

# Stable gradient-adjusted root mean square propagation on least squares problem\*

Runze Li<sup>†</sup>, Jintao Xu<sup>‡</sup>, and Wenxun Xing<sup>§</sup>

**Abstract.** Root mean square propagation (abbreviated as RMSProp) is a first-order stochastic algorithm used in machine learning widely. In this paper, a stable gradient-adjusted RMSProp (abbreviated as SGA-RMSProp) with mini-batch stochastic gradient is proposed, and its properties are studied on the linear least squares problem. R-linear convergence of the algorithm is established on the consistent linear least squares problem. The algorithm is also proved to converge R-linearly to a neighborhood of the minimizer for the inconsistent case, with the region of the neighborhood being controlled by the batch size. Furthermore, numerical experiments are conducted to compare the performances of SGA-RMSProp, stochastic gradient descent (abbreviated as SGD), and the original RMSProp with different batch sizes. The faster initial convergence rate of SGA-RMSProp is observed through numerical experiments and an adaptive strategy for switching from SGA-RMSProp to SGD is proposed, which combines the benefits of these two algorithms.

**Key words.** root mean square propagation, least squares, stochastic gradient, linear convergence

**MSC codes.** 90C06, 90C30, 68T09, 68W20

**1. Introduction.** To address the following finite-sum optimization problem

$$(1.1) \quad \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) := \sum_{i=1}^n f_i(\mathbf{x})$$

in machine learning, Tieleman and Hinton [34] developed a first-order stochastic optimization algorithm, called root mean square propagation (RMSProp). Its coordinate-wise version with time-varying hyperparameters is defined as follows:

$$(1.2) \quad u_{k,j} = \beta_k u_{k-1,j} + (1 - \beta_k) g_{k,j}^2,$$

$$(1.3) \quad x_{k+1,j} = x_{k,j} - \eta_k g_{k,j} / \sqrt{u_{k,j}}$$

for  $j = 1, \dots, d$ , where  $x_{k,j}$  is the  $j$ th component of the  $k$ th iteration point  $\mathbf{x}_k$  and  $g_{k,j}$  denotes the  $j$ th component of the stochastic gradient  $\mathbf{g}_k$ . Due to the large data size  $n$ , the stochastic gradient is derived by sampling from  $\{\nabla f_i(\mathbf{x}_k)\}_{i=1}^n$ , which will be discussed later. The  $j$ th component of the moving average  $\mathbf{u}_k$ , denoted as  $u_{k,j}$ , is used to adjust the current stochastic gradient, which can be expressed as a linear combination of  $u_{0,j}, g_{1,j}^2, \dots, g_{k,j}^2$  as in

\*Submitted to the editors DATE.

**Funding:** This work was funded in part by the National Natural Science Foundation of China Grant No. 11771243, PolyU postdoc matching fund scheme of the Hong Kong Polytechnic University Grant No. 1-W35A, and Huawei's Collaborative Grants "Large scale linear programming solver" and "Solving large scale linear programming models for production planning".

<sup>†</sup>Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China (lirz22@mails.tsinghua.edu.cn).

<sup>‡</sup> Corresponding author. Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China (xujtmath@163.com).

<sup>§</sup>Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China (wxing@tsinghua.edu.cn).

(1.2). The hyperparameters  $\beta_k$  and  $\eta_k$  are referred to as the discounting factor and the step size, respectively.

In this paper, we focus on the convergence rate of RMSProp (1.2)-(1.3) with the mini-batch stochastic gradient on the linear least squares problem (LLSP). It is well-known that LLSP has become one of the cornerstone problems in the optimization, machine learning, and statistics communities. In practice, it is widely used in computer vision [28], video signal processing [41], calibration [21], linear classification [6], and financial markets [18]. Specifically, let  $\mathbf{A}$  be an  $n \times d$  full column rank real matrix. Then, LLSP can be formulated as

$$(1.4) \quad \min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2,$$

where  $\mathbf{x} \in \mathbb{R}^d$  represents the parameters to be estimated in the model, while  $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)^\top$  and  $\mathbf{b} = (b_1, \dots, b_n)^\top$  are the observed data. Clearly, (1.4) can be viewed as a specific case of (1.1) with  $f_i(\mathbf{x}) = \frac{1}{2}(\mathbf{a}_i^\top \mathbf{x} - b_i)^2$ ,  $i = 1, \dots, n$ . Throughout this paper, the consistent (inconsistent) linear system represents  $\mathbf{Ax}^* = \mathbf{b}$  ( $\mathbf{Ax}^* \neq \mathbf{b}$ ) and the corresponding (1.4) is referred to as the consistent (inconsistent) LLSP, where  $\mathbf{x}^*$  is the minimizer of (1.4).

At the  $k$ th iteration, let  $\xi_1^{(k)}, \dots, \xi_B^{(k)}$  be a set of  $B$  independent and identically distributed random variables, in which  $\xi_i^{(k)}$  takes values in  $\{1, \dots, n\}$  with  $\mathbb{P}(\xi_i^{(k)} = j) = p_j > 0$ ,  $i = 1, \dots, B$ ,  $j = 1, \dots, n$ . Then the mini-batch stochastic gradient is given by

$$(1.5) \quad \mathbf{g}_k := \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \nabla f_{\xi_i^{(k)}}(\mathbf{x}_k) = \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \left( \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}_k - b_{\xi_i^{(k)}} \right).$$

In addition, aggregating (1.3) across all dimensions yields the vector form

$$(1.6) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{D}_k \mathbf{g}_k,$$

where  $\mathbf{D}_k := \text{diag}(\frac{1}{\sqrt{u_{k,1}}}, \dots, \frac{1}{\sqrt{u_{k,d}}})$ .

Equation (1.6) reduces to the iteration scheme of stochastic gradient descent (SGD) when  $\mathbf{D}_k \equiv \mathbf{I}$ . Strohmer and Vershynin [33] proposed the randomized Kaczmarz algorithm and proved that it has a Q-linear convergence rate on consistent linear systems, while Needell et al. [29] showed that studying the randomized Kaczmarz algorithm on linear systems can be viewed as analyzing a special case of SGD on the associated LLSP. Notice that the objective function of LLSP in (1.4) is strongly convex and its gradient is Lipschitz continuous. Moulines and Bach [27] proved that SGD converges to a neighborhood of the minimizer R-linearly for such objective functions. Needell et al. [29] reduced the constant in this convergence rate.

In the above studies on SGD,  $\eta_k$  is set to either a constant or a multiple of  $1/k^\ell$  for  $\ell \in [0, 1]$ . The Adagrad-norm algorithm studied by Ward et al. [37] uses a different step size and is more similar to RMSProp. Its iterative schemes are

$$(1.7) \quad v_k = v_{k-1} + \|\mathbf{g}_k\|_2^2,$$

$$(1.8) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\eta_k}{v_k} \mathbf{g}_k,$$

where  $v_k$  is a scalar. The distinction between (1.7)-(1.8) and RMSProp (1.2)-(1.3) is that the former still uses the stochastic gradient as the descent direction, that is,  $\mathbf{D}_k \equiv \mathbf{I}$ , while RMSProp modifies the descent direction. Xie et al. [39] proved that the Adagrad-norm algorithm achieves an R-linear convergence rate when the objective function is strongly convex or satisfies the Polyak-Łojasiewicz inequality and  $\nabla f_i(\mathbf{x}^*) = \mathbf{0}$ ,  $i = 1, \dots, n$ . Similar research includes [38].

Moreover, some first-order preconditioned stochastic methods also have the iterative scheme of (1.6). For instance, Liu et al. [23] discussed how a fixed preconditioner  $\mathbf{D}_k \equiv \mathbf{D}$  affects the performance of stochastic variance reduction gradient descent. The sequence  $\{f(\mathbf{x}_k)\}$  generated by their algorithm was proved to converge R-linearly when each  $f_i$  is strongly convex and has a Lipschitz continuous gradient. For the adaptive sketching-based preconditioners proposed by Lacotte and Pilanci [17], the  $\mathbf{g}_k$  and  $\mathbf{D}_k$  in (1.6) are chosen as  $\nabla f(\mathbf{x}_k)$  and the inverse of  $\mathbf{A}^\top \mathbf{L}^\top \mathbf{L} \mathbf{A}$  for LLSP (1.4) respectively, where  $\mathbf{L}$  is a random matrix with entries independently drawn from specific distributions. They proved that the algorithm converges R-linearly. Besides, linear convergence rates are also attained by some other first-order stochastic algorithms, such as proximal SGD [31, 10, 15], SGD with momentum [24, 5, 36, 40], randomized  $r$ -sets-Douglas-Rachford method [12], and randomized primal-dual gradient method [19], though their iterative schemes are different from (1.6).

RMSProp constructs  $\mathbf{D}_k$  from  $\mathbf{g}_k$  and  $\mathbf{D}_{k-1}$ , making  $\mathbf{D}_k$  a random and dynamically changing matrix that uses only first-order information. Liu et al. [22] proved that, assuming a non-convex objective function and a bounded second-order moment of the infinity norm of the stochastic gradient,  $\{\mathbf{x}_k\}$  generated by RMSProp satisfies  $\frac{1}{K} \sum_{k=1}^K (\mathbb{E}[\|\nabla f(\mathbf{x}_k)\|_2^{4/3}])^{3/2} = \mathcal{O}(\frac{\log K}{\sqrt{K}})$  with  $\eta_k \equiv \frac{1}{\sqrt{K}}$  and  $\beta_k = (1 - \frac{1}{k})^\ell$ , where  $\ell$  is a positive constant and  $K$  is the number of iterations. Under the assumptions that  $f$  is gradient Lipschitz continuous and the stochastic gradient has coordinate-wise bounded noise variance, Li and Lin [20] proved that  $\frac{1}{K} \sum_{k=1}^K \mathbb{E}[\|\nabla f(\mathbf{x}_k)\|_1] \leq \mathcal{O}(\frac{\sqrt{d}}{K^{1/4}})$  with  $\eta_k \equiv \frac{\ell}{\sqrt{dK}}$  and  $\beta_k \equiv 1 - \frac{1}{K}$ , in which  $\ell$  is a positive constant. Other studies have investigated the more general Adam-style algorithm, such as [16, 42, 11, 7], whose coordinate-wise version is defined as below:

$$\begin{aligned} m_{k,j} &= \alpha_k m_{k-1,j} + (1 - \alpha_k) g_{k,j}, \\ u_{k,j} &= \beta_k u_{k-1,j} + (1 - \beta_k) g_{k,j}^2, \\ x_{k+1,j} &= x_{k,j} - \eta_k m_{k,j} / \sqrt{u_{k,j}} \end{aligned}$$

for  $j = 1, \dots, d$ . Compared with RMSProp, the Adam-style algorithm uses an additional momentum  $m_{k,j}$  to improve the performance of the algorithm. Furthermore, setting  $\alpha_k = 0$  causes the Adam-style algorithm to degenerate into RMSProp. Therefore, in the research on the convergence rate of Adam-style algorithm, allowing  $\alpha_k = 0$  makes the results applicable to RMSProp. For instance, Zou et al. [42] and Chen et al. [7] studied the convergence rate of the Adam-style algorithm under different parameter settings. In their studies, the range of  $\alpha_k$  always includes 0, so the results can apply to RMSProp. An aspect of these studies is that they dealt with general functions, and thus the resulting convergence rates are sublinear.

In numerical experiments, we observed that RMSProp, when implemented on LLSP, exhibits a linear convergence rate empirically under specific parameter settings. We further

discovered that in numerous cases, RMSProp outperforms SGD in the early stages of iterations. Related observation of acceleration can also be noted in [25]. Since tuning the discounting factor  $\beta_k$  can be cumbersome and sensitive, we introduce a stable gradient-adjusted mechanism that selects  $\beta_k$  at each iteration. In this paper, we report the aforementioned experimental findings and explore whether RMSProp can attain a faster convergence rate on LLSP theoretically, with a focus on the linear convergence rate. It is worth mentioning that the convergence rate we obtain is better than sublinear, as we consider a specific LLSP.

The main contributions of this paper are summarized as follows:

- We introduce a parameter  $\varepsilon > 0$ , which is called adjusted level hereinafter, to stably control the adjustment applied to the stochastic gradient. Combining with a specific selection strategy of  $\beta_k$ , we guarantee  $\|\mathbf{D}_k^{-1}\mathbf{D}_{k-1} - \mathbf{I}\|_2 \leq \varepsilon$  during the iterations of RMSProp. So the algorithm is named stable gradient-adjusted RMSProp (SGA-RMSProp). Other properties arising from this stable gradient-adjusted mechanism are studied as well.
- We prove that SGA-RMSProp with the mini-batch stochastic gradient achieves an R-linear convergence rate on the consistent LLSP. For the inconsistent LLSP, we prove that SGA-RMSProp converges R-linearly to a neighborhood of the minimizer  $\mathbf{x}^*$ , where the region of the neighborhood is controlled by the batch size  $B$ . Moreover, we analytically discuss how the selection of the adjusted level  $\varepsilon$  affects the performance of SGA-RMSProp.
- We conduct numerical experiments to verify the R-linear convergence rate. In addition, the performances of SGA-RMSProp, SGD, and the original RMSProp (as defined by (1.2)-(1.3) with constant hyperparameters) on LLSP are compared numerically.
- We propose an adaptive condition for switching from SGA-RMSProp to SGD. Numerical experiments show that this method generally outperforms the vanilla SGD on LLSP in terms of computational time.

The remainder of this paper is organized as below. In section 2, notations and some results from linear algebra and probability theory are provided. In section 3, we formally introduce the framework of SGA-RMSProp and discuss its properties. In section 4, we present our main results regarding the R-linear convergence rate of SGA-RMSProp. We report our numerical experiment results in section 5. Finally, concluding remarks are presented in section 6.

**2. Preliminaries.** This section introduces the notations and some established results serving as foundational components in our proofs.

**2.1. Notations.** Let  $\mathbb{R}^n$ ,  $\mathbb{R}^{n \times d}$ , and  $\mathbb{N}$  denote the sets of real  $n$ -dimensional vectors,  $n \times d$  matrices, and natural numbers, respectively. The sets of real  $n \times n$  symmetric and positive semidefinite matrices are denoted by  $\mathbb{S}^n$  and  $\mathbb{S}_+^n$ , respectively. For a vector  $\mathbf{x} \in \mathbb{R}^d$ , let  $\|\mathbf{x}\|_2 := (\sum_{i=1}^d x_i^2)^{\frac{1}{2}}$ ,  $\frac{1}{\sqrt{\mathbf{x}}} := (\frac{1}{\sqrt{x_1}}, \dots, \frac{1}{\sqrt{x_d}})^\top$ , and  $\mathbf{x}^2 := (x_1^2, \dots, x_d^2)^\top$ . The Hadamard product is denoted by  $\circ$ . Given two matrices  $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n$ ,  $\mathbf{X} \preceq \mathbf{Y}$  represents  $\mathbf{Y} - \mathbf{X} \in \mathbb{S}_+^n$ . The  $j$ th largest eigenvalue and the  $j$ th largest singular value of  $\mathbf{X}$  are denoted by  $\lambda_j$  and  $s_j$  for  $j = 1, \dots, d$ , respectively. Let  $\|\mathbf{X}\|_F$  and  $\|\mathbf{X}\|_2$  denote the Frobenius norm and the spectral norm which is equal to the maximum singular value of  $\mathbf{X}$ , respectively. Let  $\mathbf{X} = \text{diag}(x_{11}, \dots, x_{dd})$  be a diagonal matrix whose diagonal entries are  $x_{11}, \dots, x_{dd}$ . The matrix  $\mathbf{X}^\ell$  denotes the diagonal

matrix  $\text{diag}(x_{11}^\ell, \dots, x_{dd}^\ell)$  for a real number  $\ell$ . The symbols  $\mathbf{0}$ ,  $\mathbf{O}$ , and  $\mathbf{I}$  denote the zero column vector, zero matrix, and the identity matrix whose sizes vary from the context, respectively. The triple  $(\Omega, \mathcal{A}, \mathbb{P})$  denotes a probability space, where  $\Omega$ ,  $\mathcal{A}$ , and  $\mathbb{P}$  denote the sample space,  $\sigma$ -field of subsets of  $\Omega$ , and probability measure, respectively. For a random variable  $\xi$  and a  $\sigma$ -field  $\mathcal{F} \subset \mathcal{A}$ ,  $\mathbb{E}[\cdot]$  and  $\mathbb{E}[\xi|\mathcal{F}]$  denote the expectation and the conditional expectation of  $\xi$  given  $\mathcal{F}$ , respectively. In the remainder of this paper,  $f(\mathbf{x})$  represents the object function  $\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$  in (1.4), while  $f_i(\mathbf{x})$  denotes  $\frac{1}{2}(\mathbf{a}_i^\top \mathbf{x} - b_i)^2$ ,  $i = 1, \dots, n$ .

**2.2. Linear algebra.** Next, two lemmas in linear algebra are presented.

**Lemma 2.1** (Weyl [14] Theorem 4.3.1). *Suppose that  $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^d$ . It holds that*

$$(2.1) \quad \max_{i+k=j+d} \{\lambda_i(\mathbf{X}) + \lambda_k(\mathbf{Y})\} \leq \lambda_j(\mathbf{X} + \mathbf{Y}) \leq \min_{i+k=j+1} \{\lambda_i(\mathbf{X}) + \lambda_k(\mathbf{Y})\}$$

for  $1 \leq j \leq d$ , where  $\lambda_j$  represents the  $j$ th largest eigenvalue of the matrix.

The above result describes how the eigenvalues of a symmetric matrix  $\mathbf{X}$  may change when a symmetric matrix  $\mathbf{Y}$  is added [14]. The next lemma is similar but deals with singular values.

