Hybrid Stochastic Gradient Descent Algorithms for Stochastic Nonconvex Optimization

Quoc Tran-Dinh[†], Nhan H. Pham[†], Dzung T. Phan[‡], *and* Lam M. Nguyen[‡]

[†]Department of Statistics and Operations Research
The University of North Carolina at Chapel Hill, Chapel Hill, NC27599, USA.
quoctd@email.unc.edu, nhanph@live.unc.edu

[‡]IBM Research, Thomas J. Watson Research Center
Yorktown Heights, NY10598, USA.
phandu@us.ibm.com, LamNguyen.MLTD@ibm.com

May 14, 2019

Abstract

We introduce a hybrid stochastic estimator to design stochastic gradient algorithms for solving stochastic optimization problems. Such a hybrid estimator is a convex combination of two existing biased and unbiased estimators and leads to some useful property on its variance. We limit our consideration to a hybrid SARAH-SGD for nonconvex expectation problems. However, our idea can be extended to handle a broader class of estimators in both convex and nonconvex settings. We propose a new single-loop stochastic gradient descent algorithm that can achieve $\mathcal{O}\left(\max\left\{\sigma^3\varepsilon^{-1}, \sigma\varepsilon^{-3}\right\}\right)$-complexity bound to obtain an ε-stationary point under smoothness and σ^2-bounded variance assumptions. This complexity is better than $\mathcal{O}\left(\sigma^2\varepsilon^{-4}\right)$ often obtained in state-of-the-art SGDs when $\sigma < \mathcal{O}\left(\varepsilon^{-3}\right)$. We also consider different extensions of our method, including constant and adaptive step-size with single-loop, double-loop, and mini-batch variants. We compare our algorithms with existing methods on several datasets using two nonconvex models.

1 Introduction

Consider the following stochastic nonconvex optimization problem of the form:

$$\min_{x\in\mathbb{R}^p}\Big\{f(x) := \mathbb{E}_\xi\left[f(x;\xi)\right]\Big\}, \tag{1}$$

where $f(\cdot;\cdot):\mathbb{R}^p\times\Omega\to\mathbb{R}$ is a stochastic function defined such that for each $x\in\mathbb{R}^p$, $f(x;\cdot)$ is a random variable in a given probability space $(\Omega,\mathbb{P})$, while for each realization $\xi\in\Omega$, $f(\cdot;\xi)$ is smooth on $\mathbb{R}^p$; and $\mathbb{E}_\xi\left[f(x;\xi)\right]$ is the expectation of $f(x;\xi)$ w.r.t. ξ over Ω.

Our goals and assumptions: Since (1) is nonconvex, our goal in this paper is to develop a new class of stochastic gradient algorithms to find an ε-approximate stationary point $\widetilde{x}_T$ of (1) such that $\mathbb{E}\left[\|\nabla f(\widetilde{x}_T)\|^2\right] \le \varepsilon^2$ under mild assumptions as stated in Assumption 1.1.

Assumption 1.1. The objective function f of (1) satisfies the following conditions:

(a) (**Boundedness from below**) There exists a finite lower bound $f^\star := \inf_{x\in\mathbb{R}^p} f(x) > -\infty$.

(b) (L**-average smoothness**) The function $f(\cdot;\xi)$ is L-average smooth on $\mathbb{R}^p$, i.e. there exists $L\in(0,+\infty)$ such that

$$\mathbb{E}_\xi\left[\|\nabla f(x;\xi)-\nabla f(y;\xi)\|^2\right] \le L^2\|x-y\|^2, \quad \forall x,y\in\mathbb{R}^p. \tag{2}$$

(c) (**Bounded variance**) There exists $\sigma \in (0, \infty)$ such that

$$\mathbb{E}_\xi \left[\|\nabla f(x;\xi) - \nabla f(x)\|^2\right] \leq \sigma^2, \quad \forall x \in \mathbb{R}^p. \tag{3}$$

These assumptions are very standard in stochastic optimization methods [9, 13]. The L-average smoothness of f is weaker than the smoothness of f for each realization $\xi \in \Omega$. Note that our methods described in the sequel are also applicable to the finite-sum problem $\min_x \left\{f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$ as long as the above assumptions hold. However, we do not specify our methods to solve this problem. In this case, σ in (3) can be replaced by other alternatives, e.g., $\sigma_n^2 := \frac{1}{n}\sum_{i=1}^n \left[\|\nabla f_i(x)\|^2 - \|\nabla f(x)\|^2\right]$.

Our key idea: Different from existing methods, we introduce a convex combination of a biased and unbiased estimator of the gradient ∇f of f, which we call a *hybrid stochastic gradient* estimator. While the biased estimator exploited in this paper is SARAH in [15], the unbiased one can be any unbiased estimator. SARAH is a recursive biased and variance reduced estimator for ∇f. Combining it with an unbiased estimator allows us to reduce the bias and variance of the hybrid estimator. In this paper, we only focus on the standard stochastic estimator as an unbiased candidate.

Related work: Under Assumption 1.1, problem (1) covers a large number of applications in machine learning and data sciences. The stochastic gradient descent (SGD) method was first studied in [24], and becomes extremely popular in recent years. [13] seems to be the first work showing the convergence rates of robust SGD variants in the convex setting, while [14] provides an intensive complexity analysis for many optimization algorithms, including stochastic methods. Variance reduction methods have also been widely studied, see, e.g. [1, 5, 7, 10, 19, 21, 25, 26, 28]. In the nonconvex setting, [9] seems to be the first algorithm achieving $\mathcal{O}\left(\sigma^2\varepsilon^{-4}\right)$-complexity bound. Other researchers have also made significant progress in this direction, including [2, 3, 4, 8, 12, 16, 17, 18, 22, 27, 30]. A majority of these works, including [2, 3, 4, 12, 22], rely on SVRG estimator in order to obtain better complexity bounds. Hitherto, the complexity of SVRG-based methods remains worse than the best-known results, which is obtained in [8, 20, 27] via the SARAH estimator. However, as discussed in [20, 27], the method called SPIDER in [8, 12] does not practically perform well due to small step-size and its dependence on the reciprocal of the estimator's norm. [27] amends this issue by using a large constant step-size, but requires large mini-batch and does not consider the single sample case and single loop variants. [20] provides a more general framework to treat composite problems where it covers (1) as special case, but it does not consider the single loop as in SGDs.

Our contribution: To this end, our contribution can be summarized as follows:

(a) We propose a hybrid stochastic estimator for a stochastic gradient of a nonconvex function f in (1) by combining the SARAH estimator from [15] and any unbiased stochastic estimator such as SGD and SVRG. However, we only focus on the SGD estimator in this paper. We prove some key properties of this hybrid estimator that can be used to design new algorithms.

(b) We exploit our hybrid estimator to develop a single-loop SGD algorithm that can achieve an ε-stationary point $\widetilde{x}_m$ such that $\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \leq \varepsilon^2$ in at most $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^3\varepsilon^{-1}\right)$ stochastic gradient evaluations. This complexity significantly improves $\mathcal{O}\left(\sigma^2\varepsilon^{-4}\right)$ of SGD if $\sigma < \mathcal{O}\left(\varepsilon^{-3}\right)$. We extend our algorithm to a double loop variant, which requires $\mathcal{O}\left(\max\left\{\sigma\varepsilon^{-3}, \sigma^2\varepsilon^{-2}\right\}\right)$ stochastic gradient evaluations. This is the best-known complexity in the literature for stochastic gradient-type methods for solving (1).

(c) We also investigate other variants of our method, including adaptive step-sizes, and mini-batches. In all these cases, our methods achieve the best-known complexity bounds.

Let us emphasize the following points of our contribution. Firstly, although our single-loop method requires three gradients per iteration compared to standard SGDs, it can achieves better complexity bound. Secondly, it can be cast into a variance reduction method where it starts from a "good" approximation v_0 of $\nabla f(x^0)$, and aggressively reduces the variance. Thirdly, our

step-size is $\eta = \mathcal{O}\left(m^{-1/3}\right)$ which is larger than $\eta = \mathcal{O}\left(m^{-1/2}\right)$ in SGDs. Fourthly, the step-size of the adaptive variant is increasing instead of diminishing as in SGDs. Finally, our method achieves the same best-known complexity as in variance reduction methods studied in [8, 20, 27]. We believe that our approach can be extended to other estimators such as SVRG [10] and SAGA [7], and can be used for Hessians to develop second-order methods as well as to solve convex and composite problems.

Paper organization: The rest of this paper is organized as follows. Section 2 introduces our new hybrid stochastic estimator for the gradient of f and investigates its properties. Section 3 proposes a single-loop hybrid SGD-SARAH algorithm and its complexity analysis. It also considers a double-loop and mini-batch variants with rigorous complexity analysis. Section 4 provides two numerical examples to illustrate our methods and compares them with state-of-the-art methods. All the proofs and additional experiments can be found in the Supplementary Document.

Notation: We work with Euclidean spaces, $\mathbb{R}^p$, equipped with standard inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\|$. For a smooth function f (i.e., f is continuously differentiable), ∇f denotes its gradient. We use $\mathbf{U}_{\mathbf{p}}(\mathcal{S})$ to denote a distribution on $\mathcal{S}$ with probability $\mathbf{p}$. If $\mathbf{p}$ is uniform, then we simply use $\mathbf{U}(\mathcal{S})$. We also use $\mathcal{O}(\cdot)$ to present big-O notion in complexity theory, and $\sigma(\cdot)$ to denote a σ-field.

2 Hybrid stochastic gradient estimators

In this section, we propose new stochastic estimators for the gradient of a smooth function f.

Let u_t be an unbiased estimator of $\nabla f(x_t)$ formed by a realization ζ_t of ξ, i.e. $\mathbb{E}_{\zeta_t}[u_t] = \nabla f(x_t)$. We attempt to develop the following stochastic estimator for $\nabla f(x_t)$ in (1):

$$v_t := \beta_{t-1}v_{t-1} + \beta_{t-1}(\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)) + (1-\beta_{t-1})u_t, \tag{4}$$

where ξ_t and ζ_t are two independent realizations of ξ on Ω. Clearly, if $\beta_t = 0$, then we obtain a simple unbiased stochastic estimator, and $\beta_t = 1$, we obtain the SARAH estimator in [15]. We are interested in the case $\beta_t \in (0,1)$, in which we call v_t in (4) a **hybrid stochastic** estimator.

Note that we can rewrite v_t as

$$v_t := \beta_{t-1}\nabla f(x_t;\xi_t) + (1-\beta_{t-1})u_t + \beta_{t-1}(v_{t-1} - \nabla f(x_{t-1};\xi_t)).$$

The first two terms are two stochastic gradients estimated at x_t, while the third term is the difference $v_{t-1} - \nabla f(x_{t-1};\xi_t)$ of the previous estimator and a stochastic gradient at the previous iterate. Here, since $\beta_{t-1} \in (0,1)$, the main idea is to exploit more recent information than the old ones. In fact, the hybrid estimator v_t covers many other estimators, including SGD, SVRG, and SARAH. We can use one of the following two concrete unbiased estimators u_t of $\nabla f(x_t)$ as follows:

- **The SGD estimator:** $u_t := u_t^{\text{sgd}} = \nabla f(x_t;\zeta_t)$.
- **The SVRG estimator:** $u_t := u_t^{\text{svrg}} = \nabla f(\tilde{x}) + \nabla f(x_t;\zeta_t) - \nabla f(\tilde{x};\zeta_t)$, where $\nabla f(\tilde{x})$ is a full gradient evaluated at a given snapshot point $\tilde{x}$.

However, for the sake of presentation, we only focus on the SGD estimator $u_t := u_t^{\text{sgd}}$.

We first prove the following property of the estimator v_t showing how the variance is estimated.

Lemma 2.1. *Let v_t be defined by (4). Then*

$$\mathbb{E}_{(\xi_t,\zeta_t)}[v_t] = \nabla f(x_t) + \beta_{t-1}(v_{t-1} - \nabla f(x_{t-1})). \tag{5}$$

If $\beta_{t-1} \neq 0$, then v_t is a biased estimator. Moreover, we have

$$\begin{aligned}\mathbb{E}_{(\xi_t,\zeta_t)}\left[\|v_t - \nabla f(x_t)\|^2\right] &= \beta_{t-1}^2\|v_{t-1} - \nabla f(x_{t-1})\|^2 - \beta_{t-1}^2\|\nabla f(x_{t-1}) - \nabla f(x_t)\|^2\\ &+ \beta_{t-1}^2\mathbb{E}_{\xi_t}\left[\|\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)\|^2\right]\\ &+ (1-\beta_{t-1})^2\mathbb{E}_{\zeta_t}\left[\|u_t - \nabla f(x_t)\|^2\right].\end{aligned} \tag{6}$$

Remark 2.1. From (4), we can see that v_t remains a biased estimator as long as $\beta_{t-1} \in (0, 1]$. Its biased term is

$$\mathrm{Bias}(v_t) = \|\mathbb{E}_{(\xi_t,\zeta_t)}\left[v_t - \nabla f(x_t) \mid \mathcal{F}_t\right]\| = \beta_{t-1}\|v_{t-1} - \nabla f(x_{t-1})\| \leq \|v_{t-1} - \nabla f(x_{t-1})\|.$$

This shows that the bias v_t estimator is smaller than the one in the SARAH estimator $v_t^{\mathrm{sarah}} := v_{t-1}^{\mathrm{sarah}} + \nabla f(x_t; \xi_t) - \nabla f(x_{t-1}; \xi_t)$ from [15], which is $\mathrm{Bias}(v_t^{\mathrm{sarah}}) = \|v_{t-1}^{\mathrm{sarah}} - \nabla f(x_{t-1})\|$.

The following lemma bounds the second moment of $v_t - \nabla f(x_t)$ with v_t defined in (4).

Lemma 2.2. *Assume that $f(\cdot, \cdot)$ is L-smooth and u_t is an SGD estimator. Then, we have the following upper bound on the variance $\mathbb{E}\left[\|v_t - \nabla f(x_t)\|^2\right]$ of v_t:*

$$\mathbb{E}\left[\|v_t - \nabla f(x_t)\|^2\right] \leq \omega_t \mathbb{E}\left[\|v_0 - \nabla f(x^0)\|^2\right] + L^2 \sum_{i=0}^{t-1} \omega_{i,t} \mathbb{E}\left[\|x_{i+1} - x_i\|^2\right] + S_t, \tag{7}$$

where the expectation is taking over all the randomness $\mathcal{F}_t := \sigma(v_0, v_1, \cdots, v_t)$, $\omega_t := \prod_{i=1}^{t} \beta_{i-1}^2$, $\omega_{i,t} := \prod_{j=i+1}^{t} \beta_{j-1}^2$ for $i = 0, \cdots, t$, and $S_t := \sum_{i=0}^{t-1} \big(\prod_{j=i+2}^{t} \beta_{j-1}^2\big)(1 - \beta_i)^2 \sigma_{i+1}^2$ for $t \geq 0$.

Lemmas 2.1 and 2.2 provides two key properties to develop stochastic algorithm in Section 3.

3 Hybrid SARAH-SGD algorithms

In this section, we utilize our hybrid stochastic estimator v_t in (4) to develop stochastic gradient methods for solving (1). We consider three different variants using the hybrid SARAH-SGD estimator.

3.1 The generic algorithm framework

Using v_t defined by (4), we can develop a new algorithm for solving (1) as in Algorithm 1.

Algorithm 1 (Hybrid stochastic gradient descent (Hybrid-SGD) algorithm)

1: **Initialization:** An initial point x^0 and parameters b, β_t, and η_t (will be specified).
2: Generate an unbiased estimator $v_0 := \frac{1}{b}\sum_{\hat{\xi}_i \in \mathcal{B}} \nabla f(x_0; \hat{\xi}_i)$ at x_0 using a mini-batch $\mathcal{B}$.
3: Update $x_1 := x_0 - \eta_0 v_0$.
4: **For** $t := 1, \cdots, m$ **do**
5: Generate a proper sample pair (ξ_t, ζ_t) independently (single sample or mini-batch).
6: Evaluate $v_t := \beta_{t-1} v_{t-1} + \beta_{t-1}\big(\nabla f(x_t; \xi_t) - \nabla f(x_{t-1}; \xi_t)\big) + (1 - \beta_{t-1})\nabla f(x_t; \zeta_t)$.
7: Update $x_{t+1} := x_t - \eta_t v_t$.
8: **EndFor**
9: Choose $\widetilde{x}_m$ from $\{x_0, x_1, \cdots, x_m\}$ (at random or deterministic, specified later).

Algorithm 1 looks essentially the same as any SGD scheme with only one loop. The differences are at Step 3 with a mini-batch estimator v_0 and at Step 6, where we use our hybrid gradient estimator v_t. In addition, we will show in the sequel that it uses different step-sizes and leads to different variants. Unlike the inner loop of SARAH or SVRG, each iteration of Algorithm 1 requires three individual gradient evaluations instead of two as in these methods. The snapshot at Step 3 of Algorithm 1 relies on a mini-batch $\mathcal{B}$ of the size b, which is independent of (ξ_t, ζ_t) in the loop t.

3.2 Convergence analysis

We analyze two cases: constant step-size and adaptive step-size. In both cases, β_t is fixed for all t.

3.2.a Convergence of Algorithm 1 with constant step-size η and constant β
Assume that we run Algorithm 1 within m iterations $m \geq 1$. In this case, given $0 < c_1 < \sqrt{b(m+1)}$, we choose η and β in Algorithm 1 as follows:

$$\eta := \frac{2}{L(\sqrt{1+4\alpha_m^2}+1)} \quad \text{with} \quad \beta := 1 - \frac{c_1}{\sqrt{b(m+1)}} \quad \text{and} \quad \alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2}. \tag{8}$$

The following theorem estimates the complexity of Algorithm 1 to approximate an ε-stationary point of (1), whose proof is given in Subsection 2.2 of the supplementary document.

Theorem 3.1. *Let $\{x_t\}$ be the sequence generated by Algorithm 1 using the step-size η defined by (8). Let us choose $\widetilde{x}_m \sim \mathbf{U}(\{x_t\}_{t=0}^m)$. Then*

(a) *The step-size η satisfies $\eta \geq \underline{\eta} := \frac{2\sqrt{c_1}}{3L\left[b(m+1)\right]^{1/4}}$. In addition, we have*

$$\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \leq \frac{3b^{1/4}L\left[f(x^0)-f^\star\right]}{\sqrt{c_1}(m+1)^{3/4}} + \left(c_1 + \frac{1}{c_1}\right)\frac{\sigma^2}{\sqrt{b(m+1)}}. \tag{9}$$

(b) *If we choose $b := c_2\sigma^{8/3}(m+1)^{1/3}$ for any $c_2 > 0$, then to guarantee $\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \leq \varepsilon^2$, we need to choose*

$$m := \left\lfloor \frac{\sigma}{\varepsilon^3}\left[\frac{3Lc_2^{1/4}}{\sqrt{c_1}}\left[f(x^0)-f^\star\right] + \left(c_1+\tfrac{1}{c_1}\right)\tfrac{1}{\sqrt{c_2}}\right]^{3/2}\right\rfloor = \mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right). \tag{10}$$

In particular, if we choose $c_1 = 1$, then the number of oracle calls is $\mathcal{T}_{ge}$ is

$$\begin{aligned}\mathcal{T}_{ge} &:= \frac{\sigma^3}{\varepsilon}\left[3Lc_2^{9/4}\left[f(x^0)-f^\star\right] + 2c_2^{3/2}\right]^{1/2} + \frac{3\sigma}{\varepsilon^3}\left[3Lc_2^{1/4}\left[f(x^0)-f^\star\right] + \tfrac{2}{\sqrt{c_2}}\right]^{3/2} \\ &= \mathcal{O}\left(\frac{\sigma^3}{\varepsilon} + \frac{\sigma}{\varepsilon^3}\right).\end{aligned} \tag{11}$$

Moreover, the step-size η satisfies $\eta \geq \underline{\eta} := \frac{2}{3Lc_2^{1/4}\sigma^{2/3}(m+1)^{1/3}} = \mathcal{O}\left(m^{-1/3}\right)$.

Here, $\mathcal{T}_{ge}$ stands for the number of stochastic gradient evaluations of f in (1). The complexity $\mathcal{T}_{ge}$ in (11) can be written as $\mathcal{T}_{ge} = \mathcal{O}\left(\max\left\{\sigma\varepsilon^{-3}, \sigma^3\varepsilon^{-1}\right\}\right)$. If $\sigma < \mathcal{O}\left(\frac{1}{\varepsilon}\right)$, then our complexity is $\mathcal{T}_{ge} = \mathcal{O}\left(\sigma\varepsilon^{-3}\right)$. Even if $\sigma < \mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$, then our complexity is still better than $\mathcal{O}\left(\sigma^2\varepsilon^{-4}\right)$ in SGD.

3.2.b Convergence of Algorithm 1 with adaptive step-size η_t and constant β
Let $\beta := 1 - \frac{c_1}{\sqrt{b(m+1)}} \in (0,1)$ be fixed for some $0 < c_1 < \sqrt{b(m+1)}$. Instead of fixing step-size η_t as in (8), we can update it adaptively as

$$\eta_m := \frac{1}{L}, \quad \text{and} \quad \eta_t := \frac{1}{L + L^2\left[\beta^2\eta_{t+1} + \beta^4\eta_{t+2} + \cdots + \beta^{2(m-t)}\eta_m\right]} \quad \text{for } t = 0, \cdots, m-1. \tag{12}$$

It can be shown that $0 < \eta_0 < \eta_1 < \cdots < \eta_m$. Interestingly, our step-size is updated in an increasing manner instead of diminishing as in existing SGD-type methods. Moreover, given m, we can pre-compute the sequence of these step-sizes $\{\eta_t\}_{t=0}^m$ in advance within $\mathcal{O}(m)$ basic operations. Therefore, it does not significantly incur the computational cost of our method.

The following theorem states the convergence of Algorithm 1 under the adaptive update (12), whose proof is given in Subsection 2.3 of the supplementary document.

