# Riemannian Stochastic Proximal Gradient Methods for Nonsmooth Optimization over the Stiefel Manifold

Bokun Wang* Shiqian Ma† Lingzhou Xue‡

March 22, 2020

**Abstract**

Riemannian optimization has drawn a lot of attention due to its wide applications in practice. Riemannian stochastic first-order algorithms have been studied in the literature to solve large-scale machine learning problems over Riemannian manifolds. However, most of the existing Riemannian stochastic algorithms require the objective function to be differentiable, and they do not apply to the case where the objective function is nonsmooth. In this paper, we present two Riemannian stochastic proximal gradient methods for minimizing nonsmooth function over the Stiefel manifold. The two methods, named R-ProxSGD and R-ProxSPB, are generalizations of proximal SGD and proximal SpiderBoost in Euclidean setting to the Riemannian setting. Analysis on the incremental first-order oracle (IFO) complexity of the proposed algorithms is provided. Specifically, the R-ProxSPB algorithm finds an $\epsilon$-stationary point with $\mathcal{O}(\epsilon^{-3})$ IFOs in the online case, and $\mathcal{O}(n+\sqrt{n}\epsilon^{-3})$ IFOs in the finite-sum case with $n$ being the number of summands in the objective. Experimental results on online sparse PCA and robust low-rank matrix completion show that our proposed methods significantly outperform the existing methods that uses Riemannian subgradient information.

## 1 Introduction

We consider the following composite optimization problem over the Stiefel manifold $\mathcal{M} := \mathrm{St}(d,r) = \{X \in \mathbb{R}^{d\times r} \mid X^\top X = I_r\}$:

$$\min_{X\in\mathcal{M}} F(X) := f(X) + h(X), \tag{1}$$

where $f(X)$ takes one of the following two forms:

- Online case:
$$f(X) := \mathbb{E}_\pi[f(X;\pi)], \tag{2}$$
where $\mathbb{E}$ is the expectation with respect to the random variable $\pi$.

- Finite-sum case:
$$f(X) := \frac{1}{n}\sum_{i=1}^{n} f_i(X), \tag{3}$$
where $n$ denotes the number of data and is assumed to be extremely large.

*Department of Computer Science, University of California, Davis
†Department of Mathematics, University of California, Davis
‡Department of Statistics, Penn State University

Throughout this paper, we assume that $f(\cdot;\pi)$, $f_i(\cdot)$ and thus $f(\cdot)$ are all smooth, $h$ is convex and possibly nonsmooth. Here the smoothness and convexity are interpreted when the function in question is considered as a function in the ambient Euclidean space. Note that since (2) involves an expectation, and (3) involves extremely large $n$, we assume that the full gradient information of $f$ is not available and only stochastic estimators to the gradient of $f$ can be obtained.

Problem (1) with $f$ being (2) and (3) appear frequently in machine learning applications. In the online case (2), $f(X,\pi)$ denotes the loss function corresponding to data $\pi$; and in the finite-sum case (3), $f_i(X)$ denotes the loss function corresponding to the $i$-th sample data. Function $h$ is usually a regularizer that can promote certain desired structure of the solution. For example, letting $h(X) = \|X\|_1 := \sum_{ij} |X_{ij}|$ serves the purpose of promoting the sparsity of solution $X$.

One important application of (1) in the online case is the online sparse PCA, which can be cast as

$$\min_{X} \quad \mathbb{E}_{Z\in\mathcal{D}}[\|Z - XX^\top Z\|_2^2] + \mu\|X\|_1, \text{s.t.}, X \in \mathcal{M}, \tag{4}$$

where $\mu > 0$ is a weighting parameter, $\mathcal{D}$ denotes the distribution of the random online data $Z$, and the $\ell_1$ norm is used to promote the sparsity of the eigenvectors. In this case, $r$ is the desired number of principal components. Although PCA and sparse PCA have been studied extensively in the literature, studies for online sparse PCA, i.e., sparse PCA with streaming data, seem to be very limited [31, 26]. In this paper, we propose efficient stochastic Riemannian algorithms for solving this important application.

### 1.1 Related Works

Riemannian optimization has been a very active research area in the last decade, due to its wide applications in machine learning, signal processing, statistics and so on. The monograph by Absil et.al. [1] studied this topic in depth. However, most existing methods for Riemannian optimization, including the ones presented in [1], require that the objective function to be smooth in order to compute Riemannian gradient information. Algorithms for solving Riemannian optimization with nonsmooth objective functions are very limited in the literature. Moreover, when the loss function $f$ takes the expectation or finite-sum form as in (2) and (3), stochastic algorithms are usually in demand because we have only access to noisy stochastic gradients of $f$ instead of the full gradient. When the nonsmooth regularizer $h$ vanishes, that is, when (1) reduces to a smooth problem with $f$ given by (2) or (3), there exist stochastic algorithms for solving it. In particular, existing Riemannian algorithms such as R-SGD [2], R-SVRG [33], R-SRG [17] and R-SPIDER [35, 34] are amenable to solving these two smooth problems. Among these algorithms, R-SVRG, R-SRG and R-SPIDER all utilize the variance reduction techniques [16, 9] to improve the convergence rate of R-SGD. On the other hand, when the nonsmooth regularizer $h$ presents but the manifold constraint vanishes in (1), i.e., when $\mathcal{M}$ is the Euclidean space, there exist stochastic proximal gradient algorithms for solving these unconstrained problems in Euclidean space. Popular methods include ProxSGD [25], ProxSVRG [29], ProxSARAH [24] and ProxSpiderBoost [27]. However, to the best of our knowledge, when both nonsmooth regularizer $h$ and manifold constraint $X \in \mathcal{M}$ present as in (1), there is no stochastic algorithm that can solve them. In this paper, we close this gap by proposing two stochastic algorithms, namely R-ProxSGD and R-ProxSPB, for solving (1) with $f$ being (2) or (3), i.e., Riemannian optimization with nonsmooth objectives. Our algorithms are inspired by the ManPG algorithm that is recently proposed by Chen et.al. [8] for solving the nonsmooth Riemannian optimization problem (1). ManPG assumes that the full gradient of $f$ can be obtained, and thus it is a deterministic algorithm, while our R-ProxSGD and R-ProxSPB are the first stochastic algorithms for solving (1) without using subgradient information. Recently, Li et.al.

[19] showed that when the objective function is weakly convex, Riemannian stochastic subgradient Method (R-Subgrad) has $\mathcal{O}(\epsilon^{-4})$ iteration complexity for obtaining an $\epsilon$-stationary point.

| Objective | Euclidean | Riemannian |
|---|---|---|
| Smooth | SGD [21] | R-SGD [2] |
| | SVRG [16] | R-SVRG [33] |
| | SARAH [23] | R-SRG [17] |
| | SPIDER [12] | R-SPIDER [35, 34] |
| | SpiderBoost [27] | **R-SpiderBoost** (ours) |
| Non-smooth | ProxSGD [25] | **R-ProxSGD** (ours) |
| | ProxSVRG [29] | N/A |
| | ProxSARAH [24] | N/A |
| | ProxSpiderBoost [27] | **R-ProxSPB** (ours) |

Table 1: Summary of existing methods and our methods in Euclidean and Riemannian settings.

| Algorithms | Step size | Finite-sum | Online |
|---|---|---|---|
| ManPG [8] | constant | $\mathcal{O}(n\epsilon^{-2})$ | N/A |
| R-ProxSGD | diminishing | N/A | $\mathcal{O}(\epsilon^{-6})$ |
| R-ProxSPB | constant | $\mathcal{O}(n+\sqrt{n}\epsilon^{-2})$ | $\mathcal{O}(\epsilon^{-3})$ |

Table 2: Comparison of IFO complexity for nonsmooth Riemannian optimization methods over the Stiefel manifold.

### 1.2 Our Contributions

The contributions of this paper lie in several folds.

(i) First, we propose two stochastic algorithms for solving (1). These two algorithms, named R-ProxSGD and R-ProxSPB, are Riemannian generalizations of their counterparts in the Eucldiean setting: ManPG [8] and ProxSpiderBoost [27]. On the other hand, they can also be viewed as generalizations of their smooth counterparts, R-SGD and R-SpiderBoost, to the nonsmooth case. However, we emphasize here that none of these generalizations are straightforward. In Table 1 we give a summary of existing methods and our proposed methods in different cases: the objective is smooth or nonsmooth and the constraint is Riemannian manifold or Euclidean space. Note that when the nonsmooth function $h$ vanishes, our R-ProxSPB reduces to a Riemannian SpiderBoost algorithm (R-SpiderBoost) that solves Riemannian optimization with smooth objective. It seems that R-SpiderBoost is also new in the literature.

(ii) Second, we prove the convergence of the proposed two algorithms and analyze their incremental first-order oracle (IFO) complexity results. Specifically, we analyze the IFO complexity of R-ProxSGD for the online setting problem, i.e., (1) with $f$ being (2); and R-ProxSPB for both the online setting problem and the finite-sum setting problem, i.e., (1) with $f$ being (3). In Table 2 we summarize the IFO complexity results of our proposed algorithms and the existing ManPG algorithm, as they are the only algorithms that can solve the nonsmooth Riemannian optimization problem (1) with known IFO complexity results.

(iii) Third, we conduct extensive numerical experiments for solving online sparse PCA (4) and robust low-rank matrix completion problems to demonstrate the advantages of the proposed methods.

**Organization.** The rest of the paper is organized as follows. Section 2 introduces the necessary notation and assumptions. Our new algorithms and their convergence and complexity results are presented in Section 3. The experimental results are reported in Section 4. Finally, we make some concluding remarks in Section 5. The detailed proofs of the theorems and lemmas are provided in the appendix.

## 2 Preliminaries

In this work, we consider the Riemannian submanifold $(\mathcal{M}, \mathfrak{g})$ where $\mathcal{M}$ is the Stiefel manifold and $\mathfrak{g}$ is the Riemannian metric on $\mathcal{M}$ that is induced from the Euclidean inner product. For smooth function $f$, we use $\mathrm{grad} f(X)$ to denote the full Riemannian gradient of $f$ at $X$, and $\nabla f(\mathbf{X})$ represents the full Euclidean gradient of $f$ at $\mathbf{X}$. With an abuse of notation, when there is no ambiguity, we use $f_i$ to denote the component function in the online case (2), i.e., $f_i(\mathbf{X}) := f(\mathbf{X}, \pi_i)$, though it is still used as a component function in the finite-sum case (3). For a mini-batch set $\mathcal{S}$, $\nabla f_{\mathcal{S}}(X) := \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \nabla f_i(X)$ denotes the stochastic Euclidean gradient estimated on $\mathcal{S}$. We use $\mathcal{F}_t$ to denote all randomness occurred up to (include) the $t$-th iteration of any algorithm. When there is no ambiguity, we use $\|\mathbf{a}\|$ to denote the Frobenius norm when $\mathbf{a}$ is a matrix and the Euclidean norm when $\mathbf{a}$ is a vector.

A classical geometric concept in the study of manifolds is that of an exponential mapping, which defines a geodesic curve on the manifold. However, the exponential mapping is difficult to compute in general. The concept of a retraction [1], which is a first-order approximation of the exponential mapping and can be more amenable to computation, is given as follows.

**Definition 1.** *[1, Definition 4.1.1] A retraction on a differentiable manifold $\mathcal{M}$ is a smooth mapping* Retr *from the tangent bundle* $\mathrm{T}\mathcal{M}$ *onto* $\mathcal{M}$ *satisfying the following two conditions (here* $\mathrm{Retr}_X$ *denotes the restriction of* Retr *onto* $\mathrm{T}_X\mathcal{M}$*):*

1. $\mathrm{Retr}_X(0) = X, \forall X \in \mathcal{M}$*, where* $0$ *denotes the zero element of* $\mathrm{T}_X\mathcal{M}$*.*
2. *For any* $X \in \mathcal{M}$*, it holds that*

$$\lim_{\mathrm{T}_X\mathcal{M}\ni\xi\to 0} \frac{\|\mathrm{Retr}_X(\xi) - (X+\xi)\|}{\|\xi\|} = 0.$$

**Remark 1.** *Here and thereafter, when we talk about the summation $X + \xi$, we always treat $X$ and $\xi$ as elements in the ambient Euclidean space so that their sum is well defined. The second condition in Definition 1 ensures that* $\mathrm{Retr}_X(\xi) = X + \xi + \mathcal{O}(\|\xi\|^2)$ *and* $D\mathrm{Retr}_X(0) = \mathrm{Id}$*, where* $D\mathrm{Retr}_X$ *is the differential of* $\mathrm{Retr}_X$ *and* Id *denotes the identity mapping. For more details about retraction, we refer the reader to [1, 5] and the references therein.*

The retraction onto the Euclidean space is simply the identity mapping; i.e., $\mathrm{Retr}_X(\xi) = X + \xi$. For the Stiefel manifold $\mathrm{St}(n, r)$, common retractions include the exponential mapping [11]

$$\mathrm{Retr}_X^{\exp}(t\xi) = [X, Q] \exp\left(t \begin{bmatrix} -X^\top\xi & -R^\top \\ R & 0 \end{bmatrix}\right) \begin{bmatrix} I_r \\ 0 \end{bmatrix},$$

where $QR = -(I_d - XX^\top)\xi$ is the unique QR factorization; the polar decomposition

$$\mathrm{Retr}_X^{\mathrm{polar}}(\xi) = (X+\xi)(I_r + \xi^\top\xi)^{-1/2};$$

the QR decomposition

$$\mathrm{Retr}_X^{\mathrm{QR}}(\xi) = \mathrm{qf}(X+\xi),$$

where $\mathrm{qf}(A)$ is the $Q$ factor of the QR factorization of $A$; the Cayley transformation [28]

$$\mathrm{Retr}_X^{\mathrm{cayley}}(\xi) = \left(I_d - \frac{1}{2}W(\xi)\right)^{-1}\left(I_d + \frac{1}{2}W(\xi)\right)X,$$

where $W(\xi) = (I_d - \frac{1}{2}XX^\top)\xi X^\top - X\xi^\top(I_d - \frac{1}{2}XX^\top)$.