**Lemma 2.2** (Ky Fan [13] Theorem 3.3.16). *Suppose that  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times d}$ . It holds that*

$$(2.2) \quad s_{i+j-1}(\mathbf{X} + \mathbf{Y}) \leq s_i(\mathbf{X}) + s_j(\mathbf{Y}), \quad s_{i+j-1}(\mathbf{X}\mathbf{Y}) \leq s_i(\mathbf{X})s_j(\mathbf{Y})$$

for  $1 \leq i, j \leq d, i+j-1 \leq d$ , where  $s_j$  represents the  $j$ th largest singular value of the matrix.

Here are some useful facts about eigenvalues and singular values of a matrix [14]. Let  $\mathbf{X} \in \mathbb{R}^{d \times d}$  be a real matrix,  $s_j(\mathbf{X}) = \sqrt{\lambda_j(\mathbf{X}^\top \mathbf{X})}$ ,  $j = 1, \dots, d$ ,  $\|\mathbf{X}\|_2 = s_1(\mathbf{X}) = \sqrt{\lambda_1(\mathbf{X}^\top \mathbf{X})}$ . If  $\mathbf{X}$  is positive semidefinite, then  $\lambda_j(\mathbf{X}) = s_j(\mathbf{X})$ ,  $j = 1, \dots, d$ . Moreover, if  $\mathbf{X} = \text{diag}(x_{11}, \dots, x_{dd})$  is a diagonal matrix, then  $\|\mathbf{X}\|_2 = s_1(\mathbf{X}) = \max_{j=1, \dots, d} |x_{jj}|$ .

**2.3. Probability theory.** Taking  $(\Omega, \mathcal{A}, \mathbb{P})$  as the probability space, we review some notions and results from probability theory.

**Definition 2.3** (Filtration [8]). *Let  $\mathcal{K} \subset \mathbb{N}$  be an index set. A filtration  $\{\mathcal{F}_k\}_{k \in \mathcal{K}}$  is an increasing sequence of  $\sigma$ -fields, that is,  $\mathcal{F}_{k_1} \subset \mathcal{F}_{k_2} \subset \mathcal{A}$  for all  $k_1, k_2 \in \mathcal{K}$  with  $k_1 \leq k_2$ .*

Consider  $\xi, \zeta$  as  $\mathcal{A}$ -measurable random variables with  $\mathbb{E}[|\xi|], \mathbb{E}[|\zeta|], \mathbb{E}[|\xi\zeta|] < +\infty$  and  $\mathcal{F} \subset \mathcal{A}$  as a  $\sigma$ -field. Since the  $\sigma$ -fields we consider in this paper have no null sets other than the empty set, we omit the “almost surely” in equalities like  $\xi = \mathbb{E}[\zeta|\mathcal{F}]$  throughout this paper. The following are three useful propositions.

**Proposition 2.4** ([8]). *If  $\xi$  is  $\mathcal{F}$ -measurable, then*

$$(2.3) \quad \mathbb{E}[\xi\zeta|\mathcal{F}] = \xi\mathbb{E}[\zeta|\mathcal{F}], \text{ and specifically, } \mathbb{E}[\xi|\mathcal{F}] = \xi.$$

**Proposition 2.5** ([8]). *If the  $\sigma$ -field generated by  $\xi$  is independent of  $\mathcal{F}$ , which is also referred to as  $\xi$  is independent of  $\mathcal{F}$ , then*

$$(2.4) \quad \mathbb{E}[\xi|\mathcal{F}] = \mathbb{E}[\xi].$$

**Proposition 2.6** ([8]). *If  $\mathcal{G}$  is a  $\sigma$ -field with  $\mathcal{G} \subset \mathcal{F}$ , then*

$$(2.5) \quad \mathbb{E}[\mathbb{E}[\xi|\mathcal{F}|\mathcal{G}] = \mathbb{E}[\xi|\mathcal{G}].$$

To estimate the expectation of the spectral norm of independent, zero mean random matrices sum, we introduce the matrix Bernstein inequality.

**Lemma 2.7 (Matrix Bernstein [35] Theorem 6.1.1).** *Let  $\{\mathbf{Z}_i\}_{i=1}^n$  be independent, random  $d_1 \times d_2$  matrices,  $\mathbb{E}[\mathbf{Z}_i] = \mathbf{O}$ ,  $\|\mathbf{Z}_i\|_2 \leq L$ ,  $\mathbf{Z} := \sum_{i=1}^n \mathbf{Z}_i$ , and*

$$(2.6) \quad \nu(\mathbf{Z}) = \max \left\{ \left\| \sum_{i=1}^n \mathbb{E}[\mathbf{Z}_i \mathbf{Z}_i^\top] \right\|_2, \left\| \sum_{i=1}^n \mathbb{E}[\mathbf{Z}_i^\top \mathbf{Z}_i] \right\|_2 \right\}$$

*be the matrix variance statistic of the sum. Then we have*

$$\mathbb{E}[\|\mathbf{Z}\|_2] \leq \sqrt{2\nu(\mathbf{Z}) \log(d_1 + d_2)} + \frac{L \log(d_1 + d_2)}{3}.$$

**3. SGA-RMSProp.** In this section, we first present the pseudocode of **SGA-RMSProp** and then discuss some properties of the algorithm.

---

**Algorithm 3.1** SGA-RMSProp

---

**Input:** Step size  $\{\eta_k\}$ , adjusted level  $\varepsilon$ , lower and upper bounds  $\underline{u}$ ,  $\bar{u}$ , initial value  $\mathbf{x}_1$ .

- 1:  $\mathbf{u}_0 = \left( \frac{1}{\underline{u}^2}, \dots, \frac{1}{\bar{u}^2} \right)^\top$ .
- 2: **for**  $k = 1, \dots, K$  **do**
- 3:   Sample a stochastic gradient  $\mathbf{g}_k$ .
- 4:    $\beta_k = \beta\text{-Selection}(\mathbf{g}_k, \mathbf{u}_{k-1}, \varepsilon, \underline{u}, \bar{u})$ .
- 5:    $\mathbf{u}_k = \beta_k \mathbf{u}_{k-1} + (1 - \beta_k) \mathbf{g}_k^2$ .
- 6:    $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\eta_k}{\sqrt{\mathbf{u}_k}} \circ \mathbf{g}_k$ .
- 7: **end for**

**Output:** The optimal point  $\mathbf{x}_{K+1}$ .

---

The algorithm above can be viewed as a vector form of RMSProp (1.2)-(1.3), in which the discounting factor  $\beta_k$  is selected during each iteration, as shown in **Line 4**. The details of  **$\beta$ -Selection** are presented below, and its validity will be illustrated through **Proposition 3.1**.

---

**Algorithm 3.2**  $\beta$ -Selection( $\mathbf{g}_k, \mathbf{u}_{k-1}, \varepsilon, \underline{u}, \bar{u}$ )

---

**Input:** Stochastic gradient  $\mathbf{g}_k$ , moving average  $\mathbf{u}_{k-1}$ , adjusted level  $\varepsilon$ , lower and upper bounds  $\underline{u}$ ,  $\bar{u}$ .

- 1: **if**  $\exists j \in \{1, \dots, d\}$  such that  $\frac{g_{k,j}^2}{u_{k-1,j}} \leq 1$  **then**
- 2:    $\beta_k = 1$ .
- 3: **else**
- 4:   Select  $\beta_k \in \left[ \max_{j=1, \dots, d} \left\{ \frac{g_{k,j}^2 - \frac{1}{\underline{u}^2}}{g_{k,j}^2 - u_{k-1,j}}, \frac{g_{k,j}^2 - (1+\varepsilon)^2 u_{k-1,j}}{g_{k,j}^2 - u_{k-1,j}}, 0 \right\}, 1 \right]$ .
- 5: **end if**

**Output:** The discounting factor  $\beta_k$ .

---

Recall from (1.5) that the mini-batch stochastic gradient in the  $k$ th iteration is given by

$$\mathbf{g}_k := \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \nabla f_{\xi_i^{(k)}}(\mathbf{x}_k) = \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \left( \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}_k - b_{\xi_i^{(k)}} \right).$$

Let  $\mathcal{S}_k$  denote the set  $\{\xi_1^{(k)}, \dots, \xi_B^{(k)}\}$ . It can be easily verified that the randomness of  $\mathbf{g}_k$  arises not only from  $\mathcal{S}_k$  but also from  $\mathbf{x}_k$  for each  $k = 2, \dots, K$ . Let  $\mathcal{F}_k$  be the  $\sigma$ -field generated by the random variables in  $\mathcal{S}_1, \dots, \mathcal{S}_{k-1}$ , and then  $\{\mathcal{F}_k\}_{k=2}^K$  is a filtration by Definition 2.3. Since  $\mathbf{x}_k$  is  $\mathcal{F}_k$ -measurable and  $\xi_i^{(k)} \in \mathcal{S}_k$  is independent of  $\mathcal{F}_k$ ,  $i = 1, \dots, B$ , we have

$$\begin{aligned} \mathbb{E} \left[ \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \left( \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}_k - b_{\xi_i^{(k)}} \right) \middle| \mathcal{F}_k \right] &\stackrel{(2.3)}{=} \mathbb{E} \left[ \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top \middle| \mathcal{F}_k \right] \mathbf{x}_k - \mathbb{E} \left[ \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} b_{\xi_i^{(k)}} \middle| \mathcal{F}_k \right] \\ (3.1) \quad &\stackrel{(2.4)}{=} \mathbb{E} \left[ \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top \right] \mathbf{x}_k - \mathbb{E} \left[ \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} b_{\xi_i^{(k)}} \right] = \sum_{i=1}^n \mathbf{a}_i (\mathbf{a}_i^\top \mathbf{x}_k - b_i) \end{aligned}$$

for each  $i = 1, \dots, B$ , and then  $\mathbb{E}[\mathbf{g}_k | \mathcal{F}_k] = \nabla f(\mathbf{x}_k)$ ,  $k = 2, \dots, K$ . Similarly, we have  $\mathbb{E}[\mathbf{g}_1] = \nabla f(\mathbf{x}_1)$ . So the stochastic gradient is unbiased.

Next, recalling that  $\mathbf{D}_k = \text{diag}(\frac{1}{\sqrt{u_{k,1}}}, \dots, \frac{1}{\sqrt{u_{k,d}}})$ ,  $k = 0, \dots, K$ , we present the following proposition to illustrate how **SGA-RMSProp** uses  $\varepsilon$  to modulate the variation of the adjustment made to the stochastic gradient, thereby achieving stable control.

**Proposition 3.1.** *Let  $\{\mathbf{u}_k\}_{k=0}^K$  be the sequence generated by **SGA-RMSProp**. Then  $u_{k,j}$  satisfies that  $\underline{u} \leq \frac{1}{\sqrt{u_{k,j}}} \leq \bar{u}$ ,  $k = 0, \dots, K$ ,  $j = 1, \dots, d$ . Equivalently, for each  $k = 0, \dots, K$ ,*

$$(3.2) \quad \left\| \mathbf{D}_k^\ell \right\|_2 \leq \bar{u}^\ell, \quad \left\| \mathbf{D}_k^{-\ell} \right\|_2 \leq \underline{u}^{-\ell},$$

where  $\ell$  is a positive number. Moreover, the matrix  $\mathbf{D}_k$  satisfies

$$(3.3) \quad \left\| \mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_{k-1}^{-\frac{1}{2}} \right\|_2 \leq 1 \text{ and } \left\| \mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \mathbf{I} \right\|_2 \leq \varepsilon, k = 1, \dots, K.$$

**Proof.** To prove the first part of this proposition, we employ induction on the variable  $k$ . When  $k = 0$ , recalling the initial values of  $u_{0,j}$  in **SGA-RMSProp**, we have  $\underline{u} \leq \frac{1}{\sqrt{u_{0,j}}} = \bar{u}$ ,  $j = 1, \dots, d$ . Next assume that  $\underline{u} \leq \frac{1}{\sqrt{u_{k,j}}} \leq \bar{u}$  for each  $k = 0, \dots, t$ ,  $j = 1, \dots, d$ . Now we show that  $\underline{u} \leq \frac{1}{\sqrt{u_{t+1,j}}} \leq \bar{u}$ ,  $j = 1, \dots, d$ . Under the induction hypothesis,  $\beta_{t+1} \in [0, 1]$ . If there exists  $j$  such that  $\frac{g_{t+1,j}^2}{u_{t,j}} \leq 1$ , then we have  $\mathbf{u}_{t+1} = \mathbf{u}_t$  since  $\beta_{t+1} = 1$  and  $\mathbf{u}_{t+1} = \beta_{t+1} \mathbf{u}_t + (1 - \beta_{t+1}) \mathbf{g}_{t+1}^2$ . In this case, we have  $\underline{u} \leq \frac{1}{\sqrt{u_{t+1,j}}} \leq \bar{u}$ . If  $\frac{g_{t+1,j}^2}{u_{t,j}} > 1$  for each  $j = 1, \dots, d$ , then we have

$$\beta_{t+1} \geq \max_{j=1, \dots, d} \left\{ \frac{g_{t+1,j}^2 - \frac{1}{\underline{u}^2}}{g_{t+1,j}^2 - u_{t,j}}, \frac{g_{t+1,j}^2 - (1 + \varepsilon)^2 u_{t,j}}{g_{t+1,j}^2 - u_{t,j}}, 0 \right\} \geq \frac{g_{t+1,j}^2 - \frac{1}{\underline{u}^2}}{g_{t+1,j}^2 - u_{t,j}}, j = 1, \dots, d,$$

which implies that  $u_{t+1,j} = \beta_{t+1} u_{t,j} + (1 - \beta_{t+1}) g_{t+1,j}^2 \leq \frac{1}{\underline{u}^2}$ ,  $j = 1, \dots, d$ . In addition, since

$$\frac{u_{t+1,j}}{u_{t,j}} = \frac{\beta_{t+1} u_{t,j} + (1 - \beta_{t+1}) g_{t+1,j}^2}{u_{t,j}} = \beta_{t+1} + (1 - \beta_{t+1}) \frac{g_{t+1,j}^2}{u_{t,j}} \geq 1, j = 1, \dots, d,$$



we have  $u_{t+1,j} \geq u_{t,j} \geq \frac{1}{\bar{u}^2}$ ,  $j = 1, \dots, d$ . Hence,  $\underline{u} \leq \frac{1}{\sqrt{u_{k,j}}} \leq \bar{u}$  for each  $j = 1, \dots, d$ ,  $k = t+1$ . Since  $\mathbf{D}_k$  is a diagonal matrix with all its diagonal elements positive, it is invertible and  $\|\mathbf{D}_k^\ell\|_2 \leq \bar{u}^\ell$ ,  $\|\mathbf{D}_k^{-\ell}\|_2 \leq \underline{u}^{-\ell}$ ,  $k = 0, \dots, K$ .

Next, we prove the second part of the lemma. The proof of the first part yields  $u_{k,j} \geq u_{k-1,j}$ ,  $j = 1, \dots, d$ ,  $k = 1, \dots, K$ . Combined with  $\|\mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_{k-1}^{-\frac{1}{2}}\|_2 = \max_{j=1,\dots,d} \{(\frac{u_{k-1,j}}{u_{k,j}})^{\frac{1}{4}}\}$ , we conclude that  $\|\mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_{k-1}^{-\frac{1}{2}}\|_2 \leq 1$ ,  $k = 1, \dots, K$ .

For  $\|\mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \mathbf{I}\|_2$ ,  $k = 1, \dots, K$ , we have

$$(3.4) \quad \|\mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \mathbf{I}\|_2 \leq \varepsilon \iff \left| \sqrt{\frac{u_{k,j}}{u_{k-1,j}}} - 1 \right| \leq \varepsilon \iff 1 - \varepsilon \leq \sqrt{\frac{u_{k,j}}{u_{k-1,j}}} \leq 1 + \varepsilon$$

for  $j = 1, \dots, d$ . Since  $\frac{u_{k,j}}{u_{k-1,j}} \geq 1$  for each  $k = 1, \dots, K$ ,  $j = 1, \dots, d$ , it suffices to consider solely the second inequality. By the following transformation,

$$\frac{u_{k,j}}{u_{k-1,j}} \leq (1 + \varepsilon)^2 \iff \left(1 - \frac{g_{k,j}^2}{u_{k-1,j}}\right) \beta_k + \frac{g_{k,j}^2}{u_{k-1,j}} \leq (1 + \varepsilon)^2, \iff \beta_k \geq \frac{g_{k,j}^2 - (1 + \varepsilon)^2 u_{k-1,j}}{g_{k,j}^2 - u_{k-1,j}}$$

for  $j = 1, \dots, d$ ,  $k = 1, \dots, K$ , and recalling the selection of  $\beta_k$ , we have

$$\beta_k \geq \max_{j=1,\dots,d} \left\{ \frac{g_{k,j}^2 - \frac{1}{\bar{u}^2}}{g_{k,j}^2 - u_{k-1,j}}, \frac{g_{k,j}^2 - (1 + \varepsilon)^2 u_{k-1,j}}{g_{k,j}^2 - u_{k-1,j}}, 0 \right\} \geq \frac{g_{k,j}^2 - (1 + \varepsilon)^2 u_{k-1,j}}{g_{k,j}^2 - u_{k-1,j}}, \quad k = 1, \dots, K.$$

The proof is completed. ■

This proposition shows two aspects of **SGA-RMSProp**. Firstly, our algorithm can control the range of all  $u_{k,j}$ . As a result, reasonable settings of  $\underline{u}$  and  $\bar{u}$  can avoid potential numerical issues caused by the division of  $\frac{1}{\sqrt{u_{k,j}}}$ . Secondly, the adjusted level  $\varepsilon$  stably controls the impact of squared gradients on the adjustment applied to the stochastic gradient by  $\|\mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \mathbf{I}\|_2 \leq \varepsilon$ . It provides us the ability to quantitatively manage the change of  $u_{k,j}$  at each iteration. A larger  $\varepsilon$  allows  $\mathbf{D}_k$  to adjust more, while a smaller  $\varepsilon$  has the opposite effect.

