known complexity $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$.

Experiment 6 (Mini-batch comparison on (46)): To further illustrate the advantage of the increasing step-size, we run ProxSARAH and ProxSARAH-A with different mini-batch sizes and select the top two variants of each for comparison when applying to solve (46) using the loss function ℓ_2. Their results along with the chosen mini-batch sizes are depicted in Figure 7. We can see that ProxSARAH-A performs better than ProxSARAH in all three data sets which confirms the advantage of the dynamic step-size scheme.

(b) ***Large data sets:*** Next, we test these algorithms on three large data sets: `url_combined` ($n = 2,396,130$, $d = 3,231,961$), `avazu-app` ($n = 14,596,137$, $d = 999,990$), and `kddb-raw` ($n = 19,264,097$, $d = 3,231,961$).

Experiment 7 (Mini-batch comparison on large data sets): Since we use large data sets, we only test the mini-batch variants. Figure 8 presents the results on these data sets.

Again, we can observe from Figure 8 that, ProxSARAH-A-v2 achieves the best performance. ProxSpiderBoost also works well in this experiment while ProxSVRG are comparable with ProxSARAH-v1 and ProxSARAH-v2. ProxSGD also has good performance but is not as good as ProxSpiderBoost and ProxSARAH variants.

The complete results on these three data sets with three loss functions are presented in Table 3. Apart from the relative objective residuals and gradient mapping norms, the table

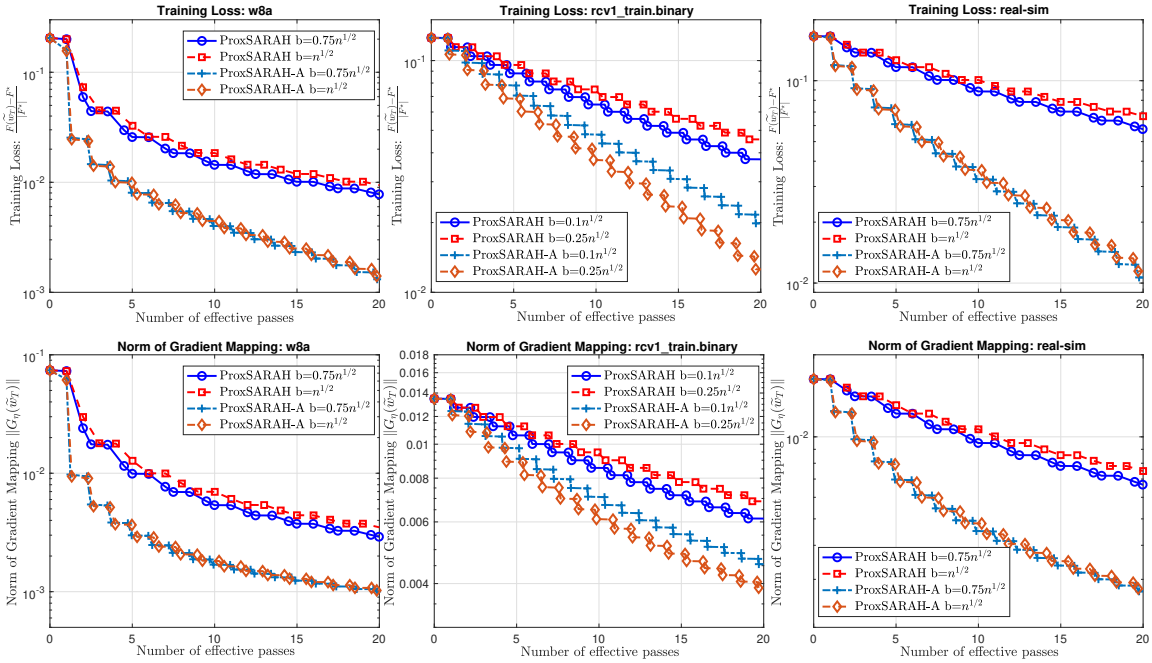

Figure 7: The relative objective residuals and gradient mapping norms of (46) on three data sets using the loss $\ell_2(s,\tau)$.

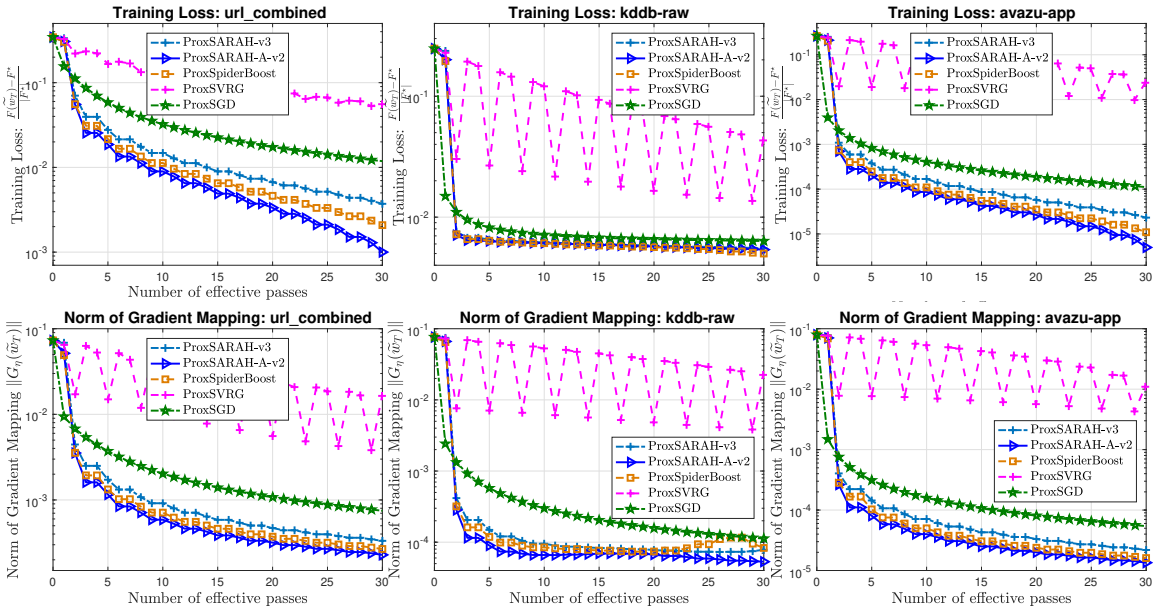

Figure 8: The relative objective residuals and gradient mapping norms of (46) on three large data sets using the loss $\ell_2(s,\tau)$ - The mini-batch case.

consists of both training and test accuracies where we use 10% of the data set to evaluate the testing accuracy.

Algorithms	$\|G_\eta(\widetilde{w}_T)\|^2$			$(F(w_T)-F^*)/\|F^*\|$			Training Accuracy			Test Accuracy		
	ℓ_1-Loss	ℓ_2-Loss	ℓ_3-Loss	ℓ_1-Loss	ℓ_2-Loss	ℓ_3-Loss	ℓ_1-Loss	ℓ_2-Loss	ℓ_3-Loss	ℓ_1-Loss	ℓ_2-Loss	ℓ_3-Loss
`url_combined` ($n = 2,396,130$, $d = 3,231,961$)												
ProxSARAH-v2	2.534e-06	5.827e-08	1.181e-07	1.941e-01	1.397e-02	8.092e-02	0.965	0.9684	0.9657	0.9636	0.9672	0.9646
ProxSARAH-v3	2.772e-06	5.515e-08	1.110e-07	2.065e-01	9.149e-03	7.399e-02	0.965	0.9685	0.9658	0.9635	0.9673	0.9647
ProxSARAH-v4	1.252e-05	6.003e-06	1.433e-05	4.749e-01	8.210e-01	1.597e+00	0.962	0.9617	0.9558	0.9614	0.9607	0.9528
ProxSARAH-v5	1.182e-05	5.595e-06	1.346e-05	4.617e-01	7.931e-01	1.546e+00	0.962	0.9617	0.9568	0.9615	0.9609	0.9537
ProxSARAH-A-v2	1.115e-06	**4.969e-08**	**5.215e-08**	9.225e-02	**1.076e-05**	**1.268e-05**	**0.966**	**0.9687**	**0.9672**	0.9645	**0.9676**	**0.9662**
ProxSARAH-A-v3	1.034e-05	3.639e-07	4.555e-07	4.325e-01	1.946e-01	2.619e-01	0.962	0.9644	0.9634	0.9616	0.9631	0.9625
ProxSpiderBoost	1.375e-06	6.454e-08	7.158e-08	1.178e-01	2.274e-02	2.947e-02	0.965	0.9681	0.9664	0.9641	0.9669	0.9653
ProxSVRG	7.391e-03	2.043e-04	2.697e-04	2.196e+00	1.091e+00	1.490e+00	0.958	0.9601	0.9595	0.9570	0.9585	0.9579
ProxSGD	**5.005e-07**	2.340e-07	5.963e-07	**4.446e-03**	1.406e-01	3.062e-01	0.968	0.9651	0.9633	**0.9667**	0.9637	0.9624
`avazu-app` ($n = 14,596,137$, $d = 999,990$)												
ProxSARAH-v2	8.647e-09	1.053e-08	5.074e-10	4.354e-04	1.958e-03	1.687e-04	0.883	0.8843	0.8834	0.8615	0.8617	0.8615
ProxSARAH-v3	9.757e-09	9.792e-09	4.776e-10	4.615e-04	1.397e-03	1.554e-04	0.883	**0.8844**	0.8834	0.8615	0.8617	0.8615
ProxSARAH-v4	9.087e-08	3.179e-07	1.841e-07	1.738e-03	5.102e-02	9.816e-03	0.883	0.8834	0.8834	0.8615	0.8615	0.8615
ProxSARAH-v5	8.568e-08	3.029e-07	1.702e-07	1.675e-03	5.036e-02	9.433e-03	0.883	0.8834	0.8834	0.8615	0.8615	0.8615
ProxSARAH-A-v2	3.062e-09	**8.724e-09**	**1.814e-10**	2.046e-04	**5.467e-07**	**1.388e-08**	0.883	**0.8844**	0.8834	0.8615	0.8617	0.8615
ProxSARAH-A-v3	7.784e-08	5.124e-08	4.405e-09	1.604e-03	2.499e-02	1.223e-03	0.883	0.8834	0.8834	0.8615	0.8615	0.8615
ProxSpiderBoost	4.050e-09	1.152e-08	2.579e-10	2.626e-04	3.090e-03	5.073e-05	0.883	0.8842	0.8834	0.8615	0.8617	0.8615
ProxSVRG	4.218e-03	1.309e-03	1.202e-04	3.137e-01	4.287e-01	2.031e-01	0.883	0.8648	0.8834	0.8615	0.8146	0.8615
ProxSGD	**9.063e-10**	2.839e-08	3.150e-09	**6.449e-06**	1.595e-02	9.536e-04	0.883	0.8835	0.8834	0.8615	0.8616	0.8615
`kddb-raw` ($n = 19,264,097$, $d = 3,231,961$)												
ProxSARAH-v2	2.013e-08	1.770e-08	5.688e-09	7.235e-04	3.455e-03	4.295e-03	0.862	0.8654	0.8619	0.8531	0.8560	0.8534
ProxSARAH-v3	2.168e-08	1.669e-08	6.105e-09	7.903e-04	2.275e-03	3.741e-03	0.862	0.8655	0.8619	0.8530	0.8561	0.8534
ProxSARAH-v4	2.265e-07	4.066e-07	2.796e-07	3.862e-03	9.196e-02	2.203e-02	0.862	0.8617	0.8615	0.8530	0.8533	0.8531
ProxSARAH-v5	2.127e-07	3.943e-07	2.600e-07	3.725e-03	9.098e-02	2.152e-02	0.862	0.8617	0.8615	0.8530	0.8533	0.8531
ProxSARAH-A-v2	7.955e-09	**1.490e-08**	**2.830e-09**	2.106e-04	**8.502e-07**	2.829e-03	0.862	**0.8656**	**0.8621**	0.8531	**0.8562**	**0.8536**
ProxSARAH-A-v3	1.951e-07	1.036e-07	9.293e-09	3.539e-03	4.887e-02	9.223e-03	0.862	0.8627	0.8616	0.8530	0.8544	0.8531
ProxSpiderBoost	9.867e-09	1.906e-08	6.889e-09	3.082e-04	5.249e-03	**5.026e-07**	0.862	0.8652	0.8619	0.8531	0.8559	0.8534
ProxSVRG	1.225e-02	1.105e-03	5.040e-04	3.541e-01	3.471e-01	2.780e-01	0.860	0.8611	0.8599	0.8518	0.8529	0.8519
ProxSGD	**6.027e-09**	8.899e-08	1.331e-08	**2.593e-05**	4.320e-02	9.937e-03	0.862	0.8629	0.8616	0.8530	0.8546	0.8531