In this paper, we adopt the assumption that the retraction that we use is invertible, the same as what is assumed in existing works [17, 35]. We use $\Gamma_X^Y$ to denote the vector transport from $X$ to $Y$ satisfying $\mathrm{Retr}_X(\xi) = Y$. Vector transport $\Gamma : \mathrm{T}\mathcal{M} \bigoplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M}$, $(\xi, \zeta) \mapsto \Gamma_X^Y(\zeta)$ is associated with the retraction Retr, where $\xi, \zeta \in \mathrm{T}_X\mathcal{M}$.

The following assumptions regarding the retraction and vector transport are necessary to our analysis.

**Assumption 1.** *(i) (see [17]). All of the iterates $\{X_t\}_{t=1}^{T+1}$ are in a totally retractive neighborhood $\mathcal{U} \subset \mathcal{M}$ of an optimum $X^*$: $\{\mathrm{Retr}_{X_t}(\xi_t)\} \in \mathcal{U}$ with $X_{t+1} = \mathrm{Retr}_{X_t}(\zeta_t)$, $\zeta_t \in \mathrm{T}_X\mathcal{M}$.*

*(ii) (see [17]). Suppose that $\mathrm{Exp}_X : \mathrm{T}_X\mathcal{M} \to \mathcal{M}$ denotes the exponential mapping and $\mathrm{Exp}_X^{-1} : \mathcal{M} \to \mathrm{T}_X\mathcal{M}$ is its inverse mapping. There exist $c_R, c_E > 0$ such that $\|\mathrm{Exp}_X^{-1}(Y) - \mathrm{Retr}_X^{-1}(Y)\| \leq c_R\|\mathrm{Retr}_X^{-1}(Y)\|$, $\forall X, Y \in \mathcal{U}$ and $\|\mathrm{Retr}_X^{-1}(Y)\| \leq c_E\|\xi\|$ if $\mathrm{Retr}_X(\xi) = Y$.*

*(iii) (see [5]). For all $X \in \mathcal{M}$ and $\xi \in \mathrm{T}_X\mathcal{M}$, there exist constants $M_1 > 0$ and $M_2 > 0$ such that the following two inequalities hold:*

$$\|\mathrm{Retr}_X(\xi) - X\| \leq M_1\|\xi\| \tag{5}$$

$$\|\mathrm{Retr}_X(\xi) - (X+\xi)\| \leq M_2\|\xi\|^2. \tag{6}$$

**Assumption 2.** *(see [17]). The vector transport is isometric on the manifold $\mathcal{M}$, i.e., $\|\Gamma_X^Y(\zeta)\| = \|\zeta\|$ for $X, Y \in \mathcal{M}$, $\xi, \zeta \in \mathrm{T}_X\mathcal{M}$ and $\mathrm{Retr}_X(\xi) = Y$.*

Besides, we impose some assumptions on $f(X)$ and its first-order oracle, which are also required in previous work on smooth Riemannian optimization with retraction and vector transport [17, 35].

**Assumption 3** (Upper-bounded Hessian of $f$)**.** *Every individual loss $f_i(X)$ is twice continuously differentiable and the individual Hessian of every $f_i(X)$ is bounded as $\|\nabla^2 f_i(X)\| \leq L_H$. $f(X)$ has upper-bounded Hessian in $\mathcal{U} \in \mathcal{M}$ with respect to the retraction $\mathrm{Retr}_X(\cdot)$ if there exists $L_R > 0$ such that $\frac{d^2 f(\mathrm{Retr}_X(t\xi))}{dt^2} \leq L_R$ for all $X \in \mathcal{U}, \xi \in \mathrm{T}_X\mathcal{M}$ with $\|\xi\| = 1$ and all $t$ such that $\mathrm{Retr}_X(\tau\xi) \in \mathcal{U}$ for all $\tau \in [0, t]$.*

**Assumption 4** (Bounded variance)**.** *Stochastic gradient oracle of every individual loss $f_i(X)$ is bounded $\|\nabla f_i(X)\| \leq G$ and its variance is also bounded $\mathbb{E}_i[\|\nabla f_i(X) - \nabla f(X)\|^2] \leq \sigma^2$.*

Moreover, we make the following assumption on the regularization term $h(X)$.

**Assumption 5.** *The regularization function $h$ is convex and $L_h$-Lipschitz continuous, i.e., $\|h(X) - h(Y)\| \leq L_h\|X - Y\|$, $\forall X, Y \in \mathcal{M}$.*

We now give the definition of the stationary point of problem (1), which is standard in the literature, see [32, 8].

**Definition 2** (Stationary point)**.** *$X \in \mathcal{M}$ is a stationary point of* (1) *if it satisfies:*

$$0 \in \hat{\partial} F(X) := \nabla f(X) + \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}} \partial h(X), \tag{7}$$

where $\hat{\partial} F(X)$ is the generalized Clarke subdifferential at $X$ (see Definition 5 in Appendix).

The computational costs of the algorithms are evaluated in terms of IFO complexity.

**Definition 3.** *An IFO takes an index $i \in \{1, \ldots, n\}$ and returns $(f_i(X), \nabla f_i(X))$ for the finite-sum case* (3)*, or $(f(X, \pi_i), \nabla_X f(X, \pi_i))$ for the online case* (2)*.*

# 3 Riemannian Stochastic Proximal Gradient Methods

In this section, we introduce our Riemannian stochastic proximal gradient algorithms and provide their non-asymptotic convergence results. Proofs of the theorems are provided in the appendix.

## 3.1 The Main Framework

The main framework of our Riemnannian stochastic proximal gradient algorithms is inspired by the ManPG algorithm [8]. The ManPG algorithm aims to solve the nonsmooth Riemannian optimization problem (1) by assuming that the full gradient of $f$ can be accessed. Therefore, it is a deterministic algorithm. ManPG is a generalization of the proximal gradient method from Euclidean setting to the Riemannian setting. The proximal gradient method for solving $\min_X F(X) := f(X) + h(X)$ in the Euclidean setting generates the iterates as follows:

$$X_{t+1} := \mathrm{argmin}_Y f(X_t) + \langle \nabla f(X_t), Y - X_t \rangle + \frac{1}{2\gamma} \|Y - X_t\|^2 + h(Y). \tag{8}$$

In other words, one minimizes the quadratic model $Y \mapsto f(X_t) + \langle \nabla f(X_t), Y - X_t \rangle + \frac{1}{2t} \|Y - X_t\|^2 + h(Y)$ of $F$ at $X_t$ in the $t$-th iteration, where $\gamma > 0$ is a parameter that can be regarded as the stepsize. It is known that the quadratic model is an upper bound of $F$ when $\gamma \leq 1/L$, where $L$ is the Lipschitz constant of $\nabla f$. The subproblem (8) corresponds to the proximal mapping of $h$ and the efficiency of the proximal gradient method relies on the assumption that (8) is easy to solve. For (1), in order to deal with the manifold constraint, one needs to ensure that the descent direction lies in the tangent space. This motivates the following subproblem for finding the descent direction $\xi_t$ in the $t$-th iteration:

$$\begin{aligned} \xi_t = \mathrm{argmin}_\xi := \langle \nabla f(X_t), \xi \rangle + \tfrac{1}{2\gamma} \|\xi\|^2 + h(X_t + \xi) \\ \text{s.t.} \quad \xi \in \mathrm{T}_{X_t}\mathcal{M}, \end{aligned} \tag{9}$$

and then a retraction step is performed to keep the iterate feasible to the manifold constraint:

$$X_{t+1} := \mathrm{Retr}_{X_t}(\eta_t \xi_t). \tag{10}$$

It is shown that the ManPG algorithm (9)-(10) finds an $\epsilon$-stationary point of (1) in $O(\epsilon^{-2})$ iterations. It was shown in [8] that ManPG performs better than some existing algorithms for solving the sparse PCA problem. The ManPG algorithm was extended successfully later to solving problems with two block variables [7] such as another sparse PCA formulation [36] and the sparse CCA problem [13].

Motivated by the success of the ManPG algorithm, when we only have the access to stochastic gradient of $f$, we design a stochastic version of ManPG to solve (1). In particular, each iteration of our proposed algorithms consists of two steps: (i) finding the descent direction, and (ii) performing retraction. The basic framework of our proposed algorithm is to simply replace the full gradient in ManPG by a stochastic estimator to the gradient. This leads to the following updating scheme of the proposed framework:

$$\begin{aligned} \zeta_t = \mathrm{argmin}_\zeta \phi_t(\zeta) := \langle V_t, \zeta\rangle + \tfrac{1}{2\gamma}\|\zeta\|^2 + h(X_t+\zeta) \\ \text{s.t.} \quad \zeta \in \mathrm{T}_{X_t}\mathcal{M}, \end{aligned} \tag{11}$$

and

$$X_{t+1} := \mathrm{Retr}_{X_t}(\eta_t \zeta_t), \tag{12}$$

where $\gamma > 0$ and $\eta_t > 0$ are step sizes, and $V_t$ denotes a stochastic estimation of the Euclidean gradient $\nabla f(X_t)$. Specific choices of $V_t$ will be discussed in Sections 3.2 and 3.3. Note that for the Stiefel manifold $\mathcal{M}$, the tangent space is given by $\mathrm{T}_X\mathcal{M} = \{\zeta \mid \zeta^\top X + X^\top \zeta = 0\}$. Therefore, the constraint in (11) is a linear equality constraint. Since we assume that $h$ is a convex function, it follows that the subproblem (11) is a convex problem. This convex problem can be efficiently solved using the semi-smooth Newton method [30]. We refer the readers to [30] and [8] for more details on how to solve (11) efficiently.

To prepare for the analysis of IFO complexity, we need to define the $\epsilon$-stationary solution and the $\epsilon$-stochastic stationary point.

**Definition 4** ($\epsilon$-stationary point and $\epsilon$-stochastic stationary point)**.** *Define*

$$G(X, \nabla f(X), \gamma) = (X - \mathrm{Retr}_X(\xi))/\gamma, \tag{13}$$

*where*

$$\xi := \mathrm{argmin}_{\xi \in \mathrm{T}_X\mathcal{M}} \{\langle \nabla f(X), \xi\rangle + \frac{1}{2\gamma}\|\xi\|^2 + h(X+\xi)\}.$$

*$X$ is called an $\epsilon$-stationary point of* (1) *if $\|G(X, \nabla f(X), \gamma)\| \le \epsilon$. When the sequence $\{X_t\}$ is generated by a stochastic algorithm (stochastic process), we call $X_t$ an $\epsilon$-stochastic stationary point if $\mathbb{E}[\|G(X_t, \nabla f(X_t), \gamma)\|] \le \epsilon$, where the expectation $\mathbb{E}$ is taken for all randomness before $X_t$ is generated.*

**Remark 2.** *Note that $\xi_t$ is the solution to* (11) *with full gradient $V_t = \nabla f(X_t)$. In the Euclidean space, $\mathrm{Retr}_{X_t}(\gamma\xi_t)$ reduces to $X_t + \gamma\xi_t$ and $\xi_t = \mathrm{prox}_{\gamma h}(X_t - \gamma\nabla f(X)) - X$. Thus, $G(X_t, \nabla f(X_t), \gamma)$ defined in* (13) *is analogous to the proximal gradient in the Euclidean space.*

### 3.2 R-ProxSGD: Riemannian Stochastic Proximal Gradient Descent Algorithm

In this section, we design the basic Riemannian proximal stochastic gradient descent method (R-ProxSGD) by choosing $V_t$ as the mini-batch stochastically sampled gradients. More specifically, in the $t$-th iteration of R-ProxSGD, we randomly sample a mini-batch set $\mathcal{S}_t$, and define $V_t = \frac{1}{|\mathcal{S}_t|}\sum_{i_t \in \mathcal{S}_t} \nabla f_{i_t}(X_t)$, which is an unbiased gradient estimator with bounded variance. That is, $\mathbb{E}[V_t] = \nabla f(X_t)$ and $\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \le \frac{\sigma^2}{|\mathcal{S}_t|}$. Our R-ProxSGD is described in Algorithm 1. Note that the step size $\eta_t$ is diminishing in this algorithm.

We have the following iteration and IFO complexity results for R-ProxSGD for solving the online case problem (1) with $f$ being (2). The proof is given in the appendix.

**Algorithm 1** R-ProxSGD

1: **Input:** initial point $X_0 \in \mathcal{M}$, parameters $\eta_0 > 0$, $\gamma > 0$
2: **for** $t = 0, 1, ..., T-1$ **do**
3: Compute the stochastic gradient by randomly sampling a mini-batch set $\mathcal{S}_t$ and calculating the unbiased stochastic gradient estimator:

$$V_t = \nabla f_{\mathcal{S}_t}(X_t) := \frac{1}{|\mathcal{S}_t|} \sum_{i_t \in \mathcal{S}_t} \nabla f_{i_t}(X_t)$$

4: Proximal step: obtain $\zeta_t$ by solving the subproblem (11).
5: Retraction step: $X_{t+1} = \mathrm{Retr}_{X_t}(\eta_t \zeta_t)$, with $\eta_t := \eta_0/\sqrt{t+1}$.
6: **end for**
7: **Output:** $X_\nu$, where $\nu$ is uniformly sampled from $\{1, ..., T\}$.