**4. Convergence analysis of SGA-RMSProp.** In this section, we first estimate the convergence rate of **SGA-RMSProp** on the consistent LLSP, then on the inconsistent LLSP.

**4.1. Consistent LLSP.** Denote the minimizer of (1.4) as  $\mathbf{x}^*$ , and then the consistency of LLSP implies  $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ , that is,  $\mathbf{a}_i^\top \mathbf{x}^* = b_i$ ,  $i = 1, \dots, n$ . Replacing  $b_i$  with  $\mathbf{a}_i^\top \mathbf{x}^*$  in (1.5), by (1.6), we have

$$\begin{aligned} \mathbf{x}_{k+1} - \mathbf{x}^* &= \mathbf{x}_k - \mathbf{x}^* - \frac{\eta_k}{B} \mathbf{D}_k \left( \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top (\mathbf{x}_k - \mathbf{x}^*) \right) \\ &= \mathbf{x}_k - \mathbf{x}^* - \eta_k \mathbf{D}_k \mathbf{M}_k (\mathbf{x}_k - \mathbf{x}^*) \\ &= (\mathbf{I} - \eta_k \mathbf{D}_k \mathbf{M}_k) (\mathbf{x}_k - \mathbf{x}^*), \end{aligned}$$

where  $\mathbf{M}_k := \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top$  and  $\mathbf{D}_k = \text{diag}(\frac{1}{\sqrt{u_{k,1}}}, \dots, \frac{1}{\sqrt{u_{k,d}}})$ .



Denote  $\mathbf{Y}_k := \mathbf{I} - \eta_k \mathbf{D}_k \mathbf{M}_k$  as the stochastic transition matrix in the  $k$ th iteration. We have  $\mathbf{x}_{K+1} - \mathbf{x}^* = \mathbf{Y}_K (\mathbf{x}_K - \mathbf{x}^*) = \dots = \mathbf{Y}_K \cdots \mathbf{Y}_1 (\mathbf{x}_1 - \mathbf{x}^*)$ , which implies that

$$(4.1) \quad \|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2 \leq \|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2 \|\mathbf{x}_1 - \mathbf{x}^*\|_2.$$

Therefore, establishing the convergence rate of **SGA-RMSProp** hinges upon the estimation of  $\mathbb{E}[\|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2]$ . To achieve this, we present the following two lemmas. The first one shows the relationship between  $p_j$  and  $\mathbf{a}_j$ ,  $j = 1, \dots, n$ .

**Lemma 4.1.** *Let  $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)^\top \in \mathbb{R}^{n \times d}$  be a non-zero matrix and  $\{p_j\}_{j=1}^n$  be the non-zero probabilities, that is,  $p_j > 0$ ,  $j = 1, \dots, n$ , and  $\sum_{j=1}^n p_j = 1$ . Then we have*

$$\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \geq \|\mathbf{A}\|_F^2.$$

*Proof.* Assume that  $\frac{\|\mathbf{a}_j\|_2^2}{p_j} < \|\mathbf{A}\|_F^2$ ,  $j = 1, \dots, n$ . It follows that  $\sum_{j=1}^n p_j > \sum_{j=1}^n \frac{\|\mathbf{a}_j\|_2^2}{\|\mathbf{A}\|_F^2} = 1$ , a contradiction. Therefore, we have  $\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \geq \|\mathbf{A}\|_F^2$ .  $\blacksquare$

Next, we give an upper bound on  $\mathbb{E}[\|\mathbf{M}_k - \mathbb{E}[\mathbf{M}_k]\|_2]$ , which is a monotonically decreasing function of the batch size, similar to Lemma 2 in [5].

**Lemma 4.2.** *Let  $\mathbf{M}_k = \frac{1}{B} \sum_{i=1}^B \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top$ , where  $\xi_i^{(k)}$ ,  $i = 1, \dots, B$ ,  $k = 1, \dots, K$ , are independent and identically distributed random variables, taking values in  $\{1, \dots, n\}$  with  $\mathbb{P}(\xi_i^{(k)} = j) = p_j$ ,  $j = 1, \dots, n$ . Then, for each  $k = 1, \dots, K$ , we have*

$$(4.2) \quad \begin{aligned} \mathbb{E}[\|\mathbf{M}_k - \mathbb{E}[\mathbf{M}_k]\|_2] &\leq \left( \frac{2 \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \|\mathbf{A}\|_2^2 \log(2d)}{B} \right)^{\frac{1}{2}} \\ &\quad + \frac{\log(2d)}{3B} \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right). \end{aligned}$$

*Proof.* Since  $\xi_i^{(k)}$ ,  $k = 1, \dots, K$ , are independent and identically distributed random variables, our proof below holds for each  $k = 1, \dots, K$ . Similar to (3.1), we have  $\mathbb{E}[\mathbf{M}_k] = \mathbf{A}^\top \mathbf{A}$ . Denoting  $\mathbf{Z}_{\xi_i^{(k)}} := \frac{1}{B} \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top - \mathbf{A}^\top \mathbf{A} \right)$  and  $\mathbf{Z}_k := \sum_{i=1}^B \mathbf{Z}_{\xi_i^{(k)}}$ , it is clear that  $\mathbb{E}[\|\mathbf{M}_k - \mathbb{E}[\mathbf{M}_k]\|_2] = \mathbb{E}(\|\mathbf{Z}_k\|_2)$  and  $\mathbb{E}[\mathbf{Z}_{\xi_i^{(k)}}] = \mathbf{O}$ . Since  $\mathbf{Z}_{\xi_i^{(k)}}$  is a real symmetric matrix,  $\|\mathbf{Z}_{\xi_i^{(k)}}\|_2 = s_1(\mathbf{Z}_{\xi_i^{(k)}}) = \max\{\lambda_1(\mathbf{Z}_{\xi_i^{(k)}}), -\lambda_d(\mathbf{Z}_{\xi_i^{(k)}})\}$ ,  $i = 1, \dots, B$ . By Lemma 2.1, we have

$$\lambda_d(\mathbf{Z}_{\xi_i^{(k)}}) = \lambda_d \left( \frac{1}{B} \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top - \mathbf{A}^\top \mathbf{A} \right) \right) \stackrel{(2.1)}{\geq} \frac{1}{B} \left( \lambda_d \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top \right) - \lambda_1(\mathbf{A}^\top \mathbf{A}) \right) \geq -\frac{\lambda_1(\mathbf{A}^\top \mathbf{A})}{B},$$

and

$$\lambda_1(\mathbf{Z}_{\xi_i^{(k)}}) \stackrel{(2.1)}{\leq} \frac{1}{B} \left( \lambda_1 \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top \right) - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \leq \frac{\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A})}{B}.$$

Therefore,  $\|\mathbf{Z}_{\xi_i^{(k)}}\|_2 \leq \max\left\{\frac{\lambda_1(\mathbf{A}^\top \mathbf{A})}{B}, \frac{\max_{j=1,\dots,n}\{\|\mathbf{a}_j\|_2^2/p_j\} - \lambda_d(\mathbf{A}^\top \mathbf{A})}{B}\right\}$ . By the definition of Frobenius norm, we have  $\|\mathbf{A}\|_F^2 = \sum_{i=1}^d s_i^2(\mathbf{A}) = \sum_{i=1}^d \lambda_i(\mathbf{A}^\top \mathbf{A})$ , which implies  $\|\mathbf{A}\|_F^2 \geq \lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})$ . By Lemma 4.1, it follows that  $\max_{j=1,\dots,n}\{\|\mathbf{a}_j\|_2^2/p_j\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \geq \lambda_1(\mathbf{A}^\top \mathbf{A})$  and

$$\|\mathbf{Z}_{\xi_i^{(k)}}\|_2 \leq \frac{\max_{j=1,\dots,n}\left\{\frac{\|\mathbf{a}_j\|_2^2}{p_j}\right\} - \lambda_d(\mathbf{A}^\top \mathbf{A})}{B}, \quad i = 1, \dots, B.$$

The next step is to estimate the matrix variance statistic (2.6) of  $\mathbf{Z}_k$ . Firstly, we have

$$\mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}} = \frac{1}{B^2} \left( \frac{\|\mathbf{a}_{\xi_i^{(k)}}\|_2^2}{p_{\xi_i^{(k)}}^2} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top - \frac{1}{p_{\xi_i^{(k)}}} \mathbf{A}^\top \mathbf{A} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top - \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{A}^\top \mathbf{A} + (\mathbf{A}^\top \mathbf{A})^2 \right)$$

for each  $i = 1, \dots, B$ . Then, the expectation of  $\mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}}$  can be estimated as

$$\mathbb{E} \left[ \mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}} \right] = \frac{1}{B^2} \left( \sum_{i=1}^n \frac{\|\mathbf{a}_i\|_2^2}{p_i} \mathbf{a}_i \mathbf{a}_i^\top - (\mathbf{A}^\top \mathbf{A})^2 \right) \preceq \frac{1}{B^2} \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \mathbf{A}^\top \mathbf{A} - (\mathbf{A}^\top \mathbf{A})^2 \right),$$

where we use  $\mathbf{A}^\top \mathbf{A} = \sum_{i=1}^n \mathbf{a}_i \mathbf{a}_i^\top$  in the first equality. Notice that  $\mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}}$  is positive semidefinite. The above implies that for each  $i = 1, \dots, B$ ,

$$\begin{aligned} \left\| \mathbb{E} \left[ \mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}} \right] \right\|_2 &\leq \frac{\left\| \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \mathbf{A}^\top \mathbf{A} - (\mathbf{A}^\top \mathbf{A})^2 \right\|_2}{B^2} \\ &\leq \frac{\|\mathbf{A}^\top \mathbf{A}\|_2 \left\| \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \mathbf{I} - \mathbf{A}^\top \mathbf{A} \right\|_2}{B^2} = \frac{\|\mathbf{A}\|_2^2 \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right)}{B^2}, \end{aligned}$$

where the last equality is derived from Lemma 4.1, which implies that  $\max_{j=1,\dots,n} \{\|\mathbf{a}_j\|_2^2/p_j\} \mathbf{I} - \mathbf{A}^\top \mathbf{A}$  is positive semidefinite. As a result, the matrix variance statistic of  $\mathbf{Z}_k$  satisfies

$$\nu(\mathbf{Z}_k) = \left\| \sum_{i=1}^B \mathbb{E} \left[ \mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}} \right] \right\|_2 \leq \sum_{i=1}^B \left\| \mathbb{E} \left[ \mathbf{Z}_{\xi_i^{(k)}}^\top \mathbf{Z}_{\xi_i^{(k)}} \right] \right\|_2 \leq \frac{\|\mathbf{A}\|_2^2 \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right)}{B}.$$

Combined with the independence of the matrices  $\{\mathbf{Z}_{\xi_i^{(k)}}\}_{i=1}^B$  and  $\mathbb{E}[\mathbf{Z}_{\xi_i^{(k)}}] = \mathbf{O}$ , applying Lemma 2.7, we obtain (4.2). ■

The next theorem shows an R-linear convergence rate of **SGA-RMSP** on the consistent LLSP, which is our main result.

**Theorem 4.3.** *Let  $\mathbf{x}^*$  be the optimal point of the consistent LLSP (1.4), and  $\{\mathbf{x}_k\}_{k=1}^K$  be the sequence generated by **SGA-RMSP** with  $\eta_k \equiv \eta := \frac{2}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ . Then we have*

$$(4.3) \quad \mathbb{E} [\|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2] \leq \rho \|\mathbf{x}_1 - \mathbf{x}^*\|_2 \gamma^K,$$

where  $\rho \leq 1$  and

$$(4.4) \quad \gamma := G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon \right),$$

in which  $\gamma_k \leq 1$ ,  $k = 1, \dots, K$ ,  $G(\gamma_1, \dots, \gamma_K) := \left( \prod_{k=1}^K \gamma_k \right)^{\frac{1}{K}}$ , and

$$(4.5) \quad \sigma := \left( \frac{2 \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \|\mathbf{A}\|_2^2 \log(2d)}{B} \right)^{\frac{1}{2}} + \frac{\log(2d)}{3B} \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right).$$

*Proof.* Revisiting (4.1), noticing that  $\mathbf{D}_k$  is a diagonal matrix for  $k = 0, \dots, K$ , which makes their multiplication commuted, we have

$$(4.6) \quad \begin{aligned} & \|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2 \\ &= \|(\mathbf{I} - \eta \mathbf{D}_K \mathbf{M}_K)(\mathbf{I} - \eta \mathbf{D}_{K-1} \mathbf{M}_{K-1}) \cdots (\mathbf{I} - \eta \mathbf{D}_1 \mathbf{M}_1)\|_2 \\ &= \|\mathbf{D}_K (\mathbf{D}_K^{-1} \mathbf{D}_{K-1} - \eta \mathbf{M}_K \mathbf{D}_{K-1}) \mathbf{D}_{K-1}^{-1} \mathbf{D}_{K-1} (\mathbf{D}_{K-1}^{-1} \mathbf{D}_{K-2} - \eta \mathbf{M}_{K-1} \mathbf{D}_{K-2}) \\ & \quad \cdots (\mathbf{D}_1^{-1} \mathbf{D}_0 - \eta \mathbf{M}_1 \mathbf{D}_0) \mathbf{D}_0^{-1}\|_2 \\ &= \|\mathbf{D}_K \mathbf{D}_{K-1}^{-\frac{1}{2}} (\mathbf{D}_K^{-1} \mathbf{D}_{K-1} - \eta \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{M}_K \mathbf{D}_{K-1}^{\frac{1}{2}}) \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{D}_{K-2}^{-\frac{1}{2}} \\ & \quad (\mathbf{D}_{K-1}^{-1} \mathbf{D}_{K-2} - \eta \mathbf{D}_{K-2}^{\frac{1}{2}} \mathbf{M}_{K-1} \mathbf{D}_{K-2}^{\frac{1}{2}}) \mathbf{D}_{K-2}^{\frac{1}{2}} \cdots \mathbf{D}_0^{-\frac{1}{2}} (\mathbf{D}_1^{-1} \mathbf{D}_0 - \eta \mathbf{D}_0^{\frac{1}{2}} \mathbf{M}_1 \mathbf{D}_0^{\frac{1}{2}}) \mathbf{D}_0^{-\frac{1}{2}}\|_2 \\ &\leq \|\mathbf{D}_K^{\frac{1}{2}}\|_2 \|\mathbf{D}_K^{\frac{1}{2}} \mathbf{D}_{K-1}^{-\frac{1}{2}}\|_2 \|\mathbf{D}_K^{-1} \mathbf{D}_{K-1} - \eta \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{M}_K \mathbf{D}_{K-1}^{\frac{1}{2}}\|_2 \|\mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{D}_{K-2}^{-\frac{1}{2}}\|_2 \\ & \quad \|\mathbf{D}_{K-1}^{-1} \mathbf{D}_{K-2} - \eta \mathbf{D}_{K-2}^{\frac{1}{2}} \mathbf{M}_{K-1} \mathbf{D}_{K-2}^{\frac{1}{2}}\|_2 \|\mathbf{D}_{K-2}^{\frac{1}{2}} \mathbf{D}_{K-3}^{-\frac{1}{2}}\|_2 \cdots \|\mathbf{D}_1^{\frac{1}{2}} \mathbf{D}_0^{-\frac{1}{2}}\|_2 \|\mathbf{D}_1^{-1} \mathbf{D}_0 - \eta \mathbf{D}_0^{\frac{1}{2}} \mathbf{M}_1 \mathbf{D}_0^{\frac{1}{2}}\|_2 \|\mathbf{D}_0^{-\frac{1}{2}}\|_2. \end{aligned}$$

Denote  $\mathbf{X}_k := \mathbf{I} - \eta \mathbf{D}_{k-1}^{\frac{1}{2}} \mathbf{M}_k \mathbf{D}_{k-1}^{\frac{1}{2}}$ ,  $k = 1, \dots, K$ . Proposition 3.1 implies that

$$(4.7) \quad \left\| \mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \eta \mathbf{D}_{k-1}^{\frac{1}{2}} \mathbf{M}_k \mathbf{D}_{k-1}^{\frac{1}{2}} \right\|_2 \leq \left\| \mathbf{I} - \eta \mathbf{D}_{k-1}^{\frac{1}{2}} \mathbf{M}_k \mathbf{D}_{k-1}^{\frac{1}{2}} \right\|_2 + \left\| \mathbf{D}_k^{-1} \mathbf{D}_{k-1} - \mathbf{I} \right\|_2 \stackrel{(3.3)}{\leq} \|\mathbf{X}_k\|_2 + \varepsilon.$$