Theorem 3.2. *Let $\{x_t\}$ be the sequence generated by Algorithm 1 using the step-size η_t defined by (12). Let $\Sigma_m := \sum_{t=0}^m \eta_t$, and $\widetilde{x}_m \sim \mathbf{U}_{\mathbf{p}}(\{x_t\}_{t=0}^m)$ with $\mathbf{p}_t := \mathbb{P}(\widetilde{x}_m = x_t) = \frac{\eta_t}{\Sigma_m}$. Then*

(a) *The sum Σ_m is bounded from below as $\Sigma_m \geq \frac{\sqrt{c_1}(m+1)^{3/4}}{2Lb^{1/4}}$.*

(b) *If we choose* $b := c_2\sigma^{8/3}(m+1)^{1/3}$ *for any* $c_2 > 0$, *then to guarantee* $\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \le \varepsilon^2$, *we need to choose* $m := \left\lfloor \frac{\sigma}{\varepsilon^3}\left[\frac{3Lc_2^{1/4}}{\sqrt{c_1}}\left[f(x^0) - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{c_2}}\right]^{3/2}\right\rfloor = \mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right)$. *Therefore, the number of stochastic gradient evaluations* $\mathcal{T}_{ge}$ *is at most the same as in* (11).

Note that in the finite sum case, i.e. $|\Omega| = n$, we set $b := \min\{n, c_2\sigma^{8/3}(m+1)^{1/3}\}$ in both Theorems 3.1 and 3.2. This complexity remains the same as in Theorem 3.1. However, the adaptive stepsize η_t potentially gives a better performance in practice as we will see in Section 4.

Algorithm 1 can be considered as a single-loop variance reduction method, which is similar to SAGA [7], but Algorithm 1 aims at solving the nonconvex problem (1). It is different from standard SGD methods, where it can be initialized by a mini-batch and then update the estimator using three individual gradients. Therefore, it has the same cost as SGD with mini-batch of size 3. As a compensation, we obtain an improvement on the complexity bound as in Theorems 3.1 and 3.2.

3.3 Convergence analysis of the double loop variant

Since the step-size η_t depends on m, it is natural to run Algorithm 1 with multiple stages. This leads to a double-loop algorithm as SVRG, SARAH, and SPIDER, where Algorithm 1 is restarted at each outer iteration s. The detail of this variant is described in Algorithm 2.

Algorithm 2 (Double-loop HSGD algorithm)

1: **Initialization:** An initial point $\widetilde{x}^0$ and parameters b, m, β_t, and η_t (will be specified).
2: **OuterLoop: For** $s := 1, 2, \cdots, S$ **do**
3: Run Algorithm 1 with an initial point $x_0^{(s)} := \widetilde{x}^{(s-1)}$.
4: Set $\widetilde{x}^{(s)} := x_{m+1}^{(s)}$ as the last iterate of Algorithm 1.
5: **EndFor**

To analyze Algorithm 2, we use $x_t^{(s)}$ to represent the iterate of Algorithm 1 at the t-th inner iteration within each stage s. From (45), we can see that each stage s, the following estimate holds

$$\frac{\eta}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \le \mathbb{E}\left[f(x_0^{(s)})\right] - \mathbb{E}\left[f(x_{m+1}^{(s)})\right] + \frac{\eta\sigma^2\sqrt{m+1}}{(1+\beta)\sqrt{b}}.$$

Here, we assume that we fix the step-size $\eta_t = \eta > 0$ for simplicity of analysis. The complexity of Algorithm 2 is given in the following theorem, whose proof is in Supplementary Document 2.4.

Theorem 3.3. *Let* $\{x_t^{(s)}\}_{t=0\to m}^{s=1\to S}$ *be the sequence generated by Algorithm 2 using constant step-size* η *in* (8). *Then, the following estimate holds*

$$\frac{1}{S(m+1)}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \le \frac{3Lb^{1/4}}{S(m+1)^{3/4}}\left[f(\widetilde{x}^0) - f^\star\right] + \frac{2\sigma^2}{\sqrt{b(m+1)}}. \tag{13}$$

Let $\widetilde{x}_T \sim \mathbf{U}(\{x_t^{(s)}\}_{t=0\to m}^{s=1\to S})$. *If we choose* $b := \frac{c_1\sigma^2}{\varepsilon^2}$ *and* $m+1 := \frac{c_2\sigma^2}{\varepsilon^2}$ *for some constants* $c_1 > 0$ *and* $c_2 > 0$ *and* $c_1c_2 > 4$, *then, to guarantee* $\mathbb{E}\left[\|\nabla f(\widetilde{x}_T)\|^2\right] \le \varepsilon^2$, *we require at most*

$$S := \left\lfloor \frac{3Lc_1^{1/4}\left[f(\widetilde{x}^0) - f^\star\right]}{c_2^{3/4}\sigma\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon}\right\rfloor \quad \textit{outer iterations.} \tag{14}$$

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

$$\mathcal{T}_{ge} := (b+3m)S = \frac{3L(c_1+3c_2)c_1^{1/4}\left[f(\widetilde{x}^0) - f^\star\right]\sigma}{c_2^{3/4}\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^3} = \mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right). \tag{15}$$

Note that the complexity (15) only holds if $\mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right) > \frac{c_1\sigma^2}{\varepsilon^2}$. Otherwise, the total complexity is $\mathcal{O}\left(\max\left\{\frac{\sigma}{\varepsilon^3}, \frac{\sigma^2}{\varepsilon^2}\right\}\right)$, where other constants independent of σ and ε, and are hidden. Practically, if β is very close to 1, one can remove the unbiased SGD term to save one stochastic gradient evaluation. In this case, our estimator reduces to SARAH but using different step-size. We observed empirically that when $\beta \approx 0.999$, the performance of our methods is not affected if we do so.

3.4 Extensions to mini-batch cases

We consider a mini-batch hybrid stochastic estimator $\hat{v}_t$ for the gradient $\nabla f(x_t)$ defined as:

$$\hat{v}_t := \beta_{t-1}\hat{v}_{t-1} + \frac{\beta_{t-1}}{\hat{b}_t}\sum_{i\in\hat{\mathcal{B}}_t}(\nabla f(x_t;\xi_i) - \nabla f(x_{t-1};\xi_i)) + (1-\beta_{t-1})u_t, \tag{16}$$

where $\beta_{t-1} \in [0,1]$, and $\hat{\mathcal{B}}_t$ is a mini-batch of the size $\hat{b}_t$ and independent of the unbiased estimator u_t. Note that u_t can also be a mini-batch unbiased estimator. For example, $u_t := \frac{1}{\tilde{b}_t}\sum_{j\in\tilde{\mathcal{B}}_t}\nabla f(x_t;\zeta_j)$ is a mini-batch SGD estimator with a mini-batch $\tilde{\mathcal{B}}_t$ of size $\tilde{b}_t$, where $\tilde{\mathcal{B}}_t$ is independent of $\hat{\mathcal{B}}_t$.

Using $\hat{v}_t$ defined by (16), we can design a mini-batch variant of Algorithms 1 to solve (1). The following corollary is obtained as a result of Theorems 3.1 for the mini-batch variant of Algorithm 1, whose proof is in Subsection 3.2 of the supplementary document.

Corollary 3.1. *Let Algorithm 1 be applied to solve* (1) *using mini-batch update* (16) *for v_t with $\hat{b}_t = \tilde{b}_t = \hat{b} \geq 1$ fixed, $0 < c_1 < \sqrt{b(m+1)}$, and the step-size*

$$\eta := \frac{2}{L\left(1+\sqrt{1+4\rho\alpha_m^2}\right)} \quad \text{with} \quad \alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2} \quad \text{and} \quad \beta := 1 - \frac{c_1}{\sqrt{\rho b(m-1)}}. \tag{17}$$

If we choose $b := c_2\sigma^{8/3}\left[\rho(m+1)\right]^{1/3}$ for any $c_2 > 0$, then to guarantee $\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \leq \varepsilon^2$, we need to choose

$$m := \left\lfloor \frac{\rho^{1/2}\sigma}{\varepsilon^3}\left[\frac{3Lc_2^{1/4}}{2c_1}\left(f(x^0) - f^\star\right) + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{2\sqrt{c_2}}\right]^{3/2}\right\rfloor = \mathcal{O}\left(\frac{\rho^{1/2}\sigma}{\varepsilon^3}\right). \tag{18}$$

Therefore, the number of oracle calls is $\mathcal{T}_{ge}$ is

$$\mathcal{T}_{ge} := \mathcal{O}\left(\frac{\rho^{1/2}\sigma^3}{\varepsilon} + \frac{\sigma}{\rho^{1/2}\varepsilon^3}\right), \tag{19}$$

where $\rho = \rho(\hat{b}) := \frac{n-\hat{b}}{(n-1)\hat{b}}$ if $n := |\Omega|$ is finite, and $\rho(\hat{b}) := \frac{1}{\hat{b}}$, otherwise. In particular, if we choose $\hat{b} := \frac{\varepsilon^2\sigma^2}{c_3^2}$ for some $0 < c_3 \leq \varepsilon\sigma$, then, the overall complexity $\mathcal{T}_{ge}$ is $\mathcal{T}_{ge} := \mathcal{O}\left(\left(c_3 + \frac{1}{c_3}\right)\frac{\sigma^2}{\varepsilon^2}\right)$.

We can also develop a mini-batch variant of Algorithm 2 and estimate its complexity as in Theorem 3.3. For more details, we refer to Subsection 3.3 in the supplementary document due to space limit.

4 Numerical experiments

We verify our algorithms on two numerical examples and compare them with several existing methods: SVRG [23], SVRG+ [11], SPIDER [8], SpiderBoost [27], and SGD [9]. Due to space limit, the detailed configuration of our experiments as well as more numerical experiments can be found in Supplementary Document D. Our numerical experiments are implemented in Python and running on a MacBook Pro. Laptop with 2.7GHz Intel Core i5 and 16Gb memory.

4.1 Logistic regression with nonconvex regularizer

Our first example is the following well-known problem used in may papers including [27]:

$$\min_{x\in\mathbb{R}^p}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n}\Big[f_i(x) := \log(1+\exp(-a_i^T x)) + \lambda\sum_{j=1}^{p}\frac{x_i^2}{1+x_i^2}\Big]\right\}, \tag{20}$$

where $a_i \in \mathbb{R}^p$ are given for $i = 1, \cdots, n$, and $\lambda > 0$ is a regularization parameter. Clearly, problem (20) fits (1) well with $L_{f_i} = \frac{\|A\|^2}{4} + 2\lambda$. In this experiment, we choose $\lambda = 0.1$ and normalize the data. One can also verify Assumption 1.1 due to the bounded Hessian of f_i.

We use three datasets from LibSVM for (20): `w8a` ($n = 49,749, p = 300$), `rcv1.binary` ($n = 20,242, p = 47,236$), and `real-sim` ($n = 72,309, p = 20,958$). We run 8 different algorithms as follows. Algorithm 1 with constant step-size (Hybrid-SGD-SL) and adaptive step-size (Hybrid-SGD-ASL) using our theoretical step-sizes (8) and (12), respectively without tuning. Hybrid-SGD-DL is Algorithm 2. SGD1 is SGD with constant step-size $\eta_t = \frac{0.1}{L}$, and SGD2 is SGD with adaptive step-size $\eta_t = \frac{0.1}{L(1+\lfloor t/n\rfloor)}$. Since the stepsize of SPIDER depends on an accuracy ε, we choose $\varepsilon = 10^{-1}$ to get a larger step-size. Our first result in the single-sample case (i.e. when $\hat{b} = 1$, not using mini-batch) is plotted in Fig. 1 after 20 epochs.

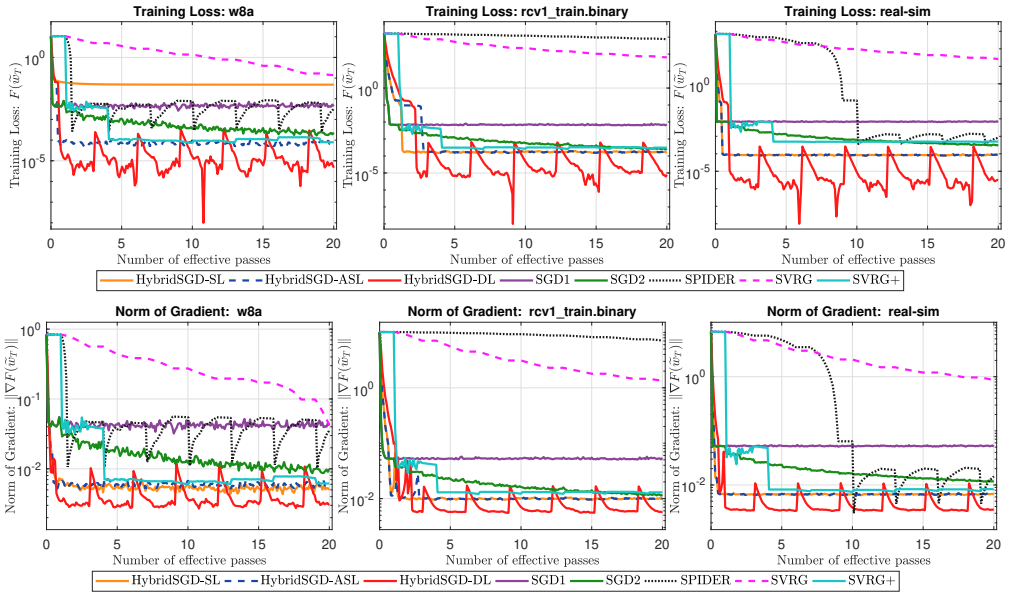

Figure 1: The training loss and gradient norms of (20): Single sample case $\hat{b} = 1$.

From Fig. 1, we observe that Hybrid-SGD-SL has similar convergence behavior as SGD1, but Hybrid-SGD-ASL works better. Hybrid-SGD-DL is the best but has some oscillation. SGD2 works better than SGD1 and is comparable with Hybrid-SGD-SL/ASL in the two last datasets. SVRG performs very poorly due to its small step-size. SVRG+ works much better than SVRG, and is comparable with our methods. SPIDER is also slow even when we have increased its step-size.

Now, we run 3 single-loop algorithms with mini-batch of the size $\hat{b} := 300$. The result is shown in Fig. 2 after 20 epochs. Fig. 2 shows similar performance between Hybrid-SGD-SL and ASL and SGD2. Clearly, these theoretical variants of Algorithm 1 are slightly better than the adaptive SGD variant (SGD2), where a careful step-size is used.

4.2 Binary classification involving nonconvex loss and Tikhonov's regularizer

We consider the following binary classification problem studied in [29] involving nonconvex loss:

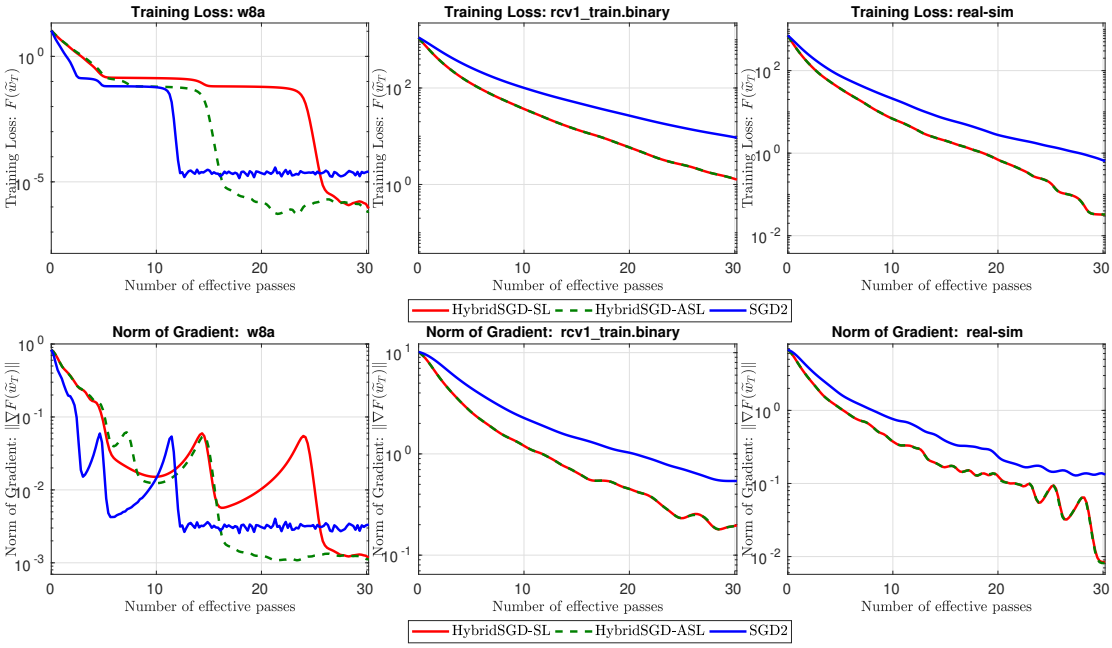

Figure 2: The training loss and gradient norms of (20): Mini-batch case $\hat{b} > 1$.

$$f^\star := \min_{x\in\mathbb{R}^p}\Big\{f(x) := \frac{1}{n}\sum_{i=1}^{n}\ell(a_i^\top x, b_i) + \frac{\lambda}{2}\|x\|^2\Big\}, \tag{21}$$

where $a_i \in \mathbb{R}^p$ and $b_i \in \{-1, 1\}$ are given data for $i = 1, \cdots, n$, $\lambda > 0$ is a regularization parameter, and ℓ is a nonconvex loss of the forms: $\ell(\tau, s) = \Big(1 - \frac{1}{1+\exp(-\tau s)}\Big)^2$ (using in two-layer neural networks). One can check that (21) satisfies Assumption 1.1 with $L \approx 0.15405\max_i\|a_i\|^2 + \lambda$. We choose $\lambda := 0.01$, and test three variants of Algorithm 2: Hybrid-SGD-DL and compare them with SpiderBoost, SVRG, and SVRG+. Due to space limit, we only plot one experiment in Fig. 3 after 20 epochs. Additional experiments can be found in Supplementary Document D.

As we can see from Fig. 3 that Algorithm 2 performs well and is slightly better than SpiderBoost. Note that SpiderBoost simply uses SARAH estimator with constant stepsize $\eta = \frac{1}{2L}$ but with mini-batch of the size $\lfloor\sqrt{n}\rfloor$. It is not surprise that SpiderBoost makes very good progress to decrease the gradient norms. Both SVRG and SVRG+ perform much worse than Hybrid-SGD-DL and SpiderBoost in this test, but SVRG+ is slightly better than SVRG. In our methods, due to the aid of SARAH part, they also make similar progress as SpiderBoost but using different step-sizes.

5 Conclusion

We have introduced a new hybrid SARAH-SGD estimator for the objective gradient of expectation optimization problems. Under standard assumptions, we have shown that this estimator has a better variance reduction property than SARAH. By exploiting such an estimator, we have developed a new Hybrid-SGD algorithm, Algorithm 1, that has better complexity bounds than state-of-the-art SGDs. Our algorithm works with both constant and adaptive step-sizes. We have also studied its double-loop and mini-batch variants. We believe that our approach can be extended to other choices of unbiased estimators, Hessian estimators for second-order stochastic methods, and adaptive β.

A Appendix: Properties of the hybrid stochastic estimator

This supplementary document provides the full proof of all the results in the main text. First, we need the following lemma in the sequel.

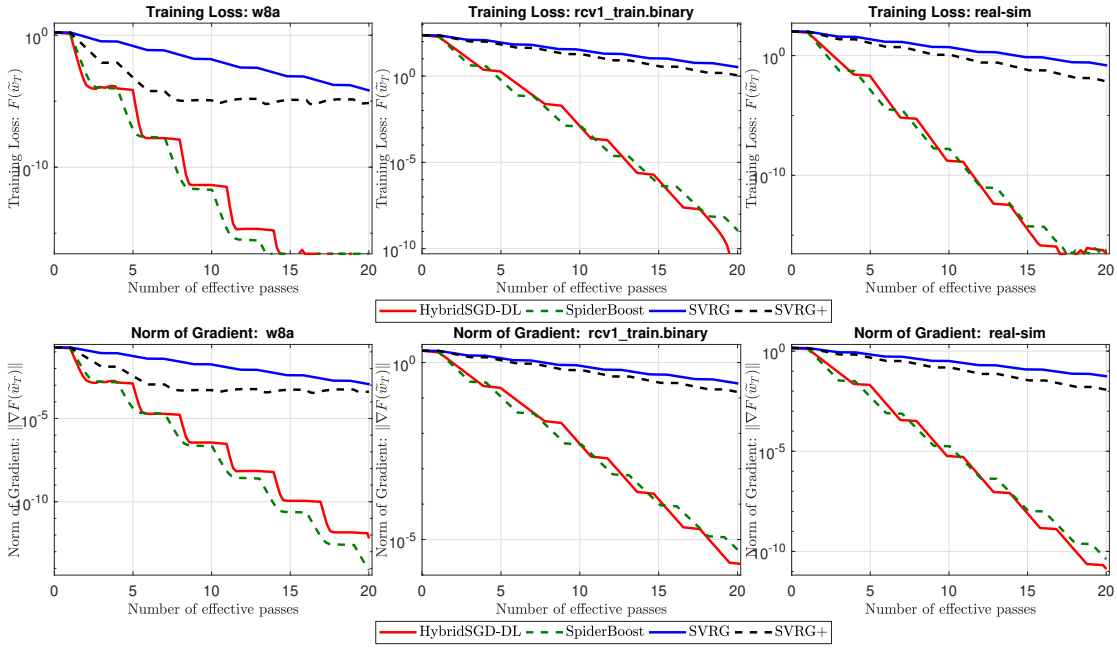

Figure 3: The training loss and gradient norms of (21): Mini-batch case $\hat{b} > 1$.