**Theorem 1.** *In R-ProxSGD, we set the batch size $|\mathcal{S}_t| := s = \mathcal{O}(\epsilon^{-2})$ for all $t$, and $\gamma$ is chosen to be sufficiently small. Under this parameter setting, the number of iterations needed by R-ProxSGD for obtaining an $\epsilon$-stochastic stationary point of the online case problem* (1) *with $f$ being* (2)*, is $T = O(\epsilon^{-4})$. Moreover, the IFO complexity of the R-ProxSGD algorithm for obtaining an $\epsilon$-stochastic stationary point in the online setting* (1) *with $f$ being* (2) *is $O(\epsilon^{-6})$.*

**Remark 3.** *In Theorem 1, since we require the batch size to be $\mathcal{O}(\epsilon^{-2})$, the results only hold for the online case problem, and do not hold for the finite-sum case problem.*

### 3.3 R-ProxSPB: Riemannian Proximal SpiderBoost Algorithm

Note that the convergence and complexity results of R-ProxSGD do not apply to the finite-sum case problem. In this section, we propose a Riemannian proximal SpiderBoost algorithm (R-ProxSPB) that can solve both the online case problem and the finite-sum case problem. More importantly, we can show that R-ProxSPB has an improved IFO complexity comparing with R-ProxSGD for the online case problem. For smooth problems in the Euclidean setting, there exist many works that use the variance reduction technique to improve the convergence speed of SGD, such as SVRG [16], SAGA [9], SARAH [23], SPIDER [12] and SpiderBoost [27]. In particular, the SpiderBoost algorithm proposed by Wang et.al. [27] achieves the same complexity bound as SPIDER, but in practice SpiderBoost converges much faster because it allows a constant step size, while SPIDER requires an $\epsilon$-dependent step size that can be too conservative in practice. Some of these algorithms have been extended to the Riemannian optimization with smooth objective functions, such as R-SVRG [33], R-SRG [17] and R-SPIDER [34, 35]. It was found that R-SRG and R-SPIDER equipped with the biased R-SARAH estimator consistently outperform the R-SVRG algorithm. Inspired by the SpiderBoost algorithm, we propose a Riemannian proximal SpiderBoost algorithm, named R-ProxSPB, which is a generalization of SpiderBoost to nonsmooth Riemannian optimization. When the nonsmooth function $h$ vanishes, our R-ProxSPB algorithm reduces to a Riemannian SpiderBoost algorithm (R-SpiderBoost) for Riemannian optimization with smooth objective function, which seems to be new in the literature as well.

Our R-ProxSPB algorithm is described in Algorithm 2. R-ProxSPB specifies a constant integer $q$. When the iteration number $t$ is a multiple of $q$, mini-batch $\mathcal{S}_t^1$ is sampled and unbiased stochastic gradient estimator is used; while for other iterations, mini-batch $\mathcal{S}_t^2$ is sampled and R-SARAH estimator (14) is used. Comparing with R-ProxSGD (Algorithm 1), a significant difference of R-ProxSPB is that it allows a constant step size $\eta$ instead of a diminishing step size. That the constant

**Algorithm 2** R-ProxSPB

**Input:** initial point $X_0 \in \mathcal{M}$, parameters $\eta > 0$, $\gamma > 0$, integers $q$, $T$.
**for** $t = 0, ..., T-1$ **do**
  **if** $\mathrm{mod}(t, q) = 0$ **then**
    Randomly sample a mini-batch $\mathcal{S}_t^1$ and calculate $V_t = \nabla f_{\mathcal{S}_t^1}(X)$ satisfying:

$$\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \leq \frac{\sigma^2}{|\mathcal{S}_t^1|}$$

  **else**
    Randomly sample a mini-batch $\mathcal{S}_t^2$ and calculate $V_t$ by the R-SARAH estimator:

$$V_t = \nabla f_{\mathcal{S}_t^2}(X_t) - \Gamma_{X_{t-1}}^{X_t}\big(\nabla f_{\mathcal{S}_t^2}(X_{t-1}) - V_{t-1}\big) \tag{14}$$

  **end if**
  Proximal step: obtain $\zeta_t$ by solving the subproblem (11).
  Retraction step: $X_{t+1} = \mathrm{Retr}_{X_t}(\eta\zeta_t)$.
**end for**
**Output:** $X_\nu$, $\nu$ is uniformly sampled from $\{1, ..., T\}$.

step size is allowed is due to the biased stochastic gradient estimator R-SARAH, which leads to variance reduction of the stochastic gradients, and thus improves the convergence rate. This has been justified in several variance reduced stochastic algorithms such as SVRG, SAGA, SPIDER and SpiderBoost and so on. A constant step size usually leads to a faster algorithm both theoretically and practically. In fact, we can prove the following convergence rate and IFO complexity results of R-ProxSPB, which indeed improve the results of R-ProxSGD.

**Theorem 2.** *In R-ProxSPB (Algorithm 2), we set* $\eta = \min\big(\frac{1}{2(L_R/2+L_hM_2)}, \frac{1}{\sqrt{2c_E\Theta^2}}\big)$, $\gamma = \frac{2}{5}$, *and* $|\mathcal{S}_t^2| = q$ *for all* $t$, *where* $\Theta$ *is a constant that will be specified in the proof. Under this parameter setting, we have the following convergence rate and IFO complexity results of R-ProxSPB.*

*(i). For the finite-sum case problem, i.e., problem* (1) *with* $f$ *being* (3), *we set* $q = \sqrt{n}$, $|\mathcal{S}_t^1| = n$, *for all* $t$. *R-ProxSPB returns an* $\epsilon$*-stochastic stationary point of* (1) *after* $T = O(\epsilon^{-2})$ *iterations. Moreover, the IFO complexity of R-ProxSPB for obtaining an* $\epsilon$*-stochastic stationary point of* (1) *is* $\mathcal{O}(\sqrt{n}\epsilon^{-2} + n)$.

*(ii). For the online case problem, i.e., problem* (1) *with* $f$ *being* (2), *we set* $q = \mathcal{O}(\epsilon^{-1})$, $|\mathcal{S}_t^1| = \mathcal{O}(\epsilon^{-2})$, *for all* $t$. *R-ProxSPB returns an* $\epsilon$*-stochastic stationary point of* (1) *after* $T = O(\epsilon^{-2})$ *iterations. Moreover, the IFO complexity of R-ProxSPB for obtaining an* $\epsilon$*-stochastic stationary point of* (1) *is* $\mathcal{O}(\epsilon^{-3})$.

**Remark 4.** *Here we summarize some comparisons of the two proposed algorithms. For the online case problem, R-ProxSPB has a better IFO complexity than R-ProxSGD. R-ProxSPB allows constant step size* $\eta$, *but R-ProxSGD needs a diminishing step size* $\eta_t$. *The convergence results of R-ProxSPB in Theorem 2 covers the finite-sum case problem, which is still lacking for the R-ProxSGD algorithm. We also need to point out that, though R-ProxSPB is faster than R-ProxSGD in theory, it involves more tuning parameters and the R-SARAH estimator might be difficult to compute for certain manifolds. Therefore, for certain applications, R-ProxSGD could be more favorable in practice.*

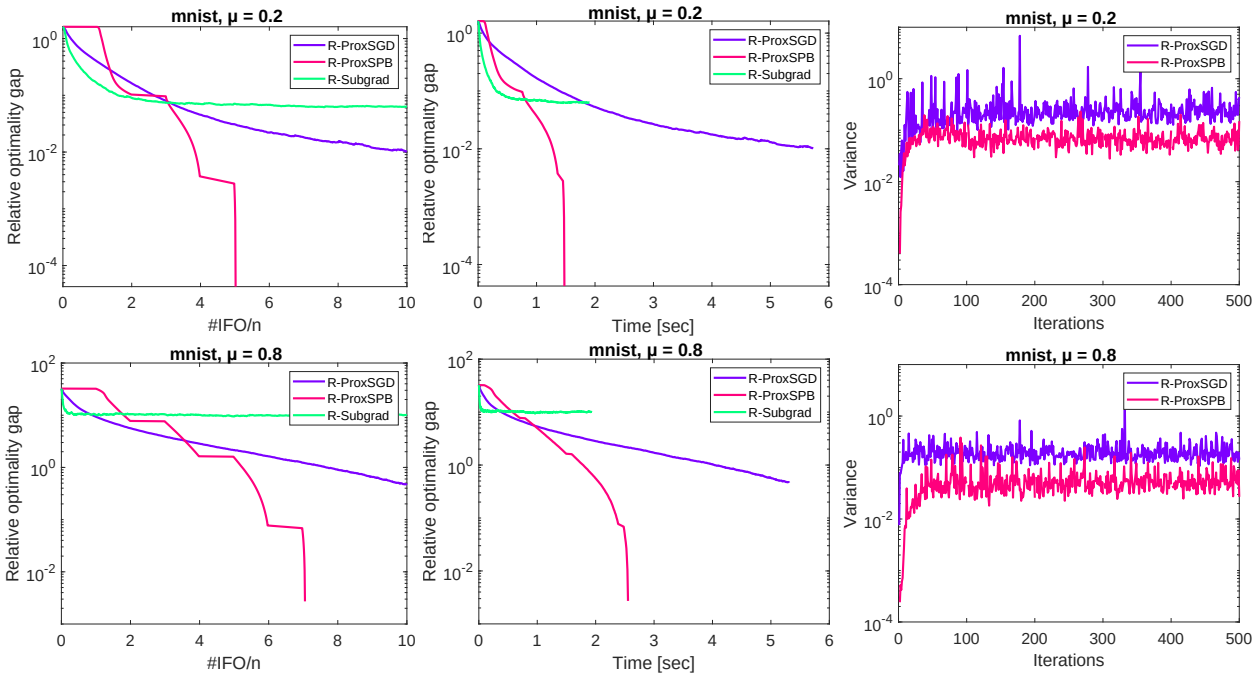



Figure 1: Experimental results on the `mnist` dataset with $\mu = 0.2, 0.8$.

# 4 Numerical Experiments

## 4.1 Online Sparse PCA Problem

First, we compare our proposed algorithms R-ProxSGD and R-ProxSPB with the Riemannian stochastic subgradient method (R-Subgrad) on the online sparse PCA problem (4). R-Subgrad for solving (4) iterates as follows:

$$\begin{aligned} \xi_t &:= -\mathrm{Proj}_X(-2Z_{i_t}Z_{i_t}^\top X_t + \mu\ \mathrm{sign}(X_t)), \\ X_{t+1} &:= \mathrm{Retr}_{X_t}(\xi_t), \end{aligned}$$

where $Z_{i_t}$ is a randomly sampled data. Here the projection operation is defined as: $\mathrm{Proj}_X(Y) = Y - X\ \mathrm{sym}(X^\top Y)$ and $\mathrm{sym}(X) = \frac{1}{2}(X + X^\top)$.

The experiments are performed on two real datasets: `coil100` [22] and `mnist` [18]. The `coil100` dataset contains $n = 7,200$ RGB images of 100 objects taken from different angles. The `mnist` dataset has $n = 80,000$ grayscale digit images of size $28 \times 28$. We set $\gamma = 10^{-3}$ for both R-ProxSGD and R-ProxSPB. For R-Subgrad and R-ProxSGD, we use the diminishing step size $\eta_t = \frac{\eta_0}{\sqrt{t+1}}$. For R-ProxSPB, we use the constant step size $\eta$ as suggested in our theory. Because some of the problem-dependent constants cannot be directly estimated from the datasets, we perform grid search to tune $\eta_0$ and $\eta$ for all algorithms from $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, ..., 1\}$. The best $\eta_0$ and $\eta$ on different datasets are reported in the Supplementary materials. For R-ProxSGD, we set $|\mathcal{S}_t| = 100$. For R-ProxSPB, we set $|\mathcal{S}_t^1| = n$ and $|\mathcal{S}_t^2| = q = 100$. We use $\gamma = 10^{-5}$ for R-ProxSGD and $\gamma = \frac{2}{5}$ for R-ProxSPB based on our theoretical results.

All algorithms are implemented in Matlab and we use the `Manopt` [3] package to compute vector transport, retraction and Riemannian gradient. Since there is no groundtruth for comparison, we run the deterministic algorithm ManPG [8] first, and denote the objective function value returned by ManPG as $F^*$. We then measure the performance of other algorithms by computing the relative

optimality gap: $\frac{F-F^*}{|F^*|}$, where $F$ denotes the objective function value returned by the algorithm in consideration.

The experimental results are shown in Figures 1-2. In particular, Figures 1 and 2 give results for $r = 10$. More specifically, in Figure 1 we report the results on the `mnist` dataset, and in Figure 2 we report the results on the `coil100` dataset, both with two choices of $\mu$: $\mu = 0.2$ and $\mu = 0.8$. Note that $\mu$ is the parameter in (4) controlling the sparsity of the solution. In the first column of Figures 1 and 2, we report the relative optimality gap versus the number of IFO divided by $n$. In the second column of Figures 1 and 2, we report the relative optimality gap versus the CPU time (in seconds). In the third column of Figures 1 and 2, we report the variance of gradient estimation versus the number of iterations, which is adopted in [10].

