AIMS Mathematics, 11(11): 37548–37584.
DOI: 10.3934/math.20261453
Received: 09 July 2026
Revised: 21 October 2026
Accepted: 06 November 2026
Published: 13 November 2026

https://www.aimspress.com/journal/Math

Research article

A variance-reduced stochastic quasi-Newton method with momentum for large-scale nonconvex stochastic optimization

Runmin Wang, Jing Gao*, Yuchang Zhang, Xiaoyu Fu, and Wenwen Wang

School of Mathematics and Statistics, Beihua University, Jilin 132013, China

* **Correspondence:** Email: gaojingjy@163.com.

Abstract: In this paper, we study large-scale unconstrained nonconvex stochastic optimization problems, in which only noisy gradient information is accessible through a stochastic first-order oracle. We develop a stochastic quasi-Newton framework that combines modified secant curvature pairs with a damping strategy and limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) updates. To further enhance both convergence behavior and empirical performance, the proposed method incorporates variance reduction together with a momentum acceleration mechanism, yielding an efficient stochastic quasi-Newton algorithm suitable for large-scale learning tasks. Under standard smoothness and sampling assumptions, we establish convergence to an ε-first-order stationary point and show that the method attains a stochastic first-order oracle complexity of $O\left(n^{2/3}\varepsilon^{-2}\right)$. Numerical experiments on several large-scale datasets corroborate the effectiveness of our approach, showing that the proposed algorithm performs well on classification problems relative to representative stochastic first-order methods and typical stochastic second-order methods.

Keywords: nonconvex stochastic optimization; stochastic quasi-Newton; limited-memory BFGS; variance reduction; momentum acceleration
Mathematics Subject Classification: 65K05, 90C31, 90C53

1. Introduction

In this paper, we consider a class of unconstrained stochastic nonconvex optimization problems. In general, the objective can be formulated in an expectation form

$$\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}_{\xi\sim\mathcal{P}}[F(x;\xi)], \tag{1.1}$$

where $x \in \mathbb{R}^d$ denotes the model parameters, ξ is a random sample drawn from an unknown distribution $\mathcal{P}$, and $F(x;\xi)$ is the corresponding sample loss. Optimization problems with such

stochastic formulations arise widely in machine learning, including logistic regression, support vector machines (SVMs), and the training of deep neural networks. Similar stochastic or sample-average structures also appear in other data-intensive domains such as financial risk modeling [1], healthcare data analytics and biostatistical inference [2], and transportation planning and scheduling [3].

A fundamental instance is empirical risk minimization (ERM), which is commonly written as

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1.2}$$

where n denotes the number of samples, and $f_i(x)$ is the loss associated with sample i.

Stochastic approximation (SA) provides a classical framework for stochastic optimization, with roots in the classical work of Robbins and Monro [4]. By replacing the full gradient $\nabla f(x)$ with a sampled or mini-batch gradient estimator, one obtains stochastic gradient descent (SGD) and its mini-batch variants [5–7]. Owing to their low per-iteration cost and strong scalability, SGD-type methods have become a cornerstone of large-scale learning [8, 9]. Extensive studies have analyzed the convergence properties and complexity of SGD, and have also demonstrated their stability and effectiveness in practical applications [10–12]. In recent years, a wide range of SGD variants and strategies have been developed, including acceleration methods (heavy-ball and its variants) [13–15], adaptive learning-rate methods (the adaptive gradient method (AdaGrad), root mean square propagation (RMSProp), and adaptive moment estimation (Adam)) [16, 17], variance-reduction methods (the stochastic variance reduced gradient method (SVRG), the stochastic recursive gradient algorithm (SARAH), the stochastic path-integrated differential estimator (SPIDER)) [18–20], and variance-reduced stochastic proximal point methods and distributed/parallel SGD methods for large-scale training [21–23].

Although SGD-type methods are computationally efficient and scalable, their reliance on noisy first-order information can lead to unstable search directions, restrictive stepsize choices, and slow convergence in large-scale nonconvex optimization. In response to these difficulties, stochastic quasi-Newton methods have been developed to incorporate curvature information at a relatively low computational cost [24]. These methods construct Hessian or inverse-Hessian approximations from curvature pairs, thereby improving the search direction and scaling without explicitly forming the Hessian matrix.

To reduce the computational and storage burden of explicitly forming and storing Hessian or inverse-Hessian approximations, Byrd et al. proposed a sampled quasi-Newton (SQN) method for large-scale optimization based on limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) updates [24]. By using a small number of curvature pairs, L-BFGS methods incorporate curvature information at a relatively low cost. Therefore, L-BFGS-based methods have been widely adopted in large-scale stochastic quasi-Newton algorithms [25–28]. However, in nonconvex stochastic optimization, the reliable construction and effective use of curvature information remain challenging. First, stochastic gradient noise may lead to inaccurate secant information and unreliable curvature pairs; in the absence of an exact line search, the curvature condition is difficult to guarantee, which may undermine the positive definiteness and numerical stability of the Hessian or inverse-Hessian approximation [29]. Second, the high variance of stochastic gradients may induce oscillatory search directions and restrict usable stepsizes in practice, thereby reducing convergence efficiency.

To address the first issue, stochastic damped quasi-Newton variants have been developed to improve

positive-definiteness preservation and numerical stability by modifying curvature pairs [30, 31], with classical damped quasi-Newton ideas providing the foundation for such stabilization strategies. For the second issue, variance reduction techniques, such as SVRG and its variants, have been incorporated into SQN frameworks to reduce gradient noise, improve search-direction stability, and enhance the quality of curvature information [30, 32]. Nevertheless, damping and variance reduction may still be insufficient to ensure reliable curvature construction in nonconvex stochastic optimization, as residual gradient noise and nonconvex curvature can make the gradient-difference vector $\hat{y}_k$ unreliable.

Motivated by this observation, we introduce a modified secant mechanism that replaces the raw gradient-difference vector $\hat{y}_k$ with a more stable and conservative $\tilde{y}_k$, thereby enhancing the quality of curvature pairs [33]. This design complements variance reduction as follows: Variance reduction primarily reduces gradient noise, whereas the modified secant mechanism directly strengthens the reliability and consistency of curvature information.

To further improve practical convergence behavior, we incorporate a momentum acceleration mechanism inspired by existing momentum-based optimization methods [34]. The purpose of this mechanism is to exploit accumulated descent information and mitigate possible oscillations caused by stochastic gradient noise and ill-conditioning. Combined with damped curvature scaling, variance-reduced gradient estimation, and modified secant curvature-pair construction, the resulting method is designed to improve convergence behavior and overall efficiency while maintaining numerical stability.

The main contributions of this study are as follows:

- We propose a stochastic quasi-Newton algorithm for large-scale nonconvex stochastic optimization. First, we develop a limited-memory curvature correction method based on modified secant pairs and damping, which improves the stability of curvature approximation and the reliability of search directions while keeping storage and computational costs low. Second, we incorporate variance-reduced stochastic gradient estimation and momentum acceleration to reduce the effect of gradient noise and improve convergence stability and practical efficiency.
- Under standard smoothness and sampling assumptions, we establish convergence to an ε-stationary point and derive a stochastic first-order oracle complexity of $O(n^{2/3}\varepsilon^{-2})$ for achieving $\mathbb{E}\|\nabla f(x)\|^2 \leq \varepsilon^2$, which is of the same order as classical SVRG-type nonconvex stochastic methods and existing SVRG-based stochastic quasi-Newton methods.
- We conduct experiments on multiple benchmark classification datasets. The numerical results show that the proposed method achieves favorable classification accuracy and competitive performance in reducing both objective values and gradient norms, while producing smoother convergence trajectories across different tasks.

The remainder of the paper is organized as follows. Section 2 introduces the preliminaries. Section 3 presents the proposed stochastic L-BFGS framework with the modified-secant curvature correction and stochastic damping. Section 4 incorporates an SVRG-type variance-reduced gradient estimator and a three-sequence momentum scheme, leading to the momentum-accelerated variance-reduced quasi-Newton algorithm and its implementation details. Section 5 establishes the expected descent recursion and the complexity analysis, proving convergence to an ε-stationary point and deriving the corresponding stochastic first-order oracle (SFO) complexity bound.

2. Preliminaries

In this section, several preliminaries used throughout the paper are introduced. In general, computing a global minimizer of a nonconvex problem is NP-hard [35]. Therefore, we focus on finding an ε-first-order stationary point, instead. For the finite-sum objective in (1.2), the full gradient is given by

$$\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x). \tag{2.1}$$

In large-scale problems, evaluating the full gradient (2.1) is computationally expensive. Accordingly, mini-batch stochastic gradient estimators are commonly used. Given a mini-batch index set $\xi_k \subset \{1, \ldots, n\}$ with batch size $b := |\xi_k|$, we define

$$g(x; \xi_k) = \frac{1}{b}\sum_{i\in\xi_k} \nabla f_i(x),$$

which serves as a stochastic estimator of $\nabla f(x)$.

Definition 2.1. *Let $\hat{x}$ be a random iterate drawn uniformly from $\{x_1, \ldots, x_K\}$, where K denotes the total number of iterations. We call $\hat{x}$ an ε-first-order stationary point if*

$$\mathbb{E}\,\|\nabla f(\hat{x})\|^2 \le \varepsilon^2,$$

where $\varepsilon > 0$ is an accuracy parameter.

Definition 2.2. *Given an index $i \in \{1, \ldots, n\}$ and a point $x \in \mathbb{R}^d$, the stochastic first-order oracle (SFO) [36] outputs the function value and gradient pair*

$$(f_i(x),\ \nabla f_i(x)).$$

Assumption 2.1. *The objective function f is bounded below on $\mathbb{R}^d$; that is,*

$$f^* := \inf_{x\in\mathbb{R}^d} f(x) > -\infty. \tag{2.2}$$

Assumption 2.2. *For each $i \in \{1, \ldots, n\}$, the function f_i is twice continuously differentiable on $\mathbb{R}^d$, and there exists a constant $L > 0$ such that, for all $x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|. \tag{2.3}$$

Assumption 2.3. *For any $k \ge 0$, the matrix H_k is $\mathcal{F}_k$-measurable, where $\mathcal{F}_k$ denotes all historical information available before the current mini-batch ξ_k is sampled.*

3. Generation of stochastic damped L-BFGS search directions with modified secant curvature pairs

In this section, we present a stochastic damped L-BFGS search-direction generation procedure based on modified secant curvature pairs within the SQN framework. In particular, we adopt the

following damped curvature construction using modified secant information. A corrected curvature vector $\tilde{y}_k$ is first obtained through the modified secant mechanism, and then a damped curvature pair is constructed for the subsequent limited-memory update. Specifically, the search direction is computed by applying the limited-memory inverse-Hessian approximation to the stochastic gradient estimator, namely,

$$d_k = H_k g_k, \tag{3.1}$$

where g_k denotes a stochastic gradient estimator under the stochastic first-order information model. While retaining the low storage and cost advantages of limited-memory L-BFGS, the proposed method combines the modified secant construction and stochastic damping to enhance the reliability of curvature-pair construction and the numerical stability of the updates in stochastic environments.

In quasi-Newton methods, the search direction is generated by using an approximation of the Hessian matrix or its inverse. A typical update can be written as

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k) \quad \text{or} \quad x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k), \tag{3.2}$$

where B_k is an approximation of the Hessian matrix $\nabla^2 f(x_k)$ at x_k, and H_k is an approximation of $[\nabla^2 f(x_k)]^{-1}$. One of the most widely used quasi-Newton methods is BFGS [37–40], which updates B_k via

$$B_k = B_{k-1} + \frac{\hat{y}_{k-1}\hat{y}_{k-1}^T}{s_{k-1}^T \hat{y}_{k-1}} - \frac{B_{k-1} s_{k-1} s_{k-1}^T B_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}}, \tag{3.3}$$

where $s_{k-1} := x_k - x_{k-1}$, and $\hat{y}_{k-1} := \nabla f(x_k) - \nabla f(x_{k-1})$. By the Sherman–Morrison–Woodbury formula, the corresponding update for $H_k = B_k^{-1}$ is

$$H_k = (I - \rho_{k-1} s_{k-1} \hat{y}_{k-1}^T) H_{k-1} (I - \rho_{k-1} \hat{y}_{k-1} s_{k-1}^T) + \rho_{k-1} s_{k-1} s_{k-1}^T, \tag{3.4}$$

where $\rho_{k-1} := \frac{1}{s_{k-1}^T \hat{y}_{k-1}}$.

In deterministic optimization, the performance of BFGS-type methods depends critically on the quality of the curvature information used in the secant approximation. To improve the standard secant relation, Cheng and Gao [33] proposed a modified secant mechanism that incorporates additional function-value and gradient information into the curvature vector.

In large-scale stochastic optimization, explicitly forming and storing the Hessian or inverse-Hessian approximation is usually impractical. Therefore, limited-memory variants are commonly adopted, where only a small number of recent curvature pairs are stored to reduce memory requirements and the per-iteration computational cost [24]. In stochastic nonconvex problems, however, curvature vectors constructed from stochastic gradient differences may be unreliable due to gradient noise, and the curvature condition may even fail.

Motivated by the modified secant idea, we introduce this construction into the stochastic L-BFGS method. Unlike deterministic optimization, in stochastic optimization, we do not directly use the raw stochastic gradient difference as the curvature vector. Instead, the gradient difference is computed on the same mini-batch, and a function-value correction is further introduced to obtain more reliable modified curvature information.

Specifically, using the same mini-batch samples $\{\xi_{k-1,i}\}_{i=1}^{b_{k-1}}$, define

$$\bar{g}_k := \frac{1}{b_{k-1}} \sum_{i=1}^{b_{k-1}} g(x_k, \xi_{k-1,i}), \qquad g_{k-1}^{c} := \frac{1}{b_{k-1}} \sum_{i=1}^{b_{k-1}} g(x_{k-1}, \xi_{k-1,i}) \tag{3.5}$$

and

$$\hat{y}_{k-1} := \bar{g}_k - g_{k-1}^{\mathrm{c}} = \frac{1}{b_{k-1}} \sum_{i=1}^{b_{k-1}} \Big(g(x_k, \xi_{k-1,i}) - g(x_{k-1}, \xi_{k-1,i}) \Big). \tag{3.6}$$

Let $\bar{f}_k$ and f_{k-1} denote the mini-batch averaged objective estimates at x_k and x_{k-1}, respectively, evaluated on the same sample set $\{\xi_{k-1,i}\}_{i=1}^{b_{k-1}}$. We define

$$\breve{\eta}_{k-1} := 2(f_{k-1} - \bar{f}_k) + s_{k-1}^T (g_{k-1}^{\mathrm{c}} + \bar{g}_k), \qquad \eta_{k-1} := \max\{0, \breve{\eta}_{k-1}\}.$$

Given constants $\tau > 0$ and $\tau_{\max} > 0$,

$$\tau_{k-1} := \begin{cases} \min\left\{\tau_{\max}, \tau \dfrac{\eta_{k-1}}{s_{k-1}^T \hat{y}_{k-1}}\right\}, & s_{k-1}^T \hat{y}_{k-1} > 0, \\ 0, & s_{k-1}^T \hat{y}_{k-1} \le 0, \end{cases}$$

and

$$\tilde{y}_{k-1} := (1 + \tau_{k-1}) \hat{y}_{k-1}, \tag{3.7}$$

where $\hat{y}_{k-1}$ is the stochastic gradient difference computed by (3.6). The modified secant equation is then written as

$$B_k s_{k-1} = \tilde{y}_{k-1}.$$

The modified curvature vector $\tilde{y}_{k-1}$ combines gradient-difference information with mini-batch function-value variation, thereby enriching the curvature information and reducing the influence of stochastic noise on the secant approximation.