Table 3: The results of 9 algorithms on three data sets: url_combined, avazu-app, and kddb-raw.

Among three loss functions, the loss ℓ_2 gives the best training and testing accuracy. The accuracy is consistent with the result reported in (Zhao et al., 2010). ProxSGD seems to give good results on the ℓ_1-loss, but ProxSARAH-A-v2 is the best for the ℓ_2 and ℓ_3-losses in the majority of the test.

5.3 Feedforward Neural Network Training Problem

We consider the following composite nonconvex optimization model arising from a feedforward neural network configuration:

$$\min_{w\in\mathbb{R}^d}\left\{F(w) := \frac{1}{n}\sum_{i=1}^{n}\ell\big(h(w,a_i),b_i\big) + \psi(w)\right\}, \tag{47}$$

where we concatenate all the weight matrices and bias vectors of the neural network in one vector of variable w, $\{(a_i,b_i)\}_{i=1}^n$ is a training data set, $h(\cdot)$ is a composition between all linear transforms and activation functions as $h(w,a) := \boldsymbol{\sigma}_l(W_l\boldsymbol{\sigma}_{l-1}(W_{l-1}\boldsymbol{\sigma}_{l-2}(\cdots\boldsymbol{\sigma}_0(W_0a+\mu_0)\cdots)+\mu_{l-1})+\mu_l)$, where W_i is a weight matrix, μ_i is a bias vector, $\boldsymbol{\sigma}_i$ is an activation function, l is the number of layers, $\ell(\cdot)$ is the soft-max cross-entropy loss, and ψ is a convex regularizer (e.g., $\psi(w) := \lambda\|w\|_1$ for some $\lambda > 0$ to obtain sparse weights). Again, by

defining $f_i(w) := \ell(h(w, a_i), b_i)$ for $i \in [n]$, we can bring (47) into the same composite finite-sum setting (2).

We implement our algorithms and other methods in TensorFlow (Abadi et al., 2015) and use two data sets `mnist` and `fashion_mnist` to evaluate their performance. In the first experiment, we use a one-hidden-layer fully connected neural network: $784 \times 100 \times 10$ for both `mnist` and `fashion_mnist`. The activation function $\boldsymbol{\sigma}_i$ of the hidden layer is ReLU and the loss function is soft-max cross-entropy. To estimate the Lipschitz constant L, we normalize the input data. The regularization parameter λ is set at $\lambda := \frac{1}{n}$ and $\psi(\cdot) := \lambda \|\cdot\|_1$.

Experiment 8 ($784 \times 100 \times 10$ *network*)*:* We first test ProxSARAH, ProxSVRG, ProxSpiderBoost, and ProxSGD using mini-batch. For ProxSGD, we use the mini-batch $\hat{b} = 245$, $\eta_0 = 0.1$, and $\tilde{\eta} = 0.5$ for both data sets. For the `mnist` data set, we tune $L = 1$ then follow the configuration in Subsection 3.4 to choose η, γ, m, and $\hat{b}$ for ProxSARAH variants. We also tune the learning rate for ProxSVRG at $\eta = 0.2$, and for ProxSpiderBoost at $\eta = 0.12$. However, for the `fashion_mnist` data set, it requires a smaller learning rate. Therefore, we choose $L = 4$ for ProxSARAH and follow the theory in Subsection 3.4 to set η, γ, m, and $\hat{b}$. We also tune the learning rate for ProxSVRG and ProxSpiderBoost until they are stabilized to obtain the best possible step-size in this example as $\eta_{\text{ProxSVRG}} = 0.11$ and $\eta_{\text{ProxSpiderBoost}} = 0.15$, respectively.

Figure 9 shows the convergence of different variants of ProxSARAH, ProxSpiderBoost, ProxSVRG, and ProxSGD on three criteria for `mnist` and `fashion_mnist`: training loss values, the absolute norm of gradient mapping, and the test accuracy. For ProxSARAH, we find that two variants with $\hat{b} = \mathcal{O}\left(n^{\frac{1}{2}}\right)$ and $\hat{b} = \mathcal{O}\left(n^{\frac{1}{3}}\right)$ perform well among other choices.

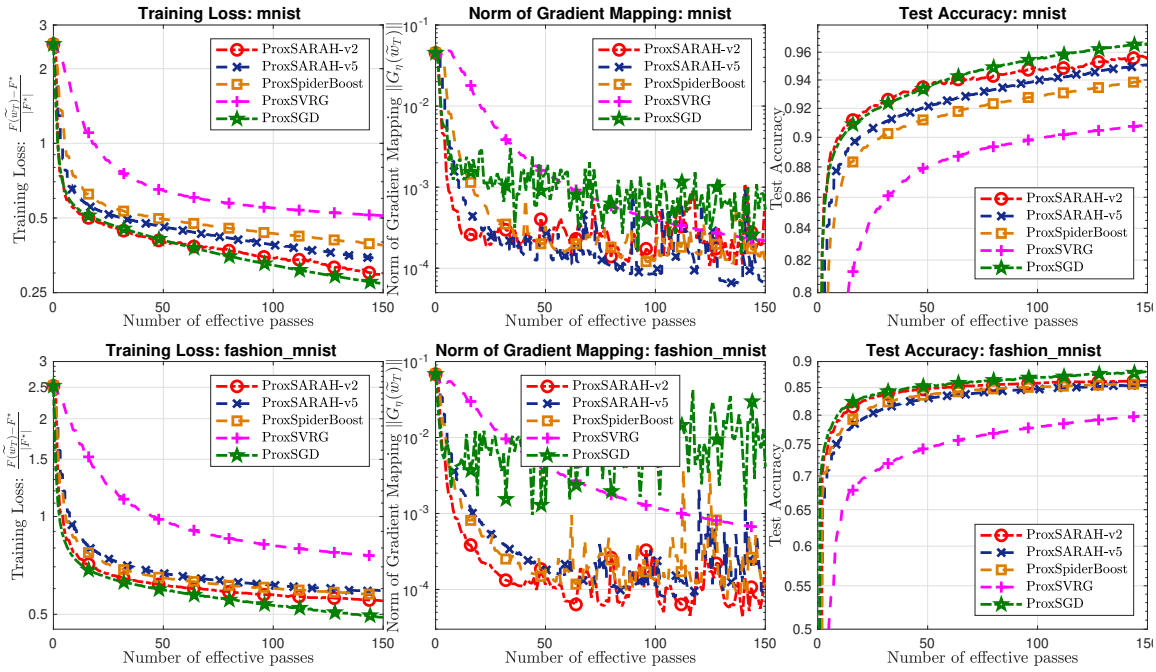

Figure 9: The training loss, gradient mapping, and test accuracy on `mnist` (top line) and `fashion_mnist` (bottom line) of 5 algorithms.

In this example, ProxSGD appears to be the best in terms of training loss and test accuracy. However, the norm of gradient mapping is rather different from others, relatively large, and oscillated. ProxSVRG is clearly slower than ProxSpiderBoost due to smaller learning rate. The two variants of ProxSARAH perform relatively well, but the variants with $\hat{b} = \mathcal{O}(\sqrt{n})$ seem to be slightly better. Note that the norm of gradient mapping tends to be decreasing but still oscillated since perhaps we are taking the last iterate instead of a random choice of intermediate iterates as stated in the theory.