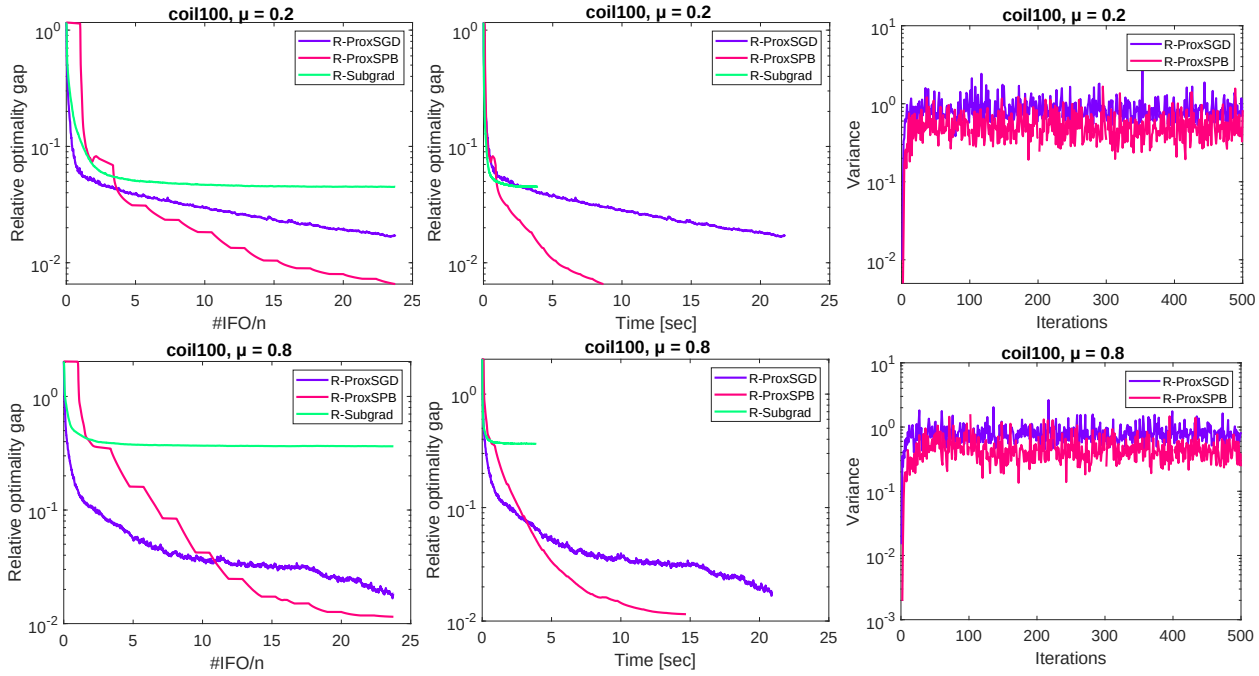



Figure 2: Experimental results on the `coil100` dataset with $\mu = 0.2, 0.8$.

All the results in Figures 1-2 indicate that both R-ProxSGD and R-ProxSPB consistently outperform R-Subgrad in terms of CPU time and the number of IFO calls. Moreover, these figures show that R-Subgrad is not able to reduce the optimality gap to a desired accuracy, comparing with R-ProxSGD and R-ProxSPB. Furthermore, these results also show that R-ProxSPB usually performs much better than R-ProxSGD, which is consistent with our theoretical results on the complexity bounds. Figures 1 and 2 also imply that R-ProxSPB is effective to reduce the variance of the stochastic gradient on both datasets.

We perform grid search to tune $\eta_0$ (used in R-Subgrad and R-ProxSGD) and $\eta$ (used in R-ProxSPB) from $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, ..., 1\}$. The best $\eta_0$ and $\eta$ on different datasets are reported in Tables 3 and 4.

Figures 3 and 4 presents more experimental results. In particular, Figure 3 shows the results for the case $r = 10$ and $\mu = 0.4$ and the meanings of the figures are the same as Figures 1 and 2. Figure 4 gives more results on the case $r = 15$ and $\mu = 0.2$, 0.4, 0.8, and here we only present the relative optimality gap versus the CPU time. These results further justify the advantages of our proposed R-ProxSGD and R-ProxSPB algorithms.

| $\mu$ | R-Subgrad | R-ProxSGD | R-ProxSPB |
|---|---|---|---|
| 0.2 | 0.01 | 0.005 | 0.005 |
| 0.4 | 0.01 | 0.01 | 0.005 |
| 0.8 | 0.05 | 0.005 | 0.005 |

Table 3: Chosen $\eta_0$ (for R-Subgrad and R-ProxSGD) and $\eta$ (for R-ProxSPB) for the reported results on `mnist` dataset.

| $\mu$ | R-Subgrad | R-ProxSGD | R-ProxSPB |
|---|---|---|---|
| 0.2 | 0.005 | 0.01 | 0.005 |
| 0.4 | 0.01 | 0.01 | 0.005 |
| 0.8 | 0.005 | 0.01 | 0.005 |

Table 4: Chosen $\eta_0$ (for R-Subgrad and R-ProxSGD) and $\eta$ (for R-ProxSPB) for the reported results on `coil100` dataset.

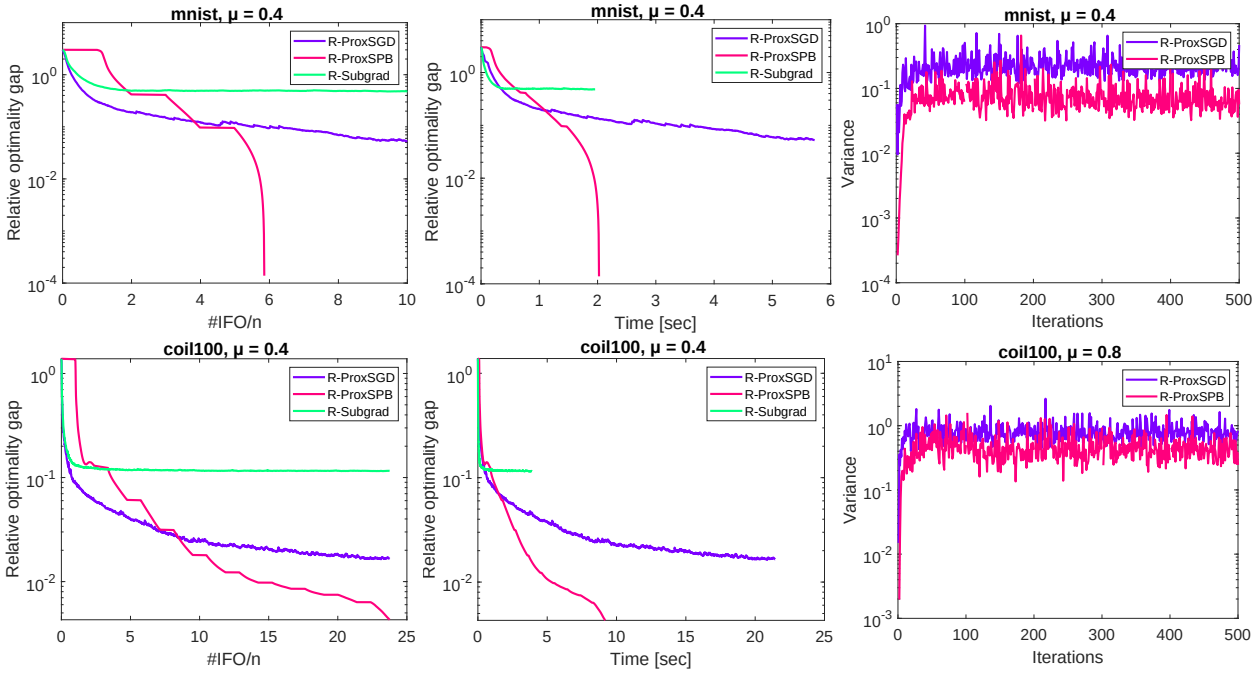



Figure 3: Experimental results on the `mnist` and `coil100` datasets with $r = 10$ and $\mu = 0.4$.

## 4.2 Robust Low-Rank Matrix Completion

Robust low-rank matrix completion is closely related to the robust PCA problem. The robust PCA aims to decompose a given matrix $M \in \mathbb{R}^{m\times n}$ into the superposition of a low-rank matrix $L$ and a sparse matrix $S$. Robust low-rank matrix completion is the same as robust PCA, except that only a subset of the entries of $M$ is observed. The convex formulations of them are studied extensively in the literature and we refer the reader to the recent survey [20]. A typical convex formulation of robust low-rank matrix completion is given as follows:

$$\min_{L,S} \ \|L\|_* + \gamma\|S\|_1, \ \text{s.t.}, \ \mathcal{P}_\Omega(L+S) = \mathcal{P}_\Omega(M), \tag{15}$$

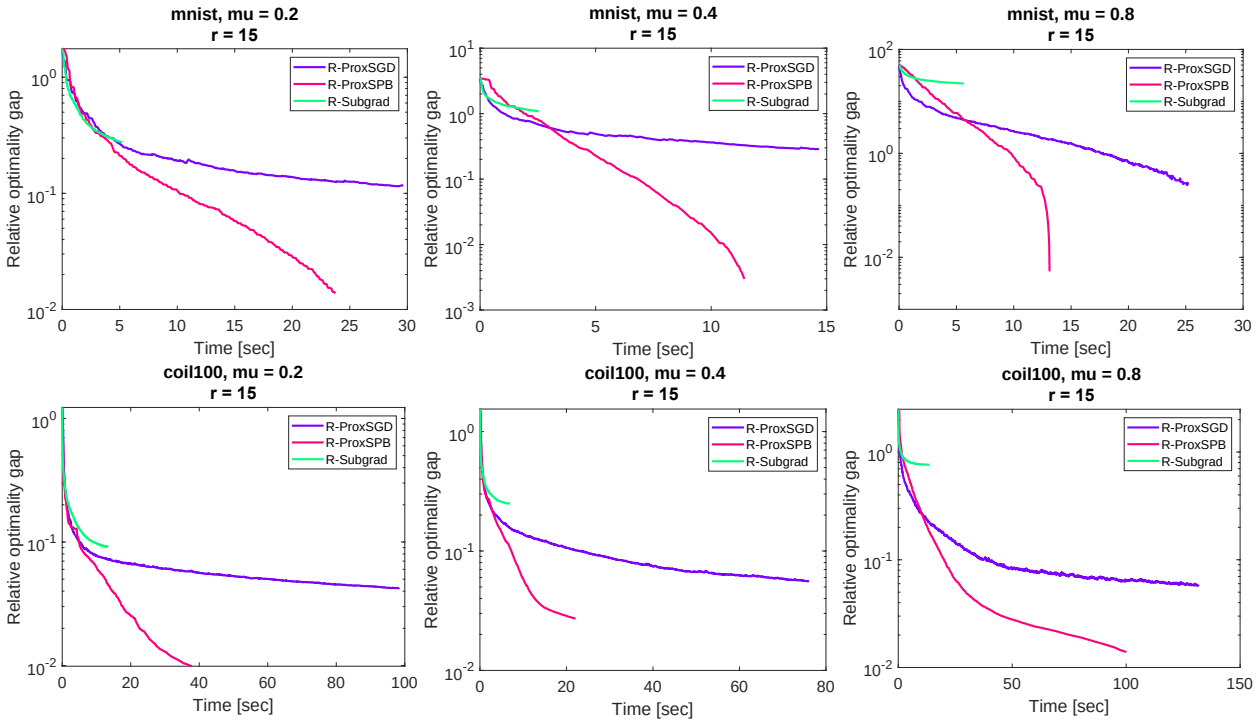



Figure 4: Relative optimality gaps versus runtime on two datasets with $r = 15$ and $\mu = 0.2$, 0.4, 0.8.

where $\|L\|_*$ denotes the nuclear norm of $L$ and it sums the singular values of $L$, $\Omega$ is a subset of the index set $\{(i,j) \mid 1 \le i \le m, 1 \le j \le n\}$, and the projection operator $\mathcal{P}_\Omega$ is defined as: $[\mathcal{P}_\Omega(Z)]_{ij} = Z_{ij}$, if $(i,j) \in \Omega$, and $[\mathcal{P}_\Omega(Z)]_{ij} = 0$ otherwise. Due to the presence of the nuclear norm in (15), algorithms for solving (15) usually requires computing the SVD of an $m \times n$ matrix in every iteration, which can be time consuming when $m$ and $n$ are large. Recently, some nonconvex formulations of robust low-rank matrix completion were proposed because they allow more efficient and scalable algorithms. In [15], the authors proposed the following nonconvex formulation of robust low-rank matrix completion:

$$\min_{U \in \mathrm{Gr}(m,r), V \in \mathbb{R}^{r \times n}, S \in \mathbb{R}^{m \times n}} \frac{1}{2}\|\mathcal{P}_\Omega(UV - M + S)\|_F^2 + \frac{\lambda}{2}\|\mathcal{P}_{\bar{\Omega}}(UV)\|_F^2 + \gamma\|\mathcal{P}_\Omega(S)\|_1, \tag{16}$$

where $\mathrm{Gr}(m,r)$ denotes the Grassmann manifold, which is the set of $r$-dimensional vector subspaces of $p\mathbb{R}^m$. In (16), the low-rank matrix $L$ is replaced by $UV$ with $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{r \times n}$, and $r$ is the estimation of the rank of $L$; the term $\frac{\lambda}{2}\|\mathcal{P}_{\bar{\Omega}}(UV)\|_F^2$ is added as a regularizer and $\lambda > 0$ is sufficiently small indicating that we have a small confidence of the components of $UV$ on $\bar{\Omega}$ being zeros; the constraint $U \in \mathrm{St}(m,r)$ is added to remove the scaling ambiguity of $U$ and $V$. The nonconvex formulation (16) was motivated by some recent works on Riemannian optimization [4, 6]. Note that, for fixed $U$ and $S$, the variable $V$ in (16) can be uniquely determined. By denoting

$$\bar{f}(U,V,S) = \frac{1}{2}\|\mathcal{P}_\Omega(UV - M + S)\|_F^2 + \frac{\lambda}{2}\|\mathcal{P}_{\bar{\Omega}}(UV)\|_F^2, \tag{17}$$

and

$$V_{U,S} := \mathrm{argmin}_V \ \bar{f}(U,V,S), \ \text{and} \ f(U,S) = \bar{f}(U, V_{U,S}, S), \tag{18}$$

we can rewrite (16) as

$$\min_{U\in \mathrm{Gr}(m,r), S\in\mathbb{R}^{m\times n}} f(U,S)+\gamma\|\mathcal{P}_{\Omega}(S)\|_1, \tag{19}$$

which is a Riemannian optimization problem with nonsmooth objective. Note that although the manifold is the Grassmann manifold instead of the Stiefel manifold, our algorithms discussed in Section 3 can be directly applied to (19). To see this, first note that as suggested in [4], without loss of generality, we can restrict matrix $U$ as orthonormal matrix. Therefore, we have

$$\|\mathcal{P}_{\bar{\Omega}}(UV)\|_F^2=\|UV\|_F^2-\|\mathcal{P}_{\Omega}\|_F^2=\|V\|_F^2-\|\mathcal{P}_{\Omega}\|_F^2,$$

and thus we can rewrite $\bar{f}(U,V,S)$ and $f(U,S)$ as

$$\bar{f}(U,V,S)=\frac{1}{2}\|\mathcal{P}_{\Omega}(UV-M+S)\|_F^2+\frac{\lambda}{2}\|V\|_F^2-\frac{\lambda}{2}\|\mathcal{P}_{\Omega}(UV)\|_F^2. \tag{20}$$

$$f(U,S)=\frac{1}{2}\|\mathcal{P}_{\Omega}(UV_{U,S}-M+S)\|_F^2+\frac{\lambda}{2}\|V_{U,S}\|_F^2-\frac{\lambda}{2}\|\mathcal{P}_{\Omega}(UV_{U,S})\|_F^2. \tag{21}$$