According to the setting  $\|\mathbf{D}_0^{-\frac{1}{2}}\|_2 = 1/\sqrt{u}$  and Proposition 3.1, we have that  $\|\mathbf{D}_K^{\frac{1}{2}}\|_2 \|\mathbf{D}_0^{-\frac{1}{2}}\|_2$  and  $\|\mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_{k-1}^{-\frac{1}{2}}\|_2$ ,  $k = 1, \dots, K$ , are bounded above by some constants at most 1. These constants, denoted by  $\rho$ ,  $\gamma_k$ ,  $k = 1, \dots, K$  respectively, only depend on the hyperparameters of SGA-RMSProp, the initial point  $\mathbf{x}_1$ , and  $\mathbf{A}$ ,  $\mathbf{b}$  in LLSP. Recalling that  $\mathcal{F}_k$  is the  $\sigma$ -field

generated by the random variables in  $\mathcal{S}_1, \dots, \mathcal{S}_{k-1}$ ,  $k = 2, \dots, K$ , taking the expectation for both sides of the inequality (4.6) and by Proposition 2.6, we have

$$(4.8) \quad \begin{aligned} \mathbb{E} [\|Y_K \cdots Y_1\|_2] &\stackrel{(4.6)}{\leq} \rho \left( \prod_{k=1}^K \gamma_k \right) \mathbb{E} \left[ \prod_{k=1}^K \left\| D_k^{-1} D_{k-1} - \eta D_{k-1}^{\frac{1}{2}} M_k D_{k-1}^{\frac{1}{2}} \right\|_2 \right] \\ &\stackrel{(4.7)}{\leq} \rho \left( \prod_{k=1}^K \gamma_k \right) \mathbb{E} \left[ \prod_{k=1}^K (\|X_k\|_2 + \varepsilon) \right] \stackrel{(2.5)}{=} \rho \left( \prod_{k=1}^K \gamma_k \right) \mathbb{E} \left[ \cdots \mathbb{E} \left[ \mathbb{E} \left[ \prod_{k=1}^K (\|X_k\|_2 + \varepsilon) \middle| \mathcal{F}_K \right] \middle| \mathcal{F}_{K-1} \right] \cdots \right]. \end{aligned}$$

By  $D_{k-1} = \text{diag}(\frac{1}{\sqrt{u_{k-1,1}}}, \dots, \frac{1}{\sqrt{u_{k-1,d}}})$  and (1.2),  $D_{k-1}$  is  $\mathcal{F}_K$ -measurable for  $k = 1, \dots, K$ . Moreover, the matrix  $M_k$  depends solely on  $\mathcal{S}_k$  and  $X_k = I - \eta D_{k-1}^{\frac{1}{2}} M_k D_{k-1}^{\frac{1}{2}}$ , implying that  $X_k$  is  $\mathcal{F}_K$ -measurable for  $k = 1, \dots, K-1$ . Therefore, we have

$$(4.9) \quad \mathbb{E} \left[ \prod_{k=1}^K (\|X_k\|_2 + \varepsilon) \middle| \mathcal{F}_K \right] \stackrel{(2.3)}{=} \mathbb{E} [\|X_K\|_2 + \varepsilon | \mathcal{F}_K] \prod_{k=1}^{K-1} (\|X_k\|_2 + \varepsilon).$$

Using triangle inequality,  $\mathbb{E} [\|X_K\|_2 + \varepsilon | \mathcal{F}_K]$  has the following upper bound estimation

$$(4.10) \quad \begin{aligned} \mathbb{E} [\|X_K\|_2 + \varepsilon | \mathcal{F}_K] &= \mathbb{E} [\|X_K - \mathbb{E}[X_K | \mathcal{F}_K] + \mathbb{E}[X_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] + \varepsilon \\ &\leq \mathbb{E} [\|X_K - \mathbb{E}[X_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] + \mathbb{E} [\|\mathbb{E}[X_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] + \varepsilon \\ &= \mathbb{E} [\|X_K - \mathbb{E}[X_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] + \mathbb{E} [\|\mathbb{E}[X_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] + \varepsilon. \end{aligned}$$

As  $D_{K-1}$  is  $\mathcal{F}_K$ -measurable and  $M_K$  is independent of  $\mathcal{F}_K$ , we have

$$(4.11) \quad \begin{aligned} \|\mathbb{E}[X_K | \mathcal{F}_K]\|_2 &= \left\| \mathbb{E} \left[ I - \eta D_{K-1}^{\frac{1}{2}} M_K D_{K-1}^{\frac{1}{2}} \middle| \mathcal{F}_K \right] \right\|_2 \\ &\stackrel{(2.3), (2.4)}{=} \left\| I - \eta D_{K-1}^{\frac{1}{2}} \mathbb{E}[M_K] D_{K-1}^{\frac{1}{2}} \right\|_2 = \left\| I - \eta D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right\|_2. \end{aligned}$$

Since  $I - \eta D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}}$  is a symmetric matrix, the spectral norm can be calculated as

$$(4.12) \quad \begin{aligned} \left\| I - \eta D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right\|_2 &= \max \left\{ \lambda_1 \left( I - \eta D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right), -\lambda_d \left( I - \eta D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) \right\} \\ &= \max \left\{ 1 - \eta \lambda_d \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right), \eta \lambda_1 \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) - 1 \right\}. \end{aligned}$$

Note that when a matrix is positive semidefinite, its singular values are the same as its eigenvalues. By Lemma 2.2 and Proposition 3.1, we have

$$(4.13) \quad \lambda_1 \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) = s_1 \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) \stackrel{(2.2)}{\leq} s_1 \left( D_{K-1}^{\frac{1}{2}} \right) s_1 \left( A^\top A \right) s_1 \left( D_{K-1}^{\frac{1}{2}} \right) \stackrel{(3.2)}{\leq} \bar{u} \lambda_1 \left( A^\top A \right),$$

and

$$(4.14) \quad \lambda_d \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) = s_d \left( D_{K-1}^{\frac{1}{2}} A^\top A D_{K-1}^{\frac{1}{2}} \right) \stackrel{(2.2)}{\leq} s_1 \left( D_{K-1}^{\frac{1}{2}} \right) s_d \left( A^\top A \right) s_1 \left( D_{K-1}^{\frac{1}{2}} \right) \stackrel{(3.2)}{\leq} \bar{u} \lambda_d \left( A^\top A \right).$$

Combined with our step size setting  $\eta = \frac{2}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ , we have

$$\begin{aligned}
 1 - \eta \lambda_d \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right) &= \frac{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})) - 2\lambda_d \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \\
 &\stackrel{(4.14)}{\geq} \frac{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})) - 2\bar{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \\
 &= \frac{2\bar{u}\lambda_1(\mathbf{A}^\top \mathbf{A}) - \bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \\
 &\stackrel{(4.13)}{\geq} \frac{2\lambda_1 \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right) - \bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \\
 &= \eta \lambda_1 \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right) - 1.
 \end{aligned}$$

Thus, (4.12) implies that

$$(4.15) \quad \left\| \mathbf{I} - \eta \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right\|_2 = 1 - \eta \lambda_d \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right).$$

Lemma 2.2 and Proposition 3.1 also imply that

$$\begin{aligned}
 \lambda_d(\mathbf{A}^\top \mathbf{A}) &= \lambda_d \left( \mathbf{D}_{K-1}^{-\frac{1}{2}} \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{D}_{K-1}^{-\frac{1}{2}} \right) \\
 &\stackrel{(2.2)}{\leq} s_1 \left( \mathbf{D}_{K-1}^{-\frac{1}{2}} \right) s_d \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right) s_1 \left( \mathbf{D}_{K-1}^{-\frac{1}{2}} \right) \stackrel{(3.2)}{\leq} \frac{\lambda_d \left( \mathbf{D}_{K-1}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{D}_{K-1}^{\frac{1}{2}} \right)}{\underline{u}}.
 \end{aligned}
 \quad (4.16)$$

Combining (4.11), (4.15) with (4.16), we have

$$(4.17) \quad \|\mathbb{E}[\mathbf{X}_K | \mathcal{F}_K]\|_2 \leq 1 - \eta \underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A}) = 1 - \frac{2\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}.$$

Moreover, given  $\sigma$  as in (4.5), by Proposition 3.1 and Lemma 4.2, we have

$$\begin{aligned}
 &\mathbb{E}[\|\mathbf{X}_K - \mathbb{E}[\mathbf{X}_K | \mathcal{F}_K]\|_2 | \mathcal{F}_K] \\
 &= \eta \mathbb{E} \left[ \left\| \mathbf{D}_{K-1}^{\frac{1}{2}} \left( -\mathbf{M}_K + \mathbf{A}^\top \mathbf{A} \right) \mathbf{D}_{K-1}^{\frac{1}{2}} \right\|_2 \middle| \mathcal{F}_K \right] \\
 &\stackrel{(3.2)}{\leq} \eta \bar{u} \mathbb{E} \left[ \left\| \left( -\mathbf{M}_K + \mathbf{A}^\top \mathbf{A} \right) \right\|_2 \middle| \mathcal{F}_K \right] \\
 &\stackrel{(2.4)}{=} \eta \bar{u} \mathbb{E} \left[ \left\| \sum_{i=1}^B \frac{1}{B} \left( -\frac{1}{p_{\xi_i(K)}} \mathbf{a}_{\xi_i(K)} \mathbf{a}_{\xi_i(K)}^\top + \mathbf{A}^\top \mathbf{A} \right) \right\|_2 \right] \\
 &\stackrel{(4.2)}{\leq} \eta \bar{u} \left( \left( \frac{2 \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \|\mathbf{A}\|_2^2 \log(2d)}{B} \right)^{\frac{1}{2}} + \frac{\log(2d) \left( \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right)}{3B} \right)
 \end{aligned}$$

$$(4.18) \stackrel{(4.5)}{=} \frac{2\sigma}{\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})}.$$

Substituting (4.18) and (4.17) into (4.10), the estimation of  $\mathbb{E}[\prod_{k=1}^K (\|\mathbf{X}_k\|_2 + \varepsilon) | \mathcal{F}_K]$  by (4.9) becomes

$$(4.19) \quad \mathbb{E} \left[ \prod_{k=1}^K (\|\mathbf{X}_k\|_2 + \varepsilon) \middle| \mathcal{F}_K \right] \leq \left( 1 - \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon \right) \prod_{k=1}^{K-1} (\|\mathbf{X}_k\|_2 + \varepsilon).$$

Notice that  $1 - \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon$  has already been a constant, so we can extend the same procedure to  $k = K-1$ , and subsequently to others. Finally, substituting the estimation of  $\mathbb{E}[\prod_{k=1}^K (\|\mathbf{X}_k\|_2 + \varepsilon)]$  into (4.8) yields

$$(4.20) \quad \mathbb{E}[\|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2] \leq \rho \gamma^K,$$

where  $\gamma = G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon \right)$  and  $G(\gamma_1, \dots, \gamma_K) = \left( \prod_{k=1}^K \gamma_k \right)^{\frac{1}{K}}$ . Combined with (4.1), we obtain (4.3)-(4.4).  $\blacksquare$

It is worth noting that (4.3)-(4.4) implies a theoretical range of  $\varepsilon$  to maintain an R-linear convergence rate of **SGA-RMSProp** on the consistent LLSP, that is,

$$0 < \varepsilon < \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}.$$

Due to the impact of multiplier  $G(\gamma_1, \dots, \gamma_K) \leq 1$ , this range may become larger. Numerical experiments on the selection of  $\varepsilon$  are presented in [subsection 5.1.2](#). Moreover,  $\sigma$  can be viewed as an upper bound on the matrix analogue of the standard deviation of  $\mathbf{M}_k$ ,  $k = 1, \dots, K$ . According to (4.5), it monotonically decreases to 0 when the batch size  $B$  increases, making the upper bound of  $\varepsilon$  positive for large  $B$ . As a result, it is straightforward to determine the values of  $B$  and  $\varepsilon$  that yield the R-linear convergence rate from (4.3)-(4.4). This is shown in the corollary below, which is also a special case of [Theorem 4.3](#).

**Corollary 4.4.** *Setting  $B \geq \frac{4\bar{u}^2 \log(2d) \max_{j=1, \dots, n} \{\|\mathbf{a}_j\|_2^2/p_j\}}{\underline{u}^2} \left( \frac{2\sqrt{2}\|\mathbf{A}\|_2}{\lambda_d(\mathbf{A}^\top \mathbf{A})} + \left( \frac{\underline{u}}{3\bar{u}\lambda_d(\mathbf{A}^\top \mathbf{A})} \right)^{\frac{1}{2}} \right)^2$  ensures that  $\sigma \leq \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{4\bar{u}}$ . Combined with setting  $0 < \varepsilon \leq \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{2\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ , we have*

$$(4.21) \quad \mathbb{E}[\|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2] \leq \rho \|\mathbf{x}_1 - \mathbf{x}^*\|_2 \left( G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right) \right)^K,$$

in which  $G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right)$  is a positive constant smaller than 1.

**Proof.** Firstly, by (4.4), setting  $0 < \sigma \leq \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{4\bar{u}}$  and  $0 < \varepsilon \leq \frac{\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A})}{2\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$  gives (4.21) directly. Next, let  $c_1 := (2(\max_{j=1, \dots, n} \{\|\mathbf{a}_j\|_2^2/p_j\} - \lambda_d(\mathbf{A}^\top \mathbf{A})))\|\mathbf{A}\|_2^2 \log(2d))^{\frac{1}{2}}$ ,  $c_2 :=$

$\frac{\log(2d)}{3} \left( \max_{j=1,\dots,n} \{ \|\mathbf{a}_j\|_2^2 / p_j \} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right)$ . Then (4.5) implies that  $-\sigma(\sqrt{B})^2 + c_1\sqrt{B} + c_2 = 0$ , so  $B = \left( \frac{c_1 + \sqrt{c_1^2 + 4\sigma c_2}}{2\sigma} \right)^2$ . In order to obtain  $\sigma \leq \frac{u\lambda_d(\mathbf{A}^\top \mathbf{A})}{4\bar{u}}$ , we need

$$(4.22) \quad B \geq \left( 2\bar{u} \left( c_1 + \sqrt{c_1^2 + \frac{uc_2\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}}} \right) \right)^2 / \left( u\lambda_d(\mathbf{A}^\top \mathbf{A}) \right)^2.$$

The following is the estimation of  $c_1 + \sqrt{c_1^2 + \frac{uc_2\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}}}$ :

$$\begin{aligned} & c_1 + \sqrt{c_1^2 + \frac{uc_2\lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}}} \\ &= \left( 2 \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \|\mathbf{A}\|_2^2 \log(2d) \right)^{\frac{1}{2}} \\ & \quad + \left( 2 \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \|\mathbf{A}\|_2^2 \log(2d) + \frac{u\lambda_d(\mathbf{A}^\top \mathbf{A})\log(2d)}{3\bar{u}} \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} - \lambda_d(\mathbf{A}^\top \mathbf{A}) \right) \right)^{\frac{1}{2}} \\ &\leq \left( 2 \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \|\mathbf{A}\|_2^2 \log(2d) \right)^{\frac{1}{2}} + \left( 2 \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \|\mathbf{A}\|_2^2 \log(2d) + \frac{\max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \log(2d) u\lambda_d(\mathbf{A}^\top \mathbf{A})}{3\bar{u}} \right)^{\frac{1}{2}} \\ &= \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \log(2d) \right)^{\frac{1}{2}} \left( \sqrt{2} \|\mathbf{A}\|_2 + \left( 2 \|\mathbf{A}\|_2^2 + \frac{u\lambda_d(\mathbf{A}^\top \mathbf{A})}{3\bar{u}} \right)^{\frac{1}{2}} \right) \\ &\leq \left( \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \log(2d) \right)^{\frac{1}{2}} \left( 2\sqrt{2} \|\mathbf{A}\|_2 + \left( \frac{u\lambda_d(\mathbf{A}^\top \mathbf{A})}{3\bar{u}} \right)^{\frac{1}{2}} \right), \end{aligned}$$

where the second inequality is due to  $(y^2 + z^2)^{\frac{1}{2}} \leq |y| + |z|$ . Combining the above with (4.22), we conclude that  $\sigma \leq \frac{u\lambda_d(\mathbf{A}^\top \mathbf{A})}{4\bar{u}}$  follows from ■

$$B \geq \frac{4\bar{u}^2 \log(2d) \max_{j=1,\dots,n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\}}{\bar{u}^2} \left( \frac{2\sqrt{2} \|\mathbf{A}\|_2}{\lambda_d(\mathbf{A}^\top \mathbf{A})} + \left( \frac{\bar{u}}{3\bar{u}\lambda_d(\mathbf{A}^\top \mathbf{A})} \right)^{\frac{1}{2}} \right)^2.$$

**Remark 4.5.** A straightforward calculation shows that the minimum of  $\max_{j=1,\dots,n} \{ \|\mathbf{a}_j\|_2^2 / p_j \}$  over all sample probabilities  $\{p_j\}$  with  $p_j > 0$ ,  $j = 1, \dots, n$ , is attained when sampling rows with probability proportional to their squared length, that is,  $p_j = \|\mathbf{a}_j\|_2^2 / \|\mathbf{A}\|_F^2$ ,  $j = 1, \dots, n$ . Under this choice, the resulting minimum is  $\|\mathbf{A}\|_F^2$ . A similar statement can be found in [29].

**4.2. Inconsistent LLSP.** For the inconsistent LLSP,  $\mathbf{A}\mathbf{x}^*$  is not equal to  $\mathbf{b}$ , meaning that  $\nabla f_i(\mathbf{x}^*) \neq \mathbf{0}$  for some  $i = 1, \dots, n$ . In this case, an extra residual term will appear in the upper bound on  $\|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2$  given by (4.1). This requires additional estimation to analyze the convergence of **SGA-RMSProp**, that is, to show that the distances between iteration points  $\{\mathbf{x}_k\}$  and a neighborhood of  $\mathbf{x}^*$  converge to zero. The following are details.