Lemma A.1. *Given $L > 0$ and $\omega \in (0,1)$. Let $\{\eta_t\}_{t=0}^m$ be the sequence updated by*

$$\eta_m := \frac{1}{L}, \quad \text{and} \quad \eta_t := \frac{1}{L + L^2\big[\omega\eta_{t+1} + \omega^2\eta_{t+2} + \cdots + \omega^{(m-t)}\eta_m\big]}, \tag{22}$$

for $t = 0, \cdots, m-1$. Then

$$0 < \eta_0 < \eta_1 < \cdots < \eta_m = \frac{1}{L}, \quad \text{and} \quad \Sigma_m := \sum_{t=0}^{m}\eta_t \geq \frac{(m+1)\sqrt{1-\omega}}{2L}. \tag{23}$$

Proof. First, from (22) it is obvious to show that $0 < \eta_0 < \cdots < \eta_{m-1} = \frac{1}{L(1+\omega)} < \eta_m = \frac{1}{L}$. At the same time, since $\omega \in (0,1)$, we have $1 \geq \omega \geq \omega^2 \geq \cdots \geq \omega^m$. By Chebyshev's sum inequality, we have

$$\begin{aligned}(m-t)\big(\omega\eta_{t+1} + \omega^2\eta_{t+2} + \cdots + \omega^{m-t}\eta_m\big) &\leq \big(\textstyle\sum_{j=t+1}^m \eta_i\big)\big(\omega + \omega^2 + \cdots + \omega^{m-t}\big)\\ &\leq \tfrac{\omega}{1-\omega}\big(\textstyle\sum_{j=t+1}^m \eta_i\big).\end{aligned} \tag{24}$$

From the update (22), we also have

$$\left\{\begin{array}{ll} L^2\eta_0(\omega\eta_1 + \omega^2\eta_2 + \cdots + \omega^m\eta_m) & = 1 - L\eta_0\\ L^2\eta_1(\omega\eta_2 + \omega^2\eta_3 + \cdots + \omega^{m-1}\eta_m) & = 1 - L\eta_1\\ \cdots & \cdots\\ L^2\eta_{m-1}\omega\eta_m & = 1 - L\eta_{m-1}\\ 0 & = 1 - L\eta_m. \end{array}\right. \tag{25}$$

Using (24) into (25), we get

$$\left\{\begin{array}{ll} \frac{\omega L^2}{1-\omega}\eta_0(\eta_0 + \eta_1 + \cdots + \eta_m) & \geq m - mL\eta_0 + \frac{\omega L^2}{1-\omega}\eta_0^2\\ \frac{\omega L^2}{1-\omega}\eta_1(\eta_0 + \eta_1 + \cdots + \eta_m) & \geq (m-1) - (m-1)L\eta_1 + \frac{\omega L^2}{1-\omega}(\eta_1\eta_0 + \eta_1^2)\\ \cdots & \cdots\\ \frac{\omega L^2}{1-\omega}\eta_{m-1}(\eta_0 + \eta_1 + \cdots + \eta_m) & \geq 1 - L\eta_{m-1} + \frac{\omega L^2}{1-\omega}(\eta_{m-1}\eta_0 + \cdots + \eta_{m-1}^2)\\ \frac{\omega L^2}{1-\omega}\eta_m(\eta_0 + \eta_1 + \cdots + \eta_m) & \geq 1 - L\eta_m + \frac{\omega L^2}{1-\omega}(\eta_m\eta_0 + \cdots + \eta_m^2). \end{array}\right.$$

Let $\Sigma_m := \sum_{t=0}^m \eta_t$ and $S_m := \sum_{t=0}^m \eta_t^2$. Summing up both sides of the above inequalities, we get

$$\frac{\omega L^2}{1-\omega}\Sigma_m^2 \geq \frac{m^2+m+2}{2} - L(m\eta_0 + (m-1)\eta_1 + \cdots + \eta_{m-1} + \eta_m) + \frac{\omega L^2}{2(1-\omega)}\big(S_m + \Sigma_m^2\big).$$

Using again Chebyshev's sum inequality, we have

$$m\eta_0 + (m-1)\eta_1 + \cdots + \eta_{m-1} + \eta_m \leq \frac{m^2+m+2}{2(m+1)}\left(\sum_{t=0}^m \eta_t\right) = \frac{(m^2+m+2)\Sigma_m}{2(m+1)}.$$

Note that $(m+1)S_m \geq \Sigma_m^2$ by Cauchy-Schwarz's inequality, which shows that $S_m + \Sigma_m^2 \geq \big(\frac{m+2}{m+1}\big)\Sigma_m^2$. Combining three last inequalities, we obtain the following quadratic inequation in Σ_m

$$\frac{m\omega L^2}{(1-\omega)}\Sigma_m^2 + L(m^2+m+2)\Sigma_m - (m+1)(m^2+m+2) \geq 0.$$

Solving this inequation with respect to $\Sigma_m > 0$, we obtain

$$\begin{aligned}
\Sigma_m &\geq \frac{(1-\omega)\Big[\sqrt{(m^2+m+2)^2 + \frac{4m(m+1)(m^2+m+2)\omega}{1-\omega}} - (m^2+m+2)\Big]}{2\omega m L} \\
&= \frac{2(m+1)}{L\left[1+\sqrt{1+\frac{4m(m+1)\omega}{(1-\omega)(m^2+m+2)}}\right]} \\
&\geq \frac{2(m+1)\sqrt{1-\omega}}{L\big[\sqrt{1-\omega}+\sqrt{1+3\omega}\big]} \qquad \text{since} \quad \tfrac{m(m+1)}{m^2+m+2} < 1 \\
&\geq \frac{2(m+1)\sqrt{1-\omega}}{L(2+\sqrt{3\omega})} \qquad \text{since} \quad \sqrt{1+3\omega} + \sqrt{1-\omega} \leq 2 + \sqrt{3\omega}.
\end{aligned}$$

Since $\omega \in (0,1)$, we can overestimate this as $\Sigma_m \geq \frac{(m+1)\sqrt{1-\omega}}{2L}$, which proves (23). □

1.1 The proof of Lemma 2.1: Properties of the hybrid SARAH estimator

By taking the expectation of both sides in (4) and using the fact that ξ_t and ζ_t are independent, we can easily obtain (5).

To prove (6), we first write

$$\begin{aligned}
v_t - \nabla f(x_t) &= \beta_{t-1}(v_{t-1} - \nabla f(x_{t-1})) + \beta_{t-1}(\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)) \\
&\quad + (1-\beta_{t-1})\big[u_t - \nabla f(x_t)\big] + \beta_{t-1}\big[\nabla f(x_{t-1}) - \nabla f(x_t)\big].
\end{aligned}$$

In this case, we have

$$\begin{aligned}
\|v_t - \nabla f(x_t)\|^2 &= \beta_{t-1}^2\|v_{t-1} - \nabla f(x_{t-1})\|^2 + \beta_{t-1}^2\|\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)\|^2 \\
&\quad + (1-\beta_{t-1})^2\|u_t - \nabla f(x_t)\|^2 + \beta_{t-1}^2\|\nabla f(x_{t-1}) - \nabla f(x_t)\|^2 \\
&\quad + 2\beta_{t-1}^2(v_{t-1} - \nabla f(x_{t-1}))^\top(\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)) \\
&\quad + 2\beta_{t-1}^2(v_{t-1} - \nabla f(x_{t-1}))^\top(\nabla f(x_{t-1}) - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(v_{t-1} - \nabla f(x_{t-1}))^\top(u_t - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t))^\top(u_t - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}^2(\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t))^\top(\nabla f(x_{t-1}) - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(u_t - \nabla f(x_t))^\top(\nabla f(x_{t-1}) - \nabla f(x_t)).
\end{aligned}$$

Let us first take expectation w.r.t. ξ_t conditioned on ζ_t to obtain

$$\begin{aligned}
\mathbb{E}_{\xi_t}\left[\|v_t - \nabla f(x_t)\|^2 \mid \zeta_t\right] &= \beta_{t-1}^2\|v_{t-1} - \nabla f(x_{t-1})\|^2 + \beta_{t-1}^2\mathbb{E}_{\xi_t}\left[\|\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)\|^2 \mid \zeta_t\right] \\
&\quad + (1-\beta_{t-1})^2\|u_t - \nabla f(x_t)\|^2 - \beta_{t-1}^2\|\nabla f(x_{t-1}) - \nabla f(x_t)\|^2 \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(v_{t-1} - \nabla f(x_{t-1}))^\top(u_t - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(\nabla f(x_t) - \nabla f(x_{t-1}))^\top(u_t - \nabla f(x_t)) \\
&\quad + 2\beta_{t-1}(1-\beta_{t-1})(u_t - \nabla f(x_t))^\top(\nabla f(x_{t-1}) - \nabla f(x_t)).
\end{aligned}$$

Now, taking the expectation over ζ_t, and noting that $\mathbb{E}_{(\xi_t,\zeta_t)}[\cdot] = \mathbb{E}_{\zeta_t}[\mathbb{E}_{\xi_t}[\cdot \mid \zeta_t]]$ and $\mathbb{E}_{\zeta_t}[u_t - \nabla f(x_t)] = 0$, we get

$$\begin{aligned}\mathbb{E}_{(\xi_t,\zeta_t)}\left[\|v_t - \nabla f(x_t)\|^2\right] &= \beta_{t-1}^2\|v_{t-1} - \nabla f(x_{t-1})\|^2 + \beta_{t-1}^2\mathbb{E}_{\xi_t}\left[\|\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)\|^2\right] \\ &\quad + (1-\beta_{t-1})^2\mathbb{E}_{\zeta_t}\left[\|u_t - \nabla f(x_t)\|^2\right] - \beta_{t-1}^2\|\nabla f(x_{t-1}) - \nabla f(x_t)\|^2,\end{aligned}$$

which is exactly (6). □

1.2 The proof of Lemma 2.2: Bound on the variance of the hybrid estimator

We first upper bound (6) by using $\sigma_t^2 := \mathbb{E}_{\zeta_t}\left[\|u_t - \nabla f(x_t)\|^2\right]$ and then taking the full expectation over $\mathcal{F}_t := \sigma(v_0, v_1, \cdots, v_t)$ as

$$\begin{aligned}\mathbb{E}\left[\|v_t - \nabla f(x_t)\|^2\right] &\leq \beta_{t-1}^2\mathbb{E}\left[\|v_{t-1} - \nabla f(x_{t-1})\|^2\right] + \beta_{t-1}^2\mathbb{E}\left[\|\nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)\|^2\right] \\ &\quad + (1-\beta_{t-1})^2\sigma_t^2 \\ &\overset{(2)}{\leq} \beta_{t-1}^2\mathbb{E}\left[\|v_{t-1} - \nabla f(x_{t-1})\|^2\right] + \beta_{t-1}^2L^2\mathbb{E}\left[\|x_t - x_{t-1}\|^2\right] + (1-\beta_{t-1})^2\sigma_t^2.\end{aligned}$$

If we define $a_t^2 := \mathbb{E}\left[\|v_t - \nabla f(x_t)\|^2\right]$, then the above inequality can lead to

$$a_t^2 \leq \beta_{t-1}^2a_{t-1}^2 + \beta_{t-1}^2L^2\mathbb{E}\left[\|x_t - x_{t-1}\|^2\right] + (1-\beta_{t-1})^2\sigma_t^2.$$

Denote $b_{t-1}^2 := \mathbb{E}\left[\|x_t - x_{t-1}\|^2\right]$. Then, we have from the last inequality that

$$a_t^2 \leq \beta_{t-1}^2a_{t-1}^2 + L^2\beta_{t-1}^2b_{t-1}^2 + (1-\beta_{t-1})^2\sigma_t^2.$$

By induction, this inequality implies

$$\begin{aligned}a_t^2 &\leq \beta_{t-1}^2a_{t-1}^2 + L^2\beta_{t-1}^2b_{t-1}^2 + (1-\beta_{t-1})^2\sigma_t^2 \\ &\leq \beta_{t-1}^2\left[\beta_{t-2}^2a_{t-2}^2 + L^2\beta_{t-2}^2b_{t-2}^2 + (1-\beta_{t-2})^2\sigma^2\right] + L^2\beta_{t-1}^2b_{t-1}^2 + (1-\beta_{t-1})^2\sigma_t^2 \\ &= \beta_{t-1}^2\beta_{t-2}^2a_{t-2}^2 + L^2\beta_{t-1}^2\beta_{t-2}^2b_{t-2}^2 + L^2\beta_{t-1}^2b_{t-1}^2 + \left[(1-\beta_{t-1})^2\sigma_t^2 + \beta_{t-1}^2(1-\beta_{t-2})^2\sigma_{t-1}^2\right] \\ &\leq \beta_{t-1}^2\beta_{t-2}^2\left[\beta_{t-3}^2a_{t-3}^2 + L^2\beta_{t-3}^2b_{t-3}^2 + (1-\beta_{t-3})^2\sigma_{t-2}^2\right] \\ &+ L^2\beta_{t-1}^2\beta_{t-2}^2b_{t-2}^2 + L^2\beta_{t-1}^2b_{t-1}^2 + \left[(1-\beta_{t-1})^2\sigma_t^2 + \beta_{t-1}^2(1-\beta_{t-2})^2\sigma_{t-1}^2\right] \\ &= \beta_{t-1}^2\beta_{t-2}^2\beta_{t-3}^2a_{t-3}^2 + L^2\beta_{t-1}^2\beta_{t-2}^2\beta_{t-3}^2b_{t-3}^2 + L^2\beta_{t-1}^2\beta_{t-2}^2b_{t-2}^2 \\ &+ L^2\beta_{t-1}^2b_{t-1}^2 + \left[(1-\beta_{t-1})^2\sigma_t^2 + \beta_{t-1}^2(1-\beta_{t-2})^2\sigma_{t-1}^2 + \beta_{t-1}^2\beta_{t-2}^2(1-\beta_{t-3})^2\sigma_{t-2}^2\right] \\ &\cdots\cdots \\ &\leq (\beta_{t-1}^2\cdots\beta_0^2)a_0^2 + L^2(\beta_{t-1}^2\cdots\beta_0^2)b_0^2 + L^2(\beta_{t-1}^2\cdots\beta_1^2)b_1^2 + \cdots + L^2\beta_{t-1}^2b_{t-1}^2 \\ &+ \left[(1-\beta_{t-1})^2\sigma_t^2 + \beta_{t-1}^2(1-\beta_{t-2})^2\sigma_{t-1}^2 + \beta_{t-1}^2\beta_{t-2}^2(1-\beta_{t-3})^2\sigma_{t-2}^2 + \cdots\right. \\ &\quad \left. + \beta_{t-1}^2\beta_{t-2}^2\cdots\beta_1^2(1-\beta_0)^2\sigma_1^2\right].\end{aligned}$$

Here, we use a convention that $\prod_{i=t+1}^t \beta_i^2 = 1$. As a result, it can be rewritten in a compact form as

$$a_t^2 \leq \Big(\prod_{i=1}^t \beta_{i-1}^2\Big)a_0^2 + L^2\sum_{i=0}^{t-1}\Big(\prod_{j=i+1}^t \beta_{j-1}^2\Big)b_i^2 + \sum_{i=0}^{t-1}\Big(\prod_{j=i+2}^t \beta_{j-1}^2\Big)(1-\beta_i)^2\sigma_{i+1}^2. \tag{26}$$

Define $\omega_t := \prod_{i=1}^t \beta_{i-1}^2$, $\omega_{i,t} := \prod_{j=i+1}^t \beta_{j-1}^2$, and $S_t := \sum_{i=0}^{t-1} s_i = \sum_{i=0}^{t-1}\big(\prod_{j=i+2}^t \beta_{j-1}^2\big)(1-\beta_i)^2\sigma_{i+1}^2$ with $s_i := (1-\beta_i)^2\sigma_{i+1}^2\big(\prod_{j=i+2}^t \beta_{j-1}^2\big)$. Then, we can rewrite (26) as

$$a_t^2 \leq \omega_t a_0^2 + L^2\sum_{i=0}^{t-1}\omega_{i,t}b_i^2 + S_t,$$

which is exactly (7). □

B Appendix: Convergence analysis of Algorithm 1 and Algorithm 2

We provide the full convergence analysis for Algorithm 1 and Algorithm 2 in the single-sample case.

2.1 The proof of Lemma B.1: One-iteration analysis

The following lemma provides a key estimate for convergence analysis of Algorithm 1.

Lemma B.1. *Let $\{x_t\}$ be the sequence generated by Algorithm 1. Then, under Assumption 1.1, we have the following estimate:*

$$\begin{aligned}\mathbb{E}\left[f(x_{m+1})\right] &\le \mathbb{E}\left[f(x_0)\right] - \frac{1}{2}\sum_{t=0}^{m}\eta_t\mathbb{E}\left[\|\nabla f(x_t)\|^2\right]\\ &+ \frac{1}{2}\Big(\sum_{t=0}^{m}\eta_t\omega_t\Big)\mathbb{E}\left[\|v_0-\nabla f(x_0)\|^2\right] + \frac{1}{2}\Big(\sum_{t=0}^{m}\eta_t S_t\Big) + \frac{1}{2}\mathcal{T}_m,\end{aligned} \tag{27}$$

where

$$\mathcal{T}_m := L^2\sum_{t=1}^{m}\eta_t\sum_{i=0}^{t-1}\omega_{i,t}\eta_i^2\mathbb{E}\left[\|v_i\|^2\right] - \sum_{t=0}^{m}\big(\eta_t - L\eta_t^2\big)\mathbb{E}\left[\|v_t\|^2\right], \tag{28}$$

and ω_t, $\omega_{i,t}$, and S_t are defined in Lemma 2.2.

Proof. First, from the L-smoothness of f, we have

$$\begin{aligned}f(x_{t+1}) &\le f(x_t) - \eta_t\langle\nabla f(x_t), v_t\rangle + \tfrac{L\eta_t^2}{2}\|v_t\|^2\\ &= f(x_t) - \tfrac{\eta_t}{2}\|\nabla f(x_t)\|^2 - \big(\tfrac{\eta_t}{2} - \tfrac{L\eta_t^2}{2}\big)\|v_t\|^2 + \tfrac{\eta_t}{2}\|v_t-\nabla f(x_t)\|^2.\end{aligned}$$

Taking the expectation over the randomness (ξ_t,ζ_t) of this estimate, we obtain

$$\begin{aligned}\mathbb{E}_{(\xi_t,\zeta_t)}\left[f(x_{t+1})\right] &\le f(x_t) - \tfrac{\eta_t}{2}\mathbb{E}_{(\xi_t,\zeta_t)}\left[\|\nabla f(x_t)\|^2\right] - \tfrac{\eta_t}{2}\big(1-L\eta_t\big)\mathbb{E}_{(\xi_t,\zeta_t)}\left[\|v_t\|^2\right]\\ &+ \tfrac{\eta_t}{2}\mathbb{E}_{(\xi_t,\zeta_t)}\left[\|v_t-\nabla f(x_t)\|^2\right].\end{aligned}$$

Taking the full expectation over the entire history up to the t-th iteration, and then using (7) and noting that $x_t - x_{t-1} = -\eta_{t-1}v_{t-1}$, we obtain

$$\begin{aligned}\mathbb{E}\left[f(x_{t+1})\right] &\le \mathbb{E}\left[f(x_t)\right] - \tfrac{\eta_t}{2}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] - \tfrac{\eta_t}{2}\big(1-L\eta_t\big)q_t^2 + \tfrac{\eta_t}{2}a_t^2\\ &\le \mathbb{E}\left[f(x_t)\right] - \tfrac{\eta_t}{2}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] - \tfrac{\eta_t}{2}\big(1-L\eta_t\big)q_t^2\\ &+ \tfrac{\eta_t}{2}\Big[\omega_t a_0^2 + L^2\textstyle\sum_{i=0}^{t-1}\omega_{i,t}\eta_i^2q_i^2 + S_t\Big],\end{aligned} \tag{29}$$

where $q_t^2 := \mathbb{E}\left[\|v_t\|^2\right]$ and $a_t^2 := \mathbb{E}\left[\|v_t-\nabla f(x_t)\|^2\right]$. Here, we use $b_{t-1}^2 := \mathbb{E}\left[\|x_t - x_{t-1}\|^2\right] = \eta_{t-1}^2\mathbb{E}\left[\|v_{t-1}\|^2\right] = \eta_{t-1}^2q_{t-1}^2$ in the last inequality.