From (18) we know that $\nabla_V \bar{f}(U,V_{U,S},S)=0$. Therefore,

$$\nabla_U f(U,S)=\nabla_U \bar{f}(U,V_{U,S},S)=\nabla_1 \hat{f}(U,V_{U,S},S),$$

where

$$\hat{f}(U,V_{U,S},S):=\frac{1}{2}\|\mathcal{P}_{\Omega}(UV_{U,S}-M+S)\|_F^2-\frac{\lambda}{2}\|\mathcal{P}_{\Omega}(UV_{U,S})\|_F^2=\sum_{(i,j)\in\Omega}\hat{f}_{ij}(U,V_{U,S},S), \tag{22}$$

and

$$\hat{f}_{ij}(U,V_{U,S},S)=\frac{1}{2}(UV_{U,S}-M+S)_{ij}^2-\frac{\lambda}{2}(UV_{U,S})_{ij}^2.$$

That is, $\hat{f}$ in (22) has a natural finite-sum structure, and a stochastic gradient approximation to $\nabla_U f(U,S)$ is given by $\nabla_1 \hat{f}_{ij}(U,V_{U,S},S)$ with randomly sampled index pair $(i,j)\in\Omega$. It is easy to verify that

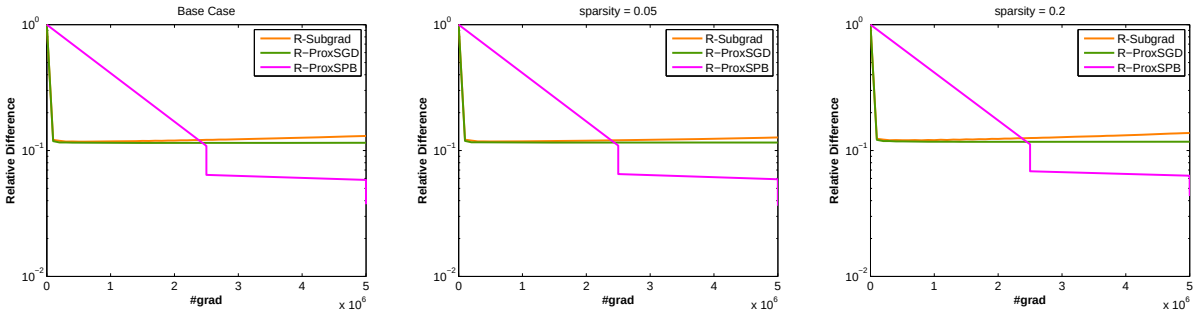
$$\nabla_1 \hat{f}_{ij}(U,V_{U,S},S)=(u_i^\top v_j-M_{ij}+S_{ij}-\lambda u_i^\top v_j)\bar{V}_j^\top,$$

where $u_i^\top$ denotes the $i$-th row of $U$, and $v_j$ denotes the $j$-th column of $V_{U,S}$, and

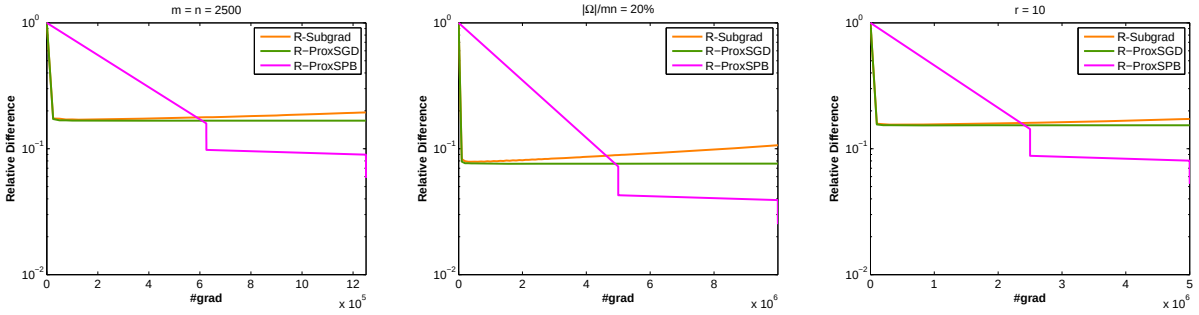
$$\bar{V}_j=\begin{bmatrix}0 & 0 & \cdots & v_j & \cdots & 0\end{bmatrix}.$$

That is, $\bar{V}_j\in\mathbb{R}^{r\times m}$ is a matrix whose $j$-th column is $v_j$ and all other columns are zeros. Clearly, when computing $\nabla_U f_{ij}(U,S)$, we only need to access $u_i^\top$ and $v_j$ and we do not need to access the whole matrix $U$ and $V_{U,S}$ and compute the matrix multiplication $UV_{U,S}$, and this is very useful when $m$ and $n$ are large.

We applied our R-ProxSGD and R-ProxSPB algorithms to solve the robust low-rank matrix completion problem (19) on some synthetic data and we again compared their performance with R-Subgrad. The synthetic data were generated in the following manner. First, the groundtruth $U^*\in\mathbb{R}^{m\times r}$ and $V^*\in\mathbb{R}^{r\times n}$ were generated as Gaussian matrices and we set $X^*=U^*V^*$. We then randomly sample a subset of indices to obtain $\Omega$, and then sample the entries of $X^*$ from $\Omega$ to get $M$. A sparse matrix $S^*$ was then added to $M$. In R-ProxSGD, we randomly sample 4% of the known entries as a batch in each iteration. In R-ProxSPB, we set $|\mathcal{S}_t^1|=|\Omega|$, $|\mathcal{S}_t^2|=100$ and $q=5$. The initial step sizes $\eta_0$ are tuned in $i/10^j,\ i=1,2,\ldots,5,\ j=0,1,\ldots,4$.




(a) Base Case (b) Sparsity = 0.05 (c) Sparsity = 0.20




(d) $m = n = 2500$ (e) $|\Omega|/(mn) = 20\%$ (f) $r = 10$

Figure 5: Relative Difference versus the first-order oracle complexity for the online robust low-rank matrix completion problem on different cases. The base case (a) has the parameters $m = n = 5000$, $r = 5$, Sparsity $= 10\%$, $|\Omega|/(mn) = 10\%$.

In Figure 5, we present the experimental results on the problem with various settings. To compare the performance of different algorithms, we calculate the relative difference at iteration $k$: $\frac{\|U^k V^k - X^*\|_F}{\|X^*\|_F}$ versus the number of stochastic Riemannian gradient. Figure 5 (a) shows the result for $m = n = 5000$, $r = 5$, Sparsity $= \frac{|S^*|}{mn} = 10\%$, $|\Omega|/(mn) = 10\%$, and we call this the base case. Figures 5 (b)-(f) report the results where there is one parameter that is different with the base case. For example, Figure 5 (b) shows the result where the sparsity is changed to 5%, and other parameters are the same as the base case. Results in Figure 5 clearly show the advantage of our R-ProxSPB algorithm over R-Subgrad algorithm.

# 5 Conclusion

In this paper, we considered the nonsmooth Riemannian optimization problems with nonsmooth regularizer in the objective. We designed Riemannian stochastic algorithms that do not need subgradiet information for solving this class of problems. Specifically, we proposed two Riemannian stochastic proximal gradient algorithms: R-ProxSGD and R-ProxSPB to solve this problem. The two proposed algorithms are generalizations of their counterparts in Euclidean space to Riemannian manifold setting. We analyzed the iteration complexity and IFO complexity of the proposed algorithms for obtaining an $\epsilon$-stationary point. Numerical results on solving online sparse PCA and robust low-rank matrix completion are conducted which demonstrate that our proposed algorithms outperform significantly the Riemannian stochastic subgradient method. Future work includes extending the current results to more general Riemannian manifolds.

## Acknowledgement

The authors would like to thank Shixiang Chen for fruitful discussions.

## A Auxiliary Definitions and Lemmas

In this section we give a few lemmas and definitions that are necessary to our analysis. These lemmas are proved in existing works, so we do not include the proof here.

**Definition 5** (Generalized Clarke subdifferential, see [14]). *For a locally Lipschitz function $F$ on the manifold $\mathcal{M}$, the Riemannian generalized directional derivative $F^\circ(X,\zeta)$ at $X \in \mathcal{M}$ in the direction $\zeta$ is defined by*

$$\limsup_{Y\to X, t\downarrow 0} \frac{F\circ\phi^{-1}(\phi(Y)+tD\phi(X)[\zeta]) - f\circ\phi^{-1}(\phi(Y))}{t}.$$

*Here $(\phi, U)$ is a coordinate chart at $X$. The Clarke subdifferential $\hat{\partial}F(X)$ at $X \in \mathcal{M}$ is:*

$$\hat{\partial}F(X) = \{\xi \in \mathrm{T}_X\mathcal{M} : \langle \xi, V\rangle \le F^\circ(X,\zeta), \forall \zeta \in \mathrm{T}_X\mathcal{M}\}.$$

**Lemma 3.** *Suppose $g_i$ is the unbiased and variance-bounded stochastic estimator of $g$ on randomly sampled instance $i$, i.e. $\mathbb{E}_i[g_i] = g$ and $\mathbb{E}_i[\|g_i - g\|^2] \le \sigma^2$. Then we can conclude that the estimator $g_{\mathcal{S}} := \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} g_i$ based on randomly sampled mini-batch $\mathcal{S}$ is also unbiased and variance-bounded:*

$$\mathbb{E}_{\mathcal{S}}[g_{\mathcal{S}}] = g, \quad \mathbb{E}_{\mathcal{S}}[\|g_{\mathcal{S}} - g\|^2] \le \frac{\sigma^2}{|\mathcal{S}|}. \tag{23}$$

The following lemmas from previous works [1, 17, 35] under Assumptions 1-5 regarding retraction and vector transport are very useful.

**Lemma 4** (Retraction $L_R$ smoothness, Lemma 3.5 in [17]). *If $f(X)$ has an upper-bounded Hessian, there exists a neighborhood $\mathcal{U}$ of any $X \in \mathcal{M}$ and a constant $L_R > 0$ such that $\forall X, Y \in \mathcal{U}, \mathrm{Retr}_X(\xi) = Y, \xi \in \mathrm{T}_X\mathcal{M}$:*

$$f(Y) \le f(X) + \langle \nabla f(X), \xi\rangle + \frac{L_R}{2}\|\xi\|^2. \tag{24}$$

**Lemma 5** (Lemma 3.7 in [17]). *Under Assumption 1(ii), there exists a constant $\theta > 0$, such that the following inequalities hold for any $X, Y \in \mathcal{U}$:*

$$\|\Gamma_\eta\xi - P_\eta\xi\| \le \theta\|\xi\|_X\|\eta\|_X, \quad \|\Gamma_\eta^{-1}\xi - P_\eta^{-1}\xi\| \le \theta\|\chi\|_X\|\eta\|_X,$$

*where $\xi, \eta \in \mathrm{T}_X\mathcal{M}$, $\chi \in \mathrm{T}_Y\mathcal{M}$, $\mathrm{Retr}_X(\eta) = Y$.*

**Lemma 6** (Lemma 4 in [35]). *Given $\hat{X} \in \mathcal{M}$ that does not depend on the update sequence $\{X_t\}$, the following inequality about the retraction and vector transport holds:*

$$\mathbb{E}_i[\|\Gamma_{X_t}^{\hat{X}}(\nabla f_i(X_t)) - \Gamma_{X_{t-1}}^{\hat{X}}\nabla f_i(X_{t-1})\|^2] \le 2\Theta^2\|\mathrm{Retr}_{X_{t-1}}^{-1}(X_t)\|^2, \tag{25}$$

*where $\Theta^2 = \theta^2G^2 + 2(1+c_R)L_H^2$ and $\theta$ is defined in Lemma 5.*

**Lemma 7** (Lemma 1 in [35]). *Let $n_t = \lceil t/q \rceil$, $(n_t - 1)q \leq t \leq n_t q$, $t_0 = (n_t - 1)q$, where $\lceil a \rceil$ denotes the smallest integer that is larger than $a$. Mini-batches $\mathcal{S}_t^1$, $\mathcal{S}_t^2$ are selected as described in Algorithm 2. Under the Assumptions 1-5, the estimation error between the R-SARAH estimator $V_t$ generated by Algorithm 2 and full gradient $\nabla f(X_t)$ is bounded by:*

$$\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \leq I\{|\mathcal{S}_t^1| < n\} \cdot \frac{\sigma^2}{|\mathcal{S}_t^1|} + \sum_{i=t_0}^{t-1} \frac{\Theta^2}{|\mathcal{S}_t^2|} \mathbb{E}[\|\mathrm{Retr}_{X_i}^{-1}(X_{i+1})\|^2],$$

*where $I\{\cdot\}$ denotes an indicator function.*

For the ease of presentation, we adopt the following notation, which is consistent with the ones used in (9) and (11).

$$\zeta_t := \mathrm{argmin}_{\zeta \in \mathrm{T}_{X_t}\mathcal{M}} \{\phi_t(\zeta) := \langle V_t, \zeta \rangle + \frac{1}{2\gamma}\|\zeta\|^2 + h(X_t + \zeta)\}, \tag{26}$$

$$\xi_t := \mathrm{argmin}_{\xi \in \mathrm{T}_{X_t}\mathcal{M}} \{\langle \nabla f(X_t), \xi \rangle + \frac{1}{2\gamma}\|\xi\|^2 + h(X_t + \xi)\}. \tag{27}$$

Moreover, note that according to the definition of $\mathcal{F}_t$, when we take conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$, $X_t$ in both R-ProxSGD and R-ProxSPB has been realized.