Denote  $\mathbf{r}^* = (r_1^*, \dots, r_d^*)^\top := \mathbf{A}\mathbf{x}^* - \mathbf{b}$ , that is,  $r_i^* := \mathbf{a}_i^\top \mathbf{x}^* - b_i$ ,  $i = 1, \dots, n$ . Then the stochastic gradient at  $\mathbf{x}_k$  can be written as

$$\begin{aligned}
 \mathbf{g}_k &= \frac{1}{B} \sum_{i=1}^B \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \left( \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}_k - b_{\xi_i^{(k)}} \right) \right) \\
 &= \frac{1}{B} \sum_{i=1}^B \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \left( \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}_k - \mathbf{a}_{\xi_i^{(k)}}^\top \mathbf{x}^* + r_{\xi_i^{(k)}}^* \right) \right) \\
 &= \frac{1}{B} \sum_{i=1}^B \left( \frac{1}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \mathbf{a}_{\xi_i^{(k)}}^\top (\mathbf{x}_k - \mathbf{x}^*) + \frac{r_{\xi_i^{(k)}}^*}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}} \right) \\
 (4.23) \quad &= \mathbf{M}_k (\mathbf{x}_k - \mathbf{x}^*) + \mathbf{h}_k,
 \end{aligned}$$

where  $\mathbf{h}_k := \frac{1}{B} \sum_{i=1}^B \frac{r_{\xi_i^{(k)}}^*}{p_{\xi_i^{(k)}}} \mathbf{a}_{\xi_i^{(k)}}$ . Below is a lemma that provides an estimation of  $\mathbb{E}[\|\mathbf{h}_k\|_2]$ , which is derived from the proof of Theorem 5 in [5].

**Lemma 4.6.** *For each  $k = 1, \dots, K$ , we have*

$$(4.24) \quad \mathbb{E}[\|\mathbf{h}_k\|_2] \leq \sqrt{\frac{2 \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \|\mathbf{r}^*\|_2^2 \log(d+1)}{B}} + \frac{\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2 |r_j^*|}{p_j} \right\} \log(d+1)}{3B}.$$

Next, we present the following theorem to analyze the convergence of **SGA-RMSProp** on the inconsistent LLSP.

**Theorem 4.7.** *Let  $\mathbf{x}^*$  be the optimal point of the inconsistent LLSP (1.4),  $\mathbf{r}^* = \mathbf{A}\mathbf{x}^* - \mathbf{b}$ ,  $0 < \varepsilon \leq \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{2\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ ,  $B \geq \frac{4\bar{u}^2 \log(2d) \max_{j=1, \dots, n} \{\|\mathbf{a}_j\|_2^2 / p_j\}}{\underline{u}^2} \left( \frac{2\sqrt{2}\|\mathbf{A}\|_2}{\lambda_d(\mathbf{A}^\top \mathbf{A})} + \left( \frac{\underline{u}}{3\bar{u}\lambda_d(\mathbf{A}^\top \mathbf{A})} \right)^{\frac{1}{2}} \right)^2$ ,  $\eta_k \equiv \eta = \frac{2}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ , and  $\{\mathbf{x}_k\}_{k=1}^K$  be the sequence generated by **SGA-RMSProp**. Then we have*

$$\mathbb{E}[\|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2] \leq \rho \|\mathbf{x}_1 - \mathbf{x}^*\|_2 \left( G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right) \right)^K + R,$$

with  $\rho$ ,  $G(\gamma_1, \dots, \gamma_K)$  defined as in **Theorem 4.3** and

$$(4.25) \quad R \leq \frac{2\bar{u}^{\frac{3}{2}}}{\underline{u}^{\frac{3}{2}} \lambda_d(\mathbf{A}^\top \mathbf{A})} \left( \sqrt{\frac{2 \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \|\mathbf{r}^*\|_2^2 \log(d+1)}{B}} + \frac{\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2 |r_j^*|}{p_j} \right\} \log(d+1)}{3B} \right).$$

**Proof.** Substituting (4.23) into (1.6), for each  $k = 1, \dots, K$ , we have

$$\begin{aligned}
 \mathbf{x}_{k+1} - \mathbf{x}^* &= \mathbf{x}_k - \mathbf{x}^* - \eta \mathbf{D}_k \mathbf{M}_k (\mathbf{x}_k - \mathbf{x}^*) - \eta \mathbf{D}_k \mathbf{h}_k \\
 &= (\mathbf{I} - \eta \mathbf{D}_k \mathbf{M}_k) (\mathbf{x}_k - \mathbf{x}^*) - \eta \mathbf{D}_k \mathbf{h}_k
 \end{aligned}$$

$$(4.26) \quad = \mathbf{Y}_k (\mathbf{x}_k - \mathbf{x}^*) - \eta \mathbf{D}_k \mathbf{h}_k$$

where, consistent with [Theorem 4.3](#), we denote  $\mathbf{Y}_k = \mathbf{I} - \eta \mathbf{D}_k \mathbf{M}_k$ . It follows that

$$(4.27) \quad \mathbf{x}_{K+1} - \mathbf{x}^* = (\mathbf{Y}_K \cdots \mathbf{Y}_1) (\mathbf{x}_1 - \mathbf{x}^*) - \eta \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right),$$

in which  $\prod_{i=K+1}^K \mathbf{Y}_i$  represents the unit matrix  $\mathbf{I}$ . We use this notation just for simplicity.

By triangle inequality, we have

$$(4.28) \quad \|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2 \leq \|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2 \|\mathbf{x}_1 - \mathbf{x}^*\|_2 + \eta \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2.$$

The term  $\|\mathbf{Y}_K \cdots \mathbf{Y}_1\|_2$  has been studied in [Theorem 4.3](#), so we focus on the second part. By [Proposition 3.1](#),  $\|\mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_{k-1}^{-\frac{1}{2}}\|_2$  is bounded above by 1 for  $k = 1, \dots, K$ , while  $\|\mathbf{D}_K^{\frac{1}{2}}\|_2 \|\mathbf{D}_j^{-\frac{1}{2}}\|_2$  is bounded above by  $\bar{u}^{\frac{1}{2}}/\underline{u}^{\frac{1}{2}}$  for  $j = 1, \dots, K-1$ . Similarly to (4.6) and (4.7), we have

$$(4.29) \quad \|\mathbf{Y}_K \mathbf{Y}_{K-1} \cdots \mathbf{Y}_{j+1}\|_2 \leq \frac{\bar{u}^{\frac{1}{2}}}{\underline{u}^{\frac{1}{2}}} \prod_{i=j+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon), \quad j = 1, \dots, K-1,$$

and

$$\begin{aligned} \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2 &\leq \sum_{j=1}^K \left\| \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right\|_2 \\ &\leq \sum_{j=1}^K \left( \left\| \prod_{i=j+1}^K \mathbf{Y}_i \right\|_2 \|\mathbf{D}_j\|_2 \|\mathbf{h}_j\|_2 \right) \stackrel{(3.2), (4.29)}{\leq} \frac{\bar{u}^{\frac{3}{2}}}{\underline{u}^{\frac{1}{2}}} \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \right), \end{aligned}$$

where  $\mathbf{X}_i = \mathbf{I} - \eta \mathbf{D}_{i-1}^{\frac{1}{2}} \mathbf{M}_i \mathbf{D}_{i-1}^{\frac{1}{2}}$ ,  $i = 2, \dots, K$ , and the notation  $\prod_{i=K+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon)$  represents 1 for simplicity. Take expectation on both sides and we have

$$(4.30) \quad \mathbb{E} \left[ \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2 \right] \leq \frac{\bar{u}^{\frac{3}{2}}}{\underline{u}^{\frac{1}{2}}} \sum_{j=1}^K \left( \mathbb{E} \left[ \left( \prod_{i=j+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \right] \right).$$

Recalling [Corollary 4.4](#), setting  $B$  and  $\varepsilon$  as in [Theorem 4.7](#) yields

$$(4.31) \quad 1 - \frac{2(\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u} \sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon \leq 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}.$$

Notice that  $\mathbf{h}_j$  is  $\mathcal{F}_{j+1}$ -measurable,  $j = 1, \dots, K-1$ . Applying the process similar to (4.8), (4.9), and (4.19),  $\mathbb{E} \left[ \left( \prod_{i=j+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \right]$  can be estimated as follows:

$$\mathbb{E} \left[ \left( \prod_{i=j+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \right]$$

$$\begin{aligned}
& \stackrel{(4.8),(4.9)}{=} \mathbb{E} \left[ \cdots \mathbb{E} \left[ \mathbb{E} [\|\mathbf{X}_K\|_2 + \varepsilon | \mathcal{F}_K] \left( \prod_{i=j+1}^{K-1} (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \middle| \mathcal{F}_{K-1} \right] \cdots \right] \\
& \stackrel{(4.19),(4.31)}{\leq} \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right) \\
& \quad \mathbb{E} \left[ \cdots \mathbb{E} \left[ \mathbb{E} [\|\mathbf{X}_{K-1}\|_2 + \varepsilon | \mathcal{F}_{K-1}] \left( \prod_{i=j+1}^{K-2} (\|\mathbf{X}_i\|_2 + \varepsilon) \right) \|\mathbf{h}_j\|_2 \middle| \mathcal{F}_{K-2} \right] \cdots \right] \\
& \leq \cdots \\
& \leq \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right)^{K-j-1} \mathbb{E} [\mathbb{E} [\|\mathbf{X}_{j+1}\|_2 + \varepsilon | \mathcal{F}_{j+1}] \|\mathbf{h}_j\|_2] \\
(4.32) \quad & \leq \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right)^{K-j} \mathbb{E} [\|\mathbf{h}_j\|_2]
\end{aligned}$$

for  $j = 1, \dots, K-1$ . Moreover, the above also holds for  $j = K$  since  $\prod_{i=K+1}^K (\|\mathbf{X}_i\|_2 + \varepsilon)$  represents 1.

As mentioned in Lemma 4.6,  $\mathbb{E}[\|\mathbf{h}_j\|_2]$  has the same value for each  $j = 1, \dots, K$ . Combining (4.30) with (4.32), we obtain

$$(4.33) \quad \mathbb{E} \left[ \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2 \right] \leq \frac{\bar{u}^{\frac{3}{2}} \mathbb{E}[\|\mathbf{h}_1\|_2]}{\underline{u}^{\frac{1}{2}}} \sum_{j=1}^K \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right)^{K-j}.$$

Finally, taking expectation on both sides of (4.28) gives

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}_{K+1} - \mathbf{x}^*\|_2] & \stackrel{(4.20)}{\leq} \rho \|\mathbf{x}_1 - \mathbf{x}^*\|_2 \left( G(\gamma_1, \dots, \gamma_K) \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right) \right)^K \\
& \quad + \eta \mathbb{E} \left[ \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2 \right].
\end{aligned}$$

According to Lemma 4.6 and our setting of  $\eta$ , we have

$$\begin{aligned}
R & := \eta \mathbb{E} \left[ \left\| \sum_{j=1}^K \left( \left( \prod_{i=j+1}^K \mathbf{Y}_i \right) \mathbf{D}_j \mathbf{h}_j \right) \right\|_2 \right] \\
& \stackrel{(4.33)}{\leq} \frac{\eta \bar{u}^{\frac{3}{2}} \mathbb{E}[\|\mathbf{h}_1\|_2]}{\underline{u}^{\frac{1}{2}}} \sum_{j=1}^K \left( 1 - \frac{\underline{u} \lambda_d(\mathbf{A}^\top \mathbf{A})}{\bar{u} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} \right)^{K-j} \\
& \leq \frac{\eta \bar{u}^{\frac{5}{2}} (\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})) \mathbb{E}[\|\mathbf{h}_1\|_2]}{\underline{u}^{\frac{3}{2}} \lambda_d(\mathbf{A}^\top \mathbf{A})} \\
& \stackrel{(4.24)}{\leq} \frac{2\bar{u}^{\frac{3}{2}}}{\underline{u}^{\frac{3}{2}} \lambda_d(\mathbf{A}^\top \mathbf{A})} \left( \sqrt{\frac{2 \max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2^2}{p_j} \right\} \|\mathbf{r}^*\|_2^2 \log(d+1)}{B}} + \frac{\max_{j=1, \dots, n} \left\{ \frac{\|\mathbf{a}_j\|_2 |r_j^*|}{p_j} \right\} \log(d+1)}{3B} \right).
\end{aligned}$$

This completes the proof. ■

Extending from the consistent case, [Theorem 4.7](#) indicates that for the inconsistent LLSP, the sequence  $\{\mathbf{x}_k\}$  generated by [SGA-RMSProp](#) still converges R-linearly, but to a neighborhood of  $\mathbf{x}^*$  with radius  $R$ . This kind of neighborhood can be viewed as the “region of confusion”, where the method fails to obtain a clear direction towards the optimal point [3]. As shown in (4.25), the region of confusion provided by [Theorem 4.7](#) can be controlled by the batch size  $B$ . An increase in the batch size results in a reduction of the region of confusion.

**5. Numerical experiments.** In this section, we evaluate the performance of [SGA-RMSProp](#) on LLSP using synthetic and real data. We compare [SGA-RMSProp](#) with SGD, showing that [SGA-RMSProp](#) generally converges faster on LLSP with the small batch size and exhibits a faster initial convergence rate with the large batch size. Based on this, a strategy for switching from [SGA-RMSProp](#) to SGD is proposed, combining the benefits of these two algorithms. Our experiments are performed in MATLAB R2023b (version 23.2.0.2365128) on a desktop equipped with 64 GB memory, an Intel Core i9-12900K (3.2GHz).

When sampling the mini-batch stochastic gradient,  $B = 50$  and 1000 are used to indicate the small and large batch sizes respectively, and let  $p_i = \frac{\|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_F^2}$  (see [Remark 4.5](#)),  $i = 1, \dots, n$ .

For parameter settings of [SGA-RMSProp](#), let  $\beta_k = \frac{1}{2} \max_{j=1, \dots, d} \left\{ \frac{g_{k,j}^2 - \frac{1}{\underline{u}^2}}{g_{k,j}^2 - u_{k-1,j}}, \frac{g_{k,j}^2 - (1+\varepsilon)^2 u_{k-1,j}}{g_{k,j}^2 - u_{k-1,j}}, 0 \right\} + \frac{1}{2}$  when  $\frac{g_{k,j}^2}{u_{k-1,j}} > 1$ ,  $j = 1, \dots, d$ , within the range defined in [SGA-RMSProp](#). We use the following step size when  $B = 50$ :

$$\eta = \begin{cases} \frac{1.1B}{\bar{u}(\|\mathbf{A}\|_F^2 + (B-1)\lambda_d(\mathbf{A}^\top \mathbf{A}))}, & \|\mathbf{A}\|_F^2 - (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) - \lambda_d(\mathbf{A}^\top \mathbf{A})) \geq 0, \\ \frac{(2.1 + \sqrt{\lambda_d(\mathbf{A}^\top \mathbf{A})/\lambda_1(\mathbf{A}^\top \mathbf{A})})B}{\bar{u}(\|\mathbf{A}\|_F^2 + (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})))}, & \|\mathbf{A}\|_F^2 - (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) - \lambda_d(\mathbf{A}^\top \mathbf{A})) < 0, \end{cases}$$

which is inspired by the step size of mini-batch SGD suggested by Moorman [26]. When  $B = 1000$ , we apply a heuristic step size based on our theoretical results. Experiments show that the algorithm always converges with  $\eta = \frac{2}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$ , which is also the step size in [Theorem 4.3](#) and [Theorem 4.7](#), and a slightly larger  $\eta$  may speed up convergence. Therefore, we set  $\eta = \frac{2 + \sqrt{\lambda_d(\mathbf{A}^\top \mathbf{A})/\lambda_1(\mathbf{A}^\top \mathbf{A})}}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}$  when  $B = 1000$ . The effect of  $\bar{u}$ ,  $\underline{u}$ , and  $\varepsilon$  is discussed in [subsection 5.1.2](#).

In each experiment, the algorithm will be run 100 times. Unless otherwise specified, we regard the algorithm as converged if  $\frac{\|\mathbf{x}_k - \mathbf{x}^*\|_2}{\|\mathbf{x}_1 - \mathbf{x}^*\|_2} \leq 10^{-4}$ . When presenting the results, we calculate the mean using values from the 5th to 95th percentile to avoid extreme cases skewing the average.

**5.1. Experiments for consistent LLSP.** In this subsection, we present the convergence results of [SGA-RMSProp](#) for different batch sizes, showing the effect of  $\bar{u}$ ,  $\underline{u}$ , and  $\varepsilon$ . Next, the performances of [SGA-RMSProp](#) and other algorithms are compared on the consistent LLSP.

**5.1.1. Data generation.** We generate the synthetic data similar to [32, 5]. Let  $\mathbf{A} \in \mathbb{R}^{n \times d}$  with  $n = 10^6$  and  $d = 10^2$ . Its thin singular value decomposition is  $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ , where  $\mathbf{U} \in \mathbb{R}^{n \times d}$  has orthonormal columns,  $\mathbf{V}$  is orthogonal, both chosen uniformly at random as in Proposition 7.2 of [9], and  $\mathbf{\Sigma} = \text{diag}(s_1, \dots, s_d)$ . Then the singular values of  $\mathbf{A}^\top \mathbf{A}$  are  $\{s_1^2, \dots, s_d^2\}$ , set

to exponential or algebraic decay. Specifically, given the condition number  $\kappa = \frac{\lambda_1(\mathbf{A}^\top \mathbf{A})}{\lambda_d(\mathbf{A}^\top \mathbf{A})}$  of  $\mathbf{A}^\top \mathbf{A}$ , decay rate  $q$ , and  $\lambda_d(\mathbf{A}^\top \mathbf{A})$ , the formula for the exponential decay (ED) is:

$$s_j = \sqrt{\lambda_d(\mathbf{A}^\top \mathbf{A}) + \left(\frac{d-j}{d-1}\right) \lambda_d(\mathbf{A}^\top \mathbf{A})(\kappa-1)q^{j-1}}, \quad j = 1, \dots, d,$$

and the formula for the algebraic decay (AD) is

$$s_j = \sqrt{\lambda_d(\mathbf{A}^\top \mathbf{A}) + \left(\frac{d-j}{d-1}\right)^q \lambda_d(\mathbf{A}^\top \mathbf{A})(\kappa-1)}, \quad j = 1, \dots, d.$$

With the same condition number, different decay types and rates result in different singular value distributions [32]. We take  $\lambda_d(\mathbf{A}^\top \mathbf{A}) = 1$ ,  $\kappa = 20, 50, 100$  (small, medium, large condition numbers) and  $q = 0.2, 0.7$  for ED,  $q = 1, 2$  for AD. These problems are denoted as (decay-type,  $\kappa$ ,  $q$ ). Lastly, let the minimizer  $\mathbf{x}^*$  have independent standard normal entries and  $\mathbf{b}$  is set to  $\mathbf{A}\mathbf{x}^*$ . We have defined 12 problems, each with 3 random instances, totaling 36 instances. Since  $\mathbf{A}$  and  $\mathbf{x}^*$  differ across the instances, the data is sufficiently randomized. Hence, we fix the initial point  $\mathbf{x}_1 = (2, \dots, 2)^\top$  in all experiments. For each problem, we report the mean and standard deviation over three instances, in the form of (mean, standard deviation).