To safeguard the positive definiteness of the inverse-Hessian approximation, we further apply the stochastic damping strategy of stochastic damped L-BFGS (SdLBFGS) [30] to the modified curvature pair $(s_{k-1}, \tilde{y}_{k-1})$. For the initial scaling in the two-loop recursion, set

$$\gamma_k = \begin{cases} \min\left\{\gamma_{\max}, \max\left\{\dfrac{\tilde{y}_{k-1}^T \tilde{y}_{k-1}}{s_{k-1}^T \tilde{y}_{k-1}}, \delta\right\}\right\}, & s_{k-1}^T \tilde{y}_{k-1} > 0, \\ \delta, & s_{k-1}^T \tilde{y}_{k-1} \le 0. \end{cases}, \qquad H_{k,0} = \gamma_k^{-1} I,$$

where $\delta > 0$ is a constant, $\gamma_{\max} := \delta + (1 + \tau_{\max})\kappa$. Here, $\kappa > 0$ is a Lipschitz constant for the component gradients. By Assumption 2.2, we choose $\kappa = L$ and define

$$\sigma_{k-1} := s_{k-1}^T H_{k,0}^{-1} s_{k-1}.$$

The damped curvature vector is defined by

$$\bar{y}_{k-1} := \varphi_{k-1} \tilde{y}_{k-1} + (1 - \varphi_{k-1})\, H_{k,0}^{-1} s_{k-1}, \tag{3.8}$$

where $\varpi \in (0, 1)$ is the damping parameter, and

$$\varphi_{k-1} = \begin{cases} \dfrac{(1 - \varpi)\sigma_{k-1}}{\sigma_{k-1} - s_{k-1}^T \tilde{y}_{k-1}}, & s_{k-1}^T \tilde{y}_{k-1} < \varpi \sigma_{k-1}, \\ 1, & \text{otherwise}. \end{cases} \tag{3.9}$$

Here, the damping step is applied to the modified curvature vector rather than directly to the raw stochastic gradient difference. Hence, the final curvature pair incorporates both modified secant information and the positive-definiteness safeguard of damping.

The final curvature pair $(s_{k-1}, \bar{y}_{k-1})$ is then used in the limited-memory L-BFGS update, and the search direction is computed by the two-loop recursion. Based on the above modified secant curvature correction and stochastic damping strategy, we present the stochastic modified damped L-BFGS (SMdLBFGS) method in Algorithm 1.

Algorithm 1. SMdLBFGS

Input: Current iterate x_k. Given the stochastic gradient g^{c}_{k-1} at x_{k-1}, the mini-batch ξ_{k-1} with $|\xi_{k-1}| = b_{k-1}$. Memory length M. Stored curvature pairs $(s_j, \bar{y}_j)$ and ρ_j for $j = k-p, \ldots, k-2$, where $p = \min(M, k)$. Set $u_0 = g_k$. Parameters: $\delta > 0$, $\tau > 0$, $\tau_{\max} > 0$, $\varpi \in (0, 1)$.

1. Compute s_{k-1}. Compute $\hat{y}_{k-1}$ by (3.5) and (3.6) .
2. Compute $\tilde{y}_{k-1}$ by (3.7), and compute γ_k.
3. Compute $\bar{y}_{k-1}$ by (3.8), and set $\rho_{k-1} = (s^T_{k-1}\bar{y}_{k-1})^{-1}$.
4. **for** $i = 0, \ldots, p-1$ **do**
5. Compute $\mu_i = \rho_{k-i-1}\, u_i^T\, s_{k-i-1}$.
6. Compute $u_{i+1} = u_i - \mu_i\, \bar{y}_{k-i-1}$.
7. **end for**
8. Compute $v_0 = \gamma_k^{-1} u_p$.
9. **for** $i = 0, \ldots, p-1$ **do**
10. Compute $\nu_i = \rho_{k-p+i}\, v_i^T\, \bar{y}_{k-p+i}$.
11. Compute $v_{i+1} = v_i + (\mu_{p-i-1} - \nu_i) s_{k-p+i}$.
12. **end for**

Output: $H_k g_k = v_p$.

Remark 3.1. *Algorithm 1 computes the product $H_k g_k$ via the standard two-loop recursion, and the search direction is given by $d_k = H_k g_k = v_p$.*

Remark 3.2. *The method stores only the most recent M curvature pairs, so that $p = \min(M, k)$ in the recursion.*

4. Momentum-accelerated variance-reduced SMdLBFGS method

In this section, we develop a momentum-accelerated variance-reduced SMdLBFGS method for large-scale nonconvex stochastic optimization. Based on the modified damped L-BFGS direction constructed in the previous section, the proposed method further incorporates an SVRG-type gradient estimator and a three-sequence coupled momentum mechanism. Specifically, the variance-reduced stochastic gradient is evaluated at an extrapolated point, and the search direction is then generated by the modified damped L-BFGS recursion. This unified design aims to improve the reliability of curvature information, reduce stochastic gradient variance, and alleviate oscillatory behavior in large-scale nonconvex stochastic optimization.

To construct the variance-reduced gradient estimator used in the proposed method, we first recall the SVRG method proposed by Johnson et al. [19]. In particular, SVRG computes a full gradient at a

snapshot point $\tilde{x}$ and constructs a stochastic gradient estimator of the form

$$g_k^{\mathrm{SVRG}} = \nabla f_{\xi_k}(x_k) - \nabla f_{\xi_k}(\tilde{x}) + \nabla f(\tilde{x}), \tag{4.1}$$

where ξ_k denotes the mini-batch at iteration k, and

$$\nabla f_{\xi_k}(u) = \frac{1}{b} \sum_{i \in \xi_k} \nabla f_i(u) \tag{4.2}$$

is the corresponding mini-batch averaged gradient. This construction reduces the variance of SGD. Under the smooth and strongly convex assumption, this mechanism allows the use of a constant stepsize and yields faster convergence behavior. SVRG does not require storing gradients of all samples, and thus, it is more suitable for large-scale learning tasks such as structured prediction and neural network training. Building on this variance-reduction idea, Moritz et al. [25] embedded the SVRG-type gradient estimator into the L-BFGS framework, aiming to address the difficulties caused by stochastic variance and unreliable curvature information in large-scale finite-sum problems. This leads to an SQN method that is more suitable for large-scale stochastic environments. To further extend this method to stochastic nonconvex problems, Wang et al. [30] incorporated a damping technique into the L-BFGS scheme with SVRG-type gradient estimates and developed the SdLBFGS-VR method, which makes the approach more complete. Moreover, the modified secant technique introduced in the previous section further improves the numerical stability of this class of methods.

Variance reduction strategies can substantially decrease the variance of stochastic gradient estimators, yet for ill-conditioned landscapes, the iterates may still suffer from pronounced oscillations and slow convergence. In particular, with a large step size, gradient noise may amplify oscillations and reduce stability; with a small step size, the method often progresses slowly in the later stage, which degrades the overall efficiency. Based on the above motivation, we further adopt a three-sequence coupled momentum acceleration technique. This mechanism originates from an extrapolation-based acceleration framework, and its idea can be traced back to Nesterov's seminal work [41]. In recent years, some researchers have incorporated similar three-sequence coupled momentum mechanisms into stochastic quasi-Newton methods to improve numerical performance [32]. The proposed technique constructs the search direction at an extrapolated point and employs a two-sequence update mechanism to accumulate consistent descent trends, thereby reducing oscillations and improving practical performance. The extrapolated point is defined as

$$z_k = (1 - \alpha_{k+1}) y_k + \alpha_{k+1} x_k. \tag{4.3}$$

Accordingly, in the proposed method, the SVRG gradient estimator is evaluated at z_k as

$$g_k = \nabla f_{\xi_k}(z_k) - \nabla f_{\xi_k}(\tilde{x}) + \nabla f(\tilde{x}). \tag{4.4}$$

The two-sequence updates are given by

$$x_{k+1} = x_k - \lambda_k d_k, \qquad y_{k+1} = z_k - \beta_k d_k, \tag{4.5}$$

where $\alpha_{k+1} \in (0, 1]$ is the momentum parameter, $\lambda_k > 0$ and $\beta_k > 0$ are step size parameters, and d_k is the search direction obtained from Algorithm 1. Based on the above discussion, we present the

following algorithm, termed the momentum-accelerated variance-reduced SMdLBFGS method (M-SMdLBFGS-VR).

Algorithm 2. M-SMdLBFGS-VR

Input: Initial point $x_0 \in \mathbb{R}^d$; epoch length $q \in \mathbb{N}$; number of outer epochs $N \in \mathbb{N}$; mini-batch size b; stepsize parameters $\{\beta_k\}$ and $\{\lambda_k\}$ with $\beta_k > 0$, and there exists a constant $c > 0$ such that for any k,

$$0 < \beta_k \leq \lambda_k \leq 1, \qquad \lambda_k \leq c(1 + \alpha_{k+1})\beta_k;$$

the momentum parameter $\alpha_{k+1} = \dfrac{2}{t+2}$, where $t = k - k_0$.

1. Set $y_0 = x_0$.
2. **for** $r = 0, 1, \ldots, N-1$ **do**
3. $k_0 = rq,\ \tilde{x}_r = x_{k_0},\ y_{k_0} = \tilde{x}_r$.
4. Compute $\nabla f(\tilde{x}_r)$.
5. **for** $k = k_0, k_0 + 1, \ldots, k_0 + q - 1$ **do**
6. $t = k - k_0,\ \alpha_{k+1} = \dfrac{2}{t+2}$.
7. Compute the extrapolated point z_k by (4.3).
8. Sample $\xi_k \subset \{1, 2, \ldots, n\}$ with $|\xi_k| = b$.
9. Compute g_k by (4.4).
10. Compute the search direction d_k by Algorithm 1.
11. Update x_{k+1} and y_{k+1} by (4.5).
12. **end for**
13. $\tilde{x}_{r+1} = x_{k_0+q}$.
14. **end for**

Output (in theory): Sample z_ζ from $\{z_k : k = 0, 1, \ldots, Nq - 1\}$.
Output (in practice): x_{Nq}.

Remark 4.1. *In Step 8 of Algorithm 2, we sample the mini-batch ξ_k with replacement; that is, the indices are drawn independently and identically distributed (i.i.d.) uniformly from $\{1, \ldots, n\}$ and are conditionally independent of $\mathcal{F}_k$. This sampling rule is used in the proofs of Lemmas 5.1 and 5.2.*

Remark 4.2. *In particular, Algorithm 2 reduces to several classical methods under specific parameter choices, thereby yielding the improvement brought by the momentum mechanism. When $\alpha_{k+1} = 0$, and $\lambda_k = \beta_k$, the extrapolated point z_k introduces no additional momentum effect, and the algorithm degenerates to a nonmomentum variance-reduced quasi-Newton method. Moreover, if $H_k = I$, then the search direction satisfies $d_k = g_k$, and the resulting update further reduces to a first-order variance-reduced gradient method. These degenerate cases indicate that Algorithm 2 can be viewed as augmenting the classical variance reduced gradient update with limited-memory quasi-Newton scaling and a three-sequence coupled momentum, which improves early-stage progress while maintaining stability.*

Algorithm 2 incorporates a three-sequence coupled momentum mechanism through (4.3) and (4.5). We update x_k and y_k with two step sizes of different scales. Specifically, λ_k corresponds to a more aggressive move and is used to accelerate the decrease of the main iterate sequence, while β_k

corresponds to a more conservative move and serves to provide a stable reference sequence. Because $\lambda_k \in [\beta_k,\ c(1+\alpha_{k+1})\beta_k]$, and the momentum parameter $\alpha_{k+1} = 2/(t+2)$ decays within each epoch and is reset at the beginning of each epoch, this strategy typically enables faster progress in the early stage and gradually enhancing stability in the later stage, thereby mitigating oscillations amplified by the interaction between stochastic gradient noise and ill conditioning and improving practical convergence performance.

5. Convergence and complexity analysis

In this section, we establish the convergence of the proposed algorithm to an ε-first-order stationary point and provide a corresponding complexity analysis. Our analysis is based on the definitions and assumptions presented in Section 2.

Lemma 5.1. *Suppose that g_k is constructed by* (4.4) *and that the mini-batch ξ_k follows the sampling rule in Algorithm 2. Then,*

$$\mathbb{E}_k[g_k] = \nabla f(z_k); \tag{5.1}$$

equivalently,

$$\mathbb{E}_k[e_k] = 0, \tag{5.2}$$

where $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_k]$, and $e_k := g_k - \nabla f(z_k)$.