Summing up (29) from $t=0$ to $t=m$, we obtain

$$\begin{aligned}\mathbb{E}\left[f(x_{m+1})\right] &\le \mathbb{E}\left[f(x_0)\right] - \textstyle\sum_{t=0}^{m}\frac{\eta_t}{2}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] - \sum_{t=0}^{m}\frac{\eta_t}{2}\big(1-L\eta_t\big)q_t^2\\ &+ \tfrac{1}{2}\left(\textstyle\sum_{t=0}^{m}\omega_t\eta_t\right)a_0^2 + \tfrac{1}{2}\left(\sum_{t=0}^{m}\eta_t S_t\right) + \tfrac{L^2}{2}\sum_{t=0}^{m}\eta_t\sum_{i=0}^{t-1}\omega_{i,t}\eta_i^2q_i^2.\end{aligned} \tag{30}$$

Let us define T_m as in (28), i.e.:

$$\mathcal{T}_m := L^2\sum_{t=1}^{m}\eta_t\sum_{i=0}^{t-1}\omega_{i,t}\eta_i^2q_i^2 - \sum_{t=0}^{m}\eta_t\big(1-L\eta_t\big)q_t^2.$$

Then, we obtain from (30) the estimate (27). □

2.2 The proof of Theorem 3.1: Single-loop with constant step-size

We analyze the case $\beta_t = \beta \in (0,1)$ fixed and the step-size $\eta_t = \eta > 0$ fixed. From Lemma 2.2, we have $\omega_t = \beta^{2t}$, $\omega_{i,t} = \beta^{2(t-i)}$, and

$$\begin{aligned} s_t &:= \textstyle\sum_{i=0}^{t-1} \big(\prod_{j=i+2}^{t} \beta_{j-1}^2\big)(1-\beta_i)^2 \\ &= (1-\beta)^2\big[1+\beta^2+\beta^4+\cdots+\beta^{2(t-1)}\big] \\ &= (1-\beta)^2\Big[\tfrac{1-\beta^{2t}}{1-\beta^2}\Big] \\ &< \tfrac{1-\beta}{1+\beta}. \end{aligned}$$

In this case, by convention that $\omega_0 = 1$, we have

$$\sum_{t=0}^{m} s_t < \frac{(1-\beta)(m+1)}{1+\beta} \quad \text{and} \quad \sum_{t=0}^{m} \omega_t = 1 + \frac{\beta^2(1-\beta^{2m})}{1-\beta^2} = \frac{1-\beta^{2(m+1)}}{1-\beta^2} < \frac{1}{1-\beta^2}. \tag{31}$$

Now, to bound the quantity $\mathcal{T}_m$ defined by (28), we note that

$$\begin{aligned} \sum_{t=1}^{m}\sum_{i=0}^{t-1} \beta^{2(t-i)} q_i^2 &= \sum_{i=0}^{0} \beta^{2(1-i)} q_i^2 + \sum_{i=0}^{1} \beta^{2(2-i)} q_i^2 + \sum_{i=0}^{2} \beta^{2(3-i)} q_i^2 + \cdots + \sum_{i=0}^{m-1} \beta^{2(m-i)} q_i^2 \\ &= \beta^2 q_0^2 + \big[\beta^4 q_0^2 + \beta^2 q_1^2\big] + \big[\beta^6 q_0^2 + \beta^4 q_1^2 + \beta^2 q_0^2\big] + \cdots \\ &+ \big[\beta^{2m} q_0^2 + \beta^{2(m-1)} q_1 + \cdots + \beta^2 q_{m-1}^2\big] \\ &= \beta^2\big[1+\beta^2+\cdots+\beta^{2(m-1)}\big] q_0^2 + \beta^2\big[1+\beta^2+\cdots+\beta^{2(m-2)}\big] q_1^2 + \cdots \\ &+ \beta^2\big[1+\beta^2\big] q_{m-2}^2 + \beta^2 q_{m-1}^2 \\ &= \tfrac{\beta^2}{1-\beta^2}\Big[(1-\beta^{2m}) q_0^2 + (1-\beta^{2(m-1)}) q_1^2 + \cdots + (1-\beta^2) q_{m-1}^2\Big]. \end{aligned}$$

Using this expression, we can write $\mathcal{T}_m$ from (28) as

$$\begin{aligned} \mathcal{T}_m &= \eta\Big[\tfrac{\beta^2(1-\beta^{2m})L^2\eta^2}{1-\beta^2} - (1-L\eta)\Big] q_0^2 + \eta\Big[\tfrac{\beta^2(1-\beta^{2(m-1)})L^2\eta^2}{1-\beta^2} - (1-L\eta)\Big] q_1^2 + \cdots \\ &+ \eta\Big[\tfrac{\beta^2(1-\beta^2)L^2\eta^2}{1-\beta^2} - (1-L\eta)\Big] q_{m-1}^2 - \eta(1-L\eta) q_m^2. \end{aligned} \tag{32}$$

To guarantee $\mathcal{T}_m \le 0$, from (32), we need to choose

$$\begin{cases} \frac{L^2\eta^2\beta^2(1-\beta^{2m})}{1-\beta^2} - (1-L\eta) & \le 0 \\ \frac{L^2\eta^2\beta^2(1-\beta^{2(m-1)})}{1-\beta^2} - (1-L\eta) & \le 0 \\ \cdots & \cdots \\ \frac{L^2\eta^2\beta^2(1-\beta^2)}{1-\beta^2} - (1-L\eta) & \le 0 \\ -(1-L\eta) & \le 0. \end{cases} \tag{33}$$

Clearly, since $1-\beta^{2(m-i)} \ge 1-\beta^2$ for $i = 0, \cdots, m-1$, if we define $\alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2}$, then the condition (33) holds if $L^2\eta^2\alpha_m^2 - (1-L\eta) \le 0$. By tightening this condition, we obtain a quadratic equation $L^2\eta^2\alpha_m^2 - (1-L\eta) = 0$ in η, which leads to

$$\eta := \frac{2}{L(\sqrt{1+4\alpha_m^2}+1)} \quad \text{with} \quad \alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2}. \tag{34}$$

Note that since $\alpha_m^2 \leq \frac{\beta^2}{1-\beta^2}$, we have $\eta \geq \underline{\eta} := \frac{2\sqrt{1-\beta^2}}{L(\sqrt{1-\beta^2}+\sqrt{1+3\beta^2})}$. In that case, by using (31) and (33), (27) reduces to

$$\begin{aligned}\mathbb{E}\left[f(x_{m+1})\right] &\overset{(31)}{\leq} \mathbb{E}\left[f(x_0)\right] - \tfrac{\eta}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \\ &\quad + \tfrac{\eta(1-\beta^{2(m+1)})}{2(1-\beta^2)}\mathbb{E}\left[\|v_0 - \nabla f(x_0)\|^2\right] + \left[\tfrac{(1-\beta)(m+1)}{1+\beta}\right]\tfrac{\eta\sigma^2}{2}.\end{aligned} \tag{35}$$

Note that $\mathbb{E}\left[\|v_0 - \nabla f(x_0)\|^2\right] \leq \frac{\sigma^2}{b}$ and $\mathbb{E}\left[f(x_{m+1})\right] \geq f^\star$, we can further bound (35) as

$$\frac{\eta}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \mathbb{E}\left[f(x_0)\right] - f^\star + \frac{\eta\sigma^2}{2(1+\beta)}\Big[\frac{1}{(1-\beta)b} + (1-\beta)(m+1)\Big].$$

Multiplying both sides of this inequality by $\frac{2}{\eta(m+1)}$, and then using the lower bound of η from (34), we obtain

$$\begin{aligned}\tfrac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] &\leq \tfrac{2}{\eta(m+1)}\Big[\mathbb{E}\left[f(x_0)\right] - f^\star\Big] + \tfrac{\sigma^2}{(1+\beta)}\Big[\tfrac{1}{(1-\beta)b(m+1)} + (1-\beta)\Big] \\ &\leq \tfrac{L}{(m+1)}\left(\tfrac{\sqrt{1-\beta^2}+\sqrt{1+3\beta^2}}{\sqrt{1-\beta^2}}\right)\Big[\mathbb{E}\left[f(x_0)\right] - f^\star\Big] \\ &\quad + \tfrac{\sigma^2}{(1+\beta)}\Big[\tfrac{1}{(1-\beta)b(m+1)} + (1-\beta)\Big].\end{aligned} \tag{36}$$

Let us choose $\beta := 1 - \frac{c_1}{\sqrt{b(m+1)}}$ for some $0 < c_1 < \sqrt{b(m+1)}$. In this case, the last two terms of the right-hand side of (36) become

$$\frac{1}{(1-\beta)b(m+1)} + (1-\beta) = \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{b(m+1)}}.$$

With this choice of β, (36) leads to

$$\begin{aligned}\tfrac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] &\leq \tfrac{L}{(m+1)}\left(\tfrac{\sqrt{1-\beta^2}+\sqrt{1+3\beta^2}}{\sqrt{1-\beta^2}}\right)\Big[\mathbb{E}\left[f(x_0)\right] - f^\star\Big] \\ &\quad + \left(c_1 + \tfrac{1}{c_1}\right)\tfrac{\sigma^2}{(1+\beta)\sqrt{b(m+1)}}.\end{aligned} \tag{37}$$

(a) Since $\beta = 1 - \frac{c_1}{\sqrt{b(m+1)}} < 1$ and $c_1 < \sqrt{b(m+1)}$, we have

$$1-\beta^2 = 1 - \Big(1 - \frac{c_1}{\sqrt{b(m+1)}}\Big)^2 = \frac{2c_1}{\sqrt{b(m+1)}} - \frac{c_1^2}{b(m+1)} = \frac{2c_1\sqrt{b(m+1)} - c_1^2}{b(m+1)} > \frac{c_1}{\sqrt{b(m+1)}},$$

and $\sqrt{1-\beta^2} + \sqrt{1+3\beta^2} \leq 1 + \sqrt{1+3\beta^2} \leq 3$. On the other hand, from (34), we have

$$\eta \geq \underline{\eta} = \frac{2\sqrt{1-\beta^2}}{L(\sqrt{1-\beta^2}+\sqrt{1+3\beta^2})} \geq \frac{2\sqrt{c_1}}{3L\big[b(m+1)\big]^{1/4}}. \tag{38}$$

This proves (a).

Let us define define $f^0 := f(x_0)$. Then, using $\beta < 1$ and (38) into (37), we get

$$\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{3Lb^{1/4}}{\sqrt{c_1}(m+1)^{3/4}}\left[f^0 - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{\sigma^2}{\sqrt{b(m+1)}}.$$

(b) Let us choose $b := c_2\sigma^{8/3}(m+1)^{1/3}$ for some constant $c_2 > 0$. Then the last estimate becomes

$$\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{\sigma^{2/3}}{(m+1)^{2/3}}\left[\frac{3Lc_2^{1/4}}{\sqrt{c_1}}\left[f^0 - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{c_2}}\right]. \tag{39}$$

To guarantee $\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \varepsilon^2$, from (39) we need to set

$$\frac{\sigma^{2/3}}{(m+1)^{2/3}}\left[\frac{3Lc_2^{1/4}}{\sqrt{c_1}}\left[f^0 - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{c_2}}\right] \leq \varepsilon^2.$$

This leads to $m+1 \geq \frac{\sigma}{\varepsilon^3}\left[\frac{3Lc_2^{1/4}}{\sqrt{c_1}}\left[f^0 - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{c_2}}\right]^{3/2}$. Therefore, we can choose m as shown in (10).

Finally, if $c_1 = 1$, then the number of stochastic gradient evaluations is $\mathcal{T}_{ge}$ is

$$\begin{aligned}
\mathcal{T}_{ge} &= b + 3m = c_2\sigma^{8/3}(m+1)^{1/3} + \tfrac{3\sigma}{\varepsilon^3}\left[3Lc_2^{1/4}\left[f^0 - f^\star\right] + \tfrac{2}{\sqrt{c_2}}\right]^{3/2}\\
&= \tfrac{c_2\sigma^3}{\varepsilon}\left[3Lc_2^{1/4}\left[f^0 - f^\star\right] + \tfrac{2}{\sqrt{c_2}}\right]^{1/2} + \tfrac{3\sigma}{\varepsilon^3}\left[3Lc_2^{1/4}\left[f^0 - f^\star\right] + \tfrac{2}{\sqrt{c_2}}\right]^{3/2}\\
&= \tfrac{\sigma^3}{\varepsilon}\left[3Lc_2^{9/4}\left[f^0 - f^\star\right] + c_2^{3/2}\right]^{1/2} + \tfrac{3\sigma}{\varepsilon^3}\left[3Lc_2^{1/4}\left[f^0 - f^\star\right] + \tfrac{2}{\sqrt{c_2}}\right]^{3/2}\\
&= \mathcal{O}\left(\tfrac{\sigma}{\varepsilon^3} + \tfrac{\sigma^3}{\varepsilon}\right),
\end{aligned}$$

which proves (11). □

2.3 The proof of Theorem 3.2: Single-loop with adaptive step-size

First, from Lemma B.1, we have

$$\begin{aligned}
\mathbb{E}\left[f(x_{m+1})\right] &\leq \mathbb{E}\left[f(x_0)\right] - \frac{1}{2}\sum_{t=0}^{m}\eta_t\mathbb{E}\left[\|\nabla f(x_t)\|^2\right]\\
&\quad + \frac{1}{2}\Big(\sum_{t=0}^{m}\eta_t\omega_t\Big)\mathbb{E}\left[\|v_0 - \nabla f(x_0)\|^2\right] + \frac{1}{2}\Big(\sum_{t=0}^{m}\eta_t S_t\Big) + \frac{1}{2}\mathcal{T}_m,
\end{aligned} \tag{40}$$

where

$$\mathcal{T}_m := L^2\sum_{t=1}^{m}\eta_t\sum_{i=0}^{t-1}\omega_{i,t}\eta_i^2\mathbb{E}\left[\|v_i\|^2\right] - \sum_{t=0}^{m}\left(\eta_t - L\eta_t^2\right)\mathbb{E}\left[\|v_t\|^2\right], \tag{41}$$

and ω_t, $\omega_{i,t}$, and s_t are defined in Lemma 2.2.

If we fix $\beta_t = \beta \in (0,1)$, then we can show that $\omega_t = \beta^{2t}$, $\omega_{i,t} = \beta^{2(t-i)}$, and $s_t = (1-\beta)^2\left[\frac{1-\beta^{2t}}{1-\beta^2}\right] < \frac{1-\beta}{1+\beta}$ as in the proof of Theorem 3.1.

Now, let $u_i^2 := \mathbb{E}\left[\|v_i\|^2\right]$. To bound the quantity $\mathcal{T}_m$ defined by (28), we note that

$$\begin{aligned}
\sum_{t=1}^{m}\eta_t\sum_{i=0}^{t-1}\beta^{2(t-i)}\eta_i^2u_i^2 &= \eta_1\sum_{i=0}^{0}\beta^{2(1-i)}\eta_i^2u_i^2+\eta_2\sum_{i=0}^{1}\beta^{2(2-i)}\eta_i^2u_i^2\\
&+\eta_3\sum_{i=0}^{2}\beta^{2(3-i)}\eta_i^2u_i^2+\cdots+\eta_m\sum_{i=0}^{m-1}\beta^{2(m-i)}\eta_i^2u_i^2\\
&=\beta^2\eta_1\eta_0^2u_0^2+\eta_2\left[\beta^4\eta_0^2u_0^2+\beta^2\eta_1^2u_1^2\right]\\
&+\eta_3\left[\beta^6\eta_2^2u_0^2+\beta^4\eta_1^2u_1^2+\beta^2\eta_2^2u_2^2\right]+\cdots\\
&+\eta_m\left[\beta^{2m}\eta_0^2u_0^2+\beta^{2(m-1)}\eta_1^2u_1^2+\cdots+\beta^2\eta_{m-1}^2u_{m-1}^2\right]\\
&=\beta^2\eta_0^2\left[\eta_1+\beta^2\eta_2+\cdots+\beta^{2(m-1)}\eta_m\right]u_0^2\\
&+\beta^2\eta_1^2\left[\eta_2+\beta^2\eta_3+\cdots+\beta^{2(m-2)}\eta_m\right]u_1^2+\cdots\\
&+\beta^2\eta_{m-2}^2\left[\eta_{m-1}+\beta^2\eta_m\right]u_{m-2}^2+\beta^2\eta_{m-1}^2\eta_mu_{m-1}^2.
\end{aligned}$$

Using this expression, we can write $\mathcal{T}_m$ from (28) as

$$\begin{aligned}
\mathcal{T}_m &= \eta_0\Big[L^2\beta^2\eta_0\left[\eta_1+\beta^2\eta_2+\cdots+\beta^{2(m-1)}\eta_m\right]-(1-L\eta_0)\Big]u_0^2\\
&+\eta_1\Big[L^2\beta^2\eta_1\left[\eta_2+\beta^2\eta_3+\cdots+\beta^{2(m-2)}\eta_m\right]-(1-L\eta_1)\Big]+\cdots\\
&+\eta_{m-1}\Big[L^2\beta^2\eta_{m-1}\eta_m-(1-L\eta_{m-1})\Big]u_{m-1}^2-\eta_m(1-L\eta_m)u_m^2.
\end{aligned}$$

To guarantee $\mathcal{T}_m\le 0$, from the last expression of $\mathcal{T}_m$, we can impose the following condition:

$$\begin{cases}
L^2\beta^2\eta_0\left[\eta_1+\beta^2\eta_2+\cdots+\beta^{2(m-1)}\eta_m\right]-(1-L\eta_0) &= 0\\
L^2\beta^2\eta_1\left[\eta_2+\beta^2\eta_3+\cdots+\beta^{2(m-2)}\eta_m\right]-(1-L\eta_1) &= 0\\
\cdots & \cdots\\
L^2\beta^2\eta_{m-1}\eta_m-(1-L\eta_{m-1}) &= 0\\
-(1-L\eta_m) &= 0.
\end{cases}\tag{42}$$

The condition (42) leads to the following update of η_t:

$$\eta_m := \frac{1}{L},\quad \text{and}\quad \eta_t := \frac{1}{L+L^2\left[\beta^2\eta_{t+1}+\beta^4\eta_{t+2}+\cdots+\beta^{2(m-t)}\eta_m\right]},\quad t=0,\cdots,m-1,$$

which is exactly (12).

Next, note that $\beta^2=\left(1-\frac{c_1}{\sqrt{b(m+1)}}\right)^2=1-\frac{2c_1}{\sqrt{b(m+1)}}+\frac{c_1^2}{b(m+1)}$. Therefore, $1-\beta^2=\frac{2c_1}{\sqrt{b(m+1)}}-\frac{c_1^2}{b(m+1)}\ge\frac{c_1}{\sqrt{b(m+1)}}$, which implies $\sqrt{1-\beta^2}\ge\frac{\sqrt{c_1}}{(b(m+1))^{1/4}}$. Using $\sqrt{1-\omega}=\sqrt{1-\beta^2}\ge\frac{\sqrt{c_1}}{(b(m+1))^{1/4}}$ into (23) of Lemma A.1, we can show that $\Sigma_m\ge\frac{\sqrt{c_1}(m+1)^{3/4}}{2Lb^{1/4}}$ as in the first statement (a) of Theorem 3.2.

Note that $\omega_t=\beta^{2t}$, by the Chebyshev sum inequality, we have

$$\sum_{t=0}^{m}\omega_t\eta_t=\sum_{t=0}^{m}\beta^{2t}\eta_t\le\frac{\Sigma_m}{(m+1)}(1+\beta^2+\cdots+\beta^{2m})\le\frac{\Sigma_m}{(m+1)(1-\beta^2)}.$$

Utilizing this estimate, $\mathbb{E}\left[\|v_0-\nabla f(x_0)\|^2\right]\le\frac{\sigma^2}{b}$, and $S_t\le\frac{(1-\beta)\sigma^2}{1+\beta}$ into (40), and noting that $\mathcal{T}_m\le 0$, we have

$$\frac{1}{2}\sum_{t=0}^{m}\eta_t\mathbb{E}\left[\|\nabla f(x_t)\|^2\right]\le f(x_0)-\mathbb{E}\left[f(x_{m+1})\right]+\frac{\Sigma_m\sigma^2}{2(1-\beta^2)b(m+1)}+\frac{(1-\beta)\sigma^2}{2(1+\beta)}\Sigma_m.$$

Since $\mathbb{E}\left[f(x_{m+1})\right] \geq f^\star$, using this into the last estimate, and multiplying the result by $\frac{2}{\Sigma_m}$, we obtain

$$\frac{1}{\Sigma_m}\sum_{t=0}^{m}\eta_t\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{2}{\Sigma_m}[f(x_0) - f^\star] + \frac{\sigma^2}{(1+\beta)}\left[\frac{1}{b(m+1)(1-\beta)} + (1-\beta)\right]. \tag{43}$$

Since $\left[\frac{1}{b(m+1)(1-\beta)} + (1-\beta)\right] = \left(c_1 + \frac{1}{c_1}\right)\frac{1}{\sqrt{b(m+1)}}$ for $\beta = 1 - \frac{c_1}{\sqrt{b(m+1)}}$, (43) leads to

$$\frac{1}{\Sigma_m}\sum_{t=0}^{m}\eta_t\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{4Lb^{1/4}}{\sqrt{c_1}(m+1)^{3/4}}\left[f(x_0) - f^\star\right] + \left(c_1 + \frac{1}{c_1}\right)\frac{\sigma^2}{\sqrt{b(m+1)}}. \tag{44}$$

The second statement (b) of Theorem 3.2 is proved similarly as in Theorem 3.1 using (44), and we omit the details. □

2.4 The proof of Theorem 3.3: Double-loop with constant step-size

Similar to the proof of (37) in Theorem 3.1, we have

$$\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \leq \frac{2}{\eta}\Big[\mathbb{E}\left[f(x_0^{(s)})\right] - \mathbb{E}\left[f(x_{m+1}^{(s)})\right]\Big] + \frac{2(m+1)\sigma^2}{\sqrt{b(m+1)}}, \tag{45}$$

where we use the superscript s to indicate the stage s in Algorithm 2. Summing up this inequality from $s = 1$ to $s = S$, and then multiplying the result by $\frac{1}{(m+1)S}$ and using $\mathbb{E}\left[f(x_{m+1}^{(S)})\right] \geq f^\star > -\infty$, we get

$$\begin{aligned}\tfrac{1}{S(m+1)}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] &\leq \tfrac{2}{\eta S(m+1)}\left[f(\tilde{x}^0) - f^\star\right] + \tfrac{2\sigma^2}{\sqrt{b(m+1)}}\\ &\leq \tfrac{3Lb^{1/4}}{S(m+1)^{3/4}}\left[f(\tilde{x}^0) - f^\star\right] + \tfrac{2\sigma^2}{\sqrt{b(m+1)}}.\end{aligned} \tag{46}$$

Here, we use the fact that $\eta \geq \frac{2}{3L\left[b(m+1)\right]^{1/4}}$ from (38) in the last inequality.

If we choose $b := \frac{c_1\sigma^2}{\varepsilon^2}$ and $m+1 := \frac{c_2\sigma^2}{\varepsilon^2}$ for some constants $c_1 > 0$ and $c_2 > 0$ and $c_1c_2 > 4$, then, from (46), to guarantee $\frac{1}{S(m+1)}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \leq \varepsilon^2$, we require

$$\begin{aligned}\frac{3Lb^{1/4}}{S(m+1)^{3/4}}\left[f^0 - f^\star\right] + \frac{2\sigma^2}{\sqrt{b(m+1)}} &= \frac{3Lc_1^{1/4}\sigma^{1/2}}{\varepsilon^{1/2}}\cdot\frac{\varepsilon^{3/2}}{Sc_2^{3/4}\sigma^{3/2}}\left[f^0 - f^\star\right] + \frac{2\sigma^2\varepsilon^2}{\sigma^2\sqrt{c_1c_2}} = \varepsilon^2\\ &\Leftrightarrow \frac{3Lc_1^{1/4}\varepsilon}{Sc_2^{3/4}\sigma}\left[f^0 - f^\star\right] = \left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^2\\ &\Leftrightarrow S = \frac{3Lc_1^{1/4}\left[f^0 - f^\star\right]}{c_2^{3/4}\sigma\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon}.\end{aligned}$$

Consequently, the total complexity is

$$\begin{aligned}\mathcal{T}_{ge} &:= (b+3m)S = (c_1+3c_2)\tfrac{\sigma^2}{\varepsilon^2}\tfrac{3Lc_1^{1/4}\left[f^0-f^\star\right]}{c_2^{3/4}\sigma\left(1-\frac{2}{\sqrt{c_1c_2}}\right)\varepsilon}\\ &= \tfrac{3L(c_1+3c_2)c_1^{1/4}\left[f^0-f^\star\right]\sigma}{c_2^{3/4}\left(1-\frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^3} = \mathcal{O}\left(\tfrac{\sigma}{\varepsilon^3}\right).\end{aligned}$$

Since we choose $b := \frac{c_1\sigma^2}{\varepsilon^2}$, the final complexity is $\mathcal{O}\left(\max\left\{\frac{\sigma}{\varepsilon^3}, \frac{\sigma^2}{\varepsilon^2}\right\}\right)$, where other constants independent of σ and ε are hidden. □

C Appendix: The convergence analysis of the mini-batch variants

In this supplementary document, we provide a full analysis of the mini-batch variants of Algorithm 1 and Algorithm 2.

3.1 Variance bound of mini-batch hybrid estimators

For $\hat{v}_t$ defined by (16), we have the following property.