Experiment 9 ($784 \times 800 \times 10$ *network*)*:* Finally, we test the above algorithm on `mnist` using a $784 \times 800 \times 10$ network as known to give a better test accuracy. We run all algorithms for 300 epochs and the results are given in Figure 10.

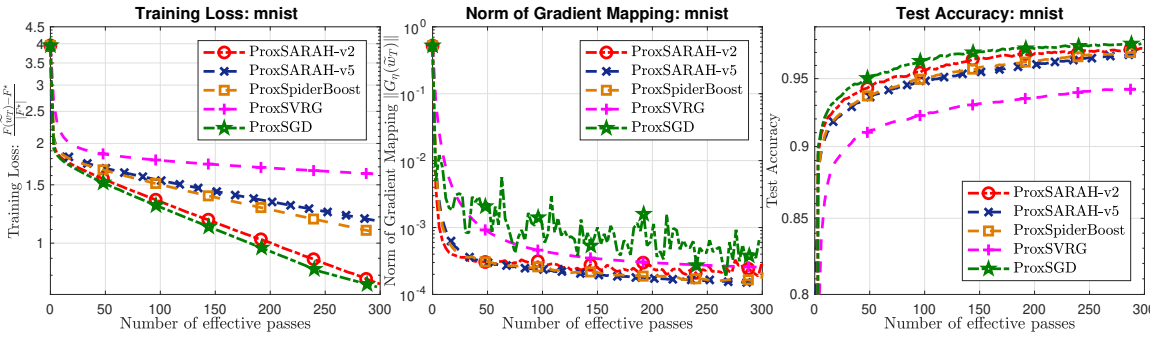

Figure 10: The training loss, gradient mapping, and test accuracy on `mnist` of 5 algorithms on a $784 \times 800 \times 10$ neural network (See http://yann.lecun.com/exdb/mnist/).

As we can see from Figure 10 that ProxSARAH-v2, ProxSARAH-v3, and ProxSGD performs really well in terms of training loss and test accuracy. However, our method can achieve lower as well as less oscillated gradient mapping norm than ProxSGD. Also, ProxSpiderBoost has similar performance to ProxSARAH-v4 and ProxSARAH-v5. ProxSVRG again does not have a good performance in this example in terms of loss and test accuracy but is slightly better than ProxSGD regarding gradient mapping norm.

6. Conclusions

We have proposed a unified stochastic proximal-gradient framework using the SARAH estimator to solve both the composite expectation problem (1) and the composite finite sum problem (2). Our algorithm is different from existing stochastic proximal gradient-type methods such as ProxSVRG and ProxSpiderBoost at which we have an additional averaging step. Moreover, it can work with both single sample and mini-batch using either constants or dynamic step-sizes. Our dynamic step-size is updated in an increasing fashion as opposed to a diminishing step-size in ProxSGD. We have established the best-known complexity bounds for all cases. We believe that our methods give more flexibility to trade-off between step-sizes and mini-batch in order to obtain good performance in practice. The numerical experiments have shown that our methods are comparable or even outperform existing methods, especially in the single sample case.

Acknowledgements

The work of Q. Tran-Dinh has partly been supported by the National Science Foundation (NSF), grant no. DMS-1619884, and the Office of Naval Research (ONR), grant no. N00014-20-1-2088 (2020-2023).

Appendix A. Technical Lemmas

This appendix provides the missing proofs of Lemma 2 and one elementary result, Lemma 12, used in our analysis in the sequel.

Lemma 12 *Given three positive constants ν, δ, and L, let $\{\gamma_t\}_{t=0}^m$ be a positive sequence satisfying the following conditions:*

$$\begin{cases} L\gamma_m - \delta & \le 0, \\ \nu L^2\gamma_t\sum_{j=t+1}^m \gamma_j - \delta + L\gamma_t & \le 0, \quad t = 0, \cdots, m-1. \end{cases} \tag{48}$$

Then, the following statements hold:

(a) *The sequence $\{\gamma_t\}_{t=0}^m$ computed recursively in a backward mode as*

$$\gamma_m := \frac{\delta}{L}, \quad \text{and} \quad \gamma_t := \frac{\delta}{L\big[1 + \nu L\sum_{j=t+1}^m \gamma_j\big]}, \quad t = 0, \cdots, m-1, \tag{49}$$

tightly satisfies (48). *Moreover, we have $\frac{\delta}{L(1+\delta\nu m)} < \gamma_0 < \gamma_1 < \cdots < \gamma_m$ and*

$$\Sigma_m := \sum_{t=0}^m \gamma_t \ge \frac{2\delta(m+1)}{L\big[\sqrt{1 + 2\delta\nu m} + 1\big]}. \tag{50}$$

(b) *The constant sequence $\{\gamma_t\}_{t=0}^m$ with $\gamma_t := \frac{2\delta}{L(\sqrt{1+4\delta\nu m}+1)}$ satisfies* (48).

Proof (a) The sequence $\{\gamma_t\}_{t=0}^m$ given by (49) is in fact computed from (48) by setting all the inequalities "$\le$" to equalities "$=$". Hence, it automatically satisfies (48). Moreover, it is obvious that $\gamma_0 < \gamma_1 < \cdots < \gamma_m$. Since $\sum_{t=1}^m \gamma_t < m\gamma_m = \frac{m\delta}{L}$, we have $\gamma_0 > \frac{\delta}{L(1+\delta\nu m)}$.

Let $\Sigma_m := \sum_{t=0}^m \gamma_t$. Using Σ_m into (48) with all equalities, we can rewrite it as

$$\begin{cases} \nu L^2\gamma_m\Sigma_m & = \delta - L\gamma_m & + \nu L^2(\gamma_m^2 + \gamma_m\gamma_{m-1} + \gamma_m\gamma_{m-2} + \cdots + \gamma_m\gamma_0) \\ \nu L^2\gamma_{m-1}\Sigma_m & = \delta - L\gamma_{m-1} & + \nu L^2(\gamma_{m-1}^2 + \gamma_{m-1}\gamma_{m-2} + \gamma_{m-1}\gamma_{m-3} + \cdots + \gamma_{m-1}\gamma_0) \\ \cdots & \cdots & \cdots \\ \nu L^2\gamma_1\Sigma_m & = \delta - L\gamma_1 & + \nu L^2(\gamma_1^2 + \gamma_1\gamma_0) \\ \nu L^2\gamma_0\Sigma_m & = \delta - L\gamma_0 & + \nu L^2\gamma_0^2. \end{cases}$$

Summing up both sides of these equations, and using the definition of Σ_m and $S_m^2 := \sum_{t=0}^m \hat{\eta}_t^2$, we obtain

$$\nu L^2\Sigma_m^2 = (m+1)\delta - L\Sigma_m + \frac{\nu L^2}{2}(\Sigma_m^2 + S_m^2).$$

Since $(m+1)S_m^2 \geq \Sigma_m^2$ by the Cauchy-Schwarz inequality, the last expression leads to

$$\nu L^2\Sigma_m^2 + 2L\Sigma_m - 2\delta(m+1) = \nu L^2 S_m^2 \geq \frac{\nu L^2\Sigma_m^2}{m+1}.$$

Therefore, by solving the quadratic inequation $\nu m L^2\Sigma_m^2 + 2(m+1)L\Sigma_m - 2\delta(m+1)^2 \geq 0$ in Σ_m with $\Sigma_m > 0$, we obtain

$$\Sigma_m \geq \frac{2\delta(m+1)}{L\big[1+\sqrt{1+2\delta\nu m}\big]},$$

which is exactly (50).

(b) Let $\gamma_t := \gamma > 0$ for $t = 0, \cdots, m$. Then (48) holds if $\nu L^2\gamma^2 m - \delta + L\gamma = 0$. Solving this quadratic equation in γ and noting that $\gamma > 0$, we obtain $\gamma = \frac{2\delta}{L(\sqrt{1+4\delta\nu m}+1)}$. ■

Proof (***The proof of Lemma 2: Properties of stochastic estimators***): We only prove (23), since other statements were proved in Harikandeh et al. (2015); Lohr (2009); Nguyen et al. (2017b, 2018a). The proof of (23) for the finite-sum case (2) was also given in Nguyen et al. (2018a) but under the L-smoothness of each f_i, we conduct this proof here by following the same path as in Nguyen et al. (2018a) for completeness.