Proof. It follows from (4.2) and (4.4) that

$$g_k = \frac{1}{b}\sum_{i\in\xi_k}(\nabla f_i(z_k) - \nabla f_i(\tilde{x}_r)) + w_r, \qquad \text{where } w_r = \nabla f(\tilde{x}_r).$$

For any $\mathcal{F}_k$-measurable point x, using the uniform sampling rule, we have

$$\mathbb{E}_k[\nabla f_i(x)] = \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x) = \nabla f(x).$$

Taking $\mathbb{E}_k[\cdot]$ in the expression of g_k, we obtain

$$\begin{aligned}\mathbb{E}_k[g_k] &= \nabla f(z_k) - \nabla f(\tilde{x}_r) + w_r\\ &= \nabla f(z_k).\end{aligned}$$

Thus, by the definition of e_k, we get (5.2). □

Lemma 5.2. *Under Assumption 2.2 and the sampling rule in Algorithm 2, for any iteration k in epoch r, the estimation error e_k satisfies*

$$\mathbb{E}_k\|e_k\|^2 \le \frac{L^2}{b}\|z_k - \tilde{x}_r\|^2. \tag{5.3}$$

Proof. For each $i \in \{1,\dots,n\}$, define

$$\delta_i := \nabla f_i(z_k) - \nabla f_i(\tilde{x}_r), \qquad \bar{\delta} := \frac{1}{n}\sum_{j=1}^{n}\delta_j. \tag{5.4}$$

By (2.1), (4.2), and (4.4), we have

$$g_k = \frac{1}{b}\sum_{i\in\xi_k}\delta_i + \nabla f(\tilde{x}_r), \qquad \bar{\delta} = \nabla f(z_k) - \nabla f(\tilde{x}_r). \tag{5.5}$$

It then follows that

$$e_k = g_k - \nabla f(z_k) = \frac{1}{b}\sum_{i\in\xi_k}\delta_i - \bar{\delta}.$$

According to the sampling rule stated in Remark 4.1, we have $\xi_k = \{i_{k,1}, \ldots, i_{k,b}\}$ and define

$$u_{k,j} := \delta_{i_{k,j}} - \bar{\delta}, \qquad j = 1, \ldots, b;$$

then, $\mathbb{E}_k[u_{k,j}] = 0$, and

$$e_k = \frac{1}{b}\sum_{j=1}^{b} u_{k,j}.$$

By the conditional independence of $\{u_{k,j}\}_{j=1}^{b}$, we have

$$\mathbb{E}_k\|e_k\|^2 = \frac{1}{b^2}\sum_{j=1}^{b}\mathbb{E}_k\|u_{k,j}\|^2 = \frac{1}{b}\mathbb{E}_k\|\delta_{i_{k,1}} - \bar{\delta}\|^2 \le \frac{1}{b}\mathbb{E}_k\|\delta_{i_{k,1}}\|^2. \tag{5.6}$$

Under Assumption 2.2, for any $i \in \{1, \ldots, n\}$,

$$\|\delta_i\|^2 = \|\nabla f_i(z_k) - \nabla f_i(\tilde{x}_r)\|^2 \le L^2\|z_k - \tilde{x}_r\|^2. \tag{5.7}$$

Combining (5.6) and (5.7), we obtain (5.3). The proof of the lemma is finished. □

Lemmas 5.1 and 5.2 establish the key statistical properties of the variance-reduced estimator g_k, namely unbiasedness and a controlled variance bound. These properties will be used in the subsequent expected-descent analysis.

Lemma 5.3. *By the damping strategy in* (3.8) *and* (3.9)*, suppose that* $s_{k-1} \neq 0$*,* $H_{k,0}^{-1} = \gamma_k I$ *with* $\gamma_k > 0$*, and* $\varpi \in (0, 1)$*. Denote*

$$\sigma_{k-1} := s_{k-1}^T H_{k,0}^{-1} s_{k-1}.$$

Then, the damped curvature pair $(s_{k-1}, \bar{y}_{k-1})$ *satisfies*

$$s_{k-1}^T \bar{y}_{k-1} \ge \varpi\,\sigma_{k-1} > 0. \tag{5.8}$$

Proof. From (3.8), we have

$$s_{k-1}^T \bar{y}_{k-1} = \varphi_{k-1} s_{k-1}^T \tilde{y}_{k-1} + (1 - \varphi_{k-1})\sigma_{k-1}.$$

If $s_{k-1}^T \tilde{y}_{k-1} \ge \varpi\sigma_{k-1}$, then (3.9) gives $\varphi_{k-1} = 1$, and hence,

$$s_{k-1}^T \bar{y}_{k-1} = s_{k-1}^T \tilde{y}_{k-1} \ge \varpi\sigma_{k-1}.$$

If $s_{k-1}^T \tilde{y}_{k-1} < \varpi \sigma_{k-1}$, then

$$\varphi_{k-1} = \frac{(1-\varpi)\sigma_{k-1}}{\sigma_{k-1} - s_{k-1}^T \tilde{y}_{k-1}}.$$

Inserting this expression into the identity for $s_{k-1}^T \bar{y}_{k-1}$, we obtain

$$s_{k-1}^T \bar{y}_{k-1} = \sigma_{k-1} - (1-\varpi)\sigma_{k-1} = \varpi \sigma_{k-1}.$$

Combining both cases, we always have $s_{k-1}^T \bar{y}_{k-1} \geq \varpi \sigma_{k-1}$.

Because $H_{k,0}^{-1} = \gamma_k I$, $\gamma_k > 0$, and $s_{k-1} \neq 0$, $\sigma_{k-1} = s_{k-1}^T H_{k,0}^{-1} s_{k-1} = \gamma_k \|s_{k-1}\|^2 > 0$. Using $\varpi \in (0,1)$, we easily get (5.8). □

Lemma 5.4. *Suppose that the conditions of Lemma 5.3 hold. Let $\bar{y}_{k-1}$ be defined by* (3.8) *and* (3.9) *and used in the inverse BFGS update* (3.4)*. If $H_0 \succ 0$, then H_k is symmetric positive definite for any k.*

Proof. Suppose that H_{k-1} is symmetric. It follows from (3.4) that

$$\begin{aligned} H_k^T &= (I - \rho_{k-1}\bar{y}_{k-1}s_{k-1}^T)H_{k-1}^T(I - \rho_{k-1}s_{k-1}\bar{y}_{k-1}^T) + \rho_{k-1}s_{k-1}s_{k-1}^T \\ &= (I - \rho_{k-1}\bar{y}_{k-1}s_{k-1}^T)H_{k-1}(I - \rho_{k-1}s_{k-1}\bar{y}_{k-1}^T) + \rho_{k-1}s_{k-1}s_{k-1}^T \\ &= H_k. \end{aligned}$$

By Lemma 5.3, $\rho_{k-1} > 0$. Because $\gamma_k \geq \delta > 0$, and $H_{k,0} = \gamma_k^{-1} I$, we have $H_{k,0} \succ 0$. For any nonzero vector $\psi \in \mathbb{R}^d$, define

$$v := (I - \rho_{k-1}\bar{y}_{k-1}s_{k-1}^T)\psi. \tag{5.9}$$

Using (3.4) and (5.9), we obtain

$$\begin{aligned} \psi^T H_k \psi &= \psi^T (I - \rho_{k-1}s_{k-1}\bar{y}_{k-1}^T)H_{k-1}(I - \rho_{k-1}\bar{y}_{k-1}s_{k-1}^T)\psi + \rho_{k-1}\psi^T s_{k-1}s_{k-1}^T\psi \\ &= v^T H_{k-1} v + \rho_{k-1}(s_{k-1}^T\psi)^2. \end{aligned} \tag{5.10}$$

If $s_{k-1}^T \psi \neq 0$, then (5.10) gives $\psi^T H_k \psi > 0$ by $\rho_{k-1} > 0$ and $H_{k-1} \succ 0$.

If $s_{k-1}^T \psi = 0$, then $v = \psi \neq 0$, $\psi^T H_k \psi > 0$ by $H_{k-1} \succ 0$ because $\psi \neq 0$, $H_k \succ 0$.

Together with the symmetry shown above, the result follows by induction. □

Lemma 5.5. *Suppose that Assumption 2.2 holds.*

Then, for any k,

$$\|\tilde{y}_{k-1}\| \leq (1 + \tau_{\max})\kappa \|s_{k-1}\| \tag{5.11}$$

and

$$\delta \leq \gamma_k \leq \delta + (1 + \tau_{\max})\kappa. \tag{5.12}$$

Proof. Denote

$$\tilde{\kappa} := (1 + \tau_{\max})\kappa. \tag{5.13}$$

According to Assumption 2.2 and the construction of $\hat{y}_{k-1}$ in (3.5) and (3.6), we have

$$\|\hat{y}_{k-1}\| \leq \kappa \|s_{k-1}\|.$$

Because $\tilde{y}_{k-1} = (1 + \tau_{k-1})\hat{y}_{k-1}$, and $0 \leq \tau_{k-1} \leq \tau_{\max}$, it follows that

$$\|\tilde{y}_{k-1}\| \leq (1 + \tau_{\max})\kappa\|s_{k-1}\| = \tilde{\kappa}\|s_{k-1}\|.$$

Moreover, by the definition of $\gamma_{\max}$, we have

$$\gamma_{\max} = \delta + (1 + \tau_{\max})\kappa \geq \delta.$$

If $s_{k-1}^T \tilde{y}_{k-1} > 0$, it follows from the definition of γ_k that

$$\delta \leq \gamma_k \leq \gamma_{\max}.$$

If $s_{k-1}^T \tilde{y}_{k-1} \leq 0$, then $\gamma_k = \delta$. Therefore,

$$\delta \leq \gamma_k \leq \gamma_{\max} = \delta + (1 + \tau_{\max})\kappa.$$

The proof of the lemma is finished. □

Lemma 5.6. *Denote $\{H_k\}$ to be generated by the M-SMdLBFGS-VR update with damping parameter $\varpi \in (0, 1)$ and memory length M. Then, there exists a constant $\underline{\mu} > 0$ such that, for any k,*

$$\lambda_{\min}(H_k) \geq \underline{\mu}. \tag{5.14}$$

Specifically, we set

$$\underline{\mu} := \left[((1 + \tau_{\max})\kappa + \delta) + \frac{M(2(1 + \tau_{\max})\kappa + \delta)^2}{\varpi\delta}\right]^{-1}. \tag{5.15}$$

Proof. Denote $B_k := H_k^{-1}$. By Lemma 5.4, $B_k \succ 0$, and

$$\lambda_{\min}(H_k) = \frac{1}{\lambda_{\max}(B_k)}.$$

For any accepted curvature pair $(s_j, \bar{y}_j)$, Lemmas 5.3 and 5.5 imply

$$s_j^T \bar{y}_j \geq \varpi s_j^T H_{j+1,0}^{-1} s_j = \varpi\gamma_{j+1}\|s_j\|^2.$$

Because $\gamma_{j+1} \geq \delta$, we have

$$s_j^T \bar{y}_j \geq \varpi\delta\|s_j\|^2. \tag{5.16}$$

Moreover, from (3.8) and Lemma 5.5,

$$\|\bar{y}_j\| \leq \|\tilde{y}_j\| + \gamma_{j+1}\|s_j\| \leq (2\tilde{\kappa} + \delta)\|s_j\|.$$

Together with (5.16),

$$\frac{\|\bar{y}_j\|^2}{s_j^T \bar{y}_j} \leq \frac{(2\tilde{\kappa} + \delta)^2}{\varpi\delta}. \tag{5.17}$$

For a single BFGS update using the modified damped curvature pair $(s, \bar{y})$, we have

$$B^+ = B - \frac{Bss^T B}{s^T Bs} + \frac{\bar{y}\bar{y}^T}{s^T \bar{y}};$$

the subtracted term is positive semidefinite. Hence, by (5.17), we obtain

$$\lambda_{\max}(B^+) \leq \lambda_{\max}(B) + \frac{\|\bar{y}\|^2}{s^T\bar{y}} \leq \lambda_{\max}(B) + \frac{(2\tilde{\kappa}+\delta)^2}{\varpi\delta}.$$

Because at most M curvature pairs are used and $B_{k,0} = H_{k,0}^{-1} = \gamma_k I$, repeated application yields

$$\lambda_{\max}(B_k) \leq \gamma_k + \frac{M(2\tilde{\kappa}+\delta)^2}{\varpi\delta}.$$

Using $\gamma_k \leq \delta + \tilde{\kappa}$, we obtain

$$\lambda_{\max}(B_k) \leq (\tilde{\kappa}+\delta) + \frac{M(2\tilde{\kappa}+\delta)^2}{\varpi\delta}.$$

Thus,

$$\lambda_{\min}(H_k) \geq \left[(\tilde{\kappa}+\delta) + \frac{M(2\tilde{\kappa}+\delta)^2}{\varpi\delta}\right]^{-1}.$$

Substituting $\tilde{\kappa} = (1+\tau_{\max})\kappa$ and (5.15), we get (5.14). □

Lemma 5.7. *Suppose that Lemmas 5.3–5.5 hold. Then, there exists a constant $\bar{\mu} > 0$ such that, for any $k \geq 1$,*

$$\lambda_{\max}(H_k) = \|H_k\| \leq \bar{\mu}. \tag{5.18}$$

Specifically, we set

$$\alpha := 1 + \frac{2(1+\tau_{\max})\kappa + \delta}{\varpi\delta}, \tag{5.19}$$

and define

$$\bar{\mu} := \frac{\alpha^{2M}}{\delta} + \left(1 + \alpha^2 + \cdots + \alpha^{2(M-1)}\right)\frac{1}{\varpi\delta}. \tag{5.20}$$

Proof. For an inverse BFGS update

$$H^+ = (I - \rho s\bar{y}^T)H(I - \rho\bar{y}s^T) + \rho ss^T, \qquad \rho = (s^T\bar{y})^{-1}.$$

Then,

$$\|H^+\| \leq \|I - \rho s\bar{y}^T\|^2\|H\| + \rho\|s\|^2,$$

$$\|I - \rho s\bar{y}^T\| \leq 1 + \rho\|s\|\,\|\bar{y}\| = 1 + \sqrt{\rho\|s\|^2}\sqrt{\frac{\|\bar{y}\|^2}{s^T\bar{y}}}.$$

According to (5.16) and (5.17),

$$\rho\|s\|^2 \leq \frac{1}{\varpi\delta}, \qquad \frac{\|\bar{y}\|^2}{s^T\bar{y}} \leq \frac{\left(2(1+\tau_{\max})\kappa + \delta\right)^2}{\varpi\delta}.$$

By the definition of α in (5.19), we have

$$\|I - \rho s\bar{y}^T\| \leq \alpha, \qquad \|H^+\| \leq \alpha^2\|H\| + \frac{1}{\varpi\delta}.$$

Because at most M curvature pairs are used in the limited-memory recursion, by repeated application of the above estimate, we obtain

$$\|H_k\| \le \alpha^{2M}\|H_{k,0}\| + \left(1 + \alpha^2 + \cdots + \alpha^{2(M-1)}\right)\frac{1}{\varpi\delta}.$$

By Lemma 5.5, $H_{k,0} = \gamma_k^{-1} I$, and $\gamma_k \ge \delta$, and thus, $\|H_{k,0}\| \le 1/\delta$. Furthermore,

$$\|H_k\| \le \frac{\alpha^{2M}}{\delta} + \left(1 + \alpha^2 + \cdots + \alpha^{2(M-1)}\right)\frac{1}{\varpi\delta} = \bar{\mu}.$$

□

Lemma 5.8. *Suppose that Lemmas 5.6 and 5.7 hold, and $d_k = H_k g_k$ as in* (3.1)*. There exist constants $0 < \underline{\mu} \le \bar{\mu}$ such that*

$$\underline{\mu} I \preceq H_k \preceq \bar{\mu} I.$$

Consequently, for any $v \in \mathbb{R}^d$,

$$\underline{\mu}\|v\|^2 \le v^T H_k v \le \bar{\mu}\|v\|^2, \qquad \|H_k v\| \le \bar{\mu}\|v\|. \tag{5.21}$$

In particular,

$$g_k^T d_k = g_k^T H_k g_k \ge \underline{\mu}\|g_k\|^2, \qquad \|d_k\|^2 = \|H_k g_k\|^2 \le \bar{\mu}^2\|g_k\|^2. \tag{5.22}$$

Proof. By $\underline{\mu} I \preceq H_k \preceq \bar{\mu} I$, we have

$$\underline{\mu}\|v\|^2 \le v^T H_k v \le \bar{\mu}\|v\|^2.$$

Because H_k is symmetric, and $H_k \preceq \bar{\mu} I$, we have $\|H_k\| = \lambda_{\max}(H_k) \le \bar{\mu}$, and therefore,

$$\|H_k v\| \le \|H_k\|\|v\| \le \bar{\mu}\|v\|.$$

Taking $v = g_k$ and using $d_k = H_k g_k$, we obtain (5.22). □

Lemmas 5.3–5.8 provide the uniform spectral bounds and stability of the quasi-Newton sequence H_k, which allow the curvature-scaled terms in the descent analysis to be controlled uniformly.