**Table 1**

*Numbers of iterations required for SGA-RMSProp to converge ( $\varepsilon = \varepsilon_1$ ), reported as (mean, standard deviation) over three instances for  $\bar{u}_i, u_i, i = 1, \dots, 4$ .*

Batch sizes	Problems	$(\underline{u}_1, \bar{u}_1)$	$(\underline{u}_2, \bar{u}_2)$	$(\underline{u}_3, \bar{u}_3)$	$(\underline{u}_4, \bar{u}_4)$
$B = 50$	(ED, 20, 0.7)	(113, 0.6)	(115, 0.6)	(117, 0.0)	(121, 1.2)
	(ED, 20, 0.2)	(113, 1.0)	(113, 0.6)	(116, 1.2)	(121, 2.0)
	(ED, 50, 0.7)	(255, 2.5)	(257, 3.2)	(266, 3.0)	(275, 3.6)
	(ED, 50, 0.2)	(261, 5.1)	(266, 2.0)	(276, 3.0)	(287, 3.0)
	(ED, 100, 0.7)	(493, 4.5)	(492, 2.0)	(506, 2.5)	(525, 2.3)
	(ED, 100, 0.2)	(531, 13.3)	(550, 13.5)	(575, 18.3)	(600, 14.0)
	(AD, 20, 1)	(149, 8.5)	(153, 9.5)	(157, 10.0)	(162, 11.7)
	(AD, 20, 2)	(145, 2.1)	(142, 0.6)	(144, 0.6)	(147, 2.0)
	(AD, 50, 1)	(327, 41.0)	(333, 39.9)	(341, 46.5)	(350, 49.4)
	(AD, 50, 2)	(304, 5.7)	(311, 6.4)	(319, 7.2)	(327, 9.0)
	(AD, 100, 1)	(541, 46.1)	(559, 53.4)	(569, 48.3)	(583, 59.3)
	(AD, 100, 2)	(558, 39.2)	(565, 45.5)	(582, 43.8)	(595, 49.2)
$B = 1000$	(ED, 20, 0.7)	(97, 1.7)	(97, 3.1)	(101, 3.1)	(109, 4.0)
	(ED, 20, 0.2)	(98, 2.1)	(100, 2.1)	(106, 2.5)	(119, 5.7)
	(ED, 50, 0.7)	(233, 3.0)	(240, 3.1)	(251, 5.9)	(268, 4.9)
	(ED, 50, 0.2)	(236, 1.5)	(244, 5.3)	(259, 9.5)	(281, 19.0)
	(ED, 100, 0.7)	(459, 2.5)	(463, 3.5)	(477, 9.9)	(500, 13.6)
	(ED, 100, 0.2)	(467, 5.6)	(479, 13.1)	(496, 18.9)	(530, 27.5)
	(AD, 20, 1)	(209, 8.1)	(127, 5.1)	(80, 1.0)	(74, 3.5)
	(AD, 20, 2)	(173, 20.0)	(110, 8.7)	(84, 5.3)	(78, 2.1)
	(AD, 50, 1)	(254, 29.0)	(191, 18.4)	(172, 17.3)	(171, 21.0)
	(AD, 50, 2)	(266, 27.9)	(206, 9.9)	(195, 4.0)	(195, 3.5)
	(AD, 100, 1)	(366, 4.7)	(316, 16.9)	(298, 25.5)	(294, 33.8)
	(AD, 100, 2)	(371, 21.9)	(360, 24.7)	(361, 23.1)	(362, 27.8)

**5.1.2. Effect of  $\bar{u}$ ,  $\underline{u}$ , and  $\varepsilon$ .** Values of  $\bar{u}$  are chosen as  $\bar{u}_i = (10^{-i} \min_{j=1,\dots,d} \{g_{1,j}^2 | g_{1,j} \neq 0\})^{-\frac{1}{2}}$ ,  $i = 1, \dots, 4$ . Note that  $\mathbf{g}_1$  is directly computed in the initial step, with no parameters required. This setting allows the upper bound  $\bar{u}$  and the initial value  $\mathbf{u}_0$  to match the scale of the observed stochastic gradient. For each  $\bar{u}_i$ , we set  $\underline{u}$  to  $\underline{u}_i = \frac{\bar{u}_i}{5}$  and  $\underline{u}'_i = \frac{\bar{u}_i}{10}$ ,  $i = 1, \dots, 4$ . Let  $\varepsilon$  take three values,  $\varepsilon_1 = \frac{\lambda_d(\mathbf{A}^\top \mathbf{A})}{\lambda_1(\mathbf{A}^\top \mathbf{A})}$ ,  $\varepsilon_2 = \frac{5\lambda_d(\mathbf{A}^\top \mathbf{A})}{\lambda_1(\mathbf{A}^\top \mathbf{A})}$ ,  $\varepsilon_3 = \frac{10\lambda_d(\mathbf{A}^\top \mathbf{A})}{\lambda_1(\mathbf{A}^\top \mathbf{A})}$ , representing low, medium, and high levels, respectively. The effect of these parameter combinations is tested in the following experiments.

Table 1 and Table 2 show the numbers of iterations required for SGA-RMSProp to converge with  $\varepsilon = \varepsilon_1$ , given different combinations of  $\bar{u}$  and  $\underline{u}$ . The mean is rounded to the nearest integer and the standard deviation is rounded to the nearest tenth when presenting the results.

Table 2

Numbers of iterations required for SGA-RMSProp to converge ( $\varepsilon = \varepsilon_1$ ), reported as (mean, standard deviation) over three instances for  $\bar{u}_i, \underline{u}'_i$ ,  $i = 1, \dots, 4$ .

Batch sizes	Problems	$(\underline{u}'_1, \bar{u}_1)$	$(\underline{u}'_2, \bar{u}_2)$	$(\underline{u}'_3, \bar{u}_3)$	$(\underline{u}'_4, \bar{u}_4)$
$B = 50$	(ED, 20, 0.7)	(114, 0.6)	(115, 0.0)	(117, 0.0)	(121, 1.2)
	(ED, 20, 0.2)	(114, 0.6)	(113, 0.6)	(116, 1.2)	(121, 2.0)
	(ED, 50, 0.7)	(256, 0.6)	(257, 1.5)	(266, 3.0)	(275, 3.6)
	(ED, 50, 0.2)	(260, 3.8)	(262, 2.3)	(276, 3.0)	(287, 3.0)
	(ED, 100, 0.7)	(494, 4.4)	(489, 1.0)	(506, 2.5)	(525, 2.3)
	(ED, 100, 0.2)	(528, 12.1)	(541, 9.8)	(575, 18.3)	(600, 14.0)
	(AD, 20, 1)	(150, 9.3)	(152, 9.5)	(157, 10.0)	(162, 11.7)
	(AD, 20, 2)	(146, 1.5)	(143, 1.5)	(144, 0.6)	(147, 2.0)
	(AD, 50, 1)	(329, 36.4)	(327, 47.2)	(341, 46.5)	(350, 49.4)
	(AD, 50, 2)	(305, 3.8)	(311, 6.2)	(319, 7.2)	(327, 9.0)
	(AD, 100, 1)	(555, 33.0)	(559, 52.1)	(569, 48.3)	(583, 59.3)
	(AD, 100, 2)	(558, 40.0)	(565, 43.2)	(582, 43.8)	(595, 49.2)
$B = 1000$	(ED, 20, 0.7)	(97, 1.7)	(97, 2.9)	(100, 3.2)	(110, 3.8)
	(ED, 20, 0.2)	(98, 2.1)	(100, 2.0)	(106, 2.5)	(118, 4.7)
	(ED, 50, 0.7)	(233, 3.0)	(239, 7.0)	(251, 5.9)	(267, 4.6)
	(ED, 50, 0.2)	(236, 1.5)	(245, 4.0)	(259, 9.1)	(281, 19.0)
	(ED, 100, 0.7)	(459, 2.5)	(464, 5.3)	(477, 9.9)	(500, 13.6)
	(ED, 100, 0.2)	(467, 5.6)	(481, 12.9)	(496, 18.9)	(530, 27.5)
	(AD, 20, 1)	(209, 8.1)	(136, 15.9)	(80, 1.0)	(74, 3.5)
	(AD, 20, 2)	(173, 20.0)	(112, 11.9)	(84, 5.3)	(78, 2.1)
	(AD, 50, 1)	(254, 29.0)	(195, 16.2)	(172, 17.3)	(171, 21.0)
	(AD, 50, 2)	(266, 27.9)	(211, 6.0)	(195, 4.0)	(195, 3.5)
	(AD, 100, 1)	(366, 4.7)	(313, 12.5)	(298, 25.5)	(294, 33.8)
	(AD, 100, 2)	(371, 21.9)	(360, 25.1)	(361, 23.1)	(362, 27.8)

The results show that, under our parameters selection strategy where  $\bar{u}$  is specified first and  $\underline{u}$  is chosen accordingly, the algorithm exhibits similar performance when using the parameters  $(\underline{u}_i, \bar{u}_i)$  and  $(\underline{u}'_i, \bar{u}_i)$  respectively,  $i = 1, \dots, 4$ . Therefore, we use  $\underline{u}_i = \frac{\bar{u}_i}{5}$ ,  $i = 1, \dots, 4$ , in the following experiments.

Table 1-Table 4 indicate that the algorithm generally performs better with  $\bar{u} = \bar{u}_1$ , the smallest value among the considered parameters, when  $B = 50$ . According to (3.2) in Proposition 3.1, we have  $\|\mathbf{D}_k\|_2 \leq \bar{u}$ ,  $k = 1, \dots, K$ . Since the uncertainty of the stochastic gradient is higher with a smaller batch size, the results indicate that choosing a small  $\bar{u}$  restricts potential

Table 3

Numbers of iterations required for SGA-RMSProp to converge ( $\varepsilon = \varepsilon_2$ ), reported as (mean, standard deviation) over three instances for  $\bar{u}_i, \underline{u}_i, i = 1, \dots, 4$ .

Batch sizes	Problems	$(\underline{u}_1, \bar{u}_1)$	$(\underline{u}_2, \bar{u}_2)$	$(\underline{u}_3, \bar{u}_3)$	$(\underline{u}_4, \bar{u}_4)$
$B = 50$	(ED, 20, 0.7)	(140, 1.0)	(177, 3.1)	(233, 7.9)	(288, 5.5)
	(ED, 20, 0.2)	(144, 4.6)	(183, 4.4)	(253, 9.5)	(298, 5.9)
	(ED, 50, 0.7)	(305, 11.9)	(364, 21.0)	(507, 26.7)	(601, 36.3)
	(ED, 50, 0.2)	(309, 35.7)	(386, 36.0)	(504, 67.2)	(635, 49.7)
	(ED, 100, 0.7)	(541, 2.3)	(634, 24.0)	(796, 31.8)	(1005, 13.1)
	(ED, 100, 0.2)	(603, 18.2)	(765, 44.7)	(981, 56.9)	(1246, 43.7)
	(AD, 20, 1)	(167, 14.2)	(191, 19.0)	(224, 28.6)	(297, 45.3)
	(AD, 20, 2)	(154, 3.1)	(177, 7.5)	(215, 5.1)	(267, 15.9)
	(AD, 50, 1)	(345, 49.6)	(378, 52.5)	(428, 72.9)	(512, 115.5)
	(AD, 50, 2)	(331, 10.7)	(370, 16.4)	(427, 22.6)	(543, 37.6)
	(AD, 100, 1)	(568, 54.9)	(607, 53.9)	(650, 65.0)	(732, 73.9)
	(AD, 100, 2)	(587, 44.3)	(631, 61.7)	(720, 74.7)	(866, 141.7)
$B = 1000$	(ED, 20, 0.7)	(143, 20.6)	(209, 29.6)	(281, 28.1)	(308, 8.7)
	(ED, 20, 0.2)	(176, 19.7)	(246, 28.9)	(304, 8.3)	(324, 1.5)
	(ED, 50, 0.7)	(299, 6.7)	(414, 10.1)	(603, 26.9)	(739, 29.5)
	(ED, 50, 0.2)	(322, 66.8)	(482, 124.4)	(664, 126.6)	(784, 68.7)
	(ED, 100, 0.7)	(485, 13.1)	(584, 41.8)	(777, 63.8)	(1093, 95.0)
	(ED, 100, 0.2)	(537, 37.2)	(710, 128.5)	(1005, 258.0)	(1354, 321.7)
	(AD, 20, 1)	(136, 31.5)	(81, 5.0)	(83, 9.0)	(89, 11.8)
	(AD, 20, 2)	(116, 8.7)	(88, 2.9)	(92, 3.5)	(111, 3.6)
	(AD, 50, 1)	(207, 21.1)	(180, 24.3)	(184, 23.8)	(193, 30.9)
	(AD, 50, 2)	(245, 8.6)	(206, 3.1)	(215, 9.1)	(238, 5.5)
	(AD, 100, 1)	(323, 29.0)	(293, 48.5)	(295, 50.0)	(304, 50.5)
	(AD, 100, 2)	(366, 28.0)	(368, 29.8)	(384, 32.3)	(411, 38.9)

Table 4

Numbers of iterations required for SGA-RMSProp to converge ( $\varepsilon = \varepsilon_3$ ), reported as (mean, standard deviation) over three instances for  $\bar{u}_i, \underline{u}_i, i = 1, \dots, 4$ .

Batch sizes	Problems	$(\underline{u}_1, \bar{u}_1)$	$(\underline{u}_2, \bar{u}_2)$	$(\underline{u}_3, \bar{u}_3)$	$(\underline{u}_4, \bar{u}_4)$
$B = 50$	(ED, 20, 0.7)	(198, 16.2)	(280, 6.6)	(336, 12.7)	(362, 1.7)
	(ED, 20, 0.2)	(203, 13.3)	(287, 7.9)	(344, 5.9)	(357, 2.3)
	(ED, 50, 0.7)	(380, 32.7)	(566, 63.9)	(738, 41.4)	(839, 5.5)
	(ED, 50, 0.2)	(418, 69.8)	(567, 56.2)	(723, 58.4)	(814, 18.2)
	(ED, 100, 0.7)	(606, 9.8)	(890, 26.2)	(1196, 28.3)	(1503, 23.0)
	(ED, 100, 0.2)	(753, 54.0)	(1054, 103.7)	(1320, 119.4)	(1540, 135.6)
	(AD, 20, 1)	(194, 20.1)	(254, 33.6)	(341, 33.5)	(448, 67.5)
	(AD, 20, 2)	(181, 15.0)	(237, 10.0)	(315, 11.6)	(395, 10.6)
	(AD, 50, 1)	(377, 52.5)	(457, 88.0)	(552, 136.3)	(782, 205.6)
	(AD, 50, 2)	(369, 22.5)	(458, 26.9)	(617, 20.7)	(846, 42.4)
	(AD, 100, 1)	(606, 65.9)	(648, 85.8)	(771, 107.9)	(960, 143.4)
	(AD, 100, 2)	(621, 51.8)	(697, 72.4)	(917, 107.3)	(1281, 207.6)
$B = 1000$	(ED, 20, 0.7)	(199, 34.0)	(294, 17.5)	(331, 9.5)	(335, 6.1)
	(ED, 20, 0.2)	(241, 24.6)	(314, 18.6)	(343, 2.1)	(344, 3.1)
	(ED, 50, 0.7)	(413, 23.1)	(575, 39.6)	(764, 31.1)	(807, 27.6)
	(ED, 50, 0.2)	(431, 104.1)	(644, 152.5)	(807, 89.9)	(864, 21.2)
	(ED, 100, 0.7)	(547, 50.5)	(720, 90.4)	(1110, 126.3)	(1484, 52.4)
	(ED, 100, 0.2)	(666, 111.9)	(949, 280.3)	(1368, 300.7)	(1637, 146.3)
	(AD, 20, 1)	(97, 3.1)	(85, 9.7)	(93, 10.0)	(114, 15.6)
	(AD, 20, 2)	(108, 7.9)	(100, 3.1)	(120, 5.2)	(155, 2.5)
	(AD, 50, 1)	(200, 16.0)	(186, 23.1)	(201, 32.8)	(232, 60.4)
	(AD, 50, 2)	(229, 11.9)	(218, 7.9)	(249, 20.0)	(328, 32.1)
	(AD, 100, 1)	(324, 47.1)	(300, 42.2)	(314, 51.8)	(336, 62.8)
	(AD, 100, 2)	(371, 29.8)	(388, 34.6)	(421, 35.2)	(480, 50.3)



excessively large components in the descent direction  $\mathbf{D}_k \mathbf{g}_k$  in (1.6) that may result from the strong randomness in  $\mathbf{g}_k$ , thereby improving the stability of the algorithm.