Our goal is to prove (24) by upper bounding the following quantity:

$$\mathcal{A}_t := \mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] - \|\nabla f(w_t) - \nabla f(w_{t-1})\|^2. \tag{51}$$

Let $\mathcal{F}_t := \sigma(w_0^{(s)}, \mathcal{B}_1, \cdots, \mathcal{B}_{t-1})$ be the σ-field generated by $w_0^{(s)}$ and mini-batches $\mathcal{B}_1, \cdots, \mathcal{B}_{t-1}$, and $\mathcal{F}_0 = \mathcal{F}_1 = \sigma(w_0^{(s)})$. If we define $\Xi_i := \nabla f_i(w_t) - \nabla f_i(w_{t-1})$, then using the update rule (17), we can upper bound $\mathcal{A}_t$ in (51) as

$$\begin{aligned}
\mathcal{A}_t &= \mathbb{E}\Big[\|\tfrac{1}{b_t}\textstyle\sum_{i\in\mathcal{B}_t}\Xi_i\|^2 \mid \mathcal{F}_t\Big] - \|\tfrac{1}{n}\textstyle\sum_{i=1}^n\Xi_i\|^2\\
&= \tfrac{1}{b_t^2}\mathbb{E}\Big[\textstyle\sum_{i\in\mathcal{B}_t}\sum_{j\in\mathcal{B}_t}\langle\Xi_i,\Xi_j\rangle \mid \mathcal{F}_t\Big] - \tfrac{1}{n^2}\textstyle\sum_{i=1}^n\sum_{j=1}^n\langle\Xi_i,\Xi_j\rangle\\
&= \tfrac{1}{b_t^2}\mathbb{E}\Big[\textstyle\sum_{i,j\in\mathcal{B}_t, i\neq j}\langle\Xi_i,\Xi_j\rangle + \sum_{i\in\mathcal{B}_t}\|\Xi_i\|^2 \mid \mathcal{F}_t\Big] - \tfrac{1}{n^2}\textstyle\sum_{i=1}^n\sum_{j=1}^n\langle\Xi_i,\Xi_j\rangle\\
&= \tfrac{1}{b_t^2}\Big[\tfrac{b_t(b_t-1)}{n(n-1)}\textstyle\sum_{i,j=1,i\neq j}^n\langle\Xi_i,\Xi_j\rangle + \tfrac{b_t}{n}\textstyle\sum_{i=1}^n\|\Xi_i\|^2\Big] - \tfrac{1}{n^2}\textstyle\sum_{i=1}^n\sum_{j=1}^n\langle\Xi_i,\Xi_j\rangle\\
&= \tfrac{(b_t-1)}{b_t n(n-1)}\textstyle\sum_{i,j=1}^n\langle\Xi_i,\Xi_j\rangle + \tfrac{(n-b_t)}{b_t n(n-1)}\textstyle\sum_{i=1}^n\|\Xi_i\|^2 - \tfrac{1}{n^2}\textstyle\sum_{i=1}^n\sum_{j=1}^n\langle\Xi_i,\Xi_j\rangle\\
&= \tfrac{(n-b_t)}{b_t n(n-1)}\textstyle\sum_{i=1}^n\|\Xi_i\|^2 - \tfrac{(n-b_t)}{(n-1)b_t}\|\tfrac{1}{n}\textstyle\sum_{i=1}^n\Xi_i\|^2\\
&= \tfrac{(n-b_t)}{b_t(n-1)}\tfrac{1}{n}\textstyle\sum_{i=1}^n\|\nabla f_i(w_t) - \nabla f_i(w_{t-1})\|^2 - \tfrac{(n-b_t)}{(n-1)b_t}\|\nabla f(w_t) - \nabla f(w_{t-1})\|^2,
\end{aligned}$$

where we use the facts that

$$\begin{aligned}
&\mathbb{E}\Big[\textstyle\sum_{i,j\in\mathcal{B}_t, i\neq j}\langle\Xi_i,\Xi_j\rangle \mid \mathcal{F}_t\Big] = \tfrac{b_t(b_t-1)}{n(n-1)}\textstyle\sum_{i,j=1,i\neq j}^n\langle\Xi_i,\Xi_j\rangle\\
\text{and}\quad &\mathbb{E}\big[\textstyle\sum_{i\in\mathcal{B}_t}\|\Xi_i\|^2 \mid \mathcal{F}_t\big] = \tfrac{b_t}{n}\textstyle\sum_{i=1}^n\|\Xi_i\|^2
\end{aligned}$$

in the third line of the above derivation. Rearranging the estimate $\mathcal{A}_t$, we obtain (23).

To prove (24), we define $\Xi_i := \nabla_w \mathbf{f}(w_t;\xi_i) - \nabla_w \mathbf{f}(w_{t-1};\xi_i)$. Clearly, $\mathbb{E}\left[\Xi_i \mid \mathcal{F}_t\right] = \nabla f(w_t) - \nabla f(w_{t-1})$ and $v_t - v_{t-1} = \frac{1}{b_t}\sum_{i\in\mathcal{B}_t}\Xi_i$. Similar to (20), we have

$$\mathbb{E}\left[\|(v_t - v_{t-1}) - \mathbb{E}\left[\Xi_i \mid \mathcal{F}_t\right]\|^2 \mid \mathcal{F}_t\right] = \tfrac{1}{b_t}\mathbb{E}\left[\|\Xi_i - \mathbb{E}\left[\Xi_i \mid \mathcal{F}_t\right]\|^2 \mid \mathcal{F}_t\right].$$

Using the fact that $\mathbb{E}\left[\|X - \mathbb{E}\left[X\right]\|^2\right] = \mathbb{E}\left[\|X\|^2\right] - \|\mathbb{E}\left[X\right]\|^2$, after rearranging, we obtain from the last expression that

$$\begin{aligned}\mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] &= \left(1 - \tfrac{1}{b_t}\right)\|\nabla f(w_t) - \nabla f(w_{t-1})\|^2\\ &\quad + \tfrac{1}{b_t}\mathbb{E}\left[\|\nabla_w \mathbf{f}(w_t;\xi) - \nabla_w \mathbf{f}(w_{t-1};\xi)\|^2 \mid \mathcal{F}_t\right],\end{aligned}$$

which is indeed (24). ■

Appendix B. The Proof of Technical Results in Section 3

We provide the full proof of the results in Section 3.

B.1 The Proof of Lemma 3: The Analysis of The Inner Loop

Proof From the update $w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t \widehat{w}_{t+1}^{(s)}$, we have $w_{t+1}^{(s)} - w_t^{(s)} = \gamma_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})$. Firstly, using the L-smoothness of f from (6) of Assumption 2.2, we can derive

$$\begin{aligned}f(w_{t+1}^{(s)}) &\le f(w_t^{(s)}) + \langle \nabla f(w_t^{(s)}), w_{t+1}^{(s)} - w_t^{(s)}\rangle + \tfrac{L}{2}\|w_{t+1}^{(s)} - w_t^{(s)}\|^2\\ &= f(w_t^{(s)}) + \gamma_t\langle \nabla f(w_t^{(s)}), \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle + \tfrac{L\gamma_t^2}{2}\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2.\end{aligned} \tag{52}$$

Next, using the convexity of ψ, one can show that

$$\psi(w_{t+1}^{(s)}) \le (1-\gamma_t)\psi(w_t^{(s)}) + \gamma_t\psi(\widehat{w}_{t+1}^{(s)}) \le \psi(w_t^{(s)}) + \gamma_t\langle \nabla\psi(\widehat{w}_{t+1}^{(s)}), \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle, \tag{53}$$

where $\nabla\psi(\widehat{w}_{t+1}^{(s)}) \in \partial\psi(\widehat{w}_{t+1}^{(s)})$.

By the optimality condition of $\widehat{w}_{t+1}^{(s)} := \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})$, we have $\nabla\psi(\widehat{w}_{t+1}^{(s)}) = -v_t^{(s)} - \frac{1}{\eta_t}(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})$ for some $\nabla\psi(\widehat{w}_{t+1}^{(s)}) \in \partial\psi(\widehat{w}_{t+1}^{(s)})$. Substituting this expression into (53), we obtain

$$\psi(w_{t+1}^{(s)}) \le \psi(w_t^{(s)}) + \gamma_t\langle v_t^{(s)}, w_t^{(s)} - \widehat{w}_{t+1}^{(s)}\rangle - \frac{\gamma_t}{\eta_t}\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2. \tag{54}$$

Combining (52) and (54), and then using $F(w) := f(w) + \psi(w)$ yields

$$F(w_{t+1}^{(s)}) \le F(w_t^{(s)}) + \gamma_t\langle \nabla f(w_t^{(s)}) - v_t^{(s)}, \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle - \Big(\frac{\gamma_t}{\eta_t} - \frac{L\gamma_t^2}{2}\Big)\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2. \tag{55}$$

Also, the following expression holds

$$\begin{aligned}\langle \nabla f(w_t^{(s)}) - v_t^{(s)}, \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle &= \tfrac{1}{2}\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2 + \tfrac{1}{2}\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\\ &\quad - \tfrac{1}{2}\|\nabla f(w_t^{(s)}) - v_t^{(s)} - (\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\|^2.\end{aligned}$$

From this expression, we can rewrite (55) as

$$F(w_{t+1}^{(s)}) \le F(w_t^{(s)}) + \frac{\gamma_t}{2}\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2 - \Big(\frac{\gamma_t}{\eta_t} - \frac{L\gamma_t^2}{2} - \frac{\gamma_t}{2}\Big)\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2 - \sigma_t^{(s)},$$

where $\sigma_t^{(s)} := \frac{\gamma_t}{2}\|\nabla f(w_t^{(s)}) - v_t^{(s)} - (\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\|^2 \ge 0$.

Taking expectation both sides of this inequality over the entire history, we obtain

$$\begin{aligned}\mathbb{E}\Big[F(w_{t+1}^{(s)})\Big] &\le \mathbb{E}\Big[F(w_t^{(s)})\Big] + \tfrac{\gamma_t}{2}\mathbb{E}\Big[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\Big]\\ &\quad - \Big(\tfrac{\gamma_t}{\eta_t} - \tfrac{L\gamma_t^2}{2} - \tfrac{\gamma_t}{2}\Big)\mathbb{E}\Big[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\Big] - \mathbb{E}\Big[\sigma_t^{(s)}\Big].\end{aligned} \tag{56}$$

Next, recall from (10) that $G_\eta(w) := \frac{1}{\eta}\big(w - \mathrm{prox}_{\eta\psi}(w - \eta\nabla f(w))\big)$ is the gradient mapping of F. In this case, it is obvious that

$$\eta_t\|G_{\eta_t}(w_t^{(s)})\| = \|w_t^{(s)} - \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)}))\|.$$

Using this definition, the triangle inequality, and the nonexpansive property $\|\mathrm{prox}_{\eta\psi}(z) - \mathrm{prox}_{\eta\psi}(w)\| \le \|z - w\|$ of $\mathrm{prox}_{\eta\psi}$, we can derive that

$$\begin{aligned}\eta_t\|G_{\eta_t}(w_t^{(s)})\| &\le \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \|\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)})) - \widehat{w}_{t+1}^{(s)}\|\\ &= \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \|\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)})) - \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})\|\\ &\le \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \eta_t\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|.\end{aligned}$$