Lemma 5.9. *Suppose that Assumptions 2.2 and 2.3 hold; then, for any k,*

$$\mathbb{E}_k[f(x_{k+1})] \le f(x_k) - \lambda_k \nabla f(z_k)^T H_k \nabla f(z_k) + \frac{L\lambda_k^2}{2}\,\mathbb{E}_k[\|d_k\|^2] - \lambda_k\,\mathbb{E}_k[\Gamma_k], \tag{5.23}$$

where

$$\Gamma_k := (\nabla f(x_k) - \nabla f(z_k))^T d_k. \tag{5.24}$$

Moreover, for any $\eta > 0$,

$$-\mathbb{E}_k[\Gamma_k] \le \frac{1}{2\eta}\left\|\nabla f(x_k) - \nabla f(z_k)\right\|^2 + \frac{\eta}{2}\,\mathbb{E}_k[\|d_k\|^2]. \tag{5.25}$$

Proof. By Assumption 2.2 and the update $x_{k+1} = x_k - \lambda_k d_k$, we have

$$\begin{aligned} f(x_{k+1}) &\le f(x_k) - \lambda_k \nabla f(x_k)^T d_k + \frac{L\lambda_k^2}{2}\|d_k\|^2 \\ &= f(x_k) - \lambda_k \nabla f(z_k)^T d_k - \lambda_k \Gamma_k + \frac{L\lambda_k^2}{2}\|d_k\|^2. \end{aligned}$$

Because z_k and H_k are $\mathcal{F}_k$-measurable, it follows from Lemma 5.1 that

$$\mathbb{E}_k[\nabla f(z_k)^T H_k g_k] = \nabla f(z_k)^T H_k \mathbb{E}_k[g_k] = \nabla f(z_k)^T H_k \nabla f(z_k).$$

According to (5.24),

$$-\Gamma_k = \big(\nabla f(z_k) - \nabla f(x_k)\big)^T d_k.$$

For any $\eta > 0$, we have

$$-\Gamma_k \le \frac{1}{2\eta}\|\nabla f(x_k) - \nabla f(z_k)\|^2 + \frac{\eta}{2}\|d_k\|^2.$$

Taking $\mathbb{E}_k[\cdot]$ yields (5.25). □

Lemma 5.10. *Denoting $\Delta_k := x_k - y_k$, we have*

$$\Delta_{k+1} = (1 - \alpha_{k+1})\Delta_k - (\lambda_k - \beta_k)H_k g_k. \tag{5.26}$$

Moreover, for any $\theta > 0$,

$$\|\Delta_{k+1}\|^2 \le (1+\theta)(1-\alpha_{k+1})^2\|\Delta_k\|^2 + \Big(1 + \frac{1}{\theta}\Big)(\lambda_k - \beta_k)^2\|H_k g_k\|^2. \tag{5.27}$$

Proof. Using the updates

$$x_{k+1} = x_k - \lambda_k H_k g_k, \qquad y_{k+1} = z_k - \beta_k H_k g_k,$$

we obtain

$$\Delta_{k+1} = x_{k+1} - y_{k+1} = (x_k - z_k) - (\lambda_k - \beta_k)H_k g_k. \tag{5.28}$$

Because $z_k = (1 - \alpha_{k+1})y_k + \alpha_{k+1}x_k$, it follows that

$$x_k - z_k = (1 - \alpha_{k+1})(x_k - y_k) = (1 - \alpha_{k+1})\Delta_k. \tag{5.29}$$

Combining (5.28) and (5.29), we have (5.26).

For any $\theta > 0$, applying Young's inequality to (5.26), we get

$$\begin{aligned} \|\Delta_{k+1}\|^2 &= \big\|(1 - \alpha_{k+1})\Delta_k - (\lambda_k - \beta_k)H_k g_k\big\|^2 \\ &\le (1+\theta)(1-\alpha_{k+1})^2\|\Delta_k\|^2 + \Big(1 + \frac{1}{\theta}\Big)(\lambda_k - \beta_k)^2\|H_k g_k\|^2. \end{aligned}$$

This proves (5.27). □

Corollary. Suppose that the stepsize parameters λ_k and β_k in Algorithm 2 satisfy

$$0 \le \lambda_k - \beta_k \le \varkappa_{k+1}\beta_k, \qquad \varkappa_{k+1} := c(1+\alpha_{k+1}) - 1 > 0. \tag{5.30}$$

Then, by Lemmas 5.6, 5.7, and (5.27), it has that

$$\|\Delta_{k+1}\|^2 \le (1+\theta)(1-\alpha_{k+1})\|\Delta_k\|^2 + \Big(1+\frac{1}{\theta}\Big)\varkappa_{k+1}^2\beta_k^2\bar{\mu}^2\|g_k\|^2. \tag{5.31}$$

Lemma 5.11. *Let $h > 0$, and define*

$$\Phi_k := \mathbb{E}[f(x_k)] + h\,\mathbb{E}\Big[\|\Delta_k\|^2\Big]. \tag{5.32}$$

Denote

$$\eta := \frac{\underline{\mu}}{2\bar{\mu}^2}, \qquad \lambda_k \le \frac{\underline{\mu}}{8L\bar{\mu}^2}. \tag{5.33}$$

Suppose that for any k, the stepsize parameters satisfy

$$0 \le \lambda_k - \beta_k \le \varkappa_{k+1}\beta_k.$$

In addition, assume that there exists $\theta > 0$ such that, for any k,

$$\left(\frac{L^2\lambda_k}{2\eta} + h(1+\theta)\right)(1-\alpha_{k+1})^2 \le h, \qquad h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2 \le \frac{\underline{\mu}}{16}\lambda_k. \tag{5.34}$$

Then,

$$\Phi_{k+1} \le \Phi_k - a_g\lambda_k\,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + a_e\lambda_k\,\mathbb{E}\Big[\|e_k\|^2\Big], \tag{5.35}$$

where $a_g = \underline{\mu}/4$, and $a_e = 3\underline{\mu}/4$.

Proof. By Assumption 2.2, we obtain

$$\|\nabla f(x_k) - \nabla f(z_k)\|^2 \le L^2\|x_k - z_k\|^2.$$

Combining (5.21) and (5.29), we have

$$\begin{aligned}\mathbb{E}[f(x_{k+1})] \le \mathbb{E}[f(x_k)] &- \underline{\mu}\lambda_k\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + \Big(\frac{L\lambda_k^2}{2} + \frac{\eta\lambda_k}{2}\Big)\mathbb{E}\Big[\|d_k\|^2\Big] \\ &+ \frac{L^2\lambda_k}{2\eta}(1-\alpha_{k+1})^2\mathbb{E}\Big[\|\Delta_k\|^2\Big].\end{aligned}$$

Moreover, by (5.22) and $g_k = \nabla f(z_k) + e_k$, we have

$$\mathbb{E}\Big[\|d_k\|^2\Big] \le 2\bar{\mu}^2\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + 2\bar{\mu}^2\mathbb{E}\Big[\|e_k\|^2\Big].$$

Hence, we obtain

$$\begin{aligned}\mathbb{E}[f(x_{k+1})] \le \mathbb{E}[f(x_k)] &- \lambda_k\Big(\underline{\mu} - \bar{\mu}^2\eta - \bar{\mu}^2L\lambda_k\Big)\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \\ &+ \bar{\mu}^2(\eta\lambda_k + L\lambda_k^2)\mathbb{E}\Big[\|e_k\|^2\Big] + \frac{L^2\lambda_k}{2\eta}(1-\alpha_{k+1})^2\mathbb{E}\Big[\|\Delta_k\|^2\Big].\end{aligned} \tag{5.36}$$

By (5.33), we know that

$$\underline{\mu} - \bar{\mu}^2\eta - \bar{\mu}^2 L\lambda_k \geq \frac{3\underline{\mu}}{8}, \qquad \text{and } \bar{\mu}^2(\eta\lambda_k + L\lambda_k^2) \leq \frac{5\underline{\mu}}{8}\lambda_k.$$

Thus, (5.36) implies

$$\begin{aligned}\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] - \frac{3\underline{\mu}}{8}\lambda_k\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + \frac{5\underline{\mu}}{8}\lambda_k\mathbb{E}\Big[\|e_k\|^2\Big] \\ + \frac{L^2\lambda_k}{2\eta}(1-\alpha_{k+1})^2\mathbb{E}\Big[\|\Delta_k\|^2\Big].\end{aligned}$$

Using (5.22), (5.27), and $\|g_k\|^2 \leq 2\|\nabla f(z_k)\|^2 + 2\|e_k\|^2$, we obtain

$$\begin{aligned}\mathbb{E}\Big[\|\Delta_{k+1}\|^2\Big] \leq (1+\theta)(1-\alpha_{k+1})^2\mathbb{E}\Big[\|\Delta_k\|^2\Big] \\ + 2\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \\ + 2\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2\mathbb{E}\Big[\|e_k\|^2\Big].\end{aligned}$$

Multiplying this estimate by h, we have

$$\begin{aligned}\Phi_{k+1} \leq \Phi_k - \Big(\frac{3\underline{\mu}}{8}\lambda_k - 2h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2\Big)\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \\ + \Big(\frac{5\underline{\mu}}{8}\lambda_k + 2h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2\Big)\mathbb{E}\Big[\|e_k\|^2\Big] \\ + \left[\left(\frac{L^2\lambda_k}{2\eta} + h(1+\theta)\right)(1-\alpha_{k+1})^2 - h\right]\mathbb{E}\Big[\|\Delta_k\|^2\Big].\end{aligned} \tag{5.37}$$

By (5.34), it has that

$$2h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2 \leq \frac{\underline{\mu}}{8}\lambda_k.$$

Hence,

$$\frac{3\underline{\mu}}{8}\lambda_k - 2h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2 \geq \frac{\underline{\mu}}{4}\lambda_k,$$

and

$$\frac{5\underline{\mu}}{8}\lambda_k + 2h\Big(1+\frac{1}{\theta}\Big)(\lambda_k-\beta_k)^2\bar{\mu}^2 \leq \frac{3\underline{\mu}}{4}\lambda_k.$$

Using (5.37), $a_g = \underline{\mu}/4$, and $a_e = 3\underline{\mu}/4$, we have (5.35). □

Remark 5.1. *The parameter conditions in Lemma 5.11 can be satisfied by an appropriate choice of the step sizes and related parameters in Algorithm 2. For example, let $q \geq 1$ and $1 < c \leq 5/4$, and set*

$$\lambda_k = \lambda, \qquad \beta_k = \lambda\left(1 - \frac{1}{4q}\right), \qquad \theta = \frac{1}{q},$$

where

$$0 < \lambda \leq \min\left\{1, \frac{\underline{\mu}}{8L\bar{\mu}^2}\right\}, \qquad h = \frac{L^2\bar{\mu}^2}{\underline{\mu}}\lambda q.$$

With this choice, the step-size Conditions in Algorithm 2 and conditions (5.33) *and* (5.34) *are satisfied simultaneously.*

Lemma 5.12. *Let* $\{x_k\}$, $\{y_k\}$, *and* $\{z_k\}$ *be generated by Algorithm 2. For any* $k \in \{rq, \dots, rq + q - 1\}$ *of epoch index r, it holds that*

$$\|z_k - \tilde{x}_r\|^2 \le 2\|x_k - \tilde{x}_r\|^2 + 2\|\Delta_k\|^2. \tag{5.38}$$

Proof. Using the identity $z_k - \tilde{x}_r = (x_k - \tilde{x}_r) + (z_k - x_k)$, we obtain

$$\|z_k - \tilde{x}_r\|^2 \le 2\|x_k - \tilde{x}_r\|^2 + 2\|z_k - x_k\|^2.$$

From (5.29), we have

$$\|z_k - x_k\|^2 = (1 - \alpha_{k+1})^2\|\Delta_k\|^2 \le \|\Delta_k\|^2.$$

Then, we have (5.38). □

Lemma 5.13. *For any epoch index r, denote* $k_0 = rq$ *and* $\tilde{x}_r = x_{k_0}$. *Then, for any* $k \in \{k_0, \dots, k_0 + q\}$,

$$x_k - \tilde{x}_r = x_k - x_{k_0} = \sum_{j=k_0}^{k-1}(x_{j+1} - x_j) = -\sum_{j=k_0}^{k-1}\lambda_j d_j. \tag{5.39}$$

Moreover,

$$\sum_{k=k_0}^{k_0+q-1}\|x_k - \tilde{x}_r\|^2 \le q^2\sum_{j=k_0}^{k_0+q-1}\lambda_j^2\|d_j\|^2. \tag{5.40}$$

Proof. Summing the update $x_{j+1} = x_j - \lambda_j d_j$ over $j = k_0, \dots, k-1$, we can obtain (5.39).

Denote $\hat{u}_j := -\lambda_j d_j$. Then, $x_k - \tilde{x}_r = \sum_{j=k_0}^{k-1}\hat{u}_j$. Using the Cauchy–Schwarz inequality, we have

$$\|x_k - \tilde{x}_r\|^2 = \Big\|\sum_{j=k_0}^{k-1}\hat{u}_j\Big\|^2 \le (k - k_0)\sum_{j=k_0}^{k-1}\|\hat{u}_j\|^2 \le q\sum_{j=k_0}^{k-1}\lambda_j^2\|d_j\|^2.$$

Summing this inequality over $k = k_0, \dots, k_0 + q - 1$, we obtain

$$\begin{aligned}\sum_{k=k_0}^{k_0+q-1}\|x_k - \tilde{x}_r\|^2 &\le q\sum_{k=k_0}^{k_0+q-1}\sum_{j=k_0}^{k-1}\lambda_j^2\|d_j\|^2 \\ &= q\sum_{j=k_0}^{k_0+q-2}(k_0 + q - 1 - j)\lambda_j^2\|d_j\|^2 \\ &\le q^2\sum_{j=k_0}^{k_0+q-1}\lambda_j^2\|d_j\|^2.\end{aligned}$$

□

Lemmas 5.9–5.13 combine the smoothness-based descent estimate with the control of the momentum error Δ_k, yielding a one-step recursion for Φ_k with an explicit descent term. Summing this recursion over k reduces the convergence analysis to establishing an aggregate bound on $\sum_k \lambda_k\mathbb{E}\|\nabla f(z_k)\|^2$.