On the other hand, the algorithm shows comparable performance across different combinations of  $\bar{u}$  and  $\underline{u}$  when  $B = 1000$ . Moreover, the effect of  $\varepsilon$  appears to be more significant than that of  $\bar{u}$  and  $\underline{u}$ , so we focus on tuning  $\varepsilon$ , and set  $\bar{u}$ ,  $\underline{u}$  to  $\bar{u}_2$ ,  $\underline{u}_2$  respectively, which are not necessarily optimal for every experiment but are also not the worst choices.

Next, Table 5 concludes the numbers of iterations required for SGA-RMSProp to converge with different  $\varepsilon$  values and batch sizes.

Table 5

The numbers of iterations required for SGA-RMSProp to converge, reported as (mean, standard deviation) over three instances.

Problems	$B = 50$			$B = 1000$		
	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$
(ED, 20, 0.7)	<b>(115, 0.6)</b>	(177, 3.1)	(280, 6.6)	<b>(97, 3.1)</b>	(209, 29.6)	(294, 17.5)
(ED, 20, 0.2)	<b>(113, 0.6)</b>	(183, 4.4)	(287, 7.9)	<b>(100, 2.1)</b>	(246, 28.9)	(314, 18.6)
(ED, 50, 0.7)	<b>(257, 3.2)</b>	(364, 21.0)	(566, 63.9)	<b>(240, 3.1)</b>	(414, 10.1)	(575, 39.6)
(ED, 50, 0.2)	<b>(266, 2.0)</b>	(386, 36.0)	(567, 56.2)	<b>(244, 5.3)</b>	(482, 124.4)	(644, 152.5)
(ED, 100, 0.7)	<b>(492, 2.0)</b>	(634, 24.0)	(890, 26.2)	<b>(463, 3.5)</b>	(584, 41.8)	(720, 90.4)
(ED, 100, 0.2)	<b>(550, 13.5)</b>	(765, 44.7)	(1054, 103.7)	<b>(479, 13.1)</b>	(710, 128.5)	(949, 280.3)
(AD, 20, 1)	<b>(153, 9.5)</b>	(191, 19.0)	(254, 33.6)	(127, 5.1)	<b>(81, 5.0)</b>	(85, 9.7)
(AD, 20, 2)	<b>(142, 0.6)</b>	(177, 7.5)	(237, 10.0)	(110, 8.7)	<b>(88, 2.9)</b>	(100, 3.1)
(AD, 50, 1)	<b>(333, 39.9)</b>	(378, 52.5)	(457, 88.0)	(191, 18.4)	<b>(180, 24.3)</b>	(186, 23.1)
(AD, 50, 2)	<b>(311, 6.4)</b>	(370, 16.4)	(458, 26.9)	(206, 9.9)	<b>(206, 3.1)</b>	(218, 7.9)
(AD, 100, 1)	<b>(559, 53.4)</b>	(607, 53.9)	(648, 85.8)	(316, 16.9)	<b>(293, 48.5)</b>	(300, 42.2)
(AD, 100, 2)	<b>(565, 45.5)</b>	(631, 61.7)	(697, 72.4)	<b>(360, 24.7)</b>	(368, 29.8)	(388, 34.6)

The results show that, with a small batch size of  $B = 50$ , the algorithm performs better with  $\varepsilon_3$  for all problems. Furthermore, when  $B = 1000$ , the algorithm performs better with  $\varepsilon_3$  on problems using ED, and performs better with  $\varepsilon_2$  on problems using AD. These parameter settings will be used in future experiments.

In addition, the results from  $B = 1000$  indicate that a suitably larger  $\varepsilon$  may accelerate the convergence of SGA-RMSProp. Recalling Theorem 4.3, the convergence rate  $\gamma$  depends not only on  $\left(1 - \frac{2(\underline{u}\lambda_d(\mathbf{A}^\top \mathbf{A}) - \bar{u}\sigma)}{\bar{u}(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))} + \varepsilon\right)$  but also on  $G(\gamma_1, \dots, \gamma_K)$ . Therefore, selecting an appropriate  $\varepsilon$  may reduce  $G(\gamma_1, \dots, \gamma_K)$ , resulting in faster convergence.

**5.1.3. Linear convergence rate.** We present the convergence of SGA-RMSProp under different problems with  $B = 1000$  through Figure 1. The vertical axis represents  $\frac{\|\mathbf{x}_k - \mathbf{x}^*\|_2}{\|\mathbf{x}_1 - \mathbf{x}^*\|_2}$ , while the horizontal axis denotes the number of iterations. The dark line represents the average of the results in each iteration. The light shaded area shows the range from the 5th to 95th percentile of the results. Since the vertical axis is logarithmic scale and the plots eventually maintain a straight line under different problems, the results demonstrate that SGA-RMSProp converges R-linearly on the consistent LLSP.

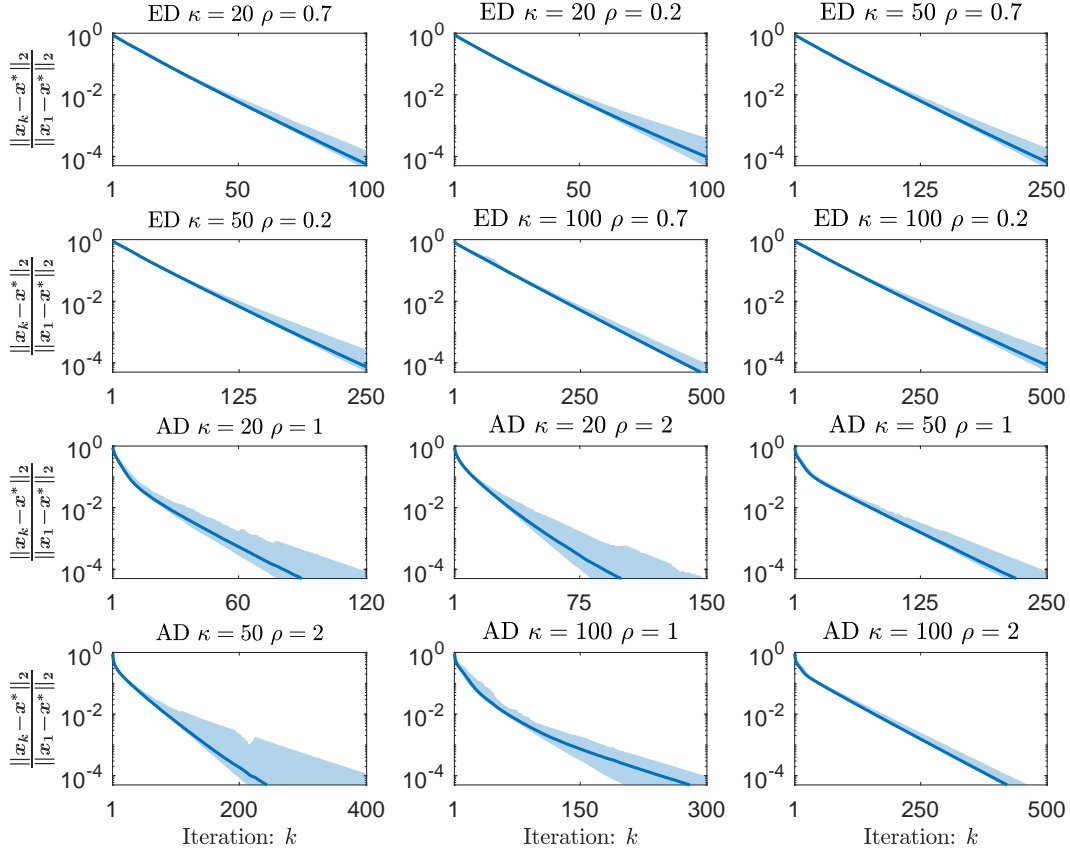


Figure 1. The  $R$ -linear convergence of SGA-RMSProp with  $B = 1000$ .

**5.1.4. Comparison of SGA-RMSProp and other algorithms.** We first compare our algorithm with the original RMSProp, whose iteration schemes follow (1.2)-(1.3) with all hyperparameters set as constants, on 12 instances derived from the problems (ED, 20, 0.7), (ED, 100, 0.7), (AD, 20, 2), and (AD, 100, 2). These instances cover both small and large condition numbers, as well as varying distributions of singular values [32]. Since there is no widely accepted parameter choice for RMSProp on LLSP, we tune its parameters as best as we can. According to the documentation of PyTorch [30], the recommended default parameters for RMSProp are  $\beta_k \equiv \beta = 0.99$  and  $\eta_k \equiv \eta = 0.01$ . Based on these, Table 6 reports the performance of RMSProp under combinations of  $\beta = 0.9, 0.99, 0.999$  and  $\eta = 0.01, 0.1, 1$ , using the notation  $\text{RMSProp}(\beta, \eta)$ . In the experiments, we find that RMSProp is already slow when  $\eta = 0.01$ , so smaller step sizes are not considered further. In Table 6, the column headings represent target accuracy level, defined as the relative error  $\frac{\|x_k - x^*\|_2}{\|x_1 - x^*\|_2}$  not exceeding 0.1, 0.01, 0.001, and 0.0001, respectively. Each cell contains a triple. The first component indicates the number of instances where the algorithm achieves the stated accuracy. The second component gives the mean number of iterations required to first hit the desired accuracy on those successful instances. The third component records the largest such iteration number among them. The triple  $(0, N, N)$  means that the algorithm does not reach the target on any instance.

Table 6

Comparison of RMSProp and SGA-RMSProp, reported as (number of successful instances, mean number of iterations, largest iteration number) over 12 instances.

Batch Sizes	Algorithm	0.1	0.01	0.001	0.0001
$B = 50$	RMSProp(0.9, 1)	(0, N, N)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.9, 0.1)	(12, 104, 142)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.9, 0.01)	(12, 692, 897)	(12, 843, 1134)	(0, N, N)	(0, N, N)
	RMSProp(0.99, 1)	(6, 57, 64)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.99, 0.1)	(12, 102, 172)	(6, 114, 147)	(5, 174, 209)	(3, 252, 260)
	RMSProp(0.99, 0.01)	(12, 793, 1087)	(9, 929, 1109)	(6, 984, 1155)	(4, 979, 1153)
	RMSProp(0.999, 1)	(12, 208, 386)	(9, 302, 621)	(9, 417, 840)	(7, 385, 1075)
	RMSProp(0.9, 0.1)	(12, 100, 184)	(12, 241, 418)	(10, 341, 696)	(9, 411, 812)
	RMSProp(0.999, 0.01)	(12, 528, 1107)	(8, 736, 1076)	(4, 777, 1149)	(3, 875, 925)
	SGA-RMSProp	(12, 65, 121)	(12, 149, 257)	(12, 239, 431)	(12, 330, 607)
$B = 1000$	RMSProp(0.9, 1)	(0, N, N)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.9, 0.1)	(12, 71, 122)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.9, 0.01)	(12, 404, 486)	(12, 520, 598)	(0, N, N)	(0, N, N)
	RMSProp(0.99, 1)	(2, 39, 40)	(0, N, N)	(0, N, N)	(0, N, N)
	RMSProp(0.99, 0.1)	(12, 63, 125)	(3, 70, 74)	(0, N, N)	(0, N, N)
	RMSProp(0.99, 0.01)	(12, 509, 704)	(12, 743, 948)	(5, 774, 852)	(3, 897, 934)
	RMSProp(0.999, 1)	(11, 127, 311)	(5, 172, 478)	(2, 155, 170)	(1, 187, 187)
	RMSProp(0.999, 0.1)	(12, 71, 176)	(9, 189, 522)	(5, 234, 640)	(1, 176, 176)
	RMSProp(0.999, 0.01)	(12, 256, 543)	(9, 490, 988)	(7, 625, 883)	(5, 749, 897)
	SGA-RMSProp	(12, 48, 111)	(12, 113, 228)	(12, 183, 347)	(12, 256, 470)

The results show that **SGA-RMSProp** generally performs better than the original RMSProp. Therefore, in the following experiments, we do not include comparisons between SGA-RMSProp and RMSProp, but instead focus on comparing **SGA-RMSProp** with SGD.

Note that the SGD in our experiments also uses the mini-batch stochastic gradient (1.5). We use the following suggested step size [26] for SGD when  $B = 50$ :

$$\eta = \begin{cases} \frac{B}{\|\mathbf{A}\|_F^2 + (B-1)\lambda_d(\mathbf{A}^\top \mathbf{A})}, & \|\mathbf{A}\|_F^2 - (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) - \lambda_d(\mathbf{A}^\top \mathbf{A})) \geq 0, \\ \frac{2B}{\|\mathbf{A}\|_F^2 + (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A}))}, & \|\mathbf{A}\|_F^2 - (B-1)(\lambda_1(\mathbf{A}^\top \mathbf{A}) - \lambda_d(\mathbf{A}^\top \mathbf{A})) < 0. \end{cases}$$

Under our problems with the large batch size  $B = 1000$ , we find that SGD generally performs better with  $\eta = \frac{2}{\lambda_1(\mathbf{A}^\top \mathbf{A}) + \lambda_d(\mathbf{A}^\top \mathbf{A})}$ , the optimal fixed step size of gradient descent for the strongly convex quadratic function [2]. Therefore, this step size is used when  $B = 1000$ .

We directly compare the wall-clock time of the algorithms in Table 7. The results indicate that **SGA-RMSProp** performs better with the small batch size. Intuitively, when the number of samples is small, the stochastic gradient may not be accurate enough, whereas the correction step in **SGA-RMSProp** helps the algorithm find a better descent direction. When the batch size is large, we observe that **SGA-RMSProp** is faster than SGD in the early stages in terms of iterations, but is surpassed by SGD eventually. Figure 2 shows the early performances of **SGA-RMSProp** and SGD on the three instances of the problem (AD, 20, 1) when  $B = 1000$ .

Recalling the definition of  $\gamma_k$  from the proof of Theorem 4.3,  $\beta_k < 1$  may lead to  $\gamma_k < 1$ ,  $k = 1, \dots, K$ . This scenario frequently occurs during the early iterations of the algorithm, resulting in smaller values of  $G(\gamma_1, \dots, \gamma_k)$  at beginning, which may be the reason behind the fast initial convergence rate observed in our experiments. Therefore, starting with **SGA-**

Table 7

Wall-clock time (in seconds) of SGA-RMSProp and SGD, reported as (mean, standard deviation) over three instances.

Problems	$B = 50$		$B = 1000$	
	SGA-RMSProp	SGD	SGA-RMSprop	SGD
(ED, 20, 0.7)	<b>(0.983, 0.002)</b>	(1.012, 0.005)	(0.992, 0.027)	<b>(0.948, 0.003)</b>
(ED, 20, 0.2)	<b>(0.970, 0.006)</b>	(0.990, 0.004)	(1.031, 0.013)	<b>(0.945, 0.005)</b>
(ED, 50, 0.7)	<b>(2.238, 0.028)</b>	(2.297, 0.040)	(2.460, 0.053)	<b>(2.356, 0.014)</b>
(ED, 50, 0.2)	(2.266, 0.027)	<b>(2.193, 0.009)</b>	(2.527, 0.049)	<b>(2.365, 0.013)</b>
(ED, 100, 0.7)	<b>(4.273, 0.002)</b>	(4.310, 0.014)	(4.759, 0.086)	<b>(4.658, 0.031)</b>
(ED, 100, 0.2)	(4.689, 0.106)	<b>(4.210, 0.068)</b>	(4.913, 0.134)	<b>(4.710, 0.004)</b>
(AD, 20, 1)	<b>(1.310, 0.068)</b>	(1.323, 0.082)	(0.808, 0.064)	<b>(0.771, 0.033)</b>
(AD, 20, 2)	(1.223, 0.010)	<b>(1.219, 0.023)</b>	(0.887, 0.013)	<b>(0.804, 0.011)</b>
(AD, 50, 1)	<b>(2.863, 0.377)</b>	(2.972, 0.380)	<b>(1.825, 0.201)</b>	(1.865, 0.090)
(AD, 50, 2)	<b>(2.658, 0.048)</b>	(2.798, 0.052)	(2.140, 0.040)	<b>(1.951, 0.041)</b>
(AD, 100, 1)	<b>(4.865, 0.499)</b>	(5.004, 0.657)	<b>(2.992, 0.515)</b>	(3.131, 0.294)
(AD, 100, 2)	<b>(4.876, 0.376)</b>	(5.133, 0.430)	(3.751, 0.291)	<b>(3.732, 0.239)</b>

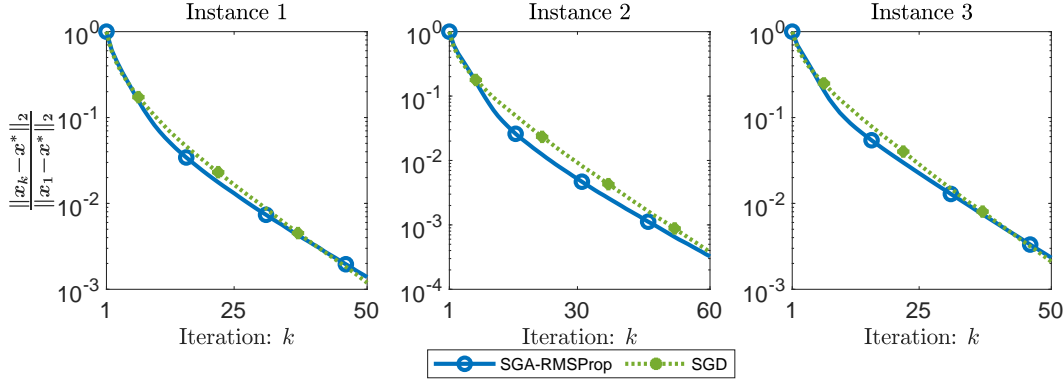


Figure 2. Comparison of SGA-RMSProp and SGD on (AD, 20, 1) with  $B = 1000$ .