Now, the last estimate leads to

$$\eta_t^2\mathbb{E}\Big[\|G_{\eta_t}(w_t^{(s)})\|^2\Big] \le 2\mathbb{E}\Big[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\Big] + 2\eta_t^2\mathbb{E}\Big[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\Big].$$

Multiplying this inequality by $\frac{\gamma_t}{2} > 0$ and adding the result to (56), we finally get

$$\begin{aligned}\mathbb{E}\Big[F(w_{t+1}^{(s)})\Big] &\le \mathbb{E}\Big[F(w_t^{(s)})\Big] - \tfrac{\gamma_t\eta_t^2}{2}\mathbb{E}\Big[\|G_{\eta_t}(w_t^{(s)})\|^2\Big]\\ &\quad + \tfrac{\gamma_t}{2}\left(1 + 2\eta_t^2\right)\mathbb{E}\Big[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\Big]\\ &\quad - \tfrac{\gamma_t}{2}\left(\tfrac{2}{\eta_t} - L\gamma_t - 3\right)\mathbb{E}\Big[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\Big] - \mathbb{E}\Big[\sigma_t^{(s)}\Big].\end{aligned}$$

Summing up this inequality from $t = 0$ to $t = m$, we obtain

$$\begin{aligned}\mathbb{E}\Big[F(w_{m+1}^{(s)})\Big] &\le \mathbb{E}\Big[F(w_0^{(s)})\Big] + \frac{1}{2}\sum_{t=0}^{m}\gamma_t\left(1 + 2\eta_t^2\right)\mathbb{E}\Big[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\Big]\\ &\quad - \frac{1}{2}\sum_{t=0}^{m}\gamma_t\Big(\frac{2}{\eta_t} - L\gamma_t - 3\Big)\mathbb{E}\Big[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\Big]\\ &\quad - \sum_{t=0}^{m}\frac{\gamma_t\eta_t^2}{2}\mathbb{E}\Big[\|G_{\eta_t}(w_t^{(s)})\|^2\Big] - \sum_{t=0}^{m}\mathbb{E}\Big[\sigma_t^{(s)}\Big].\end{aligned} \tag{57}$$

We consider two cases:

Case 1: In the finite-sum setting (2), i.e., Algorithm 1 solves (2), then from (23) of Lemma 2, the L-smoothness condition (4) in Assumption 2.2, the choice $\hat{b}_t^{(s)} = \hat{b} \geq 1$, and $w_j^{(s)} - w_{j-1}^{(s)} = \gamma_{j-1}(\widehat{w}_j^{(s)} - w_{j-1}^{(s)})$, we can estimate

$$\begin{aligned}\mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2 \mid \mathcal{F}_j\right] &\overset{(23)}{=} \tfrac{n(\hat{b}-1)}{\hat{b}(n-1)}\|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 \\ &+ \tfrac{n-\hat{b}}{\hat{b}(n-1)}\tfrac{1}{n}\textstyle\sum_{i=1}^n \|\nabla f_i(w_j^{(s)}) - \nabla f_i(w_{j-1}^{(s)})\|^2 \\ &\overset{(4)}{\leq} \|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 + \tfrac{(n-\hat{b})L^2}{\hat{b}(n-1)}\|w_j^{(s)} - w_{j-1}^{(s)}\|^2 \\ &= \|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 + \tfrac{(n-\hat{b})L^2\gamma_{j-1}^2}{\hat{b}(n-1)}\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2.\end{aligned}$$

Case 2: In the expectation setting (1), i.e., Algorithm 1 solves (1), then from (24) of Lemma 2, we have

$$\begin{aligned}\mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2 \mid \mathcal{F}_j\right] &\overset{(24)}{=} \left(1 - \tfrac{1}{\hat{b}}\right)\|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 \\ &+ \tfrac{1}{\hat{b}}\mathbb{E}\left[\|\nabla_w \mathbf{f}(w_j;\xi) - \nabla_w \mathbf{f}(w_{j-1};\xi)\|^2 \mid \mathcal{F}_j\right] \\ &\overset{(3)}{\leq} \|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 + \tfrac{L^2}{\hat{b}}\|w_j^{(s)} - w_{j-1}^{(s)}\|^2 \\ &= \|\nabla f(w_j) - \nabla f(w_{j-1})\|^2 + \tfrac{L^2\gamma_{j-1}^2}{\hat{b}}\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2.\end{aligned}$$

Using either one of the two last inequalities and (19), then taking the full expectation, we can derive

$$\begin{aligned}\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] &= \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \textstyle\sum_{j=1}^t \mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2\right] \\ &- \textstyle\sum_{j=1}^t \mathbb{E}\left[\|\nabla f(w_j) - \nabla f(w_{j-1})\|^2\right] \\ &\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \rho L^2 \textstyle\sum_{j=1}^t \gamma_{j-1}^2 \mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\ &= \bar{\sigma}^{(s)} + \rho L^2 \textstyle\sum_{j=1}^t \gamma_{j-1}^2 \mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right],\end{aligned} \tag{58}$$

where $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \geq 0$, and $\rho := \frac{1}{\hat{b}}$ if Algorithm 1 solves (1), and $\rho := \frac{n-\hat{b}}{\hat{b}(n-1)}$ if Algorithm 1 solves (2).

Substituting (58) into (57) and dropping the term $-\sum_{t=0}^m \mathbb{E}\left[\sigma_t^{(s)}\right]$ (≤ 0), we finally arrive at

$$\begin{aligned}\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \tfrac{\rho L^2}{2}\sum_{t=0}^m \gamma_t\left(1 + 2\eta_t^2\right)\sum_{j=1}^t \gamma_{j-1}^2 \mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\ &- \frac{1}{2}\sum_{t=0}^m \gamma_t\left(\frac{2}{\eta_t} - L\gamma_t - 3\right)\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \\ &- \sum_{t=0}^m \frac{\gamma_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] + \frac{1}{2}\sum_{t=0}^m \gamma_t\left(1 + 2\eta_t^2\right)\bar{\sigma}^{(s)},\end{aligned}$$

which is exactly (26). ■

B.2 The Proof of Lemma 4: The Selection of Constant Step-sizes

Proof Let us first fix all the step-sizes of Algorithm 1 as constants as follows:

$$\gamma_t := \gamma \in (0, 1] \quad \text{and} \quad \eta_t := \eta > 0.$$

We also denote $a_t^{(s)} := \mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \geq 0$.

Let $\rho := \frac{1}{\hat{b}}$ if Algorithm 1 solves (1) and $\rho := \frac{n-\hat{b}}{\hat{b}(n-1)}$ if Algorithm 1 solves (2). Using these expressions into (26), we can easily show that

$$\begin{aligned}
\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \tfrac{\rho L^2\gamma^3}{2}\left(1+2\eta^2\right)\sum_{t=0}^{m}\sum_{j=1}^{t} a_{j-1}^{(s)} \\
&\quad - \tfrac{\gamma}{2}\left(\tfrac{2}{\eta} - L\gamma - 3\right)\sum_{t=0}^{m} a_t^{(s)} - \tfrac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] \\
&\quad + \tfrac{\gamma}{2}\left(1+2\eta^2\right)(m+1)\bar{\sigma}^{(s)} \\
&= \mathbb{E}\left[F(w_0^{(s)})\right] - \tfrac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] \\
&\quad + \tfrac{\gamma}{2}\left(1+2\eta^2\right)(m+1)\bar{\sigma}^{(s)} + \mathcal{T}_m,
\end{aligned} \tag{59}$$

where $\mathcal{T}_m$ is defined as

$$\mathcal{T}_m := \frac{\rho L^2\gamma^3\left(1+2\eta^2\right)}{2}\sum_{t=0}^{m}\sum_{j=1}^{t} a_{j-1}^{(s)} - \frac{\gamma}{2}\left(\frac{2}{\eta} - L\gamma - 3\right)\sum_{t=0}^{m} a_t^{(s)}.$$

Our goal is to choose $\eta > 0$, and $\gamma \in (0, 1]$ such that $\mathcal{T}_m \leq 0$. We first rewrite $\mathcal{T}_m$ as follows:

$$\begin{aligned}
\mathcal{T}_m &= \tfrac{\rho L^2\gamma^3\left(1+2\eta^2\right)}{2}\left[m a_0^{(s)} + (m-1)a_1^{(s)} + \cdots + 2a_{m-2}^{(s)} + a_{m-1}^{(s)}\right] \\
&\quad - \tfrac{\gamma}{2}\left(\tfrac{2}{\eta} - L\gamma - 3\right)\left[a_0^{(s)} + a_1^{(s)} + \cdots + a_m^{(s)}\right].
\end{aligned}$$

By synchronizing the coefficients of the terms $a_0^{(s)}, a_1^{(s)}, \cdots, a_m^{(s)}$, to guarantee $\mathcal{T}_m \leq 0$, we need to satisfy

$$\begin{cases}
\rho\left(1+2\eta^2\right)L^2\gamma^2 m - \left(\frac{2}{\eta} - L\gamma - 3\right) & \leq 0, \\
\frac{2}{\eta} - L\gamma - 3 & \geq 0.
\end{cases} \tag{60}$$

Assume that $\frac{2}{\eta} - L\gamma - 3 = 1 > 0$. This implies that $\eta = \frac{2}{L\gamma+4}$. Next, since $L\gamma > 0$, we have $\eta \leq \frac{1}{2}$. Therefore, we can upper bound

$$\rho L^2\gamma^2 m(1+2\eta^2) - \left(\frac{2}{\eta} - L\gamma - 3\right) \leq \frac{3\rho L^2\gamma^2 m}{2} - 1 = 0.$$

The last equation and $\eta = \frac{2}{L\gamma+4}$ lead to

$$\gamma := \frac{1}{L\sqrt{\omega m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\omega m}}{4\sqrt{\omega m}+1},$$

which is exactly (27), where $\omega := \frac{3(n-\hat{b})}{2\hat{b}(n-1)}$ for (2) and $\omega := \frac{3}{2\hat{b}}$ for (1).