Lemma 5.14. *For any epoch index r, denote* $k_0 := rq$, $\tilde{x}_r := x_{k_0}$, *and* $\lambda_{\max} := \max_{k_0 \le k \le k_0+q-1} \lambda_k$. *Suppose that*

$$\frac{4L^2\bar{\mu}^2 q^2 \lambda_{\max}^2}{b} \le \frac{a_g}{2a_e}. \tag{5.41}$$

Then,

$$\sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|e_k\|^2\Big] \le \frac{a_g}{2a_e} \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + \frac{4L^2}{b} \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\Delta_k\|^2\Big]. \tag{5.42}$$

Proof. By Lemma 5.2, the SVRG-type estimator satisfies

$$\mathbb{E}\Big[\|e_k\|^2\Big] \le \frac{L^2}{b} \mathbb{E}\Big[\|z_k - \tilde{x}_r\|^2\Big].$$

Multiplying by λ_k, summing over $k = k_0, \ldots, k_0 + q - 1$, and using (5.38), we obtain

$$\sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|e_k\|^2\Big] \le \frac{2L^2}{b} \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|x_k - \tilde{x}_r\|^2\Big] + \frac{2L^2}{b} \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\Delta_k\|^2\Big]. \tag{5.43}$$

By (5.40), we have

$$\sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|x_k - \tilde{x}_r\|^2\Big] \le q^2 \lambda_{\max}^2 \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|d_k\|^2\Big]. \tag{5.44}$$

Moreover, because $d_k = H_k g_k$, $\|H_k\| \le \bar{\mu}$, and $g_k = \nabla f(z_k) + e_k$ with $\mathbb{E}[e_k] = 0$, we have

$$\sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|d_k\|^2\Big] \le \bar{\mu}^2 \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + \bar{\mu}^2 \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|e_k\|^2\Big]. \tag{5.45}$$

Denote

$$S_e := \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|e_k\|^2\Big],$$

$$S_g := \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big], \qquad c := \frac{2L^2\bar{\mu}^2 q^2 \lambda_{\max}^2}{b},$$

$$S_\Delta := \sum_{k=k_0}^{k_0+q-1} \lambda_k \mathbb{E}\Big[\|\Delta_k\|^2\Big].$$

Combining (5.43)–(5.45) yields

$$(1 - c) S_e \le c S_g + \frac{2L^2}{b} S_\Delta. \tag{5.46}$$

By (5.41), we have

$$2c = \frac{4L^2\bar{\mu}^2 q^2 \lambda_{\max}^2}{b} \le \frac{a_g}{2a_e},$$

and hence, $c \le 1/2$, and $(1 - c)^{-1} \le 2$. Therefore,

$$S_e \le 2c S_g + \frac{4L^2}{b} S_\Delta \le \frac{a_g}{2a_e} S_g + \frac{4L^2}{b} S_\Delta.$$

By use of the definitions of S_e, S_g, and S_Δ, we can conclude that (5.42). □

Lemma 5.15. *For any epoch index r, denote* $k_0 := rq$, $x_{k_0} = y_{k_0} = \tilde{x}_r$, $\Delta_{k_0} = 0$, *and*

$$\lambda_{\max} := \max_{k_0 \le k \le k_0+q-1} \lambda_k.$$

Suppose that the step-size parameters satisfy the conditions specified in Algorithm 2. Then,

$$\begin{aligned}\sum_{k=k_0}^{k_0+q-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big] \le 2q^2(2c-1)^2\bar{\mu}^{\,2}\lambda_{\max} \sum_{k=k_0}^{k_0+q-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \\ + 2q^2(2c-1)^2\bar{\mu}^{\,2}\lambda_{\max} \sum_{k=k_0}^{k_0+q-1} \lambda_k \,\mathbb{E}\Big[\|e_k\|^2\Big].\end{aligned} \tag{5.47}$$

Proof. Using (5.26) and (5.29), we have

$$\|\Delta_{k+1}\| \le (1-\alpha_{k+1})\|\Delta_k\| + |\lambda_k - \beta_k|\,\|d_k\| \le \|\Delta_k\| + |\lambda_k - \beta_k|\,\|d_k\|. \tag{5.48}$$

Because $\Delta_{k_0} = 0$, iterating (5.48) within the epoch, we get

$$\|\Delta_k\| \le \sum_{j=k_0}^{k-1} |\lambda_j - \beta_j|\,\|d_j\| \le \sum_{j=k_0}^{k_0+q-1} |\lambda_j - \beta_j|\,\|d_j\|.$$

Summing over $k = k_0, \ldots, k_0+q-1$ yields

$$\sum_{k=k_0}^{k_0+q-1} \|\Delta_k\|^2 \le q^2 \sum_{j=k_0}^{k_0+q-1} (\lambda_j - \beta_j)^2 \|d_j\|^2. \tag{5.49}$$

From the step-size condition and $\alpha_{k+1} \le 1$, we have

$$0 \le \lambda_k - \beta_k \le (c(1+\alpha_{k+1}) - 1)\beta_k \le (2c-1)\beta_k.$$

Because $\beta_k \le \lambda_k \le \lambda_{\max} \le 1$, it follows that

$$(\lambda_k - \beta_k)^2 \le (2c-1)^2\beta_k^2 \le (2c-1)^2\lambda_k. \tag{5.50}$$

Using (5.49), (5.50), and $\lambda_k \le \lambda_{\max}$, we obtain

$$\sum_{k=k_0}^{k_0+q-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big] \le q^2(2c-1)^2\lambda_{\max} \sum_{k=k_0}^{k_0+q-1} \lambda_k \,\mathbb{E}\Big[\|d_k\|^2\Big].$$

Moreover, by $d_k = H_k g_k$, $\|H_k\| \le \bar{\mu}$, and $g_k = \nabla f(z_k) + e_k$, we have

$$\mathbb{E}\Big[\|d_k\|^2\Big] \le 2\bar{\mu}^{\,2}\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + 2\bar{\mu}^{\,2}\mathbb{E}\Big[\|e_k\|^2\Big].$$

The proof of the lemma is finished. □

Combining Lemma 5.15 with the preceding recursion and summing over k, together with the random output rule $\mathbb{P}(\zeta = k) = \lambda_k / \sum_{j=0}^{Nq-1} \lambda_j$, yields the final bound on $\mathbb{E}\|\nabla f(z_\zeta)\|^2$ and the corresponding SFO complexity. This completes the transition from the accumulated descent estimate to the final stationarity result.

Theorem 5.1. *Let $K = Nq$. Suppose that Assumptions 2.1–2.3 and the conditions in Lemmas 5.11, 5.14, and 5.15 hold over all epochs. If*

$$BC_0 \le \frac{1}{15},$$

then

$$\sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \le \frac{16(\Phi_0 - f^*)}{\underline{\mu}}, \tag{5.51}$$

where denote

$$\lambda_{\max} := \max_{0\le k\le K-1} \lambda_k, \qquad C_0 := \frac{4L^2}{b}, \qquad B := 2q^2(2c-1)^2\bar{\mu}^2\lambda_{\max}.$$

The condition $BC_0 \le 1/15$ holds whenever

$$1 < c \le \frac{5}{4}, \qquad b \ge \max\left\{1, \left\lceil 270q^2L^2\bar{\mu}^2\lambda_{\max}\right\rceil\right\}.$$

Moreover, if $\zeta \in \{0, 1, \ldots, K-1\}$ is sampled according to

$$\mathbb{P}(\zeta = k) = \frac{\lambda_k}{\sum_{t=0}^{K-1}\lambda_t}, \qquad k = 0, 1, \ldots, K-1,$$

then

$$\mathbb{E}\Big[\|\nabla f(z_\zeta)\|^2\Big] \le \frac{16(\Phi_0 - f^*)}{\underline{\mu}\sum_{t=0}^{K-1}\lambda_t}. \tag{5.52}$$

Proof. Summing (5.35) over $k = 0, \ldots, K-1$ and using $\Phi_K \ge f^*$, we obtain that

$$a_g \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \le \Phi_0 - f^* + a_e \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|e_k\|^2\Big]. \tag{5.53}$$

Applying Lemma 5.14 over all epochs yields

$$\sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|e_k\|^2\Big] \le \frac{a_g}{2a_e} \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + C_0 \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big]. \tag{5.54}$$

Substituting (5.54) into (5.53), we have

$$\frac{\underline{\mu}}{8} \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \le \Phi_0 - f^* + \frac{3\underline{\mu}}{4} C_0 \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big]. \tag{5.55}$$

Similarly, by use of Lemma 5.15, we know that

$$\sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big] \le B \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] + B \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|e_k\|^2\Big]. \tag{5.56}$$

Combining (5.56) with (5.54) and using $a_g/(2a_e) = 1/6$, we obtain

$$\sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\Delta_k\|^2\Big] \le \frac{7B}{6(1 - BC_0)} \sum_{k=0}^{K-1} \lambda_k \,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big]. \tag{5.57}$$

Substituting (5.57) into (5.55), we have

$$\frac{\underline{\mu}(1-8BC_0)}{8(1-BC_0)}\sum_{k=0}^{K-1}\lambda_k\,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big] \le \Phi_0 - f^*. \tag{5.58}$$

Under the stated parameter condition, we can get (5.51).

By the sampling rule for ζ, we know that

$$\mathbb{E}\Big[\|\nabla f(z_\zeta)\|^2\Big] = \frac{\sum_{k=0}^{K-1}\lambda_k\,\mathbb{E}\Big[\|\nabla f(z_k)\|^2\Big]}{\sum_{t=0}^{K-1}\lambda_t}.$$

Applying (5.51), we complete the proof of (5.52). □

Theorem 5.1 establishes a first-order stationarity bound for the random output of the proposed method. Based on this result, we next specify the epoch length and mini-batch size to derive the SFO complexity required to reach an ε-first-order stationary point in the finite-sum setting.

Theorem 5.2. *Suppose that the conditions of Theorem 5.1 hold for the finite-sum model* (1.2) *and that* $\lambda_k \equiv \lambda$. *Choose*

$$q = \Theta(n^{1/3}), \qquad b = \Theta(n^{2/3}).$$

Let the output index $\zeta \in \{0, 1, \ldots, K-1\}$ *be sampled according to*

$$\mathbb{P}(\zeta = k) = \frac{\lambda_k}{\sum_{t=0}^{K-1}\lambda_t}, \qquad k = 0, 1, \ldots, K-1.$$

Then, to obtain

$$\mathbb{E}\Big[\|\nabla f(z_\zeta)\|^2\Big] \le \varepsilon^2,$$

the stochastic first-order oracle complexity satisfies

$$\mathrm{SFO} = \Theta\Big(n^{2/3}\varepsilon^{-2}\Big), \tag{5.59}$$

where SFO *denotes the total number of component-gradient evaluations.*

Proof. It follows from $\lambda_k \equiv \lambda$ and (5.52) that

$$\mathbb{E}\Big[\|\nabla f(z_\zeta)\|^2\Big] \le \frac{16(\Phi_0 - f^*)}{\underline{\mu} K\lambda}.$$

Thus, the condition $\mathbb{E}\Big[\|\nabla f(z_\zeta)\|^2\Big] \le \varepsilon^2$ is guaranteed whenever

$$K \ge \frac{16(\Phi_0 - f^*)}{\underline{\mu}\lambda\,\varepsilon^2}. \tag{5.60}$$

For the finite-sum model, each epoch computes one full gradient and hence costs n component-gradient evaluations, and each inner iteration costs $2b$. Because $N = K/q$, we have

$$\mathrm{SFO} = Nn + 2Kb = \frac{K}{q}n + 2Kb. \tag{5.61}$$

The parameter conditions allow the choice $b = \Theta(q^2)$. Hence,

$$\text{SFO} = \Theta\left(\frac{K}{q}n + Kq^2\right). \tag{5.62}$$

Taking $q = \Theta(n^{1/3})$ and $b = \Theta(n^{2/3})$ balances the two terms in (5.62), yielding

$$\text{SFO} = \Theta\left(Kn^{2/3}\right). \tag{5.63}$$

Using (5.60) and (5.63), we complete the proof. □

6. Numerical experiments

This section reports numerical results for the proposed algorithm on a nonconvex support vector machine (SVM) problem and a nonconvex regularized logistic regression (LR) problem. All experiments were implemented in Python 3.14 and run on a personal computer with a 1.2 GHz CPU and 32 GB of memory.

To evaluate the performance of the proposed algorithm, we test it on five public binary classification data sets and one synthetic data set. These data sets differ in sample size and feature dimension and thus provide a reasonable range of test problems. For the synthetic data set, the binary classification test data are generated by prespecifying the sample size, feature dimension, and related parameters. Detailed information on the data sets is given in Table 1.

Table 1. Information about the data sets.

Data set	No. of samples	Feature dimension	Type
a6a	11220	122	Public
a7a	16461	123	Public
w8a	49749	300	Public
realsim	72309	20958	Public
covtype	581012	54	Public
Synthetic	200000	200	Synthetic

The numerical performance of the algorithms is evaluated in terms of the objective function value, gradient norm, and classification accuracy. Because the data sets differ in sample size, feature dimension, and data characteristics, the parameter settings are selected separately for each data set. In the subsequent experiments, we compare the proposed algorithm with four other algorithms and further examine the effects of algorithmic variants and parameter settings on numerical performance. The specific parameter values used in each group of experiments will be presented together with the corresponding results. Unless otherwise stated, we choose the damping parameter as $\varpi = 0.25$, the modified-secant parameter as $\tau = 0.2$, the truncation parameter as $\tau_{\max} = 1$, and the initial scaling safeguard parameter as $\delta = 10^{-3}$. For each data set, all algorithms use the same step size. In addition, for those methods involving mini-batch sampling and limited-memory updates, the same mini-batch size and memory length are adopted on the same data set. All algorithms are run with the same number of iterations as the stopping criterion on each data set and are compared under the same evaluation metrics to ensure a fair comparison.

6.1. Numerical results on the nonconvex support vector machine problem

We first conduct numerical experiments on the following nonconvex SVM problem with a sigmoid loss, which was studied in [30]:

$$\min_{x\in\mathbb{R}^n} F(x) = \frac{1}{T}\sum_{i=1}^{T}(1-\tanh(\bar{v}_i\bar{u}_i^T x)) + \varrho\|x\|_2^2, \tag{6.1}$$

where $\varrho > 0$ is a regularization parameter, $\bar{u}_i \in \mathbb{R}^n$ denotes the feature vector associated with sample i, and $\bar{v}_i \in \{-1, +1\}$ is the corresponding binary label. In all SVM experiments, we choose $\varrho = 10^{-4}$.

