8-2019

Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds

Pan ZHOU
Singapore Management University, panzhou@smu.edu.sg

Xiao-Tong YUAN

Shuicheng YAN

Jiashi FENG

Citation

ZHOU, Pan; YUAN, Xiao-Tong; YAN, Shuicheng; and FENG, Jiashi. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. (2019). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 43, (2), 459-472.
Available at: https://ink.library.smu.edu.sg/sis_research/8990

Faster First-Order Methods for Stochastic Non-Convex Optimization on Riemannian Manifolds

Pan Zhou, Xiao-Tong Yuan, *Member, IEEE,* Shuicheng Yan, *Fellow, IEEE*, Jiashi Feng

Abstract—First-order non-convex Riemannian optimization algorithms have gained recent popularity in structured machine learning problems including principal component analysis and low-rank matrix completion. The current paper presents an efficient Riemannian Stochastic Path Integrated Differential EstimatoR (R-SPIDER) algorithm to solve the finite-sum and online Riemannian non-convex minimization problems. At the core of R-SPIDER is a recursive semi-stochastic gradient estimator that can accurately estimate Riemannian gradient under not only exponential mapping and parallel transport, but also general retraction and vector transport operations. Compared with prior Riemannian algorithms, such a recursive gradient estimation mechanism endows R-SPIDER with higher computational efficiency in first-order oracle complexity. Specifically, for finite-sum problems with n components, R-SPIDER is proved to converge to an ϵ-accuracy stationary point within $\mathcal{O}(\min(n+\frac{\sqrt{n}}{\epsilon^2}, \frac{1}{\epsilon^3}))$ stochastic gradient evaluations, beating the best-known complexity $\mathcal{O}(n+\frac{1}{\epsilon^4})$; for online optimization, R-SPIDER is shown to converge with $\mathcal{O}(\frac{1}{\epsilon^3})$ complexity which is, to the best of our knowledge, the first non-asymptotic result for online Riemannian optimization. For the special case of gradient dominated functions, we further develop a variant of R-SPIDER with improved linear rate of convergence. Extensive experimental results demonstrate the advantage of the proposed algorithms over the state-of-the-art Riemannian non-convex optimization methods.

Index Terms—Riemannian Optimization, Stochastic Variance-Reduced Algorithm, Non-convex Optimization, Online Learning

1 Introduction

Riemannian optimization problems have received broad interests in high-dimensional statistical learning [1]–[3], signal processing [4], [5] and computer vision [6]–[8]. In this paper, we are particularly interested in the following *finite-sum* or *online* Riemannian non-convex minimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x}) := \begin{cases} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) & \text{(finite-sum);} \\ \mathbb{E}[f(\boldsymbol{x};\pi)] & \text{(online),} \end{cases} \tag{1}$$

where $f : \mathcal{M} \mapsto \mathbb{R}$ is a smooth non-convex loss function on a Riemannian manifold $\mathcal{M}$. For the finite-sum problem, each individual sample has an associated loss $f_i(\boldsymbol{x})$; while under the online setting, the stochastic component $f(\boldsymbol{x};\pi)$ is indexed by a random variable π. Such a formulation encapsulates many important structural data analysis problems including principle component analysis (PCA) [9], independent component analysis [10], dictionary learning [1], [2], low-rank matrix/tensor completion/recovery [3]–[6], [8], Gaussian mixture models [7], to name a few. These wealth applications have boosted the development of general-purpose algorithms for solving (1).

One classic method [11]–[13] is to view (1) as a constrained optimization problem in the ambient Euclidean space. Accordingly, the optimizer, such as stochastic gradient descent [12], [13], alternatively minimizes the objective $f(\boldsymbol{x})$ without the constraint and projects the current solution onto the Riemannian $\mathcal{M}$. But in large-scale optimization problems, computing the projection onto certain manifolds (e.g., the one for positive-definite matrices) is rather expensive [14], limiting the usage of such methods.