Finally, using this choice (27) of the step-sizes, we can derive that

$$\mathbb{E}\left[F(w_{m+1}^{(s)}\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] - \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] + \frac{\gamma\theta}{2}(m+1)\bar{\sigma}^{(s)}, \tag{61}$$

which is exactly (28), where $\theta := 1 + 2\eta^2 \leq \frac{3}{2}$. ■

B.3 The Proof of Theorem 6: The Dynamic Step-size Case

Proof Let $\beta_t := \gamma_t\left(1+2\eta_t^2\right)$ and $\kappa_t := \gamma_t\left(\frac{2}{\eta_t} - L\gamma_t - 3\right)$. From (26) of Lemma 3 we have

$$\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] - \sum_{t=0}^{m}\frac{\gamma_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] + \frac{1}{2}\bar{\sigma}^{(s)}\Big(\sum_{t=0}^{m}\beta_t\Big) + \mathcal{T}_m, \tag{62}$$

where

$$\mathcal{T}_m := \frac{L^2(n-\hat{b})}{2\hat{b}(n-1)}\sum_{t=0}^{m}\beta_t\sum_{j=1}^{t}\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] - \frac{1}{2}\sum_{t=0}^{m}\kappa_t\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right].$$

Now, to guarantee $\mathcal{T}_m \leq 0$, let us choose all the parameters such that

$$\begin{cases} \kappa_m & = 0, \\ \frac{(n-\hat{b})}{\hat{b}(n-1)}L^2\gamma_t^2\sum_{j=t+1}^{m}\beta_j - \kappa_t & = 0, \quad t = 0, \cdots, m-1. \end{cases} \tag{63}$$

Then, (62) becomes

$$\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] - \sum_{t=0}^{m}\frac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] + \frac{1}{2}\sum_{t=0}^{m}\beta_t\bar{\sigma}^{(s)}. \tag{64}$$

If we fix $\eta_t = \eta \in (0, \frac{2}{3})$, and define $\delta := \frac{2}{\eta} - 3 > 0$, then (63) reduces to

$$\begin{cases} \delta - L\gamma_m & = 0, \\ \frac{L^2(n-\hat{b})(1+2\eta^2)}{\hat{b}(n-1)}\gamma_t\sum_{j=t+1}^{m}\gamma_j - \delta + L\gamma_t & = 0, \quad t = 0, \cdots, m-1. \end{cases} \tag{65}$$

Applying Lemma 12(a) with $\nu = \omega_\eta := \frac{(n-\hat{b})(1+2\eta^2)}{\hat{b}(n-1)}$, we obtain from (65) that

$$\gamma_m := \frac{\delta}{L}, \quad \text{and} \quad \gamma_t := \frac{\delta}{L\big[1 + \omega_\eta L\sum_{j=t+1}^{m}\gamma_j\big]}, \quad t = 0, \cdots, m-1. \tag{66}$$

Moreover, we have

$$\frac{\delta}{L(1+\omega_\eta\delta m)} < \gamma_0 < \gamma_1 < \cdots < \gamma_m, \quad \text{and} \quad \Sigma_m := \sum_{t=0}^{m}\gamma_t \geq \frac{2\delta(m+1)}{L(\sqrt{2\omega_\eta\delta m+1}+1)},$$

which proves (31).

On the other hand, since $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] = \mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_s}(\widetilde{w}_{s-1}) - \nabla f(\widetilde{w}_{s-1})\|^2\right]$, by using (21), we have $\bar{\sigma}^{(s)} \leq \left(\frac{n-b_s}{nb_s}\right)\sigma_n^2(\widetilde{w}_{s-1})$. Using this upper bound and $\beta_t := \gamma_t(1+2\eta^2) \leq \frac{3\gamma_t}{2} \leq \frac{3}{2}$ (since $\gamma_t \in [0,1]$), into the estimate (64), we can arrive at

$$\frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m}\gamma_t\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{2}{\eta^2 S\Sigma_m}\big[F(\widetilde{w}_0) - F^\star\big] + \frac{3}{2\eta^2 S}\sum_{s=1}^{S}\frac{(n-b_s)\sigma_n^2(\widetilde{w}_{s-1})}{nb_s},$$

which is exactly (32).

Now, let us choose $\eta := \frac{1}{2} \in (0, \frac{2}{3})$. Then, we have $\delta = 1$, $\omega_\eta = \frac{3(n-\hat{b})}{2\hat{b}(n-1)}$, and $\Sigma_m \geq \frac{2\delta(m+1)}{L(\sqrt{2\omega_\eta m+1}+1)}$. Using these facts, $\widetilde{w}_T \sim \mathbf{U}_p\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ with $\mathbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t} := \frac{\gamma_t}{S\Sigma_m}$, and $b_s = n$, we obtain from (32) that

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] = \frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m}\gamma_t\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{4L(\sqrt{2\omega m+1}+1)}{S(m+1)}\big[F(\widetilde{w}_0) - F^\star\big].$$

Next, using $m = \lfloor \frac{n}{\hat{b}} \rfloor$ and $\omega := \omega_\eta = \frac{3(n-\hat{b})}{2\hat{b}(n-1)}$, if $\hat{b} \leq \sqrt{n}$, then we can bound

$$\frac{\sqrt{2\omega m+1}+1}{m+1} \leq \frac{2\sqrt{\omega}}{\sqrt{m+1}} \leq \frac{\sqrt{6}}{\sqrt{n}}.$$

Using this bound, we can further bound the above estimate obtained from (32) as

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \frac{4\sqrt{6}L\left[F(\widetilde{w}_0) - F^\star\right]}{S\sqrt{n}},$$

which is (33).

To achieve $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we impose $\frac{4\sqrt{6}L[F(\widetilde{w}_0)-F^\star]}{S\sqrt{n}} = \varepsilon^2$, which shows that the number of outer iterations $S := \frac{4\sqrt{6}L[F(\widetilde{w}_0)-F^\star]}{\sqrt{n}\varepsilon^2}$. To guarantee $S \geq 1$, we need $n \leq \frac{96L^2[F(\widetilde{w}_0)-F^\star]^2}{\varepsilon^4}$.

Hence, we can estimate the number of gradient evaluations $\mathcal{T}_{\text{grad}}$ by

$$\mathcal{T}_{\text{grad}} = Sn + 2S(m+1)\hat{b} \leq 5Sn = \frac{20\sqrt{6}L\sqrt{n}\left[F(\widetilde{w}_0) - F^\star\right]}{\varepsilon^2}.$$

We can conclude that the number of stochastic gradient evaluations does not exceed $\mathcal{T}_{\text{grad}} = \mathcal{O}\left(\frac{L\sqrt{n}[F(\widetilde{w}_0)-F^\star]}{\varepsilon^2}\right)$. The number of proximal operations $\text{prox}_{\eta\psi}$ does not exceed $\mathcal{T}_{\text{prox}} := S(m+1) \leq \frac{4\sqrt{6}(\sqrt{n}+1)L[F(\widetilde{w}_0)-F^\star]}{\hat{b}\varepsilon^2}$. ■

B.4 The Proof of Theorem 8: The Constant Step-size Case

Proof If we choose $(\gamma_t, \eta_t) = (\gamma, \eta) > 0$ for all $t = 0, \cdots, m$, then, by applying Lemma 4, we can update

$$\gamma := \frac{1}{L\sqrt{\omega m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\omega m}}{4\sqrt{\omega m} + 1},$$

which is exactly (34), where $\omega := \frac{3(n-\hat{b})}{2(n-1)\hat{b}}$. With this update, we can simplify (28) as

$$\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \le \mathbb{E}\left[F(w_0^{(s)})\right] - \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] + \frac{3\gamma}{4}(m+1)\bar{\sigma}^{(s)}.$$

With the same argument as above, we obtain

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \le \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^\star\big] + \frac{3}{2\eta^2 S}\sum_{s=1}^{S}\frac{(n-b_s)\sigma_n^2(\widetilde{w}_{s-1})}{nb_s}.$$

For $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ with $T := (m+1)S$ and $b_s = n$, the last estimate implies

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] = \frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \le \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^\star\big].$$

By the update rule of η and γ, we can easily show that $\gamma\eta^2 \ge \frac{4\sqrt{\omega m}}{L(4\sqrt{\omega m}+1)^2}$. Therefore, using $m := \lfloor \frac{n}{\hat{b}} \rfloor$, we can overestimate

$$\frac{1}{\gamma\eta^2(m+1)} \le \frac{L(4\sqrt{\omega m}+1)^2}{4\sqrt{\omega m}(m+1)} \le \frac{8L\sqrt{\omega}}{\sqrt{m}} \le \frac{8\sqrt{3}L}{\sqrt{2n}}.$$

Using this upper bound, to guarantee $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \le \varepsilon^2$, we choose S and m such that $\frac{16\sqrt{3}L}{S\sqrt{2n}}\big[F(\widetilde{w}_0) - F^\star\big] = \varepsilon^2$, which leads to $S := \frac{16\sqrt{3}L}{\sqrt{2n}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big]$ as the number of outer iterations. To guarantee $S \ge 1$, we need to choose $n \le \frac{384L^2}{\varepsilon^4}\big[F(\widetilde{w}_0) - F^\star\big]^2$.