6.1.1. Parameter sensitivity analysis and parameter selection

Before comparing the proposed method with other algorithms, we first investigate the effects of the mini-batch size b and the memory length M on the numerical performance of M-SMdLBFGS-VR. The mini-batch size affects the variance level of the stochastic gradient estimator, and the memory length determines how much historical curvature information is used in the limited-memory quasi-Newton update. Therefore, both parameters play an important role in the efficiency and stability of the algorithm.

Specifically, we test the mini-batch size $b \in \{64, 128, 256, 512, 1024, 2048\}$ and the memory length $M \in \{1, 5, 10, 15, 20, 50\}$. The performance is evaluated in terms of the objective function value and the gradient norm, and the results are presented in the form of heat maps. For each data set, the parameter combination yielding favorable overall performance is selected and used in the subsequent comparison experiments.

For the four-panel figures in this subsection, panels (a) and (c) present the objective function values, whereas panels (b) and (d) present the corresponding gradient norms for the two data sets indicated in each caption.

Figure 1 indicates that the a6a and a7a data sets show similar parameter-sensitivity patterns. For these two relatively small data sets, moderate choices of b and M lead to stable performance, whereas overly large mini-batch sizes or memory lengths may introduce unnecessary sampling and curvature-updating costs without providing additional useful information. This may explain the performance deterioration observed for large values of b or M.

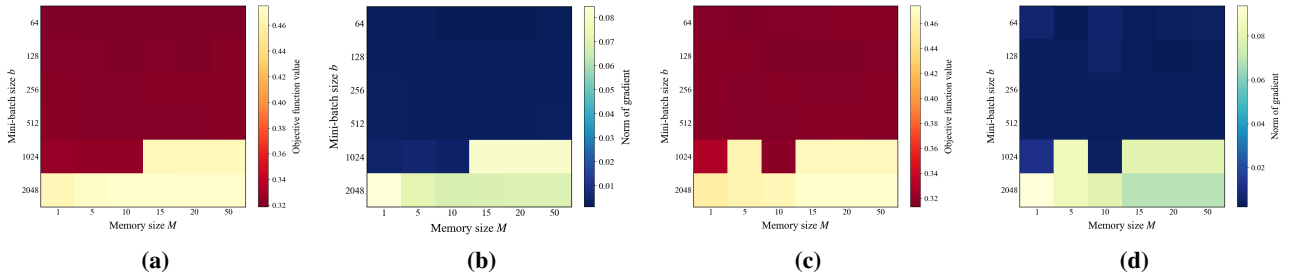


Figure 1. Parameter sensitivity of M-SMdLBFGS-VR on the a6a and a7a data sets.

Figure 2 shows different sensitivity patterns on the w8a and realsim data sets. The performance on w8a changes only mildly across most parameter combinations, suggesting that this data set is relatively insensitive to the choices of b and M. In contrast, realsim exhibits more pronounced variations in both the objective function value and the gradient norm. This may be related to its much higher feature dimension, which makes the quality of stochastic gradient estimates and curvature information more dependent on the parameter selection.

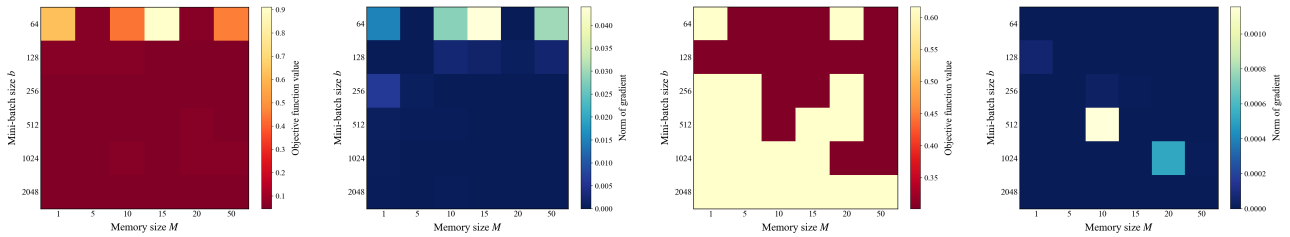


Figure 2. Parameter sensitivity of M-SMdLBFGS-VR on the w8a and realsim data sets.

Figure 3 indicates that the covtype and Synthetic data sets are more sensitive to parameter choices. This sensitivity is mainly reflected in the mini-batch size b, whereas the effect of the memory length M is relatively less pronounced. For these two large-scale data sets, moderate choices of b and M usually lead to more stable numerical performance. In contrast, excessively large mini-batch sizes do not necessarily further improve the effectiveness of stochastic gradient estimation; under a fixed iteration budget, they may reduce the flexibility of stochastic updates and lead to larger objective function values or gradient norms. These observations suggest that parameter selection should balance variance control, the use of curvature information, and iteration efficiency.

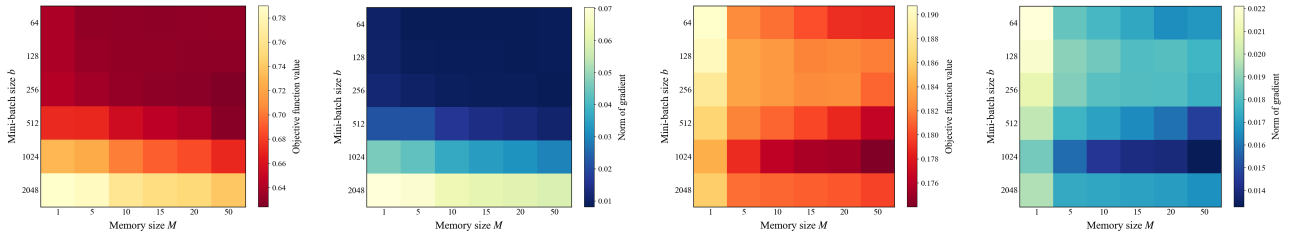


Figure 3. Parameter sensitivity of M-SMdLBFGS-VR on the covtype and Synthetic data sets.

Based on the heat-map results and the overall convergence behavior, the selected parameters for different data sets are summarized in Table 2. These parameters are used in the main comparison experiments in Section 6.1.2. Here, K denotes the iteration budget used in the comparison experiments.

Table 2. Selected parameters for different data sets.

Data set	Step size	Mini-batch size b	Memory length M	K
a6a	0.01	128	20	1000
a7a	0.01	128	20	1000
w8a	0.01	256	20	1000
realsim	0.05	256	10	1000
covtype	0.0005	512	50	10000
Synthetic	0.01	1024	50	5000

6.1.2. Comparison with other algorithms

In this subsection, we compare the proposed M-SMdLBFGS-VR algorithm with four representative algorithms on six data sets to evaluate its numerical performance. The competing algorithms include SVRG [19], Katyusha [42], SQN [24], and SdLBFGS-VR [30]. In all experiments involving Katyusha, the parameters are set to $\tau_1 = 0.5$ and $\tau_2 = 0.25$. The parameter settings for different data sets follow Table 2. For each data set, we report the objective function value and the gradient norm versus the number of iterations, and the final classification accuracies are summarized in Table 3. For compactness, in the subsequent experiments, "obj." and "gra." in the subfigure captions denote the objective function value and the gradient norm, respectively.

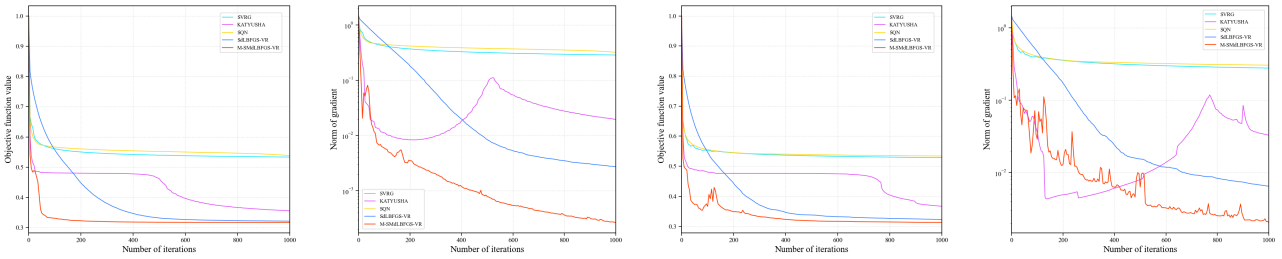
Table 3. Classification accuracies of the five algorithms on six data sets.

Data set	SVRG	Katyusha	SQN	SdLBFGS-VR	M-SMdLBFGS-VR
a6a	76.01%	83.40%	76.01%	84.73%	85.21%
a7a	76.17%	83.49%	76.17%	84.83%	85.24%
w8a	97.03%	97.03%	97.03%	97.03%	97.03%
realsim	69.25%	69.25%	69.25%	95.81%	96.27%
covtype	60.59%	63.01%	63.51%	71.13%	71.23%
Synthetic	84.06%	93.04%	84.47%	93.47%	93.51%

Figure 4 shows the comparison results on the a6a and a7a data sets. For both data sets, M-SMdLBFGS-VR achieves lower objective function values than the competing algorithms within the prescribed number of iterations. Its gradient norm also decreases rapidly and remains at a relatively low level, indicating that the proposed method is effective and stable on these relatively small-scale data sets.

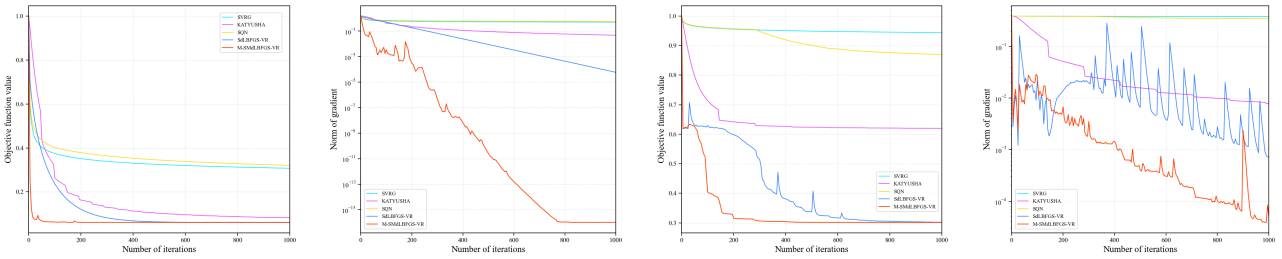
Figure 5 presents the results on the w8a and realsim data sets. For w8a, M-SMdLBFGS-VR decreases the objective function value slightly faster than the other algorithms, and its gradient norm remains at a lower level despite mild fluctuations. For realsim, the proposed method shows more evident advantages: It reaches a lower objective function value and produces a smaller final gradient norm than most competing algorithms. These results suggest that the proposed method remains effective on data sets with different feature dimensions and data characteristics.

Figure 6 reports the results on the covtype and synthetic data sets. On covtype, M-SMdLBFGS-VR and SdLBFGS-VR both show favorable convergence behavior, with the proposed method performing slightly better overall. In particular, M-SMdLBFGS-VR achieves a faster decrease in the objective function value and a lower gradient norm. On the synthetic data set, the proposed method also outperforms the other competing algorithms in terms of both the objective function value and the gradient norm. This indicates that the proposed method can retain stable numerical performance on both real and generated large-scale problems.



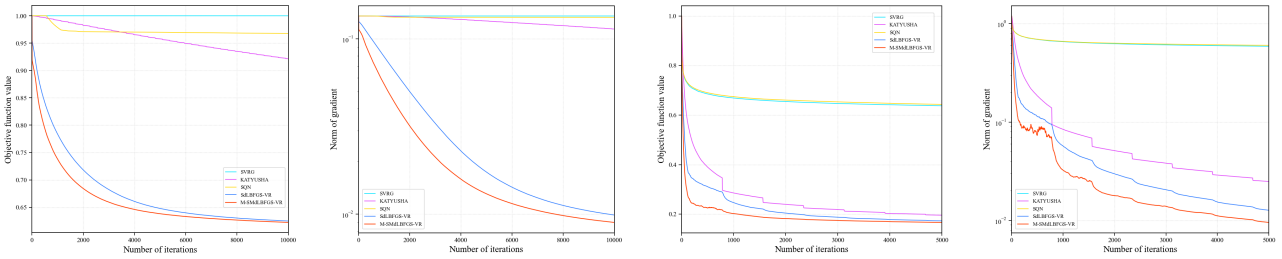
(a) a6a: obj. **(b)** a6a: gra. **(c)** a7a: obj. **(d)** a7a: gra.

Figure 4. Comparison of the five algorithms on the a6a and a7a data sets.



(a) w8a: obj. **(b)** w8a: gra. **(c)** realsim: obj. **(d)** realsim: gra.

Figure 5. Comparison of the five algorithms on the w8a and realsim data sets.



(a) covtype: obj. **(b)** covtype: gra. **(c)** Synthetic: obj. **(d)** Synthetic: gra.

Figure 6. Comparison of the five algorithms on the covtype and Synthetic data sets.

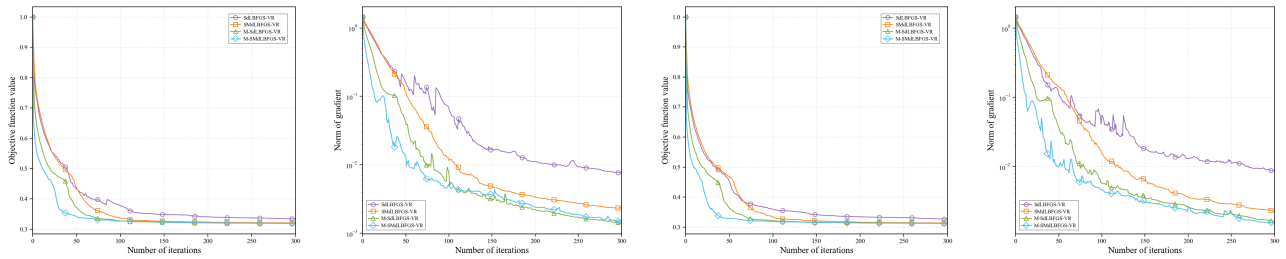
To further compare the final classification performance of the algorithms, the classification accuracies on the six data sets are reported in Table 3.

Collectively, the experimental results on the six data sets show that M-SMdLBFGS-VR achieves competitive or superior overall performance compared with the other algorithms. It generally converges faster and exhibits more stable behavior in reducing the objective function value and gradient norm. In terms of classification accuracy, M-SMdLBFGS-VR attains the highest accuracy on a6a, a7a, realsim, covtype, and synthetic data, and achieves a tied best result on w8a. These results indicate that the proposed method not only improves optimization efficiency but also yields strong classification performance for the nonconvex classification model considered in this paper.