Lemma C.1. *The mini-batch gradient estimator $\hat{v}_t$ defined by* (16) *satisfies*

$$\begin{aligned}\mathbb{E}_{(\mathcal{B}_t,\hat{\mathcal{B}}_t)}\left[\|\hat{v}_t-\nabla f(x_t)\|^2\right] &= \beta_{t-1}^2\|\hat{v}_{t-1}-\nabla f(x_{t-1})\|^2-\rho\beta_{t-1}^2\|\nabla f(x_{t-1})-\nabla f(x_t)\|^2\\ &+\rho\beta_{t-1}^2\mathbb{E}_\xi\left[\|\nabla f(x_t;\xi)-\nabla f(x_{t-1};\xi)\|^2\right]\\ &+(1-\beta_{t-1})^2\rho\sigma^2,\end{aligned}\tag{47}$$

where $\rho=\rho(\hat{b}):=\frac{n-\hat{b}}{(n-1)\hat{b}}$ if $n:=|\Omega|$ is finite, and $\rho(\hat{b}):=\frac{1}{\hat{b}}$, otherwise.

Proof. Let $\hat{v}_t$ be defined by (16). Let $z_t:=\frac{1}{b_t}\sum_{i\in\mathcal{B}_t}(\nabla f_{\xi_i}(x_t)-\nabla f_{\xi_i}(x_{t-1}))$, $\bar{z}:=\nabla f(x_t)-\nabla f(x_{t-1})$, $\Delta_t:=\hat{v}_t-\nabla f(x_t)$, and $\Delta u_t:=u_t-\nabla f(x_t)$. Clearly, we have

$$\mathbb{E}\left[z_t\right]=\bar{z}\quad\text{and}\quad\mathbb{E}\left[\Delta u_t\right]=0.$$

Moreover, we can rewrite $\hat{v}_t$ in (16) as

$$\Delta_t=\beta_{t-1}\Delta_{t-1}+\beta_{t-1}z_t+(1-\beta_{t-1})\Delta u_t-\beta_{t-1}\bar{z}.$$

Therefore, using these two expressions, we can derive

$$\begin{aligned}\mathbb{E}\left[\|\Delta_t\|^2\right] &= \beta_{t-1}^2\|\Delta_{t-1}\|^2+\beta_{t-1}^2\mathbb{E}\left[\|z_t\|^2\right]+(1-\beta_{t-1})^2\mathbb{E}\left[\|\Delta u_t\|^2\right]+\beta_{t-1}^2\|\bar{z}\|^2\\ &+2\beta_{t-1}^2\langle\Delta_{t-1},\mathbb{E}\left[z_t\right]\rangle+2\beta_{t-1}(1-\beta_{t-1})\langle\Delta_{t-1},\mathbb{E}\left[\Delta u_t\right]\rangle-2\beta_{t-1}^2\langle\Delta_{t-1},\bar{z}\rangle\\ &+2\beta_{t-1}(1-\beta_{t-1})\mathbb{E}\left[\langle z_t,\Delta u_t\rangle\right]-2\beta_{t-1}^2\langle\mathbb{E}\left[z_t\right],\bar{z}\rangle-2\beta_{t-1}(1-\beta_{t-1})\langle\mathbb{E}\left[\Delta u_t\right],\bar{z}\rangle\\ &=\beta_{t-1}^2\|\Delta_{t-1}\|^2+\beta_{t-1}^2\mathbb{E}\left[\|z_t\|^2\right]+(1-\beta_{t-1})^2\mathbb{E}\left[\|\Delta u_t\|^2\right]-\beta_{t-1}^2\|\bar{z}\|^2.\end{aligned}\tag{48}$$

For the finite-sum case, after a few elementary calculations, we can show that

$$\mathbb{E}\left[\|z_t\|^2\right]=\frac{n(b_t-1)}{(n-1)b_t}\|\bar{z}\|^2+\frac{(n-b_t)}{(n-1)b_t}\mathbb{E}_\xi\left[\|\nabla f_\xi(x_t)-\nabla f_\xi(x_{t-1})\|^2\right].$$

For the expectation case, we have

$$\mathbb{E}\left[\|z_t\|^2\right]=\big(1-\frac{1}{b_t}\big)\|\bar{z}\|^2+\frac{1}{b_t}\mathbb{E}_\xi\left[\|\nabla f_\xi(x_t)-\nabla f_\xi(x_{t-1})\|^2\right].$$

In addition, under Assumption 1.1(c), we have $\mathbb{E}\left[\|\Delta u_t\|^2\right]\leq\rho\sigma^2$.

Substituting one of the two last expressions and the bound of $\mathbb{E}\left[\|\Delta u_t\|^2\right]$ into (48), we get (47). □

The following analysis is given under fixed mini-batch sizes when we choose $\hat{b}_t=\tilde{b}_t=\hat{b}$. Similar to Lemma 2.2, we can bound the variance $\mathbb{E}\left[\|\hat{v}_t-\nabla f(x_t)\|^2\right]$ of the mini-batch hybrid estimator $\hat{v}_t$ from (16) in the following lemma.

Lemma C.2. *Assume that $f(\cdot,\cdot)$ is L-smooth and u_t is an SGD estimator, $\hat{v}$ is given in* (16)*, $\mathcal{B}_t$ and $\hat{\mathcal{B}}_t$ are mini-batches of the size $\hat{b}$. Then, we have the following upper bound on the variance $\mathbb{E}\left[\|\hat{v}_t-\nabla f(x_t)\|^2\right]$:*

$$\mathbb{E}\left[\|\hat{v}_t-\nabla f(x_t)\|^2\right]\leq\omega_t\mathbb{E}\left[\|\hat{v}_0-\nabla f(x_0)\|^2\right]+L^2\rho\sum_{i=0}^{t-1}\omega_{i,t}\mathbb{E}\left[\|x_{i+1}-x_i\|^2\right]+\rho S_t,\tag{49}$$

where the expectation is taking over all the randomness $\mathcal{F}_t := \sigma(v_0, v_1, \cdots, v_t)$, $\omega_t := \prod_{i=1}^{t} \beta_{i-1}^2$, $\omega_{i,t} := \prod_{j=i+1}^{t} \beta_{j-1}^2$ *for* $i = 0, \cdots, t$, *and* $S_t := \sum_{i=0}^{t-1} \big(\prod_{j=i+2}^{t} \beta_{j-1}^2\big)(1-\beta_i)^2\sigma$ *for* $t \geq 0$. $\rho = \frac{n-\tilde{b}}{\tilde{b}(n-1)}$ *if* $|\Omega|$ *is finite and* $\rho = \frac{1}{\tilde{b}}$ *otherwise.*

Proof. From Lemma C.1, taking the expectation with respect to $\mathcal{F}_t := \sigma(v_0, v_1, \cdots, v_t)$, we have

$$\begin{aligned}\mathbb{E}\left[\|\hat{v}_t - \nabla f(x_t)\|^2\right] &\leq \beta_{t-1}^2 \mathbb{E}\left[\|\hat{v}_{t-1} - \nabla f(x_{t-1})\|^2\right] \\ &\quad + L^2\rho\beta_{t-1}^2 \mathbb{E}\left[\|x_t - x_{t-1}\|^2\right] + \rho(1-\beta_{t-1})^2\sigma^2.\end{aligned}$$

Let $a_t^2 := \mathbb{E}\left[\|\hat{v}_t - \nabla f(x_t)\|^2\right]$ and $r_t^2 = \mathbb{E}\left[\|x_{t+1} - x_t\|^2\right]$. By following inductive step as in the proof of Lemma 2.2, we obtain

$$\begin{aligned}a_t^2 \leq &\left(\beta_{t-1}^2 \cdots \beta_0^2\right) a_0^2 + L^2\rho\left(\beta_{t-1}^2 \cdots \beta_0^2\right) r_0^2 + \cdots + L^2\rho\beta_{t-1}^2 r_{t-1}^2 \\ &+ \rho\left[\left(\beta_{t-1}^2 \cdots \beta_1^2\right)(1-\beta_0)^2 + \cdots + (1-\beta_{t-1})^2\right]\sigma^2.\end{aligned}$$

Using the definition of ω_t, $\omega_{i,t}$, and S_t in Lemma 2.2, the previous inequality becomes

$$a_t^2 \leq \omega_t a_0^2 + L^2\rho \sum_{i=0}^{t-1} \omega_{i,t} r_i^2 + \rho S_t,$$

which is the same as (49). □

3.2 The proof of Corollary 3.1: Single loop with constant step-size and mini-batches

Using Lemma C.2 and following the same path of proof of Lemma B.1, we can show that

$$\begin{aligned}\mathbb{E}\left[f(x_{m+1}\right] &\leq \mathbb{E}\left[f(x_0)\right] - \sum_{t=0}^{m} \frac{\eta}{2}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] + \frac{\eta}{2}\left(\sum_{t=0}^{m}\omega_t\right)\mathbb{E}\left[\|\hat{v}_0 - \nabla f(x_0)\|^2\right] \\ &\quad + \tfrac{\rho\eta}{2}\sum_{t=0}^{m} S_t + \frac{1}{2}\widehat{\mathcal{T}}_m,\end{aligned} \tag{50}$$

where

$$\widehat{\mathcal{T}}_m := \rho L^2\eta^3 \sum_{t=0}^{m}\sum_{i=0}^{t-1} \omega_{i,t}\mathbb{E}\left[\|\hat{v}_i\|^2\right] - \eta\sum_{t=0}^{m}\big(1 - L\eta\big)\mathbb{E}\left[\|\hat{v}_t\|^2\right].$$

Clearly, we can rewrite $\widehat{\mathcal{T}}_m$ as

$$\begin{aligned}\widehat{\mathcal{T}}_m &= \eta\Big[\frac{\beta^2(1-\beta^{2m})L^2\eta^2\rho}{1-\beta^2} - (1-L\eta)\Big]q_0^2 \\ &\quad + \eta\Big[\tfrac{\beta^2(1-\beta^{2(m-1)})L^2\eta^2\rho}{1-\beta^2} - (1-L\eta)\Big]q_1^2 + \cdots \\ &\quad + \eta\Big[\frac{\beta^2(1-\beta^2)L^2\eta^2\rho}{1-\beta^2} - (1-L\eta)\Big]q_{m-1}^2 - \eta(1-L\eta)q_m^2,\end{aligned}$$

where $q_t^2 := \mathbb{E}\left[\|\hat{v}_t\|^2\right]$. To guarantee $\widehat{\mathcal{T}}_m \leq 0$, we need to have

$$\begin{cases} \dfrac{L^2\eta^2\rho\beta^2(1-\beta^{2m})}{1-\beta^2} - (1-L\eta) & \leq 0 \\ \dfrac{L^2\eta^2\rho\beta^2(1-\beta^{2(m-1)})}{1-\beta^2} - (1-L\eta) & \leq 0 \\ \cdots & \cdots \\ \dfrac{L^2\eta^2\rho\beta^2(1-\beta^2)}{1-\beta^2} - (1-L\eta) & \leq 0 \\ -(1-L\eta) & \leq 0. \end{cases}$$

Let $\alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2}$. Since $\alpha_1^2 < \alpha_2^2 < \cdots < \alpha_m^2$, the last condition holds if $L^2\eta^2\rho\alpha_m^2 - (1-L\eta) \leq 0$. By tightening this condition, we obtain

$$\eta := \frac{2}{L\left(1+\sqrt{1+4\rho\alpha_m^2}\right)} \text{ with } \alpha_m^2 := \frac{\beta^2(1-\beta^{2m})}{1-\beta^2},$$

which is exactly (17). Since $\alpha_m^2 \leq \frac{\beta^2}{1-\beta^2}$, we have $\eta \geq \underline{\eta} := \frac{2\sqrt{1-\beta^2}}{L(\sqrt{1-\beta^2}+\sqrt{1+\beta^2(4\rho-1)})}$.

Next, we can reuse the following estimates as in the proof of Theorem 3.1:

$$\begin{aligned}\sum_{t=0}^{m} S_t &\leq \frac{\sigma^2(1-\beta)(m+1)}{1+\beta} \\ \sum_{t=0}^{m} \omega_t &= \frac{1-\beta^{2(m+1)}}{1-\beta^2} \leq \frac{1}{1-\beta^2}.\end{aligned}$$

Combining these estimate into (50) and notting that $\widehat{\mathcal{T}}_m \leq 0$ and $\mathbb{E}\left[\|v_0 - \nabla f(x_0)\|^2\right] \leq \frac{\sigma^2}{b}$, we can show that

$$\begin{aligned}\mathbb{E}\left[f(x_{m+1})\right] &\overset{(31)}{\leq} \mathbb{E}\left[f(x_0)\right] - \tfrac{\eta}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \\ &\quad + \tfrac{\eta\sigma^2}{2(1+\beta)}\Big[\tfrac{1}{(1-\beta)b} + \rho(1-\beta)(m+1)\Big].\end{aligned} \tag{51}$$

Note that $\mathbb{E}\left[f(x_{m+1})\right] \geq f^\star > -\infty$, (51) can be rewritten as

$$\frac{\eta}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \mathbb{E}\left[f(x_0)\right] - f^\star + \frac{\eta\sigma^2}{2(1+\beta)}\Big[\frac{1}{(1-\beta)b} + \rho(1-\beta)(m+1)\Big]. \tag{52}$$

If we choose $\beta := 1 - \frac{c_1}{\sqrt{\hat{\rho}b(m+1)}}$ for any $0 < c_1 < \sqrt{b(m+1)}$ such that

$$\frac{1}{(1-\beta)b(m+1)} + \rho(1-\beta) = \left(c_1 + \frac{1}{c_1}\right)\sqrt{\frac{\rho}{b(m+1)}},$$

then (52) leads to

$$\begin{aligned}\tfrac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] &\leq \tfrac{L}{(m+1)}\left(\tfrac{\sqrt{1-\beta^2}+\sqrt{1+\beta^2(4\rho-1)}}{\sqrt{1-\beta^2}}\right)\Big[\mathbb{E}\left[f(x_0)\right] - f^\star\Big] \\ &\quad + \Big(c_1 + \tfrac{1}{c_1}\Big)\tfrac{\sigma^2}{(1+\beta)}\sqrt{\tfrac{\rho}{b(m+1)}}.\end{aligned} \tag{53}$$

Since $\beta = 1 - \frac{c_1}{\sqrt{\rho b(m+1)}} < 1$ and if we choose b, $\hat{b}$, and m such that $\rho b(m+1) > c_1^2$, we have

$$1-\beta^2 = 1 - \left(1 - \tfrac{c_1}{\sqrt{\rho b(m+1)}}\right)^2 = \tfrac{2c_1}{\sqrt{\rho b(m+1)}} - \tfrac{c_1^2}{\rho b(m+1)} = \tfrac{2c_1\sqrt{\rho b(m+1)}-c_1^2}{\rho b(m+1)} > \tfrac{2c_1}{\sqrt{\rho b(m+1)}},$$

and $\sqrt{1-\beta^2} + \sqrt{1+\beta^2(4\rho-1)} \leq 1 + \sqrt{1+3\beta^2} \leq 3$ since $\rho \leq 1$. Therefore, we can bound η as

$$\eta \geq \underline{\eta} \geq \frac{2c_1}{3L\big[\rho b(m+1)\big]^{1/4}}.$$

Therefore, the inequality (53) can be rewritten as

$$\begin{aligned}\tfrac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] &\leq \tfrac{3L(\rho b)^{1/4}}{2c_1(m+1)^{3/4}}\left(\mathbb{E}\left[f(x_0)\right] - f^\star\right) + \Big(c_1 + \tfrac{1}{c_1}\Big)\tfrac{\sigma^2}{(1+\beta)}\sqrt{\tfrac{\rho}{b(m+1)}}. \\ &\leq \tfrac{3L(\rho b)^{1/4}}{2c_1(m+1)^{3/4}}\left(\mathbb{E}\left[f(x_0)\right] - f^\star\right) + \Big(c_1 + \tfrac{1}{c_1}\Big)\tfrac{\sigma^2}{2}\sqrt{\tfrac{\rho}{b(m+1)}}.\end{aligned}$$

Let $f^0 := \mathbb{E}\left[f(x_0)\right]$. We can write the bound as

$$\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \le \frac{3L(\rho b)^{1/4}}{2c_1(m+1)^{3/4}}\left(f^0 - f^\star\right) + \left(c_1 + \frac{1}{c_1}\right)\frac{\sigma^2}{2}\sqrt{\frac{\rho}{b(m+1)}}.$$

Let us choose $b := c_2\sigma^{8/3}(\rho(m+1))^{1/3}$ for some $c_2 > 0$. Then, the last inequality leads to

$$\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \le \frac{\rho^{1/3}\sigma^{2/3}}{(m+1)^{2/3}}\left[\frac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{2\sqrt{c_2}}\right]. \tag{54}$$

From (54), to guarantee $\mathbb{E}\left[\|\nabla f(\widetilde{x}_m)\|^2\right] \le \varepsilon^2$, we need to choose

$$\frac{\rho^{1/3}\sigma^{2/3}}{(m+1)^{2/3}}\left[\frac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{2\sqrt{c_2}}\right] \le \varepsilon^2,$$

which leads to

$$m+1 \ge \frac{\rho^{1/2}\sigma}{\varepsilon^3}\left[\frac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \left(c_1 + \frac{1}{c_1}\right)\frac{1}{2\sqrt{c_2}}\right]^{3/2}.$$

Hence, we can choose m as in (18).

Finally, let $c_1 = 1$. Then the number of stochastic gradient evaluations $\mathcal{T}_{ge}$ is

$$\begin{aligned}
\mathcal{T}_{ge} &= b + 3\hat{b}m \le b + \tfrac{3(m+1)}{\rho}\\
&\le c_2\sigma^{8/3}\left[\rho(m+1)\right]^{1/3} + \tfrac{3\sigma}{\rho^{1/2}\varepsilon^3}\left[\tfrac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \tfrac{1}{\sqrt{c_2}}\right]^{3/2}\\
&\le \tfrac{\rho^{1/2}\sigma^3}{\varepsilon}\left[\tfrac{3Lc_2^{9/4}}{2c_1}\left(f^0 - f^\star\right) + c_2^{3/2}\right]^{1/2} + \tfrac{3\sigma}{\rho^{1/2}\varepsilon^3}\left[\tfrac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \tfrac{1}{\sqrt{c_2}}\right]^{3/2},
\end{aligned}$$

which proves (19), where $\rho \le \frac{1}{\hat{b}}$ if $|\Omega|$ is infinite and $\rho := \frac{n-\hat{b}}{\hat{b}(n-1)}$ if $|\Omega|$ is finite. In particular, if $|\Omega|$ is infinite and we choose $\rho := \frac{c_3^2}{\sigma^2\varepsilon^2}$ for some $c_3 \le \sigma\varepsilon$, then

$$\mathcal{T}_{ge} = \frac{c_3\sigma^2}{\varepsilon^2}\left[\frac{3Lc_2^{9/4}}{2c_1}\left(f^0 - f^\star\right) + c_2^{3/2}\right]^{1/2} + \frac{3\sigma^2}{c_3\varepsilon^2}\left[\frac{3Lc_2^{1/4}}{2c_1}\left(f^0 - f^\star\right) + \frac{1}{\sqrt{c_2}}\right]^{3/2}.$$

Hence, we obtain $\mathcal{T}_{ge} = \mathcal{O}\left(\left(c_3 + \frac{1}{c_3}\right)\frac{\sigma^2}{\varepsilon^2}\right)$. □

3.3 The mini-batch variant of Algorithm 2 and its complexity

Let us consider a mini-batch variant of Algorithm 2. Similar to Theorem 3.3, we can prove the following result.