Finally, we can estimate the number of stochastic gradient evaluations $\mathcal{T}_{\text{grad}}$ as

$$\mathcal{T}_{\text{grad}} = Sn + 2S(m+1) \le 5Sn = \frac{16\sqrt{3}L\sqrt{n}}{\sqrt{2}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big] = \mathcal{O}\left(\frac{L\sqrt{n}}{\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big]\right).$$

The number of $\text{prox}_{\eta\psi}$ is $\mathcal{T}_{\text{prox}} = S(m+1) \le \frac{16\sqrt{3}L(\sqrt{n}+1)}{\hat{b}\sqrt{2}\varepsilon^2}\big[F(\widetilde{w}_0) - F^\star\big]$. ■

B.5 The Proof of Theorem 9: The Expectation Problem

Proof Summing up (28) from $s = 1$ to $s = S$, and then using $w_0^{(0)} = \widetilde{w}_0$, we obtain

$$\frac{\gamma\eta^2}{2}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \le F(\widetilde{w}_0) - \mathbb{E}\left[F(w_{m+1}^{(S)})\right] + \frac{\gamma\theta(m+1)}{2}\sum_{s=1}^{S}\bar{\sigma}^{(s)}. \tag{67}$$

Note that $\mathbb{E}\Big[F(w^{(S)}_{m+1})\Big] \geq F^\star$ by Assumption 2.1. Moreover, by (20), we have

$$\bar{\sigma}^{(s)} := \mathbb{E}\Big[\|v^{(s)}_0 - \nabla f(w^{(s)}_0)\|^2\Big] = \mathbb{E}\Big[\|\widetilde{\nabla} f_{\mathcal{B}_s}(w^{(s)}_0) - \nabla f(w^{(s)}_0)\|^2\Big] \leq \frac{\sigma^2}{b_s} = \frac{\sigma^2}{b}.$$

Recall that $\rho := \frac{1}{\hat{b}}$ for (1). Therefore, we have $\theta = 1 + \frac{8\bar{\omega}m}{(1+4\sqrt{\bar{\omega}m})^2} < \frac{3}{2}$, where $\bar{\omega} := \frac{3}{2\hat{b}}$. Using these estimates into (67), we obtain (39).

Now, since $\widetilde{w}_T \sim \mathbf{U}\big(\{w^{(s)}_t\}^{s=1\to S}_{t=0\to m}\big)$ for $T := S(m+1)$, we have

$$\begin{aligned}\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] &= \frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w^{(s)}_t)\|^2\right]\\ &\leq \frac{2}{\gamma\eta^2(m+1)S}[F(\widetilde{w}_0) - F^\star] + \frac{3\sigma^2}{2\eta^2 b}.\end{aligned}$$

Since $\eta = \frac{2\sqrt{\bar{\omega}m}}{4\sqrt{\bar{\omega}m}+1} \geq \frac{2}{5}$ and $\frac{1}{\gamma\eta^2(m+1)} \leq \frac{25L\sqrt{\bar{\omega}m}}{4(m+1)} \leq \frac{8L}{\sqrt{\hat{b}m}}$ as proved above, to guarantee $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we need to set

$$\frac{16L}{S\sqrt{\hat{b}m}}[F(\widetilde{w}_0) - F^\star] + \frac{75\sigma^2}{8b} = \varepsilon^2.$$

Let us choose b such that $\frac{75\sigma^2}{8b} = \frac{\varepsilon^2}{2}$, which leads to $b := \frac{75\sigma^2}{8\varepsilon^2}$. We also choose $m := \frac{\sigma^2}{\hat{b}\varepsilon^2}$. To guarantee $m \geq 1$, we have $\hat{b} \leq \frac{\sigma^2}{\varepsilon^2}$. Then, since $\frac{1}{\sqrt{\hat{b}m}} = \frac{\varepsilon}{\sigma}$, the above condition is equivalent to $\frac{16L\varepsilon}{S\sigma}[F(\widetilde{w}_0) - F^\star] = \frac{\varepsilon^2}{2}$, which leads to

$$S := \frac{32L}{\sigma\varepsilon}[F(\widetilde{w}_0) - F^\star].$$

To guarantee $S \geq 1$, we need to choose $\varepsilon \leq \frac{32L}{\sigma}[F(\widetilde{w}_0) - F^\star]$ if σ is sufficiently large.

Now, we estimate the total number of stochastic gradient evaluations as

$$\begin{aligned}\mathcal{T}_{\text{grad}} &= \textstyle\sum_{s=1}^{S} b_s + 2m\hat{b}S = (b + 2m\hat{b})S = \frac{32L}{\sigma\varepsilon}[F(\widetilde{w}_0) - F^\star]\left(\frac{75\sigma^2}{\varepsilon^2} + \frac{2\sigma^2}{\hat{b}\varepsilon^2}\hat{b}\right)\\ &= \frac{2464L\sigma}{\varepsilon^3}[F(\widetilde{w}_0) - F^\star].\end{aligned}$$

Hence, the number of gradient evaluations is $\mathcal{O}\left(\frac{L\sigma[F(\widetilde{w}_0)-F^\star]}{\varepsilon^3}\right)$, and the number of proximal operator calls is also $\mathcal{T}_{\text{prox}} := S(m+1) = \frac{32\sigma L}{\hat{b}\varepsilon^2}[F(\widetilde{w}_0) - F^\star]$. ■

Appendix C. The Proof of Theorem 11: The Non-Composite Cases

Proof Since $\psi = 0$, we have $\widehat{w}^{(s)}_{t+1} = w^{(s)}_t - \eta_t v^{(s)}_t$. Therefore, $\widehat{w}^{(s)}_{t+1} - w^{(s)}_t = -\eta_t v^{(s)}_t$ and $w^{(s)}_{t+1} = (1-\gamma_t)w^{(s)}_t + \gamma_t\widehat{w}^{(s)}_{t+1} = w^{(s)}_t - \gamma_t\eta_t v^{(s)}_t = w^{(s)}_t - \hat{\eta}_t v^{(s)}_t$, where $\hat{\eta}_t := \gamma_t\eta_t$. Using these relations and choose $c_t = \frac{1}{\eta_t}$, we can easily show that

$$\begin{cases}\mathbb{E}\left[\|\widehat{w}^{(s)}_{t+1} - w^{(s)}_t\|^2\right] = \eta_t^2\mathbb{E}\left[\|v^{(s)}_t\|^2\right],\\ \sigma^{(s)}_t := \frac{\gamma_t}{2c_t}\|\nabla f(w^{(s)}_t) - v^{(s)}_t - c_t(\widehat{w}^{(s)}_{t+1} - w^{(s)}_t)\|^2 = \frac{\hat{\eta}_t}{2}\|\nabla f(w^{(s)}_t) - v^{(s)}_t\|^2.\end{cases}$$

Substituting these estimates into (56) and noting that $f = F$ and $\hat{\eta}_t := \gamma_t\eta_t$, we obtain

$$\begin{aligned}\mathbb{E}\left[f(w_{t+1}^{(s)})\right] &\le \mathbb{E}\left[f(w_t^{(s)})\right] + \tfrac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right]\\ &\quad - \tfrac{\hat{\eta}_t}{2}\big(1 - L\hat{\eta}_t\big)\mathbb{E}\left[\|v_t^{(s)}\|^2\right] - \tfrac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right].\end{aligned} \tag{68}$$

On the other hand, from (19), by Assumption 2.2, (16), and $w_{t+1}^{(s)} := w_t^{(s)} - \hat{\eta}_t v_t^{(s)}$, we can derive

$$\begin{aligned}\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] &\le \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \textstyle\sum_{j=1}^t \mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2\right]\\ &\le \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]\\ &\quad + \rho\textstyle\sum_{j=1}^t \mathbb{E}\left[\|\nabla_w \mathbf{f}(w_j^{(s)};\xi_j^{(s)}) - \nabla_w \mathbf{f}(w_{j-1}^{(s)};\xi_j^{(s)})\|^2\right]\\ &\le \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \rho L^2\textstyle\sum_{j=1}^t \mathbb{E}\left[\|w_j^{(s)} - w_{j-1}^{(s)}\|^2\right]\\ &\le \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \rho L^2\textstyle\sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right],\end{aligned}$$

where $\rho := \frac{1}{\hat{b}}$ if Algorithm 1 solves (1) and $\rho := \frac{n-\hat{b}}{\hat{b}(n-1)}$ if Algorithm 1 solves (2).

Substituting this estimate into (68), and summing up the result from $t = 0$ to $t = m$, we eventually get

$$\begin{aligned}\mathbb{E}\left[f(w_{m+1}^{(s)})\right] &\le \mathbb{E}\left[f(w_0^{(s)})\right] - \sum_{t=0}^m \frac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] + \frac{1}{2}\Big(\sum_{t=0}^m \hat{\eta}_t\Big)\mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]\\ &\quad + \frac{\rho L^2}{2}\sum_{t=0}^m \hat{\eta}_t\sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right] - \sum_{t=0}^m \frac{\hat{\eta}_t(1 - L\hat{\eta}_t)}{2}\mathbb{E}\left[\|v_t^{(s)}\|^2\right].\end{aligned} \tag{69}$$

Our next step is to choose $\hat{\eta}_t$ such that

$$\rho L^2\sum_{t=0}^m \hat{\eta}_t\sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right] - \sum_{t=0}^m \hat{\eta}_t(1 - L\hat{\eta}_t)\mathbb{E}\left[\|v_t^{(s)}\|^2\right] \le 0.$$

This condition can be rewritten explicitly as

$$\begin{aligned}&\big[\rho L^2\hat{\eta}_0^2(\hat{\eta}_1 + \cdots + \hat{\eta}_m) - \hat{\eta}_0(1 - L\hat{\eta}_0)\big]\mathbb{E}\left[\|v_0^{(s)}\|^2\right]\\ &+ \big[\rho L^2\hat{\eta}_1^2(\hat{\eta}_2 + \cdots + \hat{\eta}_m) - \hat{\eta}_1(1 - L\hat{\eta}_1)\big]\mathbb{E}\left[\|v_1^{(s)}\|^2\right] + \cdots\\ &+ \big[\rho L^2\hat{\eta}_{m-1}^2\hat{\eta}_m - \hat{\eta}_{m-1}(1 - L\hat{\eta}_{m-1})\big]\mathbb{E}\left[\|v_{m-1}^{(s)}\|^2\right] - \hat{\eta}_m(1 - L\hat{\eta}_m)\mathbb{E}\left[\|v_m^{(s)}\|^2\right] \le 0.\end{aligned}$$

Similar to (48), to guarantee the last inequality, we impose the following conditions

$$\begin{cases} -\hat{\eta}_m(1 - L\hat{\eta}_m) & \le 0,\\ \rho L^2\hat{\eta}_t^2\sum_{j=t+1}^m \hat{\eta}_j - \hat{\eta}_0(1 - L\hat{\eta}_0) & \le 0.\end{cases} \tag{70}$$

Applying Lemma 48 (a) with $\nu = \rho$ and $\delta = 1$, we obtain

$$\hat{\eta}_m = \frac{1}{L}, \quad \text{and} \quad \hat{\eta}_{m-t} := \frac{1}{L\big(1 + \rho L \sum_{j=1}^{t} \hat{\eta}_{m-j+1}\big)}, \quad \forall t = 1, \cdots, m,$$

which is exactly (41). With this update, we have $\frac{1}{L(1+\rho m)} < \hat{\eta}_0 < \hat{\eta}_1 < \cdots < \hat{\eta}_m$ and $\Sigma_m \geq \frac{2(m+1)}{L(\sqrt{2\rho m+1}+1)}$.