6.1.3. Influence of the modified secant strategy and the momentum mechanism

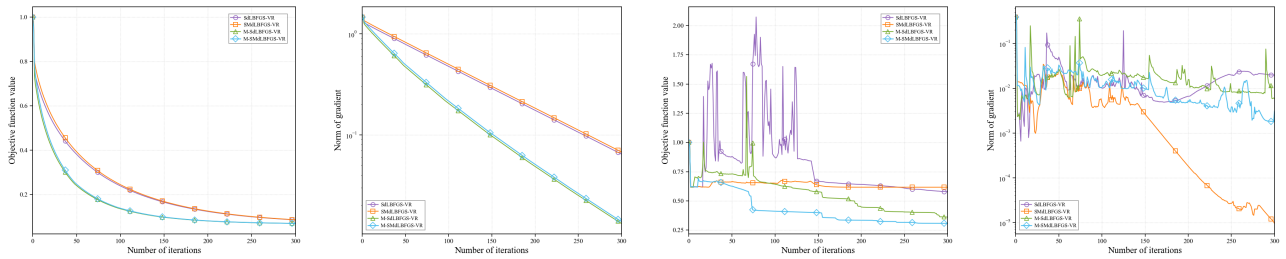
In this subsection, we examine the effects of the modified secant strategy and the momentum mechanism on the performance of the proposed method. Four variants are compared: SdLBFGS-VR, SMdLBFGS-VR, M-SdLBFGS-VR, and M-SMdLBFGS-VR. Here, SMdLBFGS-VR incorporates only the modified secant strategy into SdLBFGS-VR, M-SdLBFGS-VR incorporates only the momentum mechanism, and M-SMdLBFGS-VR combines both techniques. In the experiments of Section 6.1.3, for each data set, all variants are run with the same common parameter settings, including the mini-batch size b, memory length M, step size, and other relevant parameters. The iteration budget is set to $K = 300$ or $K = 1000$. The comparison is conducted in terms of the objective function value and the gradient norm versus the number of iterations.

Figures 7–9 show that both the modified secant strategy and the momentum mechanism affect the convergence behavior of the algorithm, although their effects vary across different data sets.



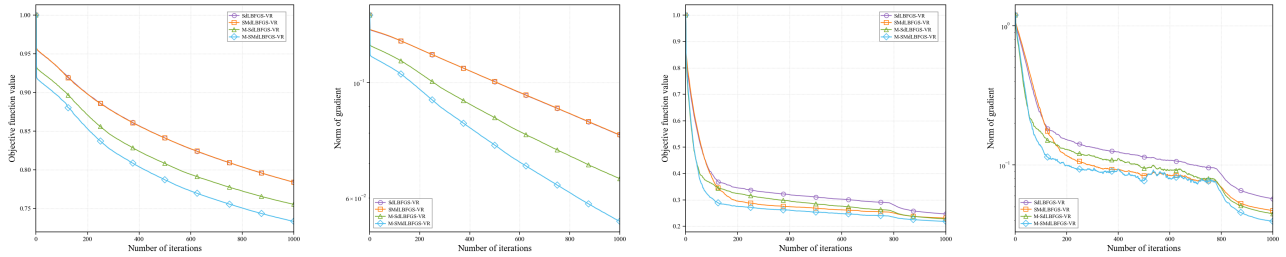
(a) a6a: obj. **(b)** a6a: gra. **(c)** a7a: obj. **(d)** a7a: gra.

Figure 7. Performance of the four variants on the a6a and a7a data sets.



(a) w8a: obj. **(b)** w8a: gra. **(c)** realsim: obj. **(d)** realsim: gra.

Figure 8. Performance of the four variants on the w8a and realsim data sets.



(a) covtype: obj. **(b)** covtype: gra. **(c)** Synthetic: obj. **(d)** Synthetic: gra.

Figure 9. Performance of the four variants on the covtype and Synthetic data sets.

Compared with the SdLBFGS-VR method, SMdLBFGS-VR improves the objective value or gradient norm in some cases, indicating that the modified secant strategy can enhance the quality of curvature information, but its effect is not uniform across all data sets. The momentum-based variant M-SdLBFGS-VR generally shows faster decrease in the early stage, suggesting that the momentum mechanism is effective in accelerating the iterative process. Overall, M-SMdLBFGS-VR achieves the best or near-best performance in most cases, which indicates that the modified secant strategy and the momentum mechanism are complementary and that their combination can improve the stability and efficiency of the stochastic quasi-Newton method.

6.2. Numerical results on the nonconvex regularized logistic regression problem

We next conduct numerical experiments on the following nonconvex regularized LR problem, which was studied in [43]:

$$\min_{x\in\mathbb{R}^n} F(x) = \frac{1}{T}\sum_{i=1}^{T}\log\left(1+\exp\left(-\bar{v}_i\bar{u}_i^T x\right)\right) + \varrho\sum_{j=1}^{n}\frac{\alpha x_j^2}{1+\alpha x_j^2}, \tag{6.2}$$

where $\varrho > 0$ is the regularization parameter, $\alpha > 0$ controls the shape of the nonconvex regularizer, $\bar{u}_i \in \mathbb{R}^n$ denotes the feature vector associated with sample i, and $\bar{v}_i \in \{-1, +1\}$ is the corresponding binary label. We choose $\varrho = 10^{-3}$ and $\alpha = 1$.

6.2.1. Parameter sensitivity analysis and parameter selection

For the nonconvex regularized LR problem, we follow the same parameter-sensitivity analysis procedure as that described in Section 6.1.1. Specifically, we test $b \in \{64, 128, 256, 512, 1024, 2048\}$ and $M \in \{1, 5, 10, 15, 20, 50\}$. The objective function values and gradient norms are reported in the form of heat maps. Based on the overall performance, an appropriate combination of b and M is selected for each data set and used in the subsequent comparison experiments.

Overall, Figures 10–12 exhibit sensitivity patterns broadly similar to those observed in Section 6.1.1. For most data sets, the numerical performance is more sensitive to the mini-batch size b than to the memory length M, although the extent of the variation differs across data sets. These results further indicate that a common parameter setting is not appropriate for all data sets, and they justify selecting b and M separately for the subsequent comparison experiments.

Based on the above parameter sensitivity results, we use the same parameter settings as those listed in Table 2 in the subsequent experiments.

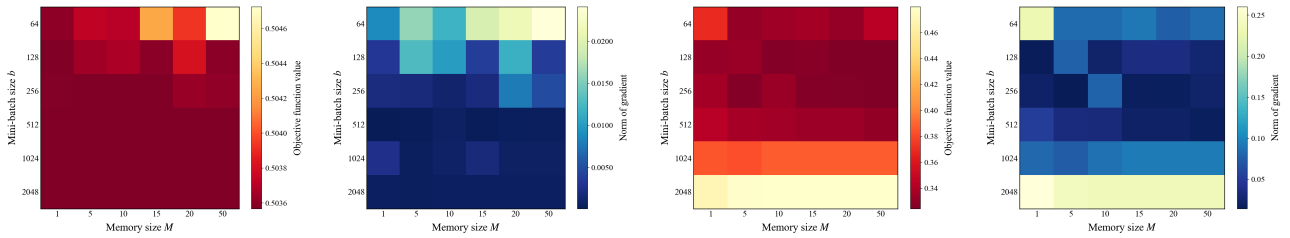


Figure 10. Parameter sensitivity of M-SMdLBFGS-VR on the a6a and a7a data sets.

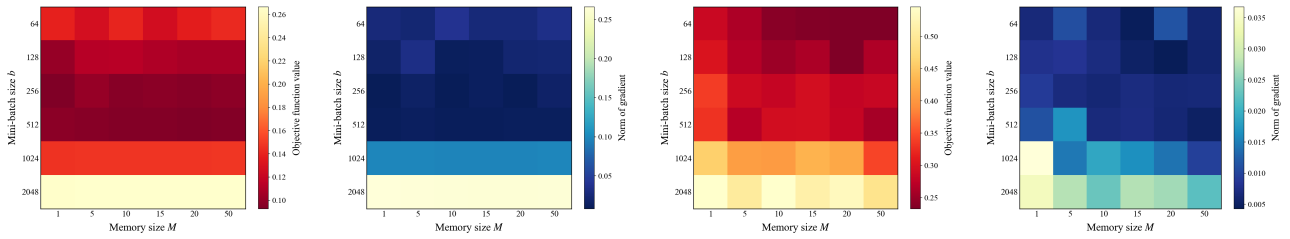


Figure 11. Parameter sensitivity of M-SMdLBFGS-VR on the w8a and realsim data sets.

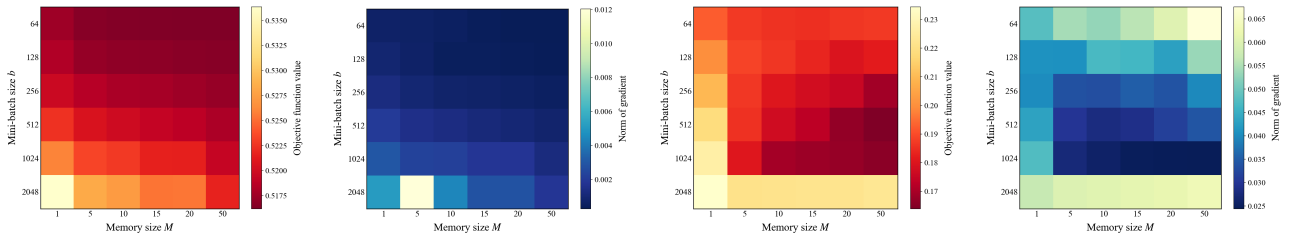
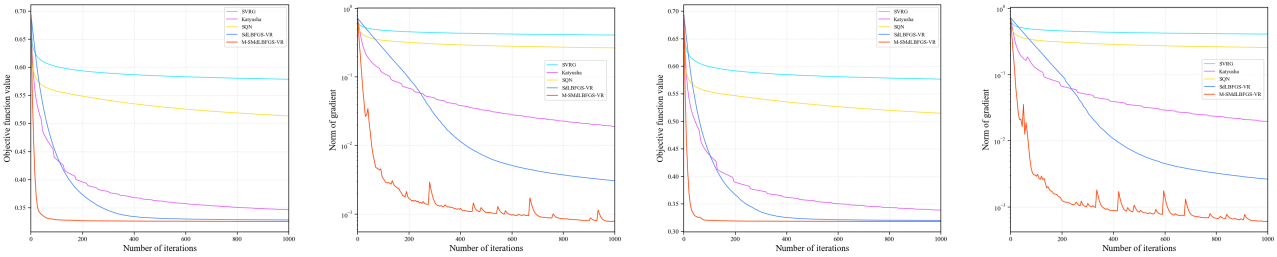


Figure 12. Parameter sensitivity of M-SMdLBFGS-VR on the covtype and Synthetic data sets.

6.2.2. Comparison with other algorithms

Following the experimental protocol used for the nonconvex SVM problem, we compare M-SMdLBFGS-VR with SVRG, Katyusha, SQN, and SdLBFGS-VR on the nonconvex regularized LR problem. The competing algorithms, parameter settings, stopping criteria, and evaluation metrics are the same as those used in Section 6.1.2. For each data set, we report the objective function value and the gradient norm versus the number of iterations, together with the final classification accuracy.

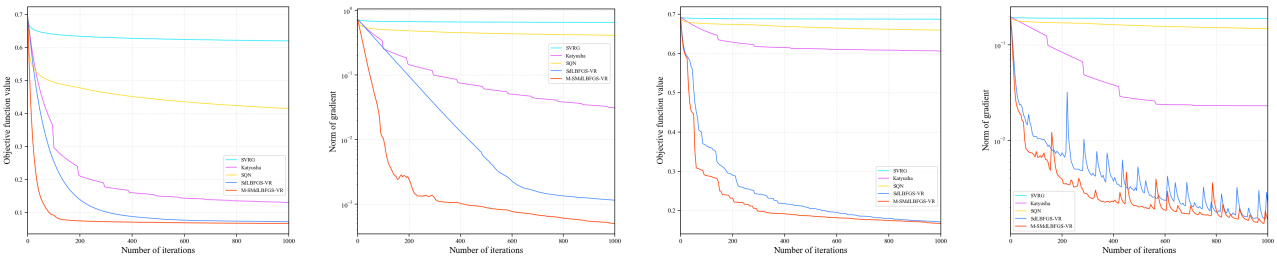
Figure 13 presents the comparison results on the a6a and a7a data sets. On both data sets, M-SMdLBFGS-VR reduces the objective function value more rapidly and reaches a lower level than the competing algorithms. Its gradient norm also decreases to the lowest overall level, demonstrating its favorable convergence performance on these two data sets.



(a) a6a: obj. **(b)** a6a: gra. **(c)** a7a: obj. **(d)** a7a: gra.

Figure 13. Comparison of the five algorithms on the a6a and a7a data sets.

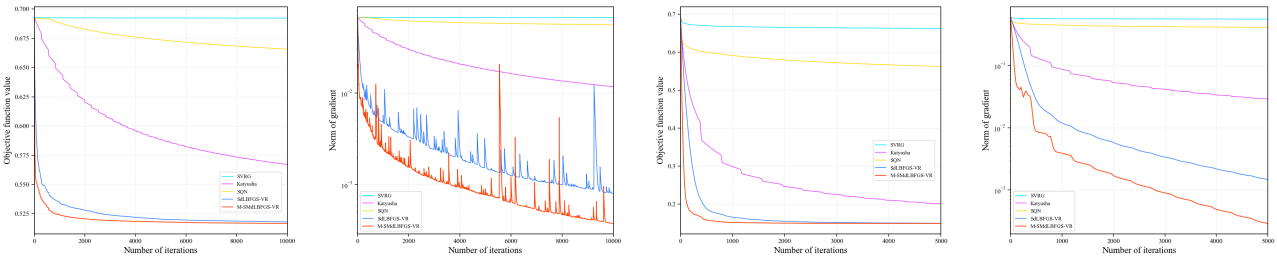
Figure 14 presents the results on the w8a and realsim data sets. On w8a, M-SMdLBFGS-VR rapidly reaches the lowest objective function value and gradient norm. On realsim, it also maintains a slight advantage over SdLBFGS-VR despite mild fluctuations, whereas the other algorithms converge considerably more slowly.



(a) w8a: obj. **(b)** w8a: gra. **(c)** realsim: obj. **(d)** realsim: gra.

Figure 14. Comparison of the five algorithms on the w8a and realsim data sets.

The results in Figure 15 further confirm the effectiveness of M-SMdLBFGS-VR on the covtype and synthetic data sets. On covtype, it maintains lower objective function values and gradient norms than SdLBFGS-VR despite occasional fluctuations. On the synthetic data set, it reaches the stable region earlier and yields the smallest final gradient norm among all the methods.



(a) covtype: obj. **(b)** covtype: gra. **(c)** Synthetic: obj. **(d)** Synthetic: gra.

Figure 15. Comparison of the five algorithms on the covtype and Synthetic data sets.

To further evaluate the final classification performance on the nonconvex regularized LR problem, the classification accuracies of the five algorithms on the six data sets are reported in Table 4.