Corollary C.1. *Let $\{x_t^{(s)}\}_{t=0\to m}^{s=1\to S}$ be the sequence generated by the mini-batch variant of Algorithm 2 using constant step-size η defined in* (17) *with $c_1 := 1$. Then, the following estimate holds*

$$\frac{1}{S(m+1)}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \le \frac{3L\rho(\hat{b})b^{1/4}}{S(m+1)^{3/4}}\left[f(\tilde{x}^0) - f^\star\right] + \frac{2\sigma^2\sqrt{\rho(\hat{b})}}{\sqrt{b(m+1)}}. \tag{55}$$

Let $\widetilde{x}_T \sim \mathbf{U}(\{x_t^{(s)}\}_{t=0\to m}^{s=1\to S})$. If we choose $b := \frac{c_1\sigma^2}{\varepsilon^2}$ and $\frac{m+1}{\hat{b}} := \frac{c_2\sigma^2}{\hat{b}^2\varepsilon^2}$ for some constants $c_1 > 0$ and $c_2 > 0$ and $c_1c_2 > 4$, then, to guarantee $\mathbb{E}\left[\|\nabla f(\widetilde{x}_T)\|^2\right] \le \varepsilon^2$, we require

$$S := \frac{3Lc_1^{1/4}\left[f(\widetilde{x}^0) - f^\star\right]}{c_2^{3/4}\hat{b}^{3/2}\sigma\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon}. \tag{56}$$

Consequently, the total number of stochastic gradient evaluations $\mathcal{T}_{ge}$ does not exceed

$$\mathcal{T}_{ge} := \left(b + 3\lfloor \tfrac{m}{\hat{b}} \rfloor\right) S = \frac{3L(c_1+3c_2)c_1^{1/4}\left[f(\widetilde{x}^0) - f^\star\right]\sigma}{c_2^{3/4}\hat{b}^{3/2}\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^3} = \mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right). \tag{57}$$

Proof. First, similar to the proof of (54), we have

$$\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \leq \frac{3L(\rho b)^{1/4}}{(m+1)^{3/4}}\left(\mathbb{E}\left[f(x_0^{(s)})\right] - \mathbb{E}\left[f(x_{m+1}^{(s)})\right]\right) + 2\sigma^2\sqrt{\frac{\rho}{b(m+1)}}.$$

Summing up this inequality from $s=1$ to $s=S$ and then using $\mathbb{E}\left[f(x_{m+1}^{(S)})\right] \geq f^\star$ and $\widetilde{x}_0 := x_0^{(1)}$, we can show that

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \leq \frac{3L(\rho b)^{1/4}}{S(m+1)^{3/4}}\left(\mathbb{E}\left[f(\widetilde{x}_0\right] - f^\star\right) + 2\sigma^2\sqrt{\frac{\rho}{b(m+1)}}.$$

If we choose $b := \frac{c_1\sigma^2}{\hat{b}^2\varepsilon^2}$ and $\frac{m+1}{\hat{b}} := \frac{c_2\sigma^2}{\varepsilon^2}$ for some constants $c_1 > 0$ and $c_2 > 0$ and $c_1c_2 > 4$, then, $\rho(m+1) = \frac{c_2\sigma^2}{\varepsilon^2}$, $b = \frac{c_1\rho^2\sigma^2}{\varepsilon^2}$, and from (46), to guarantee $\frac{1}{S(m+1)}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla f(x_t^{(s)})\|^2\right] \leq \varepsilon^2$, we require

$$\begin{aligned}
\frac{3L(\rho b)^{1/4}}{S(m+1)^{3/4}}\left[f^0 - f^\star\right] + \frac{2\sigma^2\sqrt{\rho}}{\sqrt{b(m+1)}} &= \frac{3Lc_1^{1/4}\rho^{3/4}\sigma^{1/2}}{\varepsilon^{1/2}}\cdot\frac{\rho^{3/4}\varepsilon^{3/2}}{Sc_2^{3/4}\sigma^{3/2}}\left[f^0 - f^\star\right] + \frac{2\sigma^2\varepsilon^2}{\sigma^2\sqrt{c_1c_2}} = \varepsilon^2 \\
&\Leftrightarrow \frac{3Lc_1^{1/4}\rho^{3/2}\varepsilon}{Sc_2^{3/4}\sigma}\left[f^0 - f^\star\right] = \left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^2 \\
&\Leftrightarrow S = \frac{3L\rho^{3/2}c_1^{3/4}\left[f^0 - f^\star\right]}{c_2^{3/4}\sigma\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon}.
\end{aligned}$$

Consequently, the total complexity is

$$\begin{aligned}
\mathcal{T}_{ge} &:= (b + 3\tfrac{m}{\hat{b}})S \leq (c_1+3c_2)\tfrac{\sigma^2}{\varepsilon^2}\frac{3Lc_1^{1/4}\rho^{3/2}\left[f^0 - f^\star\right]}{c_2^{3/4}\sigma\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon} \\
&= \frac{3L(c_1+3c_2)c_1^{1/4}\left[f^0 - f^\star\right]\sigma}{c_2^{3/4}\hat{b}^{3/2}\left(1 - \frac{2}{\sqrt{c_1c_2}}\right)\varepsilon^3} = \mathcal{O}\left(\frac{\sigma}{\varepsilon^3}\right).
\end{aligned}$$

Since we choose $b\hat{b}^2 := \frac{c_1\sigma^2}{\varepsilon^2}$ which shows that $b \leq \frac{c_1\sigma^2}{\varepsilon^2}$, the final complexity is $\mathcal{O}\left(\max\left\{\frac{\sigma}{\varepsilon^3}, \frac{\sigma^2}{\varepsilon^2}\right\}\right)$, where other constants independent of σ and ε are hidden. □

D Appendix: Additional numerical experiments

In this subsection, we provide more numerical examples on two examples we tested in the main text.

4.1 Experiment setup

Our algorithms: We implement the following variants of Algorithm 1 and Algorithm 2 in Python:

- **Single-loop algorithms**: We consider different variants of the single-loop algorithm, Algorithm 1. We denote them by `Hybrid-SGD-SL` for constant step-size variants, and `Hybrid-SGD-ASL` for adaptive step-size variants.

- **Double-loop algorithms**: These are variants of Algorithm 2. We denote them by `Hybrid-SGD-DL[1-3]` the variants corresponding to different snapshot gradient batch-sizes of $b = n^{2/3}$, $b = 0.1n$, and $b = n$. We also denote `Hybrid-SGD-DL` as the best variants among these three choices of the batch-size for snapshot gradient.

Competitors: We also compare our methods with the most state-of-the-art candidates from the literature. We ignore other variants since their complexity bound is worse than ours and they use complicated routines for hyper-parameter selection.

- Stochastic gradient descent (SGD): We test two variants of SGD. The first one, called SGD1, is with constant step-size $\eta_t := \frac{0.1}{L}$. The second variant, called SGD2, is with an adaptive step-size of the form $\eta_t := \frac{\eta_0}{1+\eta'\lfloor t/n \rfloor}$, where $\eta_0 > 0$ and $\eta' \geq 0$ are carefully tuned to obtain the best performance. In our tests, we use $\eta_0 := \frac{0.1}{L}$ and $\eta' := 1$.
- SVRG: This algorithmic variant is from [23], where its theoretical step-size in the single sample case is $\eta_t := \frac{1}{3nL}$, and in the mini-batch case is $\eta_t := \frac{1}{3L}$.
- SVRG+: This is a variant of SVRG studied in [11]. Its theoretical step-size in the single sample case is $\eta_t := \frac{1}{6nL}$, and in the mini-batch case is $\eta_t := \frac{1}{6L}$.
- SPIDER: SPIDER [8] is a stochastic gradient method using SARAH estimator (also called Stochastic Path-Integrated Differential EstimatoR). This method achieves the best-known complexity as Algorithm 2 but uses very different step-size $\eta_t := \min\left\{\frac{\epsilon}{Ln_0\|v^k\|}, \frac{1}{2Ln_0}\right\}$ where $n_0 = \frac{n^{1/2}}{\hat{b}}$ with $\hat{b}$ is a given mini-batch size in the range $[1, \sqrt{n}]$.
- SpiderBoost: SpiderBoost [27] is a modification of SPIDER by using a large constant step-size $\eta_t := \frac{1}{2L}$, but requires to set very specific mini-batch $\hat{b} = \lfloor\sqrt{n}\rfloor$ to achieve the best-known complexity as in Algorithm 2.

Problems: We consider three examples: The first one is the logistic regression with non-convex regularizer as in (20). The second example is a binary classification with non-convex loss as in (21).

Datasets: All the datasets used in this paper are downloaded from LibSVM [6] at https://www.csie.ntu.edu.tw/ cjlin/libsvm/. We select 6 datasets: `w8a` ($n = 49,749, p = 300$), `rcv1.binary` ($n = 20,242, p = 47,236$), `real-sim` ($n = 72,309, p = 20,958$), `news20.binary` ($n = 19,996, p = 1,355,191$), `url_combined` ($n = 2,396,130, p = 3,231,961$), and `epsilon` ($n = 400,000, p = 2,000$).

4.2 Logistic regression with non-convex regularizer

In this section, we add more numerical examples to solve problem (20). Together with the convergence of the trainning loss and gradient norms in Fig. 1, the training and test accuracies are also plotted in Fig. 4 for three datasets: `w8a`, `rcv1.binary`, and `real-sim`.

As we can observe from Fig. 4, for `w8a`, all the algorithms except for SVRG achieve similar training accuracy as well as test accuracy. SVRG eventually reaches the same accuracy after around 17 epochs. For `rcv1.binary`, HybridSGD variants, SGD2, and SVRG+ have similar training and test accuracies, but SGD2 is more oscillated than the other methods. SGD1 performs worse than our methods in this case. Both SPIDER and SVRG still perform poorly. For `real-sim`, although our methods, SGD1, and SGD2 achieve lower training accuracy, they are able to reach better test accuracy than SVRG+.

In addition, the training and testing accuracies of the mini-batch case are presented in Fig. 5, where the relative residual of the train loss and the gradient norms are shown in Fig. 2. Again, our methods achieve training and test accuracies consistently with SGD2 in `w8a` and `real-sim`, while having better accuracy in `rcv1.binary`.

We also run SVRG, SVRG+, SpiderBoost, and our double-loop variant (Algorithm 2) on three datasets: `w8a`, `rcv1.binary`, and `real-sim`. The results are plotted in Fig. 6.

In this experiment, our double-loop variant and SpiderBoost outperform SVRG and SVRG+. Although the step-size of SVRG+ is $\eta = \frac{1}{6L}$ which is smaller than $\frac{1}{3L}$ in SVRG, SVRG+ still

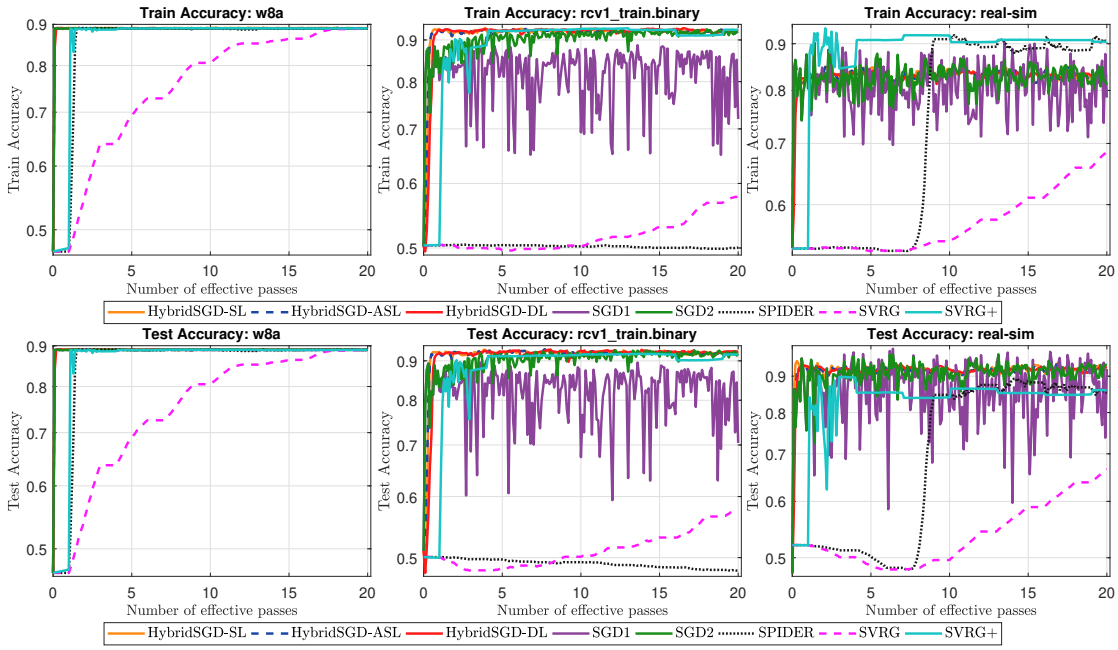

Figure 4: The training and test accuracies of (21) on three datasets: Single-sample case.

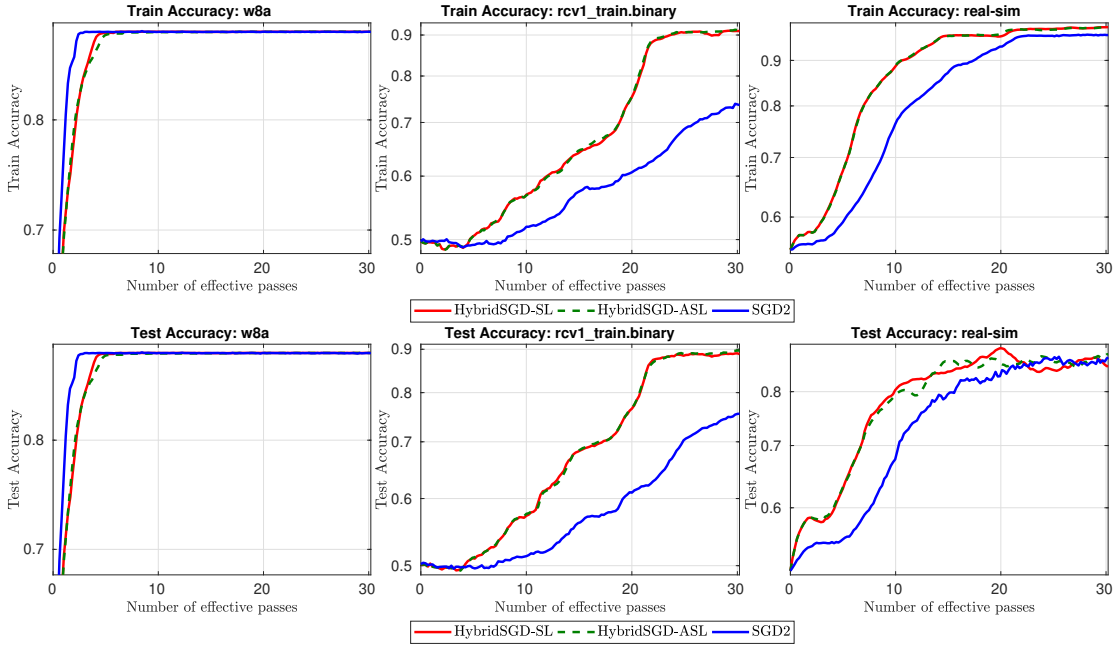

Figure 5: The training and test accuracies of (21) on three datasets: Mini-batch case.

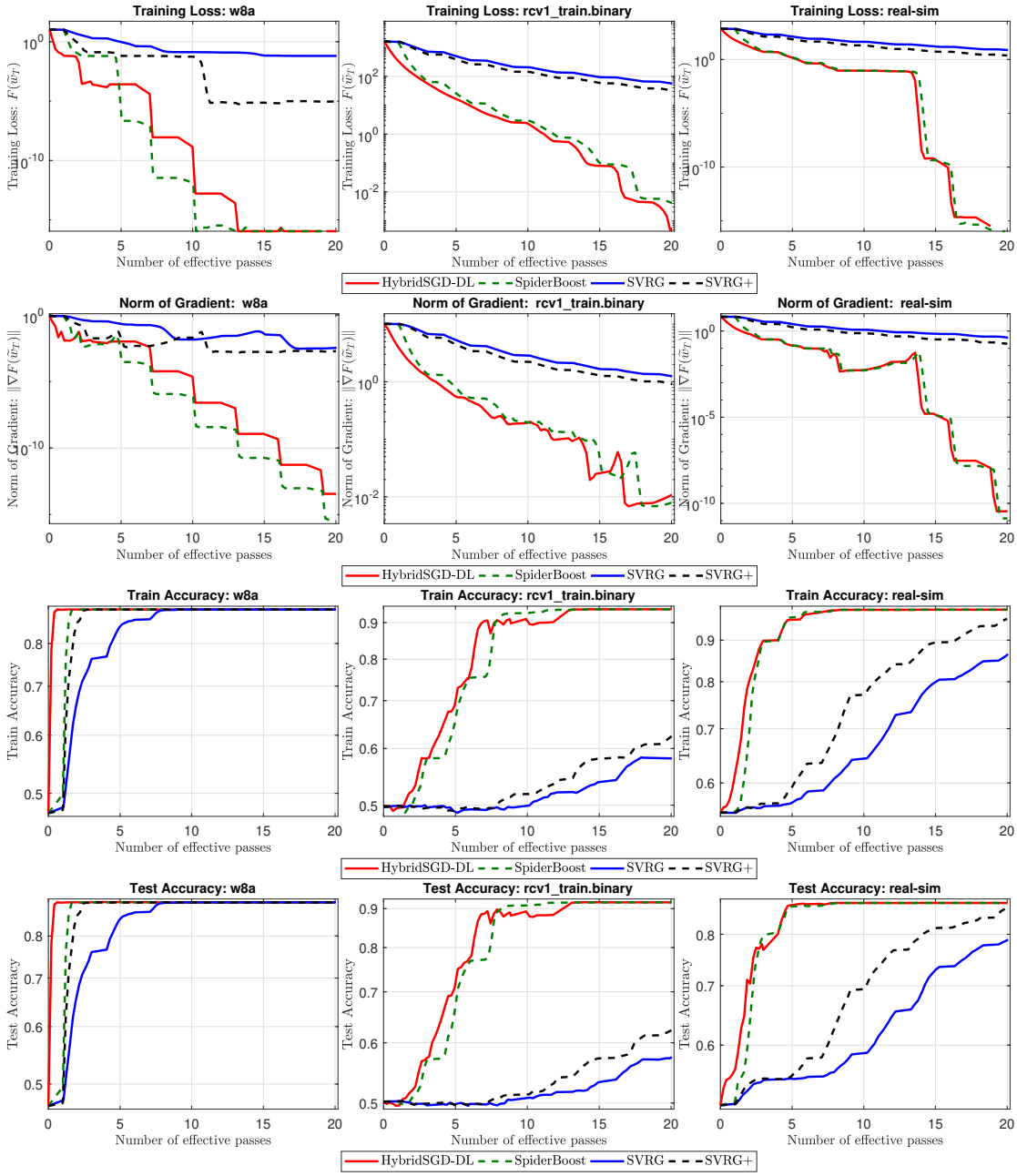

Figure 6: The results of 4 algorithms for solving (20): Mini-batch case.

performs better than SVRG. SpiderBoost uses a large step-size $\eta = \frac{1}{2L}$ and it indeed performs slightly better than ours in the `w8a` dataset, but is comparable in other two. Note that our step-size η is selected based on our theory in Theorem 3.3.

Finally, we conduct experiment on three larger datasets: `epsilon`, `url_combined`, and `news20.binary`. Since the the sample sizes are large, we only run mini-batch variants. The results of the single-loop variants are shown in Fig. 7.

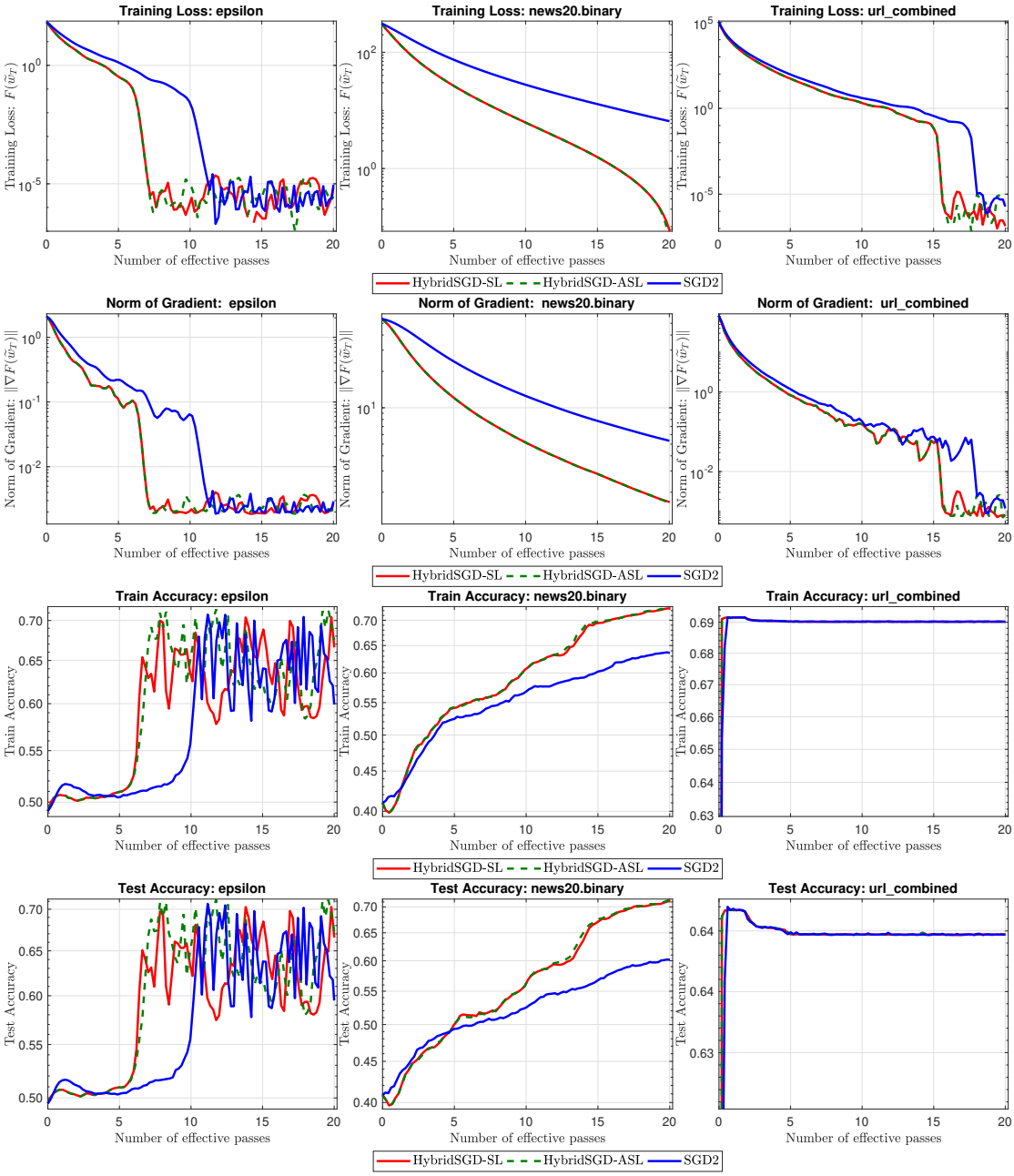

Figure 7: The results of 3 single-loop algorithms for solving (20) on large datasets: Mini-batch case.

We can observe from Fig. 7 that our single loop variants outperform SGD in all three datasets. Note that the performance of the adaptive step-size variant is similar to its fixed step-size one.

The results of the double-loop variants are also shown in Fig. 8.