Using the update (41), we can simplify (69) as follows:

$$\mathbb{E}\left[f(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[f(w_0^{(s)})\right] - \sum_{t=0}^{m} \frac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] + \frac{\sum_{t=0}^{m}\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right].$$

Let us define $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]$ and noting that $f^\star := F^\star \leq \mathbb{E}\left[f(w_{m+1}^{(S)})\right]$ and $\widetilde{w}_0 := w_0^{(0)}$. Summing up the last inequality from $s = 1$ to S and using these relations, we can further derive

$$\sum_{s=1}^{S}\sum_{t=0}^{m} \hat{\eta}_t \mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \leq 2\big[f(\widetilde{w}_0) - f^\star\big] + \Big(\sum_{t=0}^{m}\hat{\eta}_t\Big)\sum_{s=1}^{S}\hat{\sigma}_s.$$

Using the lower bound of Σ_m as $\Sigma_m \geq \frac{2(m+1)}{L(\sqrt{2\rho m+1}+1)}$, the above inequality leads to

$$\frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m} \hat{\eta}_t \mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \leq \frac{(\sqrt{2\rho m+1}+1)L}{S(m+1)}\big[f(\widetilde{w}_0) - f^\star\big] + \frac{1}{S}\sum_{s=1}^{S}\hat{\sigma}_s. \tag{71}$$

Since $\mathbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t}$ with $p_{(s-1)m+t} = \frac{\hat{\eta}_t}{S\Sigma_m}$ for $s = 1, \cdots, S$ and $t = 0, \cdots, m$, we have

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] = \frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m}\hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right].$$

Substituting this estimate into (71), we obtain (42).

Now, we consider two cases:

Case (a): If we apply this algorithm variant to solve the non-composite finite-sum problem of (2) (i.e., $\psi = 0$) using the full-gradient snapshot for the outer-loop with $b_s = n$, then $v_0^{(s)} = \nabla f(w_0^{(s)})$, which leads to $\hat{\sigma}_s = 0$. By the choice of epoch length $m = \lfloor \frac{n}{\hat{b}} \rfloor$ and $\hat{b} \leq \sqrt{n}$, we have $\frac{\sqrt{2\rho m+1}+1}{m+1} \leq \frac{2}{\sqrt{n}}$. Using these facts into (42), we obtain

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{2L}{S\sqrt{n}}\big[f(\widetilde{w}_0) - f^\star\big],$$

which is exactly (43).

To achieve $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we impose $\frac{2L}{S\sqrt{n}}\big[f(\widetilde{w}_0) - f^\star\big] = \varepsilon^2$. Hence, the maximum number of outer iterations is at most $S = \frac{2L}{\sqrt{n}\varepsilon^2}[f(\widetilde{w}_0) - f^\star]$. The number of gradient evaluations ∇f_i is at most $\mathcal{T}_{\text{grad}} := nS + 2(m+1)\hat{b}S \leq 5nS = \frac{10L\sqrt{n}}{\varepsilon^2}[f(\widetilde{w}_0) - f^\star]$.

Case (b): Let us apply this algorithm variant to solve the non-composite expectation problem of (1) (i.e., $\psi = 0$). Then, by using $\rho := \frac{1}{\hat{b}}$ and $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \leq \frac{\sigma^2}{b_s} = \frac{\sigma^2}{b}$, we have from (42) that

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{2L}{S\sqrt{\hat{b}m}}\big[f(\widetilde{w}_0) - f^\star\big] + \frac{\sigma^2}{b}.$$

This is exactly (44). Using the mini-batch $b := \frac{2\sigma^2}{\varepsilon^2}$ for the outer-loop and $m := \frac{\sigma^2}{\hat{b}\varepsilon^2}$, we can show that the number of outer iterations $S := \frac{4L}{\sigma\varepsilon}\big[f(\widetilde{w}_0) - f^\star\big]$. The number of stochastic gradient evaluations is at most $\mathcal{T}_{\rm grad} := Sb + 2S(m+1)\hat{b} = \frac{4S\sigma^2}{\varepsilon^2} = \frac{16L\sigma}{\varepsilon^3}\big[f(\widetilde{w}_0) - f^\star\big]$. This holds if $\frac{2\sigma^2}{\varepsilon^2} \leq \frac{4S\sigma^2}{\varepsilon^2} = \frac{16L\sigma}{\varepsilon^3}\big[f(\widetilde{w}_0) - f^\star\big]$ leading to $\sigma \leq \frac{8L}{\varepsilon}\big[f(\widetilde{w}_0) - f^\star\big]$. ■

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 99:1–1, 2010.

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1200–1205, Montreal, Canada, 2017.

Z. Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2675–26860, Montreal, Canada, 2018.

Z. Allen-Zhu and Y. Li. NEON2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3720–3730, Montreal, Canada, 2018.

Z. Allen-Zhu and Y. Yuan. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In *International Conference on Machine Learning (ICML)*, pages 1080–1089, New York, USA, 2016.

Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv:1912.02365*, 2019.

H. H. Bauschke and P. Combettes. *Convex Analysis and Monotone Operators Theory in Hilbert Spaces*. Springer-Verlag, 2nd edition, 2017.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics (COMPSTAT)*, pages 177–186, Paris, France, 2010.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review (SIREV)*, 60(2):223–311, 2018.

L. Bottou. Online learning and stochastic approximations. In David Saad, editor, *Online Learning in Neural Networks*, pages 9–42. Cambridge University Press, Cambridge, UK, 1998.

A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization (SIOPT)*, 28(4):2783–2808, 2018.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1646–1654, Montreal, Canada, 2014.

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

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on Learning Theory (COLT)*, pages 728–763, Paris, France 2015.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization: A generic algorithmic framework. *SIAM Journal on Optimization (SIOPT)*, 22(4):1469–1492, 2012.

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

S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT Press, 2016.

R. Harikandeh, M. O. Ahmed, A. Virani, M. Schmidt, J. Konečnỳ, and S. Sallinen. Stop-wasting my gradients: Practical SVRG. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2251–2259, Montreal, Canada, 2015.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, Lake Tahoe, NV, USA, 2013.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811, Riva del Garda, Italy, 2016.

L. Lei and M. Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In Aarti Singh and Jerry Zhu, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 54:148–156, Fort Lauderdale, FL, USA, 2017.

Z. Li and J. Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5564–5574, Montreal, Canada, 2018.

L. Lihua, C. Ju, J. Chen, and M. Jordan. Non-convex finite-sum optimization via SCSG methods. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2348–2358, Long Beach, CA, USA 2017.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3384–3392, Montreal, Canada, 2015.

S. L. Lohr. *Sampling: Design and Analysis*. Nelson Education, 2009.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization (SIOPT)*, 19(4): 1574–1609, 2009.

A. Nemirovskii and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

Y. Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning (ICML)*, PMLR 70:2613–2621, Sydney, Australia, 2017a.

L. M. Nguyen, N. H. Nguyen, D. T. Phan, J. R. Kalagnanam, and K. Scheinberg. When does stochastic gradient algorithm work well? *arXiv:1801.06159*, 2018a.

L. M. Nguyen, K. Scheinberg, and M. Takac. Inexact SARAH algorithm for stochastic optimization. *arXiv:1811.10105*, 2018b.

L. M. Nguyen, M. van Dijk, D. T. Phan, P. H. Nguyen, T.-W. Weng, and J. R. Kalagnanam. Optimal finite-sum smooth non-convex optimization with SARAH. *arXiv:1901.07648*, 2019.

L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takác. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv:1705.07261*, 2017b.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1574–1582, Montreal, Canada, 2014.

C. Paquette, H. Lin, , D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst for gradient-based nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 84:613–622, Lanzarote, Canary Islands, 2018.

S. J. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic Frank-Wolfe methods for non-convex optimization. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251, Monticello, IL, USA, 2016a.

S. J. Reddi, S. Sra, B. Póczos, and A. J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1145–1153, Barcelona, Spain, 2016b.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research (JMLR)*, 14:567–599, 2013.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning (ICML)*, PMLR 32(1):64–72, Beijing, China, 2014.

A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modelling and Theory*. SIAM, 2009.

S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. MIT Press, 2012.

Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh. SpiderBoost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems (NIPS)*, pages 2406–2416, Vancouver, Canada, 2019.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

L. Zhao, M. Mammadov, and J. Yearwood. From convex to nonconvex: a loss function analysis for binary classification. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1281–1288, Sydney Australia, 2010.

D. Zhou and Q. Gu. Lower bounds for smooth nonconvex finite-sum optimization. *International Conference on Machine Learning (ICML)*, PMLR 97:7574–7583, Long Beach, CA, USA, 2019.

D. Zhou, P. Xu, and Q. Gu. Stochastic nested variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3925–3936, Montreal, Canada, 2018.

Y. Zhou, Z. Wang, K. Ji, Y. Liang, and V. Tarokh. Momentum schemes with stochastic variance reduction for nonconvex composite optimization. *arXiv:1902.02715*, 2019.