# B Necessary Lemmas for Proving Theorem 1

**Lemma 8.** *The solution $\zeta_t$ defined in* (26) *satisfies:*

$$\mathbb{E}[\phi_t(\eta_t\zeta_t)|\mathcal{F}_{t-1}] - \phi_t(0) \leq \frac{(\eta_t - 2)\eta_t}{2\gamma} \mathbb{E}[\|\zeta_t\|^2 | \mathcal{F}_{t-1}]. \tag{28}$$

*Proof.* Note that $\phi_t(\zeta)$ is $(1/\gamma)$-strongly convex with respect to $\zeta$. For $\zeta_1, \zeta_2 \in \mathrm{T}_{X_t}\mathcal{M}$, we have:

$$\phi_t(\zeta_2) \geq \phi_t(\zeta_1) + \langle \hat{\partial}\phi_t(\zeta_1), \zeta_2 - \zeta_1 \rangle + \frac{1}{2\gamma}\|\zeta_2 - \zeta_1\|^2. \tag{29}$$

Note that the optimality conditions of (26) are given by $0 \in \mathrm{Proj}_{\mathrm{T}_{X_t}\mathcal{M}} \partial\phi_t(\zeta_t)$. Therefore,

$$\langle \hat{\partial}\phi_t(\zeta_1), \zeta_2 - \zeta_1 \rangle = \langle \mathrm{Proj}_{\mathrm{T}_{X_t}\mathcal{M}} \partial\phi_t(\zeta_1), \zeta_2 - \zeta_1 \rangle = 0, \forall \zeta_1, \zeta_2 \in \mathrm{T}_{X_t}\mathcal{M}. \tag{30}$$

Letting $\zeta_1 = \zeta_t$ and $\zeta_2 = 0$ in (29), and combining with (30), we have

$$\phi_t(0) \geq \phi_t(\zeta_t) + \frac{1}{2\gamma}\|\zeta_t\|^2,$$

which is equivalent to:

$$h(X_t + \zeta_t) - h(X_t) \leq \langle -V_t, \zeta_t \rangle - \frac{1}{\gamma}\|\zeta_t\|^2. \tag{31}$$

According to the definition of $\phi_t$, $\phi_t(\eta_t\zeta_t) - \phi_t(0)$ can be written as:

$$\phi_t(\eta_t\zeta_t) - \phi_t(0) = \eta_t\langle V_t, \zeta_t \rangle + \frac{\eta_t^2}{2\gamma}\|\zeta_t\|^2 + h(X_t + \eta_t\zeta_t) - h(X_t). \tag{32}$$

From (31) and the convexity of $h$: $h(X_t + \eta_t\zeta_t) \leq \eta_t h(X_t + \zeta_t) + (1 - \eta_t)h(X_t)$, $\eta_t \in (0, 1]$, we have

$$h(X_t + \eta_t\zeta_t) - h(X_t) \leq -\eta_t\langle V_t, \zeta_t \rangle - \frac{\eta_t}{\gamma}\|\zeta_t\|^2. \tag{33}$$

Combine (32) and (33) and take expectation conditioned on $\mathcal{F}_{t-1}$ on both sides, we get the desired result. □

The following lemma justifies why $G(X, \nabla f(X), \gamma)$ is valid for defining the $\epsilon$-stationary solution.

**Lemma 9.** *If $G(X, \nabla f(X), \gamma) = 0$, and the retraction is given by the Polar decomposition: $\mathrm{Retr}_X(\xi) = (X + \xi)(I + \xi^\top \xi)^{-\frac{1}{2}}$, then $X$ is a stationary point of problems (1), i.e., $0 \in \nabla f(X) + \mathrm{Proj}_{\mathrm{T}_X\mathcal{M}} \partial h(X)$.*

To prove Lemma 9, we first need to show the following Lemma.

**Lemma 10.** *Consider $X \in \mathcal{M}$, $\mathcal{M}$ is the Stiefel manifold and $\xi \in T_X\mathcal{M}$. If $X = \mathrm{Retr}_X(\xi)$, where the retraction is given by the Polar decomposition: $\mathrm{Retr}_X(\xi) = (X + \xi)(I + \xi^\top \xi)^{-\frac{1}{2}}$, then $\xi = \mathbf{0}_X$.*

*Proof.* If $X = \mathrm{Retr}_X(\xi) = (X + \xi)(I + \xi^\top \xi)^{-\frac{1}{2}}$, then we have

$$X + \xi = X(I + \xi^\top \xi)^{\frac{1}{2}}. \tag{34}$$

Since $X^\top X = I$, (34) leads to

$$X^\top X + \xi^\top X = (I + \xi^\top \xi)^{\frac{1}{2}} \tag{35}$$

and

$$X^\top X + X^\top \xi = (I + \xi^\top \xi)^{\frac{1}{2}}. \tag{36}$$

Since $\xi \in \mathrm{T}_X\mathcal{M}$, we have $\xi^\top X + X^\top \xi = 0$. Adding (35) and (36) gives $2I = 2(I + \xi^\top \xi)^{\frac{1}{2}}$, which implies $\xi = \mathbf{0}_X$. □

Now we are ready to give the proof of Lemma 9.

*Proof.* If $G(X_t, \nabla f(X_t), \gamma) = 0$, we have $\xi_t = \mathbf{0}_{X_t}$ because of Lemma 10. According to [32], the optimality conditions of (27) are given by

$$0 \in \nabla f(X_t) + \frac{1}{\gamma}\xi_t + \mathrm{Proj}_{\mathrm{T}_{X_t}\mathcal{M}} \partial h(X_t + \xi_t).$$

Thus, $G(X_t, \nabla f(X_t), \gamma) = 0$ leads to that $0 \in \nabla f(X_t) + \mathrm{Proj}_{\mathrm{T}_{X_t}\mathcal{M}} \partial h(X_t)$, which means $X_t$ is a stationary point of problem (1). □

The following lemma shows the progress of the algorithm in one iteration in terms of objective function value.

**Lemma 11.** *Denote $X_t^+ := X_t + \eta_t \zeta_t$. The following inequality holds:*

$$F(X_{t+1}) - F(X_t) \leq \frac{(L_R\gamma - 1)\eta_t^2}{2\gamma}\|\zeta_t\|^2 + h(X_{t+1}) - h(X_t^+) + \phi_t(\eta_t\zeta_t) - \phi_t(0) + \eta_t\langle \nabla f(X_t) - V_t, \zeta_t\rangle.$$

*Proof.* Consider the update $X_{t+1} = \mathrm{Retr}_{X_t}(\eta_t\zeta_t)$. By applying Lemma 4 with $X = X_t, Y = X_{t+1}$ and $\xi = \eta_t\zeta_t$, we get

$$f(X_{t+1}) - f(X_t) \leq \eta_t\langle \nabla f(X_t), \zeta_t\rangle + \frac{L_R\eta_t^2}{2}\|\zeta_t\|^2,$$

which leads to:

$$F(X_{t+1}) - F(X_t) \leq \frac{L_R\eta_t^2}{2}\|\zeta_t\|^2 + \eta_t\langle \nabla f(X_t), \zeta_t\rangle + h(X_{t+1}) - h(X_t). \tag{37}$$

Denote $X_t^+ := X_t + \eta_t\zeta_t$. The definition of $\phi_t$ indicates:

$$\eta_t\langle V_t, \zeta_t\rangle = \phi_t(\eta_t\zeta_t) - \phi_t(0) - \frac{\eta_t^2}{2\gamma}\|\zeta_t\|^2 - h(X_t^+) + h(X_t). \tag{38}$$

Combining (37) and (38) gives the desired result. □

The following lemma gives an upper bound to the size of $G(X_t, \nabla f(X_t), \gamma)$.

**Lemma 12.** *With $\zeta_t$ and $\xi_t$ defined in (26) and (27), for $G(X_t, \nabla f(X_t), \gamma) = \frac{1}{\gamma}(X_t - \mathrm{Retr}_{X_t}(\xi_t))$, it holds that*

$$\|G(X_t, \nabla f(X_t), \gamma)\|^2 \leq 2M_1^2(7\|\zeta_t\|^2 + 4\gamma\|V_t - \nabla f(X_t)\|^2). \tag{39}$$

*Proof.* Let $G(X_t, V_t, \gamma) = \frac{1}{\gamma}(X_t - \mathrm{Retr}_{X_t}(\gamma\zeta_t))$. We first have the following trivial inequality:

$$\|G(X_t, \nabla f(X_t), \gamma)\|^2 \leq 2\|G(X_t, V_t, \gamma)\|^2 + 2\|G(X_t, V_t, \gamma) - G(X_t, \nabla f(X_t), \gamma)\|^2. \tag{40}$$

The first term on the right hand side of (40) can be bounded based on the property of retraction in Assumption 1(iii):

$$\|G(X_t, V_t, \gamma)\|^2 = \frac{1}{\gamma^2}\|X_t - \mathrm{Retr}_{X_t}(\gamma\zeta_t)\|^2 \leq M_1^2\|\zeta_t\|^2. \tag{41}$$

The second term on the right hand side of (40) can be bounded as:

$$\|G(X_t, V_t, \gamma) - G(X_t, \nabla f(X_t), \gamma)\|^2 \leq \frac{2\|X_t - \mathrm{Retr}_{X_t}(\gamma\zeta_t)\|^2}{\gamma^2} + \frac{2}{\gamma^2}\|X_t - \mathrm{Retr}_{X_t}(\gamma\xi_t)\|^2, \tag{42}$$

which further implies

$$\|G(X_t, V_t, \gamma) - G(X_t, \nabla f(X_t), \gamma)\|^2 \leq 2M_1^2(\|\zeta_t\|^2 + \|\xi_t\|^2) \leq 2M_1^2(3\|\zeta_t\|^2 + 2\|\xi_t - \zeta_t\|^2). \tag{43}$$

The optimality conditions of (26) and (27) are given by (see [32]):

$$0 \in V_t + \frac{1}{\gamma}\zeta_t + \mathrm{Proj}_{T_{X_t}\mathcal{M}}\partial h(X_t + \zeta_t), \tag{44}$$

$$0 \in \nabla f(X_t) + \frac{1}{\gamma}\xi_t + \mathrm{Proj}_{T_{X_t}\mathcal{M}}\partial h(X_t + \xi_t). \tag{45}$$

Let $X_t^\dagger = X_t + \xi_t$ and $X_t^+ = X_t + \zeta_t$. (44) and (45) indicate that for any $\mathbf{u} \in \mathrm{T}_{X_t}\mathcal{M}$, there exist $p^+ \in \partial h(X_t^+)$ and $p^\dagger \in \partial h(X_t^\dagger)$ such that

$$\langle \tfrac{1}{\gamma}\zeta_t + V_t + \mathrm{Proj}_{T_{X_t}\mathcal{M}}p^+, \mathbf{u} - X_t^+\rangle \geq 0, \tag{46}$$

$$\langle \tfrac{1}{\gamma}\xi_t + \nabla f(X_t) + \mathrm{Proj}_{T_{X_t}\mathcal{M}}p^\dagger, \mathbf{u} - X_t^\dagger\rangle \geq 0. \tag{47}$$

Let $\mathbf{u} = X_t^\dagger$ in (46) and $\mathbf{u} = X_t^+$ in (47). Since $X_t^\dagger - X_t^+$ and $X_t^+ - X_t^\dagger$ both lie in $\mathrm{T}_{X_t}\mathcal{M}$, we have $\langle \mathrm{Proj}_{T_{X_t}\mathcal{M}}p^+, X_t^+ - X_t^\dagger\rangle = \langle p^+, X_t^+ - X_t^\dagger\rangle$ and $\langle \mathrm{Proj}_{T_{X_t}\mathcal{M}}p^\dagger, X_t^\dagger - X_t^+\rangle = \langle p^\dagger, X_t^\dagger - X_t^+\rangle$. Therefore, (46) and (47) reduce to:

$$\langle \tfrac{1}{\gamma}\zeta_t + V_t + p^+, X_t^\dagger - X_t^+\rangle \geq 0, \tag{48}$$

$$\langle \tfrac{1}{\gamma}\xi_t + \nabla f(X_t) + p^\dagger, X_t^+ - X_t^\dagger\rangle \geq 0. \tag{49}$$