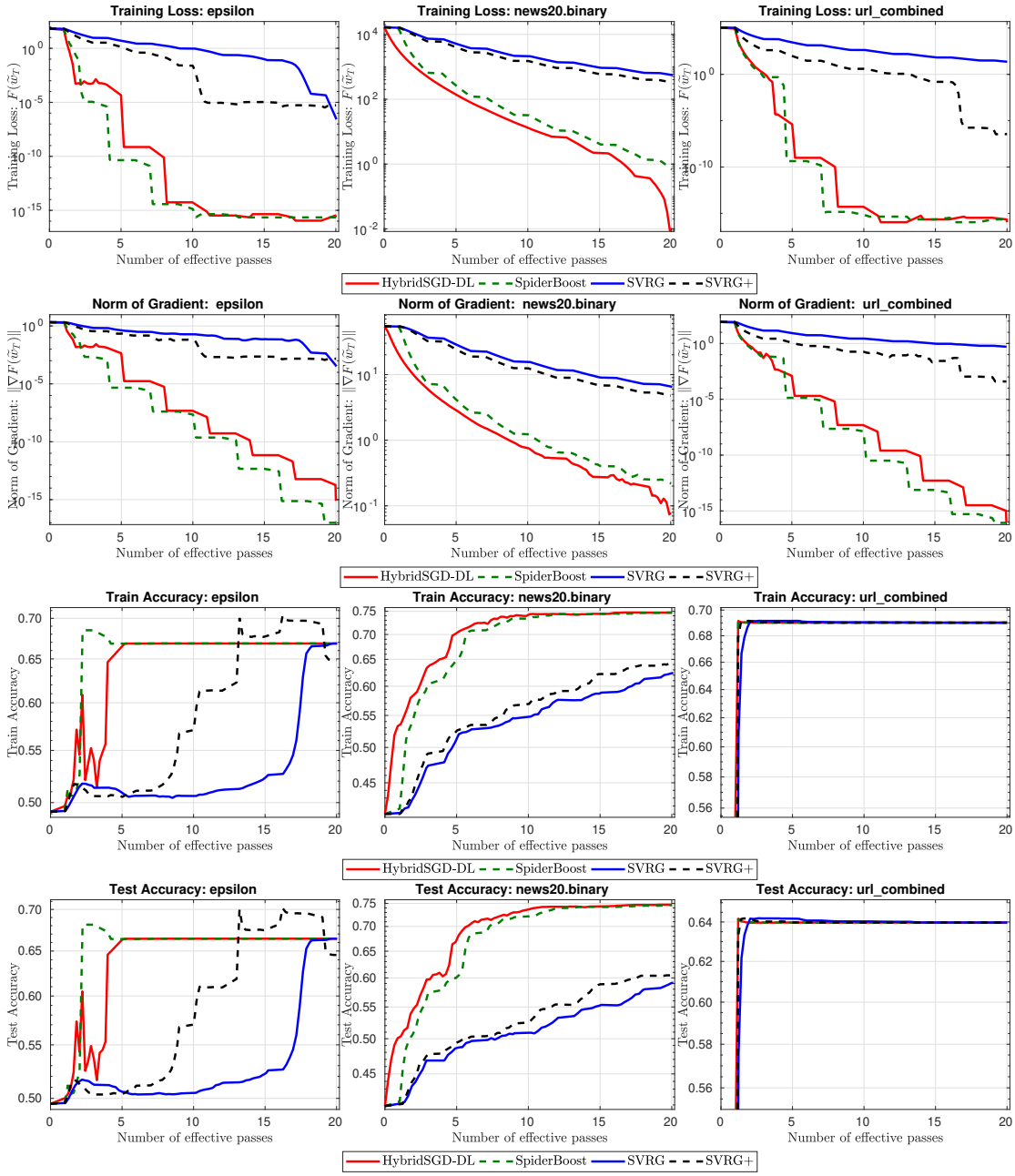

Figure 8: The results of 4 double-loop algorithms for solving (20) on large datsets: Mini-batch case.

Clearly, our double-loop variants achieve better performance than SVRG and SVRG+ due to better convergence rate. SpiderBoost is slightly better than ours in the dataset `epsilon` while they are comparable in the last two datasets since we have the same best-known convergence rate as SpiderBoost.

4.3 Binary classification involving non-convex loss and Tikhonov's regularizer

We also conduct additional experiments to test our algorithms for solving (21). We use two different non-convex loss functions as in [29] apart from the one used in the main text, which are:

- *Nonconvex loss in two-layer neural networks*: $\ell_1(\tau, s) = \left(1 - \frac{1}{1+\exp(-\tau s)}\right)^2$.
- *Logistic difference loss*: $\ell_2(\tau, s) = \log(1 + \exp(\tau s)) - \log(1 + \exp(-\tau s - 1))$.

These functions are smooth and satisfy Assumption 1.1.

Let us first test our algorithms and other methods on three datasets: `w8a`, `rcv1.binary`, and `real-sim` using single-sample setting. The results are plotted in Fig. 9 and Fig. 10. In this test, HybridSGD-DL achieves the best performance followed by HybridSGD-SL and HybridSGD-ASL. SPIDER has decent performance in the last two datasets. SGD variants also have good performance in all datasets while SGD2 is better than its fixed step-size variant. SVRG+ also has comparable performance with SGD2 whereas SVRG cannot achieve fast convergence due to its small step-size.

Next, we test mini-batch variants. On the one hand, we compare our single-loop variants HybridSGD-SL and HybridSGD-ASL with SGD. On the other hand, we compare our double-loop variants with SVRG, SVRG+, and SpiderBoost. The results for solving (21) with loss ℓ_1 are shown in Fig. 11 and Fig. 12 whereas Fig. 13 and Fig. 14 present the results when using loss ℓ_2.

Additionally, we repeat the experiments on three larger datasets: `epsilon`, `news20.binary`, and `ulr_combined`. The results are shown in Fig. 15, 16, 17, and 18.

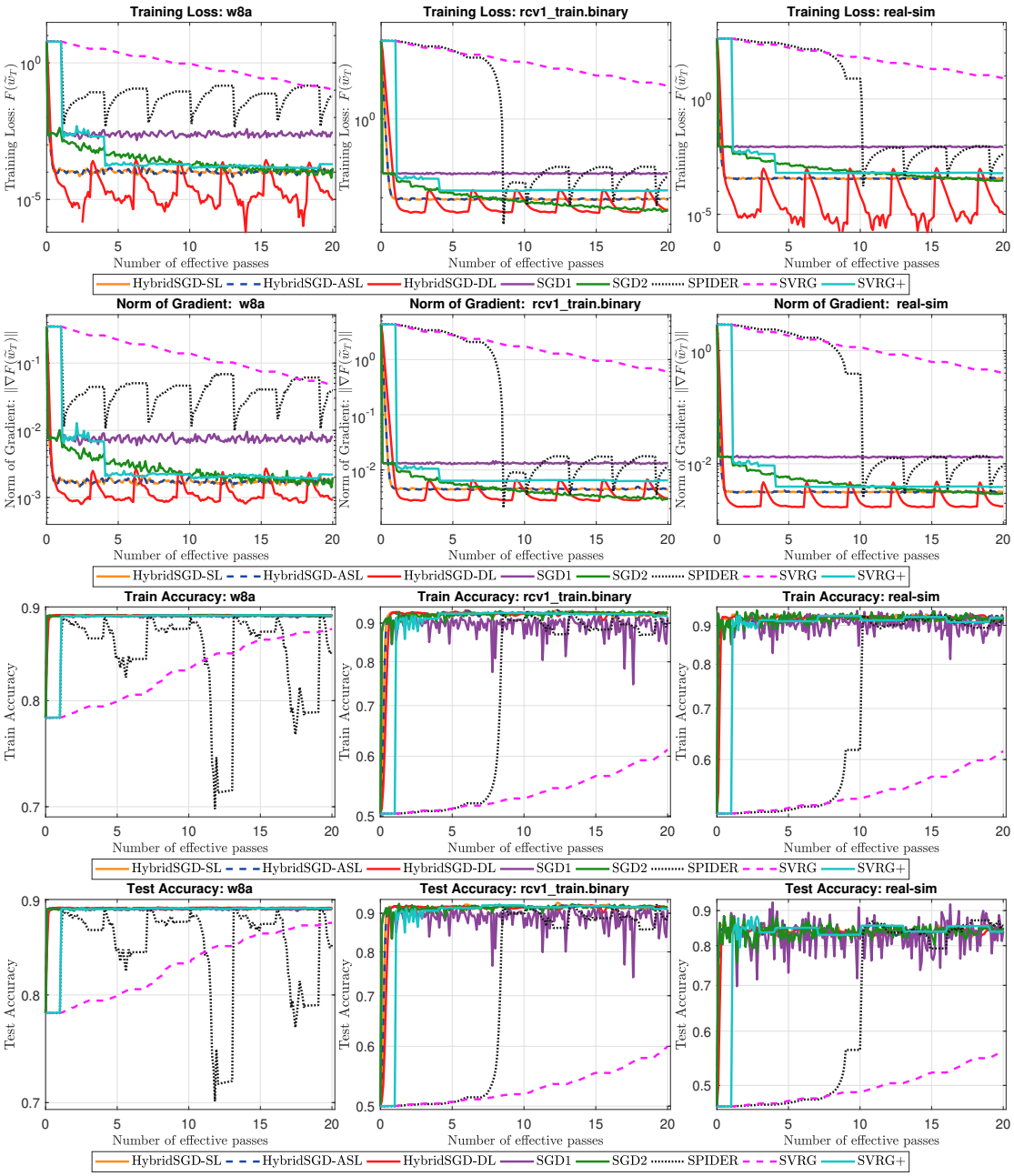

Figure 9: The training loss and gradient norms of (21) with loss ℓ_1: Single-sample.

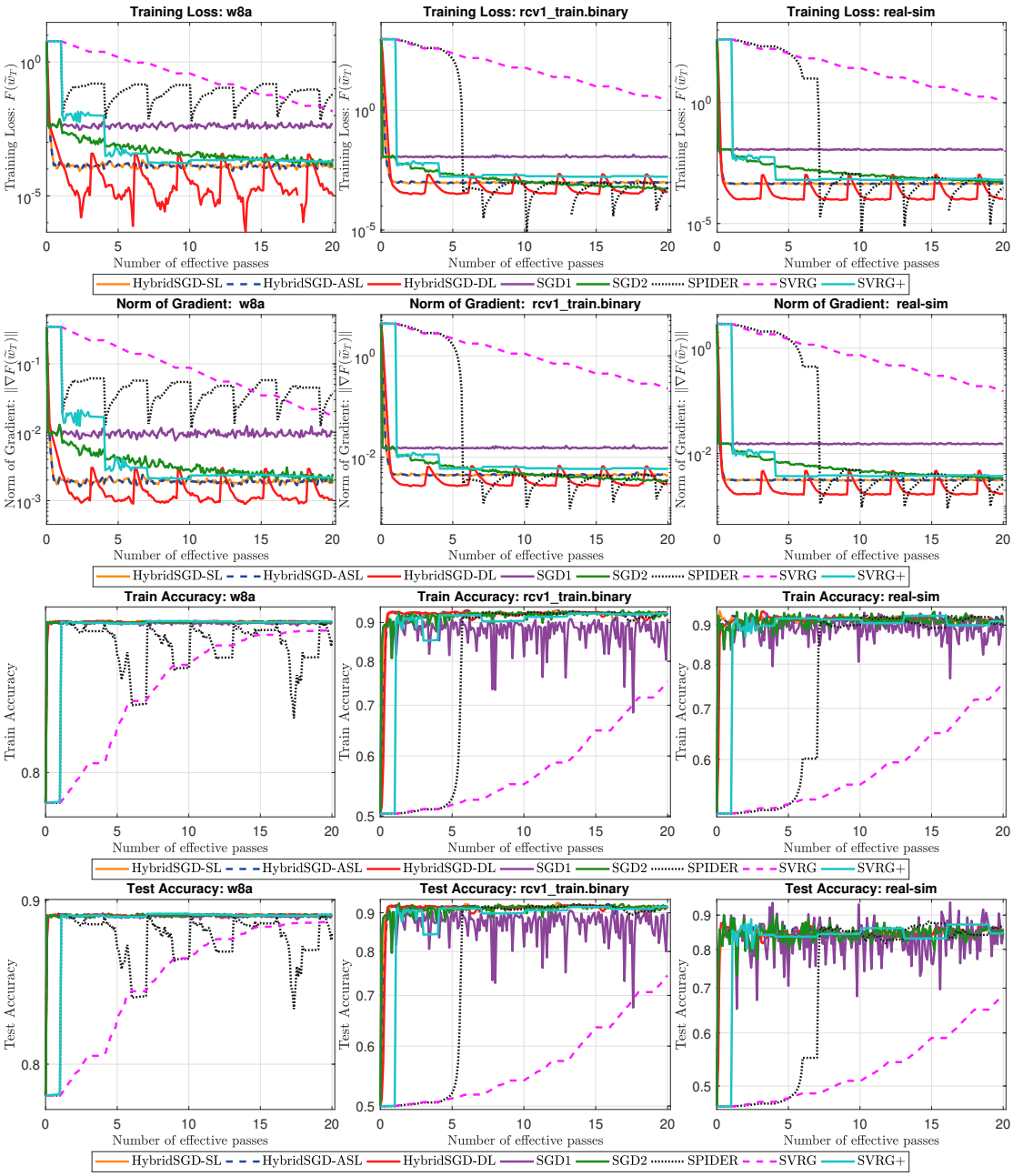

Figure 10: The training loss and gradient norms of (21) with loss ℓ_2: Single-sample.

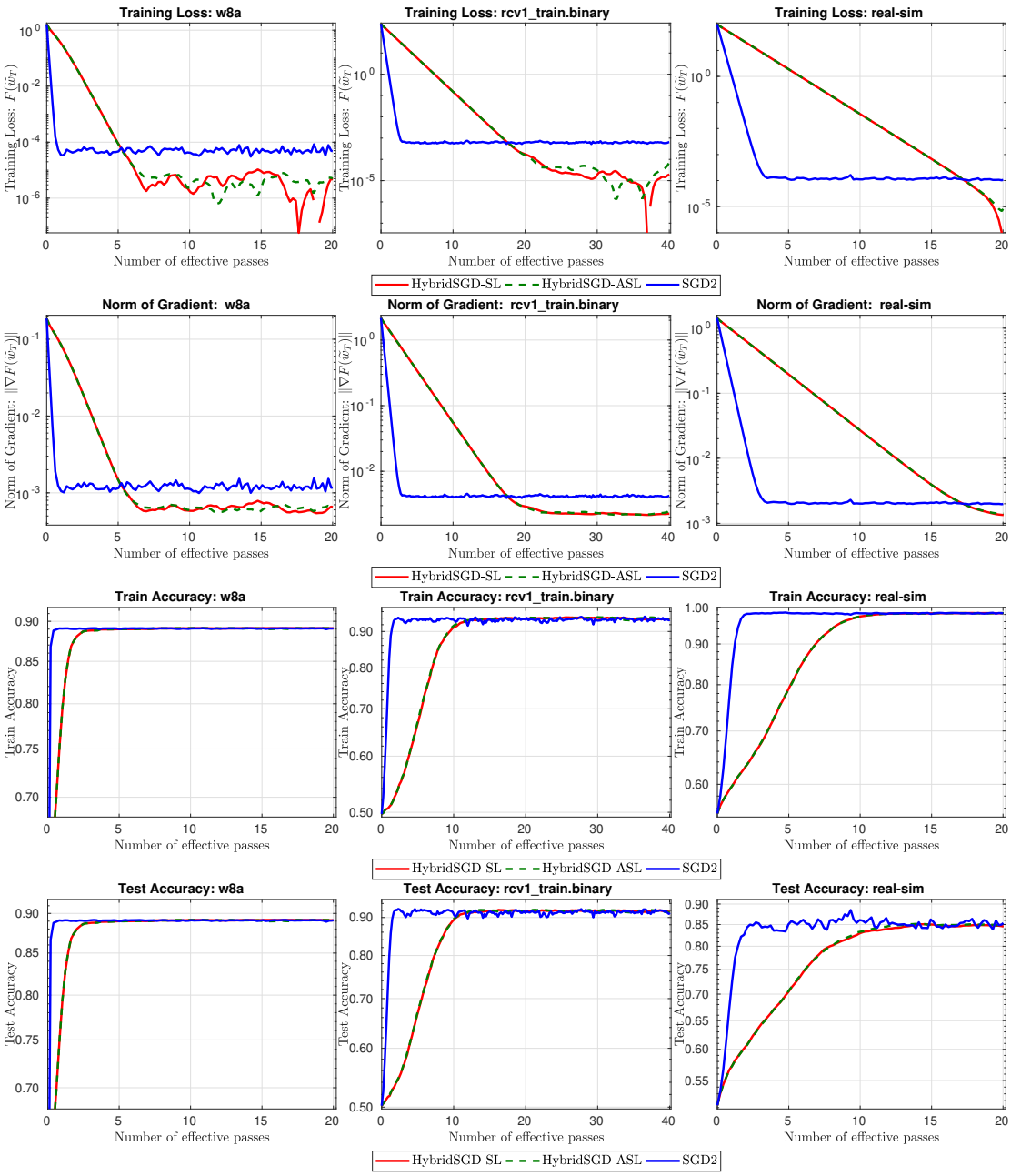

Figure 11: The training loss and gradient norms of (21) with loss ℓ_1: Mini-batch.

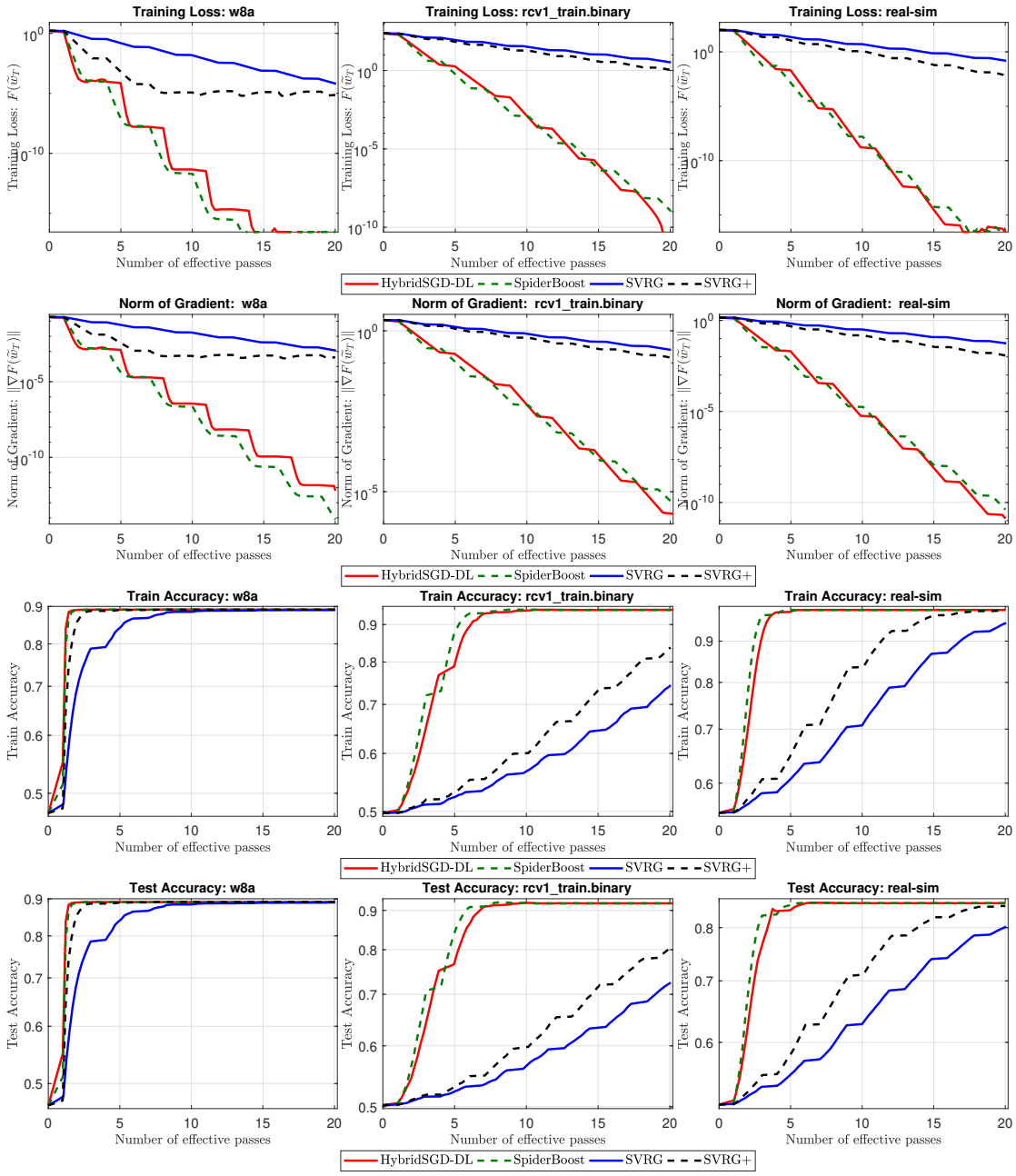

Figure 12: The training loss and gradient norms of (21) with loss ℓ_1: Mini-batch.

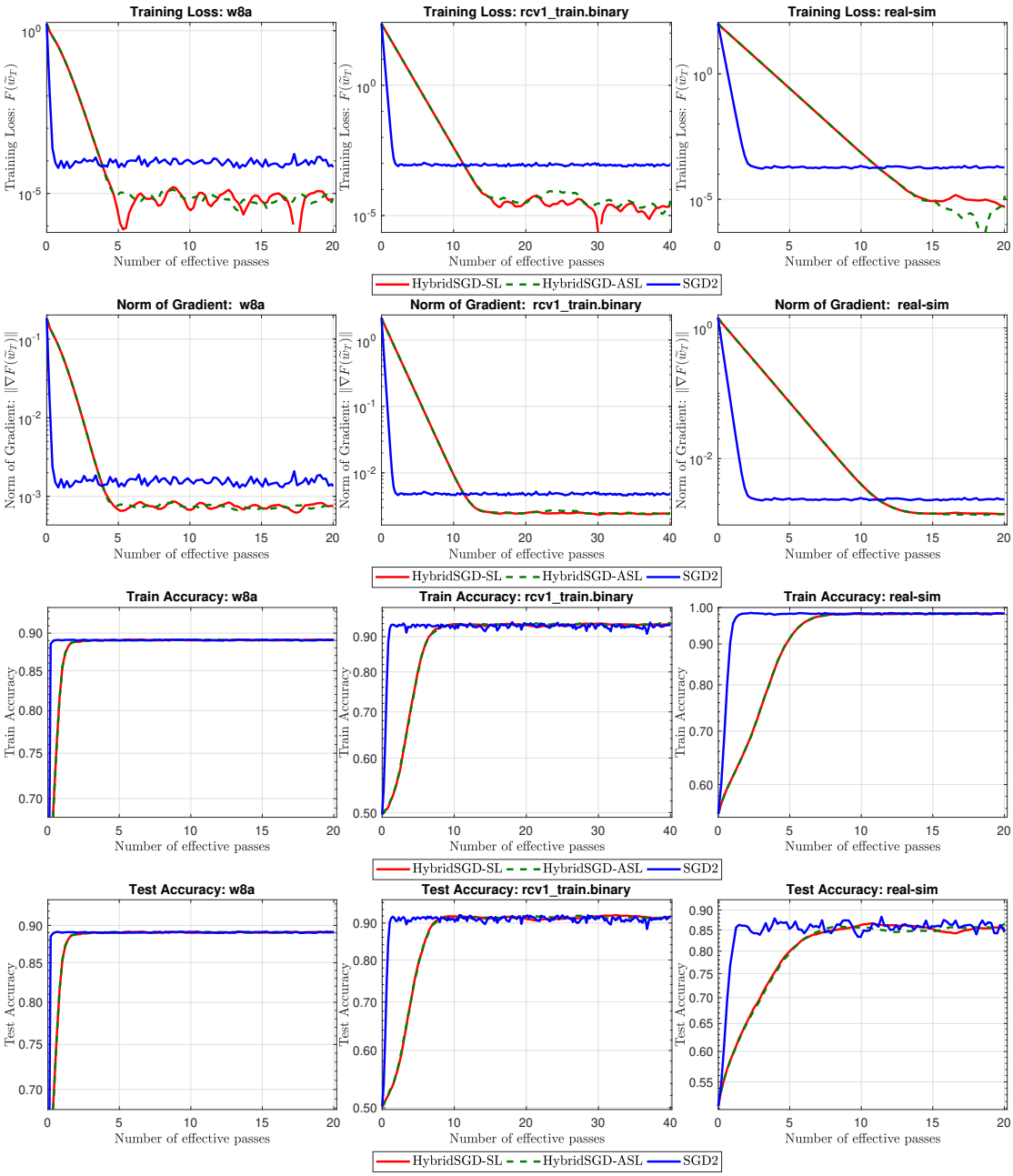

Figure 13: The training loss and gradient norms of (21) with loss ℓ_2: Mini-batch.