RMSProp and switching to SGD at an appropriate time may improve the performance of the algorithm. Based on this idea, we conduct the next experiment.

**5.2. RMSP2SGD.** We introduce RMSP2SGD as follows, a method that enhances SGA-RMSProp by adding an adaptive switching rule to SGD. The rule is that if  $\beta_k$  equals 1 for five consecutive times, then the algorithm switches to SGD. As a result, it benefits from the initial rate of SGA-RMSProp and SGD's faster convergence in the later stages.

We compare the wall-clock time of the three algorithms on the instances generated in subsection 5.1.1 with  $B = 1000$  and the parameters discussed in subsection 5.1.

Table 8 indicates that RMSP2SGD improves the performance of SGA-RMSProp on LLSP. It is also generally faster than SGD with respect to the wall-clock time, except in (ED, 20, 0.2). In addition, tuning the parameters may improve the initial convergence rate of SGA-RMSProp, thereby making RMSP2SGD faster. This will be one of the future work.

**Algorithm 5.1** RMSP2SGD

**Input:** Step size  $\{\eta_k\}$ , adjusted level  $\varepsilon$ , lower and upper bounds  $\underline{u}$ ,  $\bar{u}$ , initial value  $\mathbf{x}_1$ .

```

1:  $\mathbf{u}_0 = \left(\frac{1}{\underline{u}^2}, \dots, \frac{1}{\bar{u}^2}\right)^\top$ .
2: for  $k = 1, \dots, K$  do
3:   Sample a stochastic gradient  $\mathbf{g}_k$ .
4:    $\beta_k = \beta\text{-Selection}(\mathbf{g}_k, \mathbf{u}_{k-1}, \varepsilon, \underline{u}, \bar{u})$ .
5:   if  $\beta_k$  equals 1 for five consecutive times then
6:     Break and switch to SGD.
7:   end if
8:    $\mathbf{u}_k = \beta_k \mathbf{u}_{k-1} + (1 - \beta_k) \mathbf{g}_k^2$ .
9:    $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\eta_k}{\sqrt{\mathbf{u}_k}} \circ \mathbf{g}_k$ .
10: end for
```

**Output:** The optimal point calculated by the algorithm  $\mathbf{x}_{K+1}$ .

**Table 8**

Wall-clock time (in seconds) of the three algorithms with batch size  $B = 1000$ , reported as (mean, standard deviation) over three instances.

Problems	RMSP2SGD	SGA-RMSProp	SGD
(ED, 20, 0.7)	<b>(0.938, 0.003)</b>	(0.992, 0.027)	(0.948, 0.003)
(ED, 20, 0.2)	(0.946, 0.001)	(1.031, 0.013)	<b>(0.945, 0.005)</b>
(ED, 50, 0.7)	<b>(2.325, 0.018)</b>	(2.460, 0.053)	(2.356, 0.014)
(ED, 50, 0.2)	<b>(2.343, 0.020)</b>	(2.527, 0.049)	(2.365, 0.013)
(ED, 100, 0.7)	<b>(4.611, 0.029)</b>	(4.759, 0.086)	(4.658, 0.031)
(ED, 100, 0.2)	<b>(4.666, 0.018)</b>	(4.913, 0.134)	(4.710, 0.004)
(AD, 20, 1.0)	<b>(0.754, 0.121)</b>	(0.808, 0.064)	(0.771, 0.033)
(AD, 20, 2.0)	<b>(0.768, 0.047)</b>	(0.887, 0.013)	(0.804, 0.011)
(AD, 50, 1.0)	<b>(1.747, 0.170)</b>	(1.825, 0.201)	(1.865, 0.090)
(AD, 50, 2.0)	<b>(1.948, 0.043)</b>	(2.140, 0.040)	(1.951, 0.041)
(AD, 100, 1.0)	<b>(2.977, 0.357)</b>	(2.992, 0.515)	(3.131, 0.294)
(AD, 100, 2.0)	<b>(3.621, 0.320)</b>	(3.751, 0.291)	(3.732, 0.239)

**5.3. Experiments for inconsistent LLSP.** In this subsection, we conduct experiments for the inconsistent LLSP. We generate the matrix  $\mathbf{A}$  as described in subsection 5.1.1 and let  $\tilde{\mathbf{x}} \in \mathbb{R}^d$  have independent standard normal entries,  $\mathbf{b} = \mathbf{A}\tilde{\mathbf{x}} + \boldsymbol{\tau}$ , where  $\boldsymbol{\tau}$  represents a perturbation uniformly randomly drawn from a sphere with a radius of  $10^{-3}$ . We fix the problem at (AD, 20, 1) and present the results of the three instances. As shown in Theorem 4.7, SGA-RMSProp converges to a region of confusion in the inconsistent case, so we do not use the previous convergence criteria in this experiment. Instead, we set the maximum number of iterations to 500 and present the performances of the three algorithms in Figure 3.

The results indicate that all three algorithms converge to the region of confusion. The inset plot of each figure shows that SGA-RMSProp is still faster than SGD in the early stages, and RMSP2SGD is the fastest among three algorithms, which is similar to the consistent case.

**5.4. Real data.** We test the performance of SGA-RMSProp on real data using the YearPredictionMSD dataset from UCI machine learning data repository [1]. Previous studies used

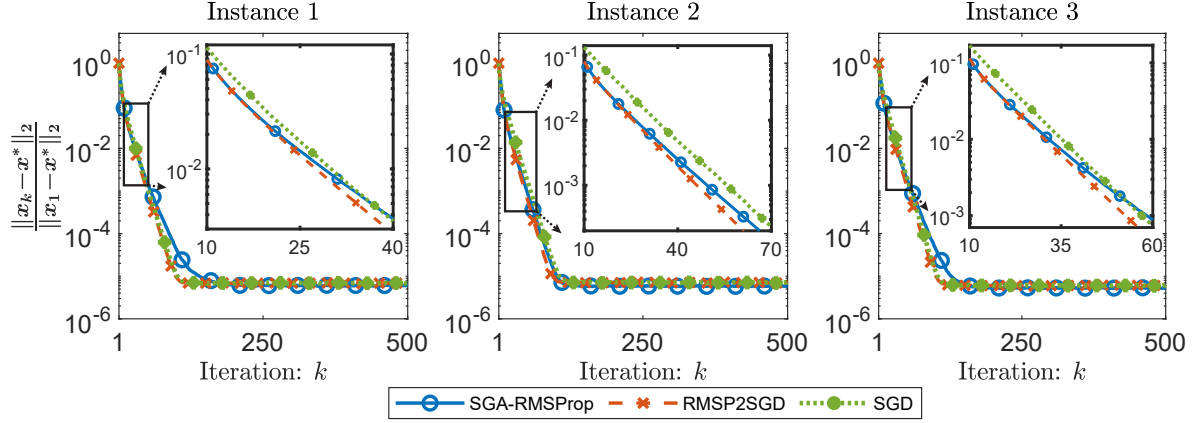


Figure 3. Comparison of the three algorithms for the inconsistent case.

this dataset to predict song release years from timbre features through linear regression, and finally formulated it as LLSP [4]. The dataset contains  $n = 463715$  records and  $d = 90$  timbre features. We apply mean-centered standardization to prepare the data for linear regression. Next, we set  $\varepsilon$  to  $\varepsilon_2$  and use a batch size  $B = 2000$ , with all other parameters remaining the same as in the previous large batch size case.

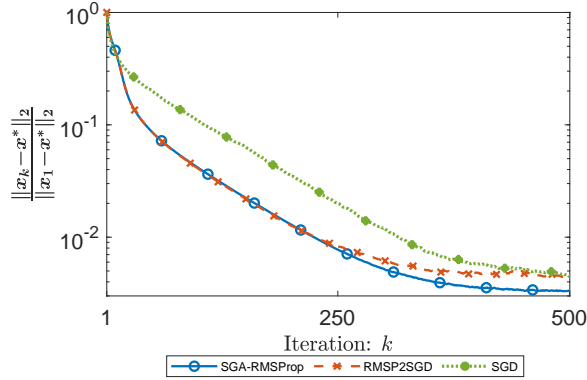


Figure 4. Comparison of the three algorithms on YearPredictionMSD.

The results are presented in Figure 4, indicating that the three algorithms converge to regions of confusion and **SGA-RMSProp** achieves a smaller region. The plot also shows that **SGA-RMSProp** is faster in the early stages. However, SGD can not surpass **SGA-RMSProp** on this dataset, possibly due to entering the region of confusion too early.

**6. Concluding remarks.** In this paper, we propose a method named stable gradient-adjusted RMSProp (**SGA-RMSProp**), which uses a parameter called adjusted level to adaptively select the discounting factor and stably control the adjustment applied to the stochastic gradient. Theoretically, the R-linear convergence rate of **SGA-RMSProp** is established on LLSP and a range of the adjusted level is provided to ensure this rate. Numerical experiments show that **SGA-RMSProp** outperforms RMSProp, and achieves better performance than SGD when the batch size is small. Moreover, adaptively switching from **SGA-RMSProp** to SGD



generally improves the performance compared to using SGD alone.

Future studies might focus on analyzing the generally faster initial convergence rate of **SGA-RMSProp** compared with SGD on LLSP from a theoretical perspective. For example, whether a tighter upper bound less than 1 for  $G(\gamma_1, \dots, \gamma_K)$  in (4.4) can be obtained? Other topics include exploring whether our method can be extended to general Adam-style algorithms or general functions.

## REFERENCES

- [1] T. BERTIN-MAHIEUX, *Year Prediction MSD*. UCI Machine Learning Repository, 2011, <https://doi.org/10.24432/C50K61>.
- [2] D. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, Belmont, 1999.
- [3] D. BERTSEKAS, *Convex Optimization Algorithms*, Athena Scientific, Belmont, 2015.
- [4] S. BHARDWAJ, R. R. CURTIN, M. EDEL, Y. MENTEKIDIS, AND C. SANDERSON, *Ensmallen: A flexible C++ library for efficient function optimization*, in Workshop on Systems for ML and Open Source Software at NeurIPS, 2018.
- [5] R. BOLLAPRAGADA, T. CHEN, AND R. WARD, *On the fast convergence of minibatch heavy ball momentum*, IMA J. Numer. Anal., drae033 (2024), <https://doi.org/10.1093/imanum/drae033>.
- [6] A. CASSIOLI, A. CHIAVAIOLI, C. MANES, AND M. SCIANDRONE, *An incremental least squares algorithm for large scale linear classification*, Eur. J. Oper. Res., 224 (2013), pp. 560–565, <https://doi.org/10.1016/j.ejor.2012.09.004>.
- [7] C. CHEN, L. SHEN, F. ZOU, AND W. LIU, *Towards practical Adam: Non-convexity, convergence theory, and mini-batch acceleration*, J. Mach. Learn. Res., 23 (2022), pp. 1–47.
- [8] R. DURRETT, *Probability: Theory and Examples*, Cambridge university press, Cambridge, 2019.
- [9] M. L. EATON, *Multivariate Statistics: A Vector Space Approach*, Institute of Mathematical Statistics, Beachwood, 2007.
- [10] E. GORBUNOV, F. HANZELY, AND P. RICHTÁRIK, *A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent*, in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 680–690.
- [11] Z. GUO, Y. XU, W. YIN, R. JIN, AND T. YANG, *A novel convergence analysis for algorithms of the Adam family and beyond*, preprint, [arXiv:2104.14840](https://arxiv.org/abs/2104.14840), 2021.
- [12] D. HAN, Y. SU, AND J. XIE, *Randomized Douglas-Rachford methods for linear systems: Improved accuracy and efficiency*, SIAM J. Optim., 34 (2024), pp. 1045–1070, <https://doi.org/10.1137/23M1567503>.
- [13] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge university press, Cambridge, 1994.
- [14] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge university press, Cambridge, 2012.
- [15] A. KHALED, O. SEBBOUH, N. LOIZOU, R. M. GOWER, AND P. RICHTÁRIK, *Unified analysis of stochastic gradient methods for composite convex and smooth optimization*, J. Optim. Theory Appl., 199 (2023), pp. 499–540, <https://doi.org/10.1007/s10957-023-02297-y>.
- [16] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, preprint, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980), 2014.
- [17] J. LACOTTE AND M. PILANCI, *Fast convex quadratic optimization solvers with adaptive sketching-based preconditioners*, preprint, [arXiv:2104.14101](https://arxiv.org/abs/2104.14101), 2021.
- [18] T. L. LAI AND H. XING, *Statistical models and methods for financial markets*, Springer, New York, 2008.
- [19] G. LAN AND Y. ZHOU, *An optimal randomized incremental gradient method*, Math. Program., 171 (2018), pp. 167–215, <https://doi.org/10.1007/s10107-017-1173-0>.
- [20] H. LI AND Z. LIN, *On the  $\mathcal{O}\left(\frac{\sqrt{d}}{T^{1/4}}\right)$  convergence rate of RMSProp and its momentum extension measured by  $l_1$  norm: Better dependence on the dimension*, preprint, [arXiv:2402.00389](https://arxiv.org/abs/2402.00389), 2024.
- [21] S. LING AND T. STROHMER, *Self-calibration and bilinear inverse problems via linear least squares*, SIAM J. Imag. Sci., 11 (2018), pp. 252–292, <https://doi.org/10.1137/16M1103634>.
- [22] J. LIU, D. XU, H. ZHANG, AND D. MANDIC, *On hyper-parameter selection for guaranteed convergence of RMSProp*, Cognit. Neurodyn., (2022), <https://doi.org/10.1007/s11571-022-09845-8>.



- [23] Y. LIU, F. FENG, AND W. YIN, *Acceleration of SVRG and Katyusha X by inexact preconditioning*, in Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 4003–4012.
- [24] N. LOIZOU AND P. RICHTÁRIK, *Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods*, Comput. Optim. Appl., 77 (2020), pp. 653–710, <https://doi.org/10.1007/s10589-020-00220-z>.
- [25] C. MA, L. WU, AND W. E, *A qualitative study of the dynamic behavior for adaptive gradient algorithms*, in Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference, PMLR, 2022, pp. 671–692.
- [26] J. D. MOORMAN, T. K. TU, D. MOLITOR, AND D. NEEDELL, *Randomized Kaczmarz with averaging*, BIT Numer. Math., 61 (2021), pp. 337–359, <https://doi.org/10.1007/s10543-020-00824-1>.
- [27] E. MOULINES AND F. BACH, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, in Advances in Neural Information Processing Systems, 2011, pp. 451–459.
- [28] I. NASEEM, R. TOGNERI, AND M. BENNAMOUN, *Linear regression for face recognition*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 2106–2112, <https://doi.org/10.1109/TPAMI.2010.128>.
- [29] D. NEEDELL, N. SREBRO, AND R. WARD, *Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm*, Math. Program., 155 (2016), pp. 549–573, <https://doi.org/10.1007/s10107-015-0864-7>.
- [30] A. PASZKE, *Pytorch: An imperative style, high-performance deep learning library*, preprint [arXiv:1912.01703](https://arxiv.org/abs/1912.01703), 2019.
- [31] P. RICHTÁRIK AND M. TAKÁC, *Stochastic reformulations of linear systems: Algorithms and convergence theory*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 487–524, <https://doi.org/10.1137/18M1179249>.
- [32] Z. STRAKOŠ, *On the real convergence rate of the conjugate gradient method*, Linear Algebra Appl., 154 (1991), pp. 535–549, [https://doi.org/10.1016/0024-3795\(91\)90393-B](https://doi.org/10.1016/0024-3795(91)90393-B).
- [33] T. STROHMER AND R. VERSHYNIN, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), pp. 262–278, <https://doi.org/10.1007/s00041-008-9030-4>.
- [34] T. TIELEMAN AND G. HINTON, *Lecture 6.5 RMSProp: Divide the gradient by a running average of its recent magnitude*, COURSE: Neural networks for machine learning, 4 (2012), pp. 26–31.
- [35] J. A. TROPP, *An Introduction to Matrix Concentration Inequalities*, now Publishers Inc., Hanover, 2015.
- [36] X. WANG, M. JOHANSSON, AND T. ZHANG, *Generalized Polyak step size for first order optimization with momentum*, in Proceedings of the 40th International Conference on Machine Learning, PMLR, 2023, pp. 35836–35863.
- [37] R. WARD, X. WU, AND L. BOTTOU, *Adagrad stepsizes: Sharp convergence over nonconvex landscapes*, J. Mach. Learn. Res., 21 (2020), pp. 1–30.
- [38] X. WU, Y. XIE, S. S. DU, AND R. WARD, *Adaloss: A computationally-efficient and provably convergent adaptive gradient method*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 8691–8699, <https://doi.org/10.1609/aaai.v36i8.20848>.
- [39] Y. XIE, X. WU, AND R. WARD, *Linear convergence of adaptive stochastic gradient descent*, in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 1475–1485.
- [40] Y. ZENG, D. HAN, Y. SU, AND J. XIE, *On adaptive stochastic heavy ball momentum for solving linear systems*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 1259–1286, <https://doi.org/10.1137/23M1575883>.
- [41] Z. ZHANG, Y. LI, L. LI, Z. LI, AND S. LIU, *Multiple linear regression for high efficiency video intra coding*, in 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2019, pp. 1832–1836.
- [42] F. ZOU, L. SHEN, Z. JIE, W. ZHANG, AND W. LIU, *A sufficient condition for convergences of Adam and RMSProp*, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Los Alamitos, 2019, IEEE Computer Society, pp. 11119–11127.