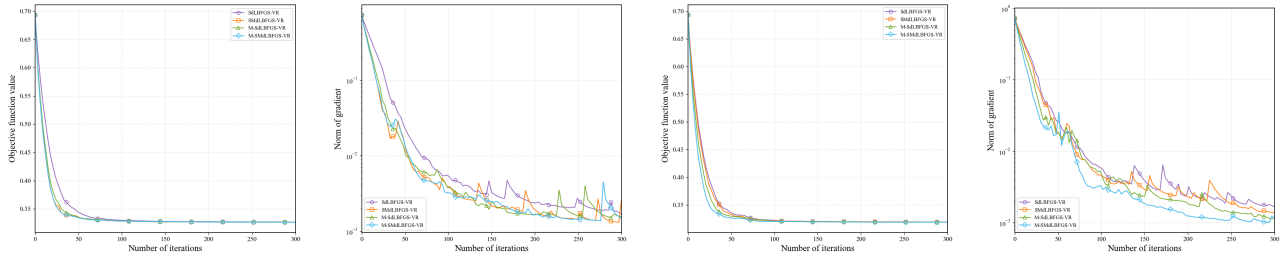
Table 4. Classification accuracies of the five algorithms on six data sets for the nonconvex regularized LR problem.

Data set	SVRG	Katyusha	SQN	SdLBFGS-VR	M-SMdLBFGS-VR
a6a	76.01%	83.70%	76.02%	84.55%	84.70%
a7a	76.17%	84.31%	76.17%	85.00%	85.07%
w8a	97.03%	97.03%	97.03%	98.13%	98.44%
realsim	69.25%	69.25%	69.25%	69.25%	69.25%
covtype	61.04%	70.41%	70.69%	75.57%	75.62%
Synthetic	82.81%	92.58%	85.86%	93.59%	93.60%

In terms of classification accuracy, M-SMdLBFGS-VR achieves the best results on five of the six data sets, demonstrating its consistently strong classification performance. On realsim, all the compared algorithms attain the same accuracy of 69.25%.

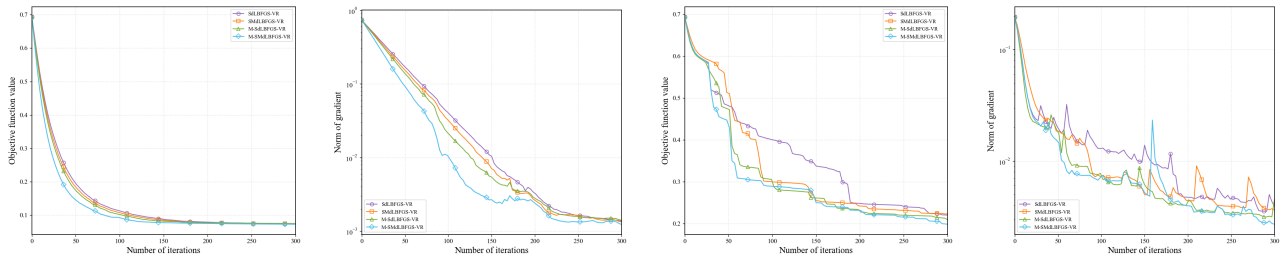
6.2.3. Influence of the modified secant strategy and the momentum mechanism

We compare four variants, SdLBFGS-VR, SMdLBFGS-VR, M-SdLBFGS-VR, and M-SMdLBFGS-VR, using the same experimental settings as those described in Section 6.1. The results show that both the modified secant strategy and the momentum mechanism improve the convergence behavior of SdLBFGS-VR. SMdLBFGS-VR generally provides better curvature information, whereas M-SdLBFGS-VR accelerates the decrease of the objective function value and gradient norm. By combining these two techniques, M-SMdLBFGS-VR achieves the best or near-best performance on most data sets, demonstrating the complementary effects of the two strategies.



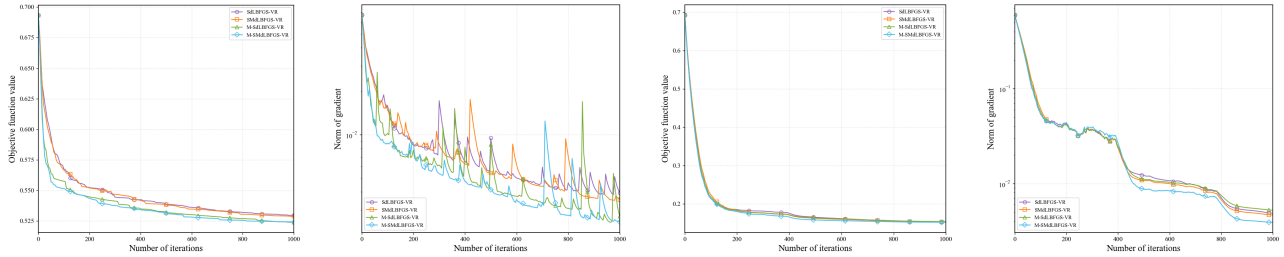
(a) a6a: obj. **(b)** a6a: gra. **(c)** a7a: obj. **(d)** a7a: gra.

Figure 16. Comparison of the algorithms on the a6a and a7a data sets.



(a) w8a: obj. **(b)** w8a: gra. **(c)** realsim: obj. **(d)** realsim: gra.

Figure 17. Comparison of the algorithms on the w8a and realsim data sets.



(a) covtype: obj. **(b)** covtype: gra. **(c)** Synthetic: obj. **(d)** Synthetic: gra.

Figure 18. Comparison of the algorithms on the covtype and Synthetic data sets.

7. Conclusions

This paper considers large-scale stochastic nonconvex optimization problems and proposes the M-SMdLBFGS-VR method by incorporating a modified secant strategy and a momentum mechanism into the SdLBFGS-VR framework. The proposed method attains an SFO complexity of $O(n^{2/3}\varepsilon^{-2})$. Numerical results show that the proposed method achieves favorable performance in terms of the objective function value, gradient norm, and classification accuracy, and the modified secant strategy and momentum mechanism both make positive contributions to the improvement of the algorithm. Building on these results, we will examine the applicability of the proposed method to more complex machine learning settings, including multiclass classification and larger-scale datasets. We will also investigate its use in online learning and deep learning, with a focus on computational cost, stability, and convergence.

Author contributions

Runmin Wang: Methodology, Writing-original draft; Jing Gao: Writing-original draft, Writing-review and editing; Yuchang Zhang: Visualization; Xiaoyu Fu: Data curation; Wenwen Wang: Software. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors would like to thank the Youth Science and Technology Talent Cultivation Project of Jilin Province (20250602005RC), the Doctoral Research Project Start-up Fund of Beihua University, and the Graduate Innovation Project of Beihua University (2024038).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. X. Tian, Z. Tian, S. F. A. Khatib, Y. Wang, Machine learning in internet financial risk management: A systematic literature review, *PLoS ONE*, **19** (2024), e0300195. https://doi.org/10.1371/journal.pone.0300195
2. L. Wang, E. Demeulemeester, Simulation optimization in healthcare resource planning: A literature review, *IISE Trans.*, **55** (2023), 985–1007. https://doi.org/10.1080/24725854.2022.2147606
3. X. Sui, H. Yan, S. Pan, X. Li, X. Gu, Bus system optimization for timetables, routes, charging, and facilities: A summary, *Digit. Transp. Saf.*, **4** (2025), 1–9. https://doi.org/10.48130/dts-0024-0024
4. H. Robbins, S. Monro, A stochastic approximation method, *Ann. Math. Statist.*, **22** (1951), 400–407.
5. S. Ghadimi, G. Lan, Stochastic first- and zeroth-order methods for nonconvex stochastic programming, *SIAM J. Optim.*, **23** (2013), 2341–2368. https://doi.org/10.1137/120880811
6. B. T. Polyak, A. B. Juditsky, Acceleration of stochastic approximation by averaging, *SIAM J. Control Optim.*, **30** (1992), 838–855. https://doi.org/10.1137/0330046
7. T. Xiao, G. Yang, A convergence study of SGD-type methods for stochastic optimization, *Numer. Math. Theory Methods Appl.*, **16** (2023), 914–930. https://doi.org/10.4208/nmtma.oa-2022-0179
8. L. Bottou, F. E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, *SIAM Rev.*, **60** (2018), 223–311. https://doi.org/10.1137/16M1080173
9. L. Bottou, Stochastic gradient descent tricks, In: *Neural Networks: Tricks of the Trade*, 2 Eds., Berlin, Heidelberg: Springer, 2012. http://doi.org/10.1007/978-3-642-35289-8_25
10. A. Nemirovski, A. Juditsky, G. Lan, A. Shapiro, Robust stochastic approximation approach to stochastic programming, *SIAM J. Optim.*, **19** (2009), 1574–1609. https://doi.org/10.1137/070704277
11. R. L. Karandikar, M. Vidyasagar, Convergence rates for stochastic approximation: Biased noise with unbounded variance, and applications, *J. Optim. Theory Appl.*, **203** (2024), 2412–2450. https://doi.org/10.1007/s10957-024-02547-7
12. L. Madden, E. Dall'Anese, S. Becker, High probability convergence bounds for non-convex stochastic gradient descent with sub-Weibull noise, *J. Mach. Learn. Res.*, **25** (2024), 1–36.
13. B. T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.*, **4** (1964), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5
14. I. Sutskever, J. Martens, G. E. Dahl, G. E. Hinton, On the importance of initialization and momentum in deep learning, In: *International Conference on Machine Learning*, 2013, 1139–1147.
15. X. Li, Z. Yang, Y. Wang, Stochastic gradient accelerated by negative momentum for training deep neural networks, *Appl. Intell.*, **55** (2025), 970. https://doi.org/10.1007/s10489-025-06900-9

16. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, **12** (2011), 2121–2159.

17. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, In: *3rd International Conference on Learning Representations, San Diego, USA, Conference Track Proceedings*, 2015. https://doi.org/10.48550/arXiv.1412.6980

18. C. Fang, C. J. Li, Z. Lin, T. Zhang, SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator, In: *Advances in Neural Information Processing Systems*, **31** (2018), 689–699.

19. R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, In: *Neural Information Processing Systems*, **26** (2013), 315–323.

20. L. M. Nguyen, J. Liu, K. Scheinberg, M. Takáč, SARAH: A novel method for machine learning problems using stochastic recursive gradient, In: *Proceedings of the 34th International Conference on Machine Learning*, **70** (2017), 2613–2621.

21. C. Traoré, V. Apidopoulos, S. Salzo, S. Villa, Variance reduction techniques for stochastic proximal point algorithms, *J. Optim. Theory Appl.*, **203** (2024), 1910–1939. https://doi.org/10.1007/s10957-024-02502-6

22. B. Recht, C. Ré, S. J. Wright, F. Niu, HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent, In: *Neural Information Processing Systems*, 2011.

23. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, et al., Large scale distributed deep networks, In: *Neural Information Processing Systems*, 2012.

24. R. H. Byrd, S. L. Hansen, J. Nocedal, Y. Singer, A stochastic quasi-Newton method for large-scale optimization, *SIAM J. Optim.*, **26** (2016), 1008–1031. https://doi.org/10.1137/140954362

25. P. Moritz, R. Nishihara, M. I. Jordan, A linearly-convergent stochastic L-BFGS algorithm, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, **51** (2016), 249–258.

26. R. Bollapragada, J. Nocedal, D. Mudigere, H. J. M. Shi, P. T. P. Tang, A progressive batching L-BFGS method for machine learning, In: *Proceedings of the 35th International Conference on Machine Learning*, **80** (2018), 620–629.

27. A. Mokhtari, A. Ribeiro, RES: Regularized stochastic BFGS algorithm, *IEEE Trans. Signal Process.*, **62** (2014), 6089–6104. https://doi.org/10.1109/tsp.2014.2357775

28. D. Goldfarb, Y. Ren, A. Bahamou, Practical quasi-Newton methods for training deep neural networks, In: *Advances in Neural Information Processing Systems*, 2020.

29. H. J. Mao, S. Xie, Y. Xie, R. Byrd, J. Nocedal, A noise-tolerant quasi-Newton algorithm for unconstrained optimization, *SIAM J. Optim.*, **32** (2022), 29–55. https://doi.org/10.1137/20m1373190

30. X. Wang, S. Ma, D. Goldfarb, W. Liu, Stochastic quasi-Newton methods for nonconvex stochastic optimization, *SIAM J. Optim.*, **27** (2017), 927–956. https://doi.org/10.1137/15M1053141

31. H. Chen, H. C. Wu, S. C. Chan, W. H. Lam, A stochastic quasi-Newton method for large-scale nonconvex optimization with applications, *IEEE Trans. Neural Netw. Learn. Syst.*, **31** (2020), 4776–4790. https://doi.org/10.1109/TNNLS.2019.2957843

32. Q. Zhang, F. Huang, C. Deng, H. Huang, Faster stochastic quasi-Newton methods, *IEEE Trans. Neural Netw. Learn. Syst.*, **33** (2022), 4388–4397. https://doi.org/10.1109/TNNLS.2021.3056947

33. Y. Cheng, J. Gao, An efficient augmented memoryless quasi-Newton method for solving large-scale unconstrained optimization problems, *AIMS Math.*, **9** (2024), 25232–25252. https://doi.org./10.3934/math.20241231

34. D. Makmuang, S. Suppalap, R. Wangkeeree, The regularized stochastic Nesterov's accelerated quasi-Newton method with applications, *J. Comput. Appl. Math.*, **428** (2023), 115190. https://doi.org/10.1016/j.cam.2023.115190

35. C. J. Hillar, L. H. Lim, Most tensor problems are NP-hard, *J. ACM*, **60** (2013), 45. https://doi.org/10.1145/2512329

36. S. J. Reddi, A. Hefny, S. Sra, B. Póczos, A. Smola, Stochastic variance reduction for nonconvex optimization, In: *Proceedings of The 33rd International Conference on Machine Learning*, **48** (2016), 314–323.

37. C. G. Broyden, The convergence of a class of double-rank minimization algorithms, *IMA J. Appl. Math.*, **6** (1970), 76–90. https://doi.org/10.1093/imamat/6.1.76

38. R. Fletcher, A new approach to variable metric algorithms, *Comput. J.*, **13** (1970), 317–322. https://doi.org/10.1093/comjnl/13.3.317

39. D. Goldfarb, A family of variable-metric methods derived by variational means, *Math. Comp.*, **24** (1970), 23–26.

40. D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comp.*, **24** (1970), 647–656.

41. Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, New York, NY: Springer, 2013. http://doi.org/10.1007/978-1-4419-8853-9

42. Z. Allen-Zhu, Katyusha: The first direct acceleration of stochastic gradient methods, *J. Mach. Learn. Res.*, **18** (2018), 1–51.

43. S. J. Reddi, S. Sra, B. Póczos, A. Smola, Fast incremental method for smooth nonconvex optimization, In: *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, 1971–1977. https://doi.org/10.1109/CDC.2016.7798553



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)