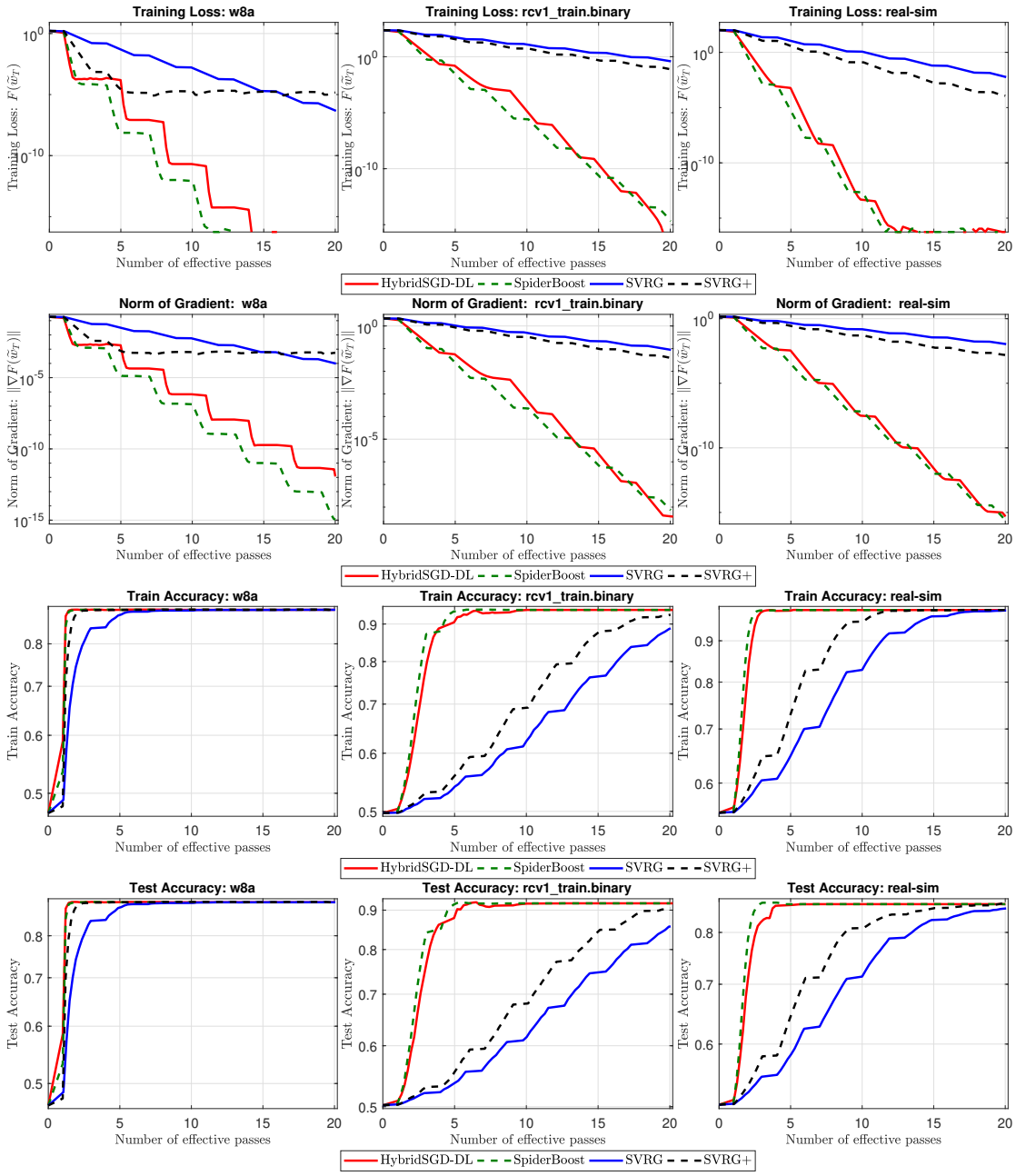

Figure 14: The training loss and gradient norms of (21) with loss ℓ_2: Mini-batch.

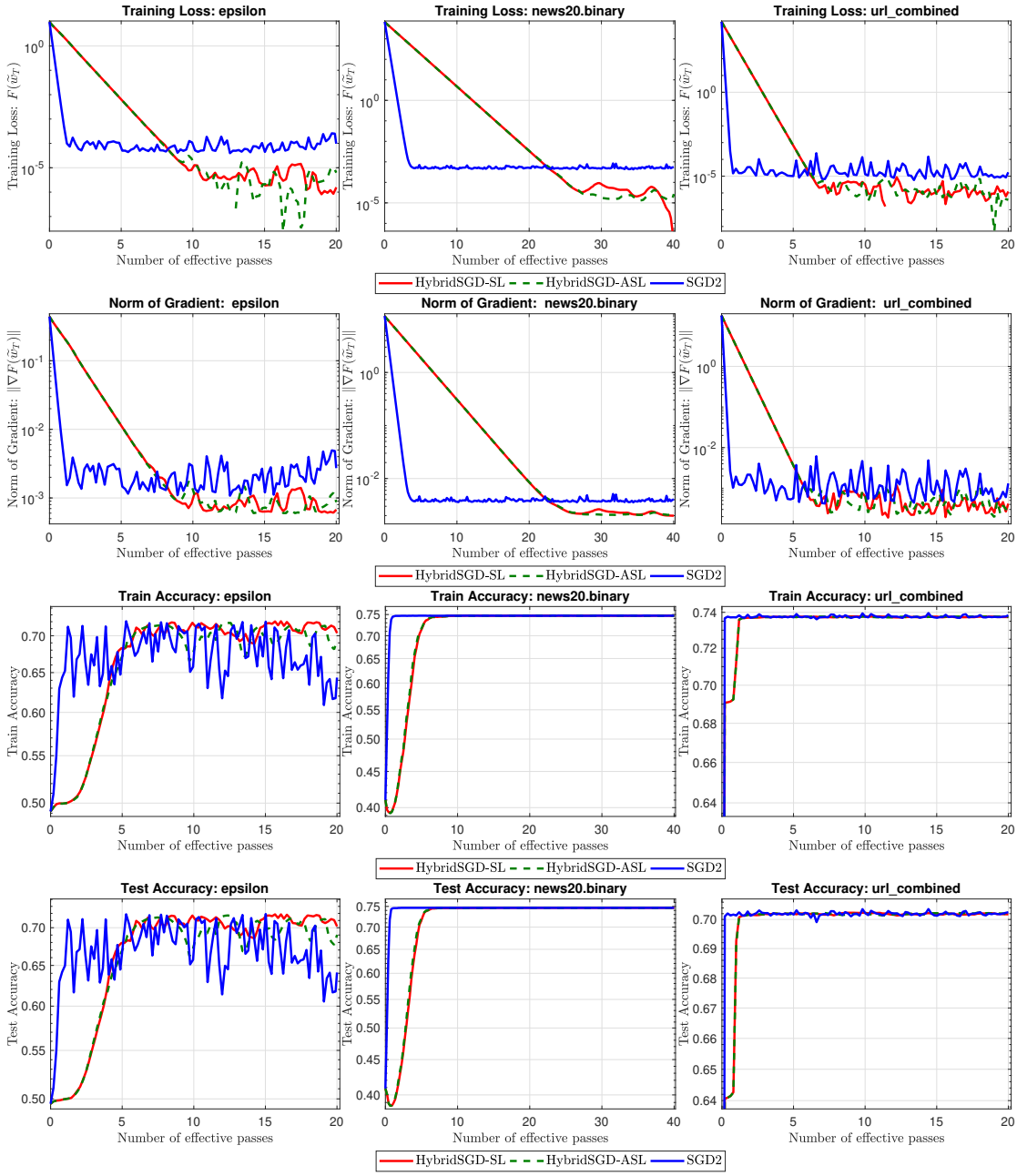

Figure 15: The training loss and gradient norms of (21) with loss ℓ_1: Mini-batch.

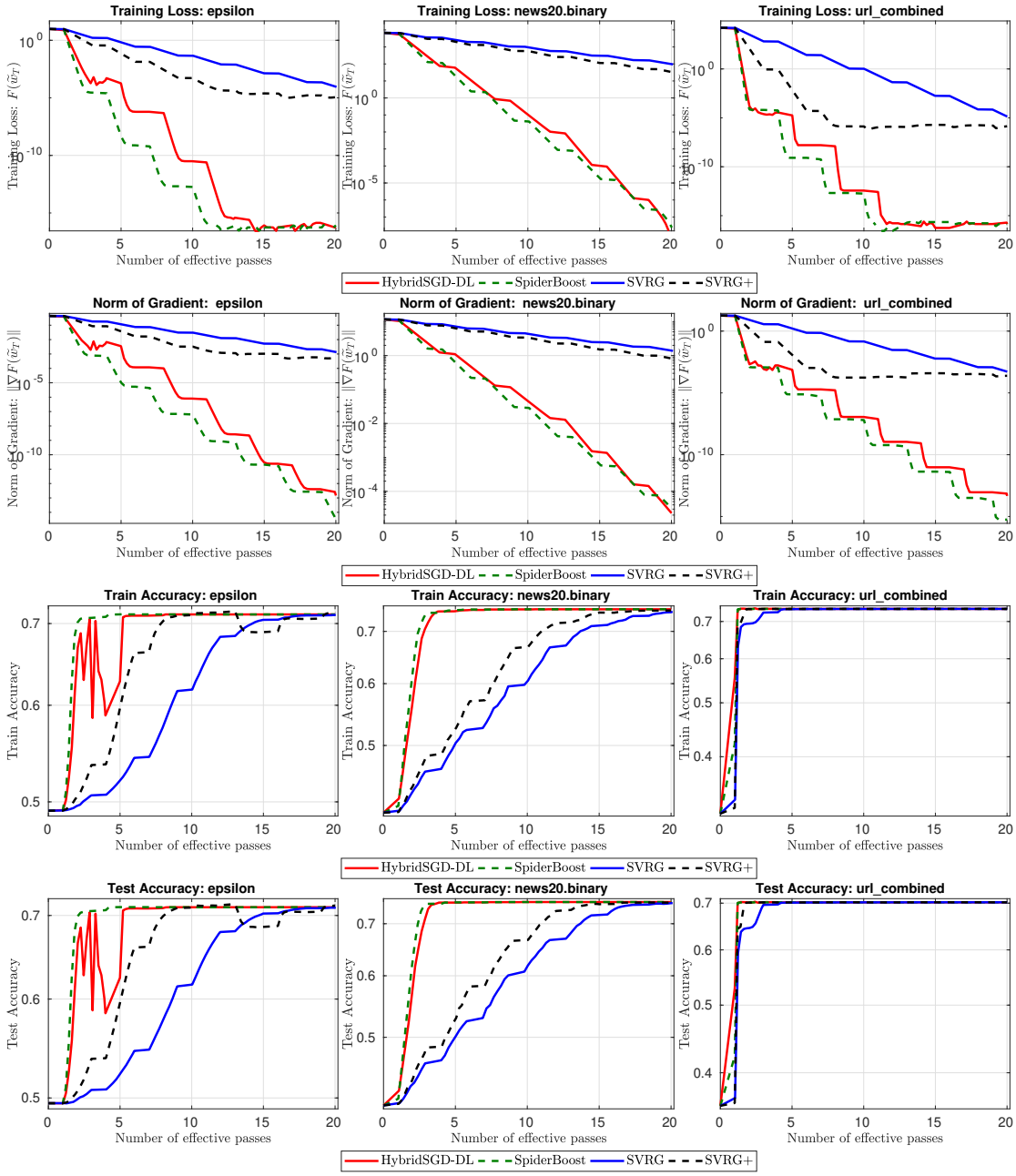

Figure 16: The training loss and gradient norms of (21) with loss ℓ_1: Mini-batch.

In this experiment, although SGD2 has faster decrease during the first few epochs, our HybridSGD-SL and HybridSGD-ASL eventually achieve lower training loss and gradient norm in all datasets while reaching similar training and testing accuracies as SGD2.

Regarding the double-loop variants, our HybridSGD-DL once again has better performance than SVRG and SVRG+ while having comparable performance with SpiderBoost in terms of training loss, gradient norm, and accuracies.

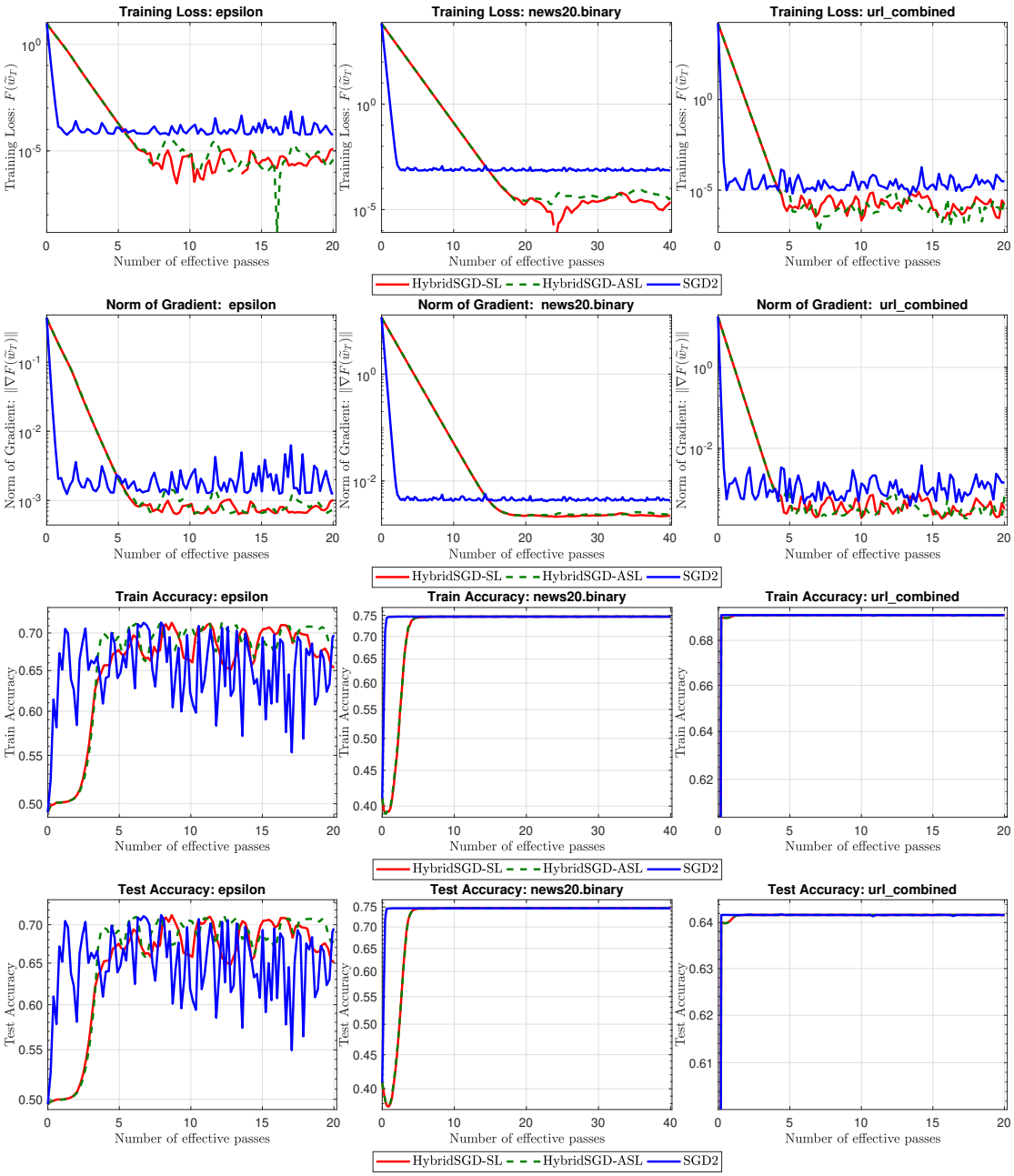

Figure 17: The training loss and gradient norms of (21) with loss ℓ_2: Mini-batch.

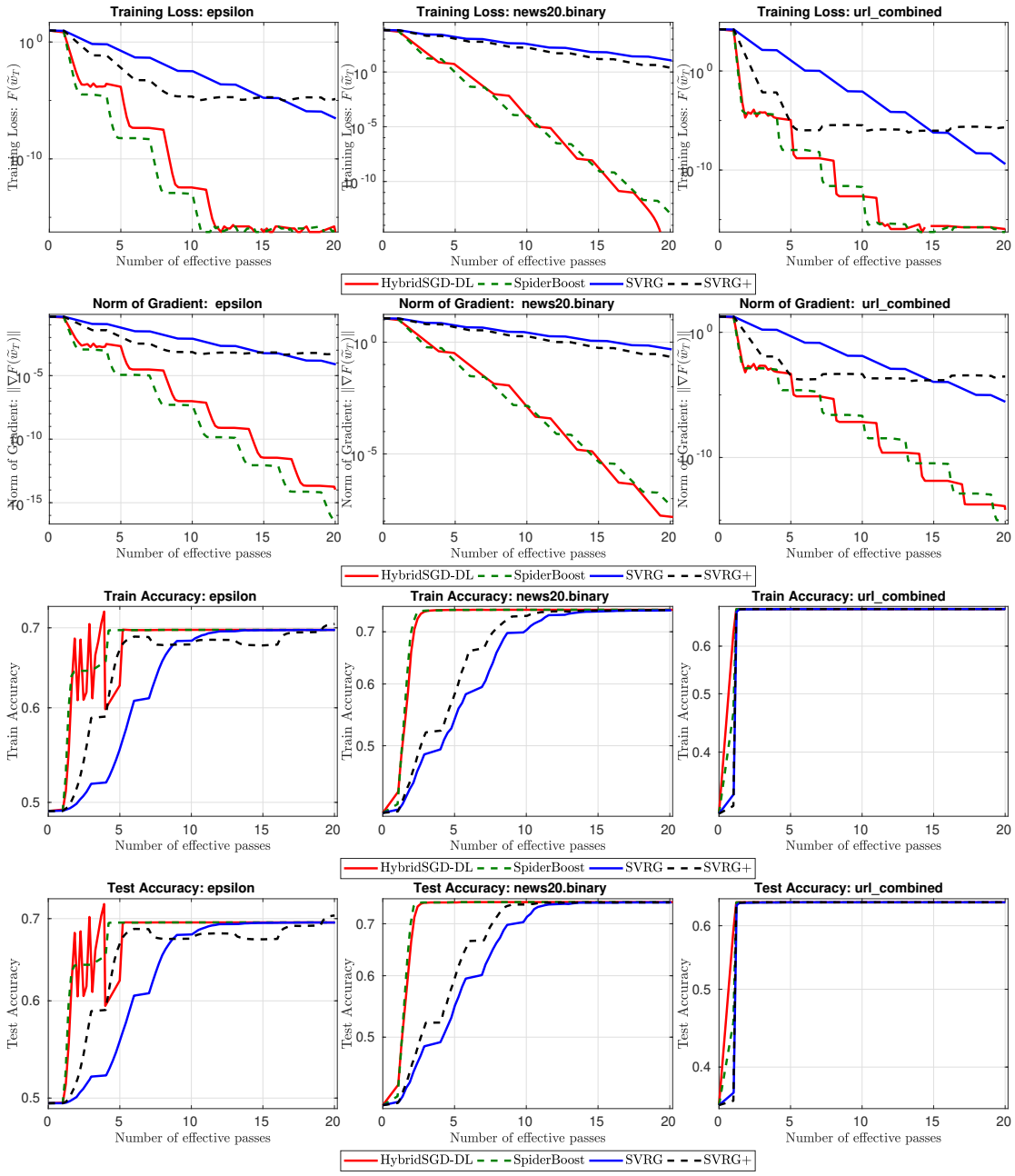

Figure 18: The training loss and gradient norms of (21) with loss ℓ_2: Mini-batch.

References

1. Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1200–1205, June 2017. Montreal, Canada.

2. Z. Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. *arXiv preprint:1708.08694*, 2017.

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

4. Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*, pages 1080–1089, 2016.

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

6. 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.

7. A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

8. C. Fang, C. J. Li, Z. Lin, and T. Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*, 2018.

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

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

11. Z. Li and J. Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1802.04477*, 2018.

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

13. A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2009.

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

15. L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*, 2017.

16. L. M. Nguyen, K. Scheinberg, and M. Takac. Inexact SARAH Algorithm for Stochastic Optimization. *arXiv preprint arXiv:1811.10105*, 2018.

17. 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 preprint arXiv:1901.07648*, 2019.

18. L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takác. Stochastic recursive gradient algorithm for nonconvex optimization. *CoRR*, abs/1705.07261, 2017.

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

20. N. H. Pham, L. M. Nguyen, D. T. Phan, and Q. Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1902.05679*, 2019.

21. S. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. *arXiv preprint arXiv:1607.08254*, 2016.

22. 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*, pages 1145–1153, 2016.

23. Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. In *ICML*, pages 314–323, 2016.

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

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

26. S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *J. Mach. Learn. Res.*, 14:567–599, 2013.

27. Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

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

29. 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. IEEE, 2010.

30. D. Zhou, P. Xu, and Q. Gu. Stochastic nested variance reduction for nonconvex optimization. *arXiv preprint arXiv:1806.07811*, 2018.