By using the convexity of $h(X)$, we have $\langle p^+, X_t^+ - X_t^\dagger\rangle \geq h(X_t^+) - h(X_t^\dagger)$, and $\langle p^\dagger, X_t^\dagger - X_t^+\rangle \geq h(X_t^\dagger) - h(X_t^+)$. Therefore, (48) and (49) reduce to:

$$\langle V_t, X_t^\dagger - X_t^+\rangle \geq \tfrac{1}{\gamma}\langle \zeta_t, X_t^+ - X_t^\dagger\rangle + h(X_t^+) - h(X_t^\dagger), \tag{50}$$

$$\langle \nabla f(X_t), X_t^+ - X_t^\dagger\rangle \geq \tfrac{1}{\gamma}\langle \xi_t, X_t^\dagger - X_t^+\rangle + h(X_t^\dagger) - h(X_t^+). \tag{51}$$

Summing up (50) and (51) gives: (note that $X_t^\dagger - X_t^+ = \xi_t - \zeta_t$):

$$\|V_t - \nabla f(X_t)\|\|X_t^+ - X_t^\dagger\| \geq \langle V_t - \nabla f(X_t), X_t^\dagger - X_t^+\rangle \geq \frac{1}{\gamma}\langle \xi_t - \zeta_t, X_t^\dagger - X_t^+\rangle = \frac{1}{\gamma}\|X_t^\dagger - X_t^+\|^2, \quad (52)$$

which further implies $\|\xi_t - \zeta_t\| = \|X_t^\dagger - X_t^+\| \leq \gamma\|V_t - \nabla f(X_t)\|$. We hence have:

$$\|G(X_t, V_t, \gamma) - G(X_t, \nabla f(X_t), \gamma)\|^2 \leq 2M_1^2(3\|\zeta_t\|^2 + 2\|\xi_t - \zeta_t\|^2) \leq 6M_1^2\|\zeta_t\|^2 + 4M_1^2\gamma\|V_t - \nabla f(X_t)\|^2,$$

which combining with (40) and (41) completes the proof. □

The following lemma shows the progress of R-ProxSGD in one iteration in terms of objective function value.

**Lemma 13.** *The sequences $\{X_t\}_{t=1}^{T+1}$ and $\{\zeta_t\}_{t=1}^{T}$ generated by R-ProxSGD (Algorithm 1) satisfy the following inequality:*

$$\mathbb{E}[F(X_{t+1}) - F(X_t)] \leq \big(\tilde{L}\eta_t^2 - \frac{1}{\gamma}\eta_t + \frac{1}{2}\big)\mathbb{E}[\|\zeta_t\|^2] + \frac{\eta_t^2\sigma^2}{2|\mathcal{S}_t|}, \quad (53)$$

*where $\tilde{L} = (L_R/2 + L_h M_2)$.*

*Proof.* Denote $X_t^+ = X_t + \eta_t\zeta_t$. Assumptions 1(iii) and 5 yield the following inequalities:

$$h(X_{t+1}) - h(X_t^+) \leq L_h\|X_{t+1} - X_t^+\| \leq L_h M_2\eta_t^2\|\zeta_t\|^2,$$

which together with Lemma 11 and Young's inequality gives

$$F(X_{t+1}) - F(X_t) \leq \left(\frac{L_R\eta_t^2}{2} - \frac{\eta_t^2}{2\gamma} + L_h M_2\eta_t^2 + \frac{1}{2}\right)\|\zeta_t\|^2 + \frac{\eta_t^2}{2}\|\nabla f(X_t) - V_t\|^2 + \phi_t(\eta_t\zeta_t) - \phi_t(0). \quad (54)$$

Taking expectation conditioned on $\mathcal{F}_{t-1}$ to both side of (54), we get:

$$\begin{aligned}\mathbb{E}[F(X_{t+1}) \mid \mathcal{F}_{t-1}] - F(X_t) \leq &\left(\bar{L}\eta_t^2 + \frac{1}{2}\right)\mathbb{E}[\|\zeta_t\|^2 \mid \mathcal{F}_{t-1}] + \frac{\eta_t^2}{2}\mathbb{E}[\|\nabla f(X_t) - V_t\|^2 \mid \mathcal{F}_{t-1}] \\ &+ \mathbb{E}[\phi_t(\eta_t\zeta_t) \mid \mathcal{F}_{t-1}] - \phi_t(0),\end{aligned} \quad (55)$$

where $\bar{L} := \frac{L_R}{2} - \frac{1}{2\gamma} + L_h M_2$. Using Lemma 8 and taking the whole expectation on both sides of (55) completes the proof. □

# C Proof of Theorem 1

We can re-arrange terms in (53) as follows for $0 < \eta_t \leq 1$ (note $|\mathcal{S}_t| = s$ for all $t$):

$$\big(\frac{1}{\gamma}\eta_t - \tilde{L}\eta_t^2 - \frac{1}{2}\big)\mathbb{E}[\|\zeta_t\|^2] \leq \mathbb{E}[F(X_t)] - \mathbb{E}[F(X_{t+1})] + \frac{\eta_t^2\sigma^2}{2s}. \quad (56)$$

If we choose $\gamma$ small enough such that $\gamma \leq \frac{2\eta_t}{2\tilde{L}\eta_t^2 + \eta_t + 1}$ for all $t = 0, \ldots, T-1$, then

$$\frac{1}{\gamma}\eta_t - \tilde{L}\eta_t^2 - \frac{1}{2} \geq \frac{\eta_t}{2}, t = 0, \ldots, T-1. \quad (57)$$

Combining (56) and (57) yields:

$$\frac{1}{2}\mathbb{E}[\|\zeta_t\|^2] \leq \frac{\mathbb{E}[F(X_t)] - \mathbb{E}[F(X_{t+1})]}{\eta_t} + \frac{\eta_t \sigma^2}{2s} \leq \frac{\mathbb{E}[F(X_t)] - \mathbb{E}[F(X_{t+1})]}{\eta_T} + \frac{\eta_t \sigma^2}{2s}, \tag{58}$$

where the second inequality is due to $\eta_t = \eta_0/\sqrt{t+1}$. Denote $\Delta_0 := F(X_0) - F(X^*)$, where $X^*$ is a global optimal to the problem (1). Summing up (58) for $t = 0, \ldots, T-1$ and dividing both sides by $T$, we get:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\zeta_t\|^2] \leq \frac{2\Delta_0}{T\eta_T} + \frac{\sigma^2\sqrt{T}\eta_0}{Ts} \leq \Big(\frac{2\Delta_0}{\eta_0} + \frac{\sigma^2\eta_0}{s}\Big)\frac{1}{\sqrt{T}}, \tag{59}$$

where the first inequality is due to $\mathbb{E}[F(X_0)] - \mathbb{E}[F(X_T)] \leq \Delta_0$. Moreover, (23) indicates that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|V_t - \nabla f(X_t)\|^2 \leq \frac{\sigma^2}{s}. \tag{60}$$

Combining (39), (59) and (60) yields:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|G(X_t, \nabla f(X_t), \gamma)\|^2] \leq 14M_1^2\Big(\frac{2\Delta_0}{\eta_0} + \frac{\sigma^2\eta_0}{s}\Big)\frac{1}{\sqrt{T}} + 8M_1^2\gamma\sigma^2/s,$$

which together with Jensen's inequality and the convexity of $\|\cdot\|^2$ implies that:

$$\begin{aligned}
&\left(\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t, \nabla f(X_t), \gamma)\|\right]\right)^2 \\
\leq &\mathbb{E}\left[\left(\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t, \nabla f(X_t), \gamma)\|\right)^2\right] \\
\leq &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|G(X_t, \nabla f(X_t), \gamma)\|^2] \\
\leq &14M_1^2\Big(\frac{2\Delta_0}{\eta_0} + \frac{\sigma^2\eta_0}{s}\Big)\frac{1}{\sqrt{T}} + 8M_1^2\gamma\sigma^2/s.
\end{aligned} \tag{61}$$

By setting $s = (16M_1^2\gamma\sigma^2)\epsilon^{-2}$, we know that as long as

$$T \geq 4\left(14M_1^2\Big(\frac{2\Delta_0}{\eta_0} + \frac{\sigma^2\eta_0}{s}\Big)\right)^2\epsilon^{-4}, \tag{62}$$

the right hand side of (61) is upper bounded by $\epsilon^2$, that is:

$$\left(\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t, \nabla f(X_t), \gamma)\|\right]\right) \leq \epsilon. \tag{63}$$

Therefore, for an index $\nu$ that is uniformly sampled from $\{0, \ldots, T-1\}$, we have $\mathbb{E}[\|G(X_\nu, \nabla f(X_\nu), \gamma)\|] \leq \epsilon$, i.e., $X_\nu$ is an $\epsilon$-stochastic stationary point of problem (1). Condition (62) shows that the number of iterations needed by R-ProxSGD for obtaining an $\epsilon$-stochastic stationary point is $T = O(\epsilon^{-4})$, which immediately implies that the total IFO complexity is $O(\epsilon^{-6})$. This completes the proof of Theorem 1.

## D Necessary Lemma for Proving Theorem 2

Similar to Lemma 13, the following lemma gives the progress of R-ProxSPB in one iteration in terms of the objective function value.

**Lemma 14.** *The sequences $\{X_t\}_{t=1}^{T+1}$ and $\{\zeta_t\}_{t=1}^{T}$ generated by R-ProxSPB (Algorithm 2) satisfy the following inequality:*

$$\mathbb{E}[F(X_{t+1}) - F(X_t)] \leq \eta\big(\tilde{L}\eta - \frac{1}{\tilde{\gamma}}\big)\mathbb{E}[\|\zeta_t\|^2] + I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|} + \sum_{i=(n_t-1)q}^{t}\frac{\Theta^2\eta^3 c_E}{2|\mathcal{S}_t^2|}\mathbb{E}[\|\zeta_i\|^2], \quad (64)$$

*where $\tilde{L} = L_R/2 + L_h M_2$ and $\tilde{\gamma} = \frac{2\gamma}{2-\gamma}$.*

*Proof.* Similar to the proof of Lemma 13, by using Lemma 11, Assumptions 1(iii) and 5, and Young's inequality, we have:

$$\begin{aligned} &F(X_{t+1}) - F(X_t) \quad (65)\\ \leq&(\frac{L_R\eta^2}{2} - \frac{\eta^2}{2\gamma} + L_h M_2\eta^2 + \frac{\eta}{2})\|\zeta_t\|^2 + \frac{\eta}{2}\|V_t - \nabla f(X_t)\|^2 + \phi_t(\eta\zeta_t) - \phi_t(0). \end{aligned}$$

Taking conditional expectation on both sides of (65) conditioned on $\mathcal{F}_{t-1}$, we have:

$$\begin{aligned} &\mathbb{E}[F(X_{t+1}) \mid \mathcal{F}_{t-1}] - F(X_t) \quad (66)\\ \leq&\eta(\bar{L}\eta + \frac{1}{2})\mathbb{E}[\|\zeta_t\|^2 \mid \mathcal{F}_{t-1}] + \frac{\eta}{2}\mathbb{E}[\|V_t - \nabla f(X_t)\|^2 \mid \mathcal{F}_{t-1}] + \mathbb{E}[\phi_t(\eta\zeta_t) \mid \mathcal{F}_{t-1}] - \phi_t(0), \end{aligned}$$

where $\bar{L} := \frac{L_R}{2} - \frac{1}{2\gamma} + L_h M_2$. Taking the whole expectation on both sides of (66) yields:

$$\begin{aligned} &\mathbb{E}[F(X_{t+1}) - F(X_t)]\\ \overset{(i)}{\leq}&\eta\big(\tilde{L}\eta - \frac{1}{\tilde{\gamma}}\big)\mathbb{E}[\|\zeta_t\|^2] + \frac{\eta}{2}\mathbb{E}[\|V_t - \nabla f(X_t)\|^2]\\ \overset{(ii)}{\leq}&\eta\big(\tilde{L}\eta - \frac{1}{\tilde{\gamma}}\big)\mathbb{E}[\|\zeta_t\|^2] + I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|} + \sum_{i=t_0}^{t}\frac{\Theta^2\eta}{2|\mathcal{S}_t^2|}\mathbb{E}[\|\mathrm{Retr}_{X_i}^{-1}(X_{i+1})\|^2]\\ \overset{(iii)}{\leq}&\eta\big(\tilde{L}\eta - \frac{1}{\tilde{\gamma}}\big)\mathbb{E}[\|\zeta_t\|^2] + I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|} + \sum_{i=t_0}^{t}\frac{\Theta^2\eta^3 c_E}{2|\mathcal{S}_t^2|}\mathbb{E}[\|\zeta_i\|^2], \end{aligned}$$

where (i) is from Lemma 8, (ii) is due to Lemma 7, and (iii) is due to the update $X_{t+1} = \mathrm{Retr}_{X_t}(\eta\zeta_t)$ and the Assumption 1(ii). This completes the proof. □

## E Proof of Theorem 2

Let $n_t = \lceil t/q \rceil, t_0 = (n_t - 1)q$. Since the length of recursion of $V_t$ is $q$ in R-ProxSPB, we calculate the telescoping sum of (64) from $t_0 = (n_t - 1)q$ to $t + 1 \leq n_t q$:

$$\begin{aligned} &\mathbb{E}[F(X_{t+1}) - F(X_{t_0})] \quad (67)\\ \leq&\eta\left(\tilde{L}\eta - \frac{1}{\tilde{\gamma}}\right)\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2] + \sum_{i=t_0}^{t} I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|} + \frac{\Theta^2 c_E\eta^3}{2|\mathcal{S}_t^2|}\sum_{j=t_0}^{t}\sum_{i=t_0}^{j}\mathbb{E}[\|\zeta_i\|^2]. \end{aligned}$$

By noting $\sum_{j=t_0}^{t}\sum_{i=t_0}^{j}\mathbb{E}[\|\zeta_i\|^2] \leq q\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2]$, $\tilde{\gamma} = 2\gamma/(2-\gamma) = 1/2$ (since $\gamma = 2/5$), and $|\mathcal{S}_t^2| = q$ for all $t$, (67) can be reduced to:

$$\mathbb{E}[F(X_{t+1}) - F(X_{t_0})] \leq \sum_{i=t_0}^{t} I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|} + \eta\left(\frac{c_E\Theta^2\eta^2}{2} + \tilde{L}\eta - 2\right)\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2]. \tag{68}$$

Moreover, the choice of $\eta$: $0 < \eta \leq (-\tilde{L} + \sqrt{\tilde{L}^2 + 2c_E\Theta^2})/(c_E\Theta^2)$ guarantees that

$$\frac{c_E\Theta^2\eta^2}{2} + \tilde{L}\eta - 2 \leq -1.$$

Therefore, (68) reduces to

$$\eta\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2] \leq -\mathbb{E}[F(X_{t+1}) - F(X_{t_0})] + \sum_{i=t_0}^{t} I\{|\mathcal{S}_t^1| < n\}\frac{\eta\sigma^2}{2|\mathcal{S}_t^1|}. \tag{69}$$

### E.1 Finite-sum case

In the finite-sum case, we have $|\mathcal{S}_t^1| = n$, which implies that $I\{|\mathcal{S}_t^1| < n\} = 0$. Therefore, (69) reduces to:

$$\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2] \leq \frac{\mathbb{E}[F(X_{(n_t-1)q}) - F(X_{t+1})]}{\eta}. \tag{70}$$

We now calculate the telescoping sum for (70) for all length-$q$ epochs that $t+1 = q, 2q, ..., Kq$ ($K = \lfloor\frac{T}{q}\rfloor$) and the telescoping sum from $t = Kq+1$ to $T-1$. This results in:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\zeta_t\|^2] = \frac{1}{T}\left(\sum_{t=0}^{Kq-1}\mathbb{E}[\|\zeta_t\|^2] + \sum_{t=Kq}^{T-1}\mathbb{E}[\|\zeta_t\|^2]\right) \leq \frac{\mathbb{E}[F(X_0) - F(X_T)]}{T\eta} \leq \frac{\Delta_0}{\eta T}. \tag{71}$$

Moreover, Lemma 7 yields that

$$\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \leq \sum_{i=t_0}^{t-1}\frac{\Theta^2 c_E^2\eta^2}{q}\mathbb{E}[\|\zeta_i\|^2]. \tag{72}$$

Summing up (72) over $t = 0, \ldots, T-1$, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \leq \frac{1}{T}\sum_{t=1}^{T-1}\sum_{i=t_0}^{t-1}\frac{\Theta^2 c_E^2\eta^2}{q}\mathbb{E}[\|\zeta_i\|^2] \leq \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=t_0}^{t}\frac{\Theta^2 c_E^2\eta^2}{q}\mathbb{E}[\|\zeta_i\|^2]. \tag{73}$$

Note that $\sum_{j=t_0}^{t}\sum_{i=t_0}^{j}\mathbb{E}[\|\zeta_i\|^2] \leq q\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2]$. This together with (73) yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|V_t - \nabla f(X_t)\|^2] \leq \frac{\Theta^2 c_E^2\eta^2}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\zeta_t\|^2]. \tag{74}$$

Now combining Lemma 12, (71) and (74), we have that:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|G(X_t, \nabla f(X_t), \gamma)\|^2] \leq 14M_1^2\frac{\Delta_0}{\eta T} + 8M_1^2\gamma\Theta^2 c_E^2\eta\frac{\Delta_0}{T}. \tag{75}$$

Again, using Jensen's inequality and the convexity of $\|\cdot\|^2$, (75) gives:

$$\begin{aligned}
&\left(\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t,\nabla f(X_t),\gamma)\|\right]\right)^2 \\
\leq&\mathbb{E}\left[\left(\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t,\nabla f(X_t),\gamma)\|\right)^2\right] \\
\leq&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|G(X_t,\nabla f(X_t),\gamma)\|^2] \\
\leq&14M_1^2\frac{\Delta_0}{\eta T}+8M_1^2\gamma\Theta^2c_E^2\eta\frac{\Delta_0}{T}.
\end{aligned} \tag{76}$$

Hence, we know that as long as

$$T\geq\left(14M_1^2\frac{\Delta_0}{\eta}+8M_1^2\gamma\Theta^2c_E^2\eta\Delta_0\right)\epsilon^{-2}, \tag{77}$$

the right hand side of (76) is upper bounded by $\epsilon^2$, which implies that if index $\nu$ is uniformly sampled from $\{0,\ldots,T-1\}$, then

$$\mathbb{E}\left[\|G(X_\nu,\nabla f(X_\nu),\gamma)\|\right]\leq\epsilon.$$

That is, $X_\nu$ is an $\epsilon$-stochastic stationary point of problem (1). Equation (77) then implies that the number of iterations needed by R-ProxSPB for obtaining an $\epsilon$-stochastic stationary point of problem (1) in the finite-sum case is $T=O(\epsilon^{-2})$. Furthermore, the IFO complexity of R-ProxSPB under the finite-sum setting is:

$$\lceil T/q\rceil\cdot|\mathcal{S}_t^1|+T\cdot|\mathcal{S}_t^2|\leq\frac{T+q}{q}n+T\sqrt{n}=\mathcal{O}(\sqrt{n}\epsilon^{-2}+n), \tag{78}$$

where the equality is due to $q=\sqrt{n}$.

### E.2 Online setting

In the online case, $I\{|\mathcal{S}_t^1|<n\}=1$. Since $|\mathcal{S}_t^1|$ is the same for all $t$, we denote $s:=|\mathcal{S}_t^1|$. In this case, (69) reduces to

$$\sum_{i=t_0}^{t}\mathbb{E}[\|\zeta_i\|^2]\leq\frac{\mathbb{E}[F(X_{(n_t-1)q})-F(X_{t+1})]}{\eta}+\frac{1}{2}\sum_{i=t_0}^{t}\frac{\sigma^2}{|\mathcal{S}_t^1|}. \tag{79}$$

We calculate the telescoping sum for (79) for all length-$q$ epochs that $t+1=q,2q,\ldots,Kq$ ($K=\lfloor\frac{T}{q}\rfloor$) and the telescoping sum from $t=Kq$ to $T-1$. This gives:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\zeta_t\|^2]=\frac{1}{T}\left(\sum_{t=0}^{Kq-1}\mathbb{E}[\|\zeta_t\|^2]+\sum_{t=Kq}^{T-1}\mathbb{E}[\|\zeta_t\|^2]\right)\leq\frac{\Delta_0}{\eta T}+\frac{\sigma^2}{2s}. \tag{80}$$

Note that Lemma 7 gives:

$$\mathbb{E}[\|V_t-\nabla f(X_t)\|^2]\leq\frac{\sigma^2}{s}+\sum_{i=t_0}^{t-1}\frac{\Theta^2c_E^2\eta^2}{|\mathcal{S}_t^2|}\mathbb{E}[\|\zeta_i\|^2],$$

which further implies:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|V_t-\nabla f(X_t)\|^2]\leq\frac{\sigma^2}{s}+\frac{\Theta^2c_E^2\eta^2}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\zeta_t\|^2]. \tag{81}$$

Now combining Lemma 12, (80) and (81), we have that:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|G(X_t,\nabla f(X_t),\gamma)\|^2]\leq\left(14M_1^2+8M_1^2\gamma\Theta^2c_E^2\eta^2\right)\frac{\Delta_0}{\eta T}+M_1^2\frac{59\sigma^2}{5s}. \tag{82}$$

Again, using Jensen's inequality and the convexity of $\|\cdot\|^2$, (82) gives:

$$\begin{aligned}&\left(\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t,\nabla f(X_t),\gamma)\|\right]\right)^2 \\ \leq&\mathbb{E}\left[\left(\frac{1}{T}\sum_{t=0}^{T-1}\|G(X_t,\nabla f(X_t),\gamma)\|\right)^2\right] \\ \leq&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|G(X_t,\nabla f(X_t),\gamma)\|^2] \\ \leq&\left(14M_1^2+8M_1^2\gamma\Theta^2c_E^2\eta^2\right)\frac{\Delta_0}{\eta T}+M_1^2\frac{59\sigma^2}{5s}.\end{aligned} \tag{83}$$

Now, by choosing

$$T=\left(\frac{2(14M_1^2+8M_1^2\gamma\Theta^2c_E^2\eta^2)\Delta_0}{\eta}\right)\epsilon^{-2},\quad\text{and}\quad s=\frac{118M_1^2\sigma^2}{5}\epsilon^{-2}, \tag{84}$$

we know that the right hand side of (83) is equal to $\epsilon^2$, which implies that if index $\nu$ is uniformly sampled from $\{0,\ldots,T-1\}$, then

$$\mathbb{E}\left[\|G(X_\nu,\nabla f(X_\nu),\gamma)\|\right]\leq\epsilon.$$

That is, $X_\nu$ is an $\epsilon$-stochastic stationary point of problem (1). Equation (84) then implies that the number of iterations needed by R-ProxSPB for obtaining an $\epsilon$-stochastic stationary point of problem (1) in the finite-sum case is $T=O(\epsilon^{-2})$, and moreover, this needs to require the batch size $|\mathcal{S}_t^1|=s=O(\epsilon^{-2})$ for all $t$. Furthermore, the IFO complexity of R-ProxSPB under the online setting is given by:

$$\lceil T/q\rceil\cdot|\mathcal{S}_t^1|+T\cdot|\mathcal{S}_t^2|\leq\frac{T+q}{q}O(\epsilon^{-2})+Tq=O(\epsilon^{-3}),$$

where the equality is due to $q=\epsilon^{-1}$.

# References


[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[3] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[4] Nicolas Boumal and Pierre-Antoine Absil. RTRMC: A riemannian trust-region method for low-rank matrix completion. In *Advances in neural information processing systems*, pages 406–414, 2011.

[5] Nicolas Boumal, Pierre-Antoine Absil, and Coralia Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, 2018.

[6] L. Cambier and P.-A. Absil. Robust low-rank matrix completion by Riemannian optimization. *SIAM J. Sci. Comput.*, 38(5):S440–S460, 2016.

[7] S. Chen, S. Ma, L. Xue, and H. Zou. An alternating manifold proximal gradient method for sparse PCA and sparse CCA. *https://arxiv.org/abs/1903.11576*, 2019.

[8] Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang. Proximal gradient method for manifold optimization. *arXiv preprint arXiv:1811.00980*, 2018.

[9] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.

[10] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *NeurIPS*, 2019.

[11] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353 (electronic), 1999.

[12] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[13] David R Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. *Machine Learning*, 83(3):331–353, 2011.

[14] S Hosseini and MR Pouryayevali. Generalized gradients and characterization of epi-lipschitz sets in Riemannian manifolds. *Nonlinear Analysis: Theory, Methods & Applications*, 74(12):3884–3895, 2011.

[15] M. Huang, S. Ma, and L. Lai. Robust low-rank matrix completion via an alternating manifold proximal gradient continuation method. In *submitted to ICML*, 2020.

[16] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[17] Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian stochastic recursive gradient algorithm with retraction and vector transport and its convergence analysis. In *International Conference on Machine Learning*, pages 2521–2529, 2018.

[18] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[19] Xiao Li, Shixiang Chen, Zengde Deng, Qing Qu, Zhihui Zhu, and Anthony Man Cho So. Nonsmooth optimization over stiefel manifold: Riemannian subgradient methods. *arXiv preprint arXiv:1911.05047*, 2019.

[20] S. Ma and N. S. Aybat. Efficient optimization algorithms for robust principal component analysis and its variants. *Proceedings of the IEEE*, 106(8):1411–1426, 2018.

[21] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[22] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). *Technical report*, 1996.

[23] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takac. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621. JMLR. org, 2017.

[24] Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. Proxsarah: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1902.05679*, 2019.

[25] Lorenzo Rosasco, Silvia Villa, and Bang Công Vũ. Convergence of stochastic proximal gradient algorithm. *arXiv preprint arXiv:1403.5074*, 2014.

[26] C. Wang and Y. M. Lu. Online learning for sparse PCA in high dimensions: Exact dynamics and phase transitions. *https://arxiv.org/pdf/1609.02191.pdf*, 2016.

[27] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

[28] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

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

[30] Xiantao Xiao, Yongfeng Li, Zaiwen Wen, and Liwei Zhang. A regularized semi-smooth newton method with projection steps for composite convex programs. *Journal of Scientific Computing*, 76(1):364–389, 2018.

[31] W. Yang and H. Xu. Streaming sparse principal component analysis. In *ICML*, 2015.

[32] Wei Hong Yang, Lei-Hong Zhang, and Ruyi Song. Optimality conditions for the nonlinear programming problems on riemannian manifolds. *Pacific Journal of Optimization*, 10(2):415–434, 2014.

[33] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638, 2016.

[34] Jingzhao Zhang, Hongyi Zhang, and Suvrit Sra. R-spider: A fast riemannian stochastic optimization algorithm with curvature independent rate. *arXiv preprint arXiv:1811.04194*, 2018.

[35] Pan Zhou, Xiao-Tong Yuan, and Jiashi Feng. Faster first-order methods for stochastic non-convex optimization on riemannian manifolds. *TPAMI*, 2019.

[36] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15(2):265–286, 2006.