As an appealing alternative, the Riemannian optimization methods [14]–[20] have recently gained wide attention. In contrast to the Euclidean-projection based methods that alternatively perform variable update and projection, the Riemannian methods directly move the iterative solution along a geodesic path towards the optimum and thus better preserve the geometric structure of the problem [14], [16]. Riemannian gradient descent (R-GD) is a classic example. At each iteration, it moves the iteration along the geodesic path decided by the Riemannian gradient $\nabla f(\boldsymbol{x}_k)$ and enjoys provable sublinear rate of convergence for geodesically convex problems [16]. Later, to avoid the expensive full gradient computation in R-GD, stochastic Riemannian optimization algorithms [14], [17]–[20] were developed that leverage the decomposable structure of problem (1) to compute gradient efficiently. For instance, Bonnabel *et al.* [17] proposed Remannian stochastic GD (R-SGD) that only evaluates gradient of one (or a mini-batch of) randomly selected sample for variable update per iteration. Afterwards, more stable and efficient variance-reduced Riemannian algorithms are developed. For instance, Riemannian stochastic variance-reduced gradient (R-SVRG) algorithm [14], [20] and Riemannian stochastic recursive gradient (R-SRG) algorithm [18] respectively adapt the variance-reduced techniques [21]–[23] into R-SGD for solving problem (1) more efficiently.

1.1 Motivation

Recently, Fang *et al.* [24] proposed the Stochastic Path Integrated Differential EstimatoR (SPIDER) method for non-convex finite-sum/online optimization in Euclidean space, which has provably substantially lower first-order oracle complexity than SVRG [21]

- *P. Zhou, S. Yan and J. Feng are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (email: pzhou@u.nus.edu, eleyans@nus.edu.sg, elefjia@nus.edu.sg).*
- *X.-T. Yuan (✉) is with the School of Automation at Nanjing University of Information Science & Technology, China (email: xtyuan1980@gmail.com).*

and SRG [22], [23], and is confirmed to be nearly optimal in certain large-scale learning settings. Inspired by this record-braking advance, we are interested in the potential of generalizing SPIDER to Riemannian manifold to beat the prior state-of-the-art non-convex Riemannian optimization algorithms including R-SVRG [14], [20] and R-SRG [18] in computational complexity.

In addition to improving the first-order oracle complexity, we are simultaneously interested in analyzing non-convex Riemannian optimization methods with general retraction and vector transport operations beyond *exponential mapping* and *parallel transport* which are only effective on a limited number of manifolds. For an instance, parallel transport has no closed-form expression in Stiefel and fixed-rank manifolds and thus is computationally daunting in these cases [18]. In contrast, the QR based vector transport can handle these two manifolds more efficiently [4]. As another example, rigging transport is more preferable when the dimension d of the sub-mannifold of an m-dimensional Euclidean space is much larger than the codimension $(m-d)$ [25]. For non-convex optimization over Grassmann manifolds, Fig. 1 demonstrates the advantage of our algorithm implementation with polar retraction/transport over the implementation with exponential mapping and parallel transport: both implementations have similar first-order oracle computational efficiency while the former is considerably more efficient than the latter in overall computation time. Therefore, it is desirable to provide a unified convergence analysis of non-convex Riemannian optimization methods implemented with computationally efficient retraction and vector transport instead of the more restrictive exponential mapping and parallel transport.

1.2 Overview of our algorithm and results

In this paper, we propose the *Riemannian Stochastic Path Integrated Differential EstimatoR* (R-SPIDER) to efficiently solve the non-convex Riemannian minimization problem (1) under general retraction and vector transport. Inspired by SPIDER, R-SPIDER employs a recursive estimation to track the history full gradients with significantly reduced computational cost. Specifically, for a proper positive integer p, at each iteration k with $\mod(k,p)\equiv 0$, R-SPIDER first samples a large data batch $\mathcal{S}_1$ and estimates the initial full Riemannian gradient $\nabla f(\boldsymbol{x}_k)$ as $\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_1}(\boldsymbol{x}_k) = \frac{1}{|\mathcal{S}_1|}\sum_{i\in\mathcal{S}_1} f_i(\boldsymbol{x}_k)$. Then at each of the next $p-1$ iterations, it samples a smaller mini-batch $\mathcal{S}_2$ and estimates $\nabla f(\boldsymbol{x}_k)$:

$$\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_2}(\boldsymbol{x}_k) - \Gamma_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k}\left(\nabla f_{\mathcal{S}_2}(\boldsymbol{x}_{k-1}) - \widetilde{\boldsymbol{v}}_{k-1}\right), \qquad (2)$$

where the vector transport $\Gamma_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y})$ (as defined in Section 2) transports $\boldsymbol{y}$ from the tangent space at $\boldsymbol{x}$ to that at the point $\boldsymbol{z}$. Next, the variable is updated via the normalized gradient decent $\boldsymbol{x}_{k+1} = \mathrm{R}_{\boldsymbol{x}_k}\big(-\eta_k \frac{\widetilde{\boldsymbol{v}}_k}{\|\widetilde{\boldsymbol{v}}_k\|}\big)$ where the retraction $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{y})$ (as defined in Section 2) moves $\boldsymbol{x}$ to $\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{y})$ along a geodesic curve decided by $\boldsymbol{y}$. By carefully setting the learning rate η_k and mini-batch sizes of $\mathcal{S}_1$ and $\mathcal{S}_2$, R-SPIDER only requires a necessary number of samples for accurately estimating Riemannian gradient and sufficiently decreasing the objective at each iteration. Consequently, R-SPIDER achieves sharper bounds of incremental first order oracle (IFO, see Definition 3) complexity than state-of-the-arts as summarized in Table 1.

For the finite-sum setting of problem (1) with general non-convex functions, the IFO complexity of R-SPIDER with *vector transport* to achieve $\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|\right] \le \epsilon$ is $\mathcal{O}\big(\min\big(n+\frac{\Theta\sqrt{n}}{\epsilon^2}, \frac{L\sigma}{\epsilon^3}\big)\big)$ which matches the lower IFO complexity bound in Euclidean space [24] and is also faster than R-SRG by a factor of $\mathcal{O}\big(\frac{1}{\epsilon}\big)$.

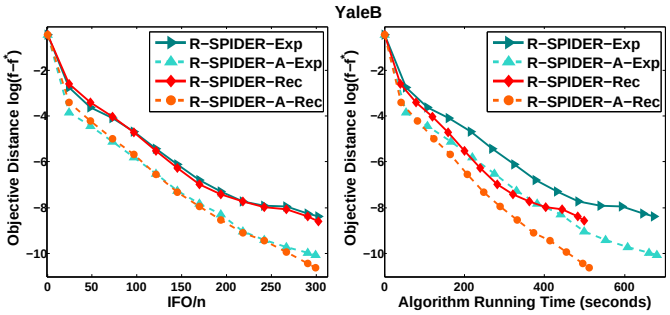


Fig. 1: Comparison between R-SPIDER using exponential mapping and parallel transport (R-SPIDER-Exp for short) and R-SPIDER using polar retraction and vector transport (R-SPIDER-Rec) on the low-rank matrix completion problem. R-SPIDER-Rec and its learning-rate adaptive version, R-SPIDER-A-Rec, respectively achieve very similar IFO complexity as R-SPIDER-Exp and R-SPIDER-A-Exp, but run much faster than R-SPIDER-Exp and R-SPIDER-A-Exp in terms of the algorithm execution time.

Such results are missing in R-SVRG. Under the particular *parallel transport*, the IFO complexity bounds of R-SRG and R-SVRG are $\mathcal{O}\big(n+\frac{L^2}{\epsilon^4}\big)$ and $\mathcal{O}\big(n+\frac{\zeta n^{2/3}}{\epsilon^2}\big)$, respectively. It can be verified that R-SPIDER improves over R-SRG by a factor of $\mathcal{O}\big(\frac{1}{\epsilon}\big)$ and R-SVRG by a factor $\mathcal{O}\big(n^{1/6}\big)$.

When $f(\boldsymbol{x})$ is a τ-gradient dominated function with finite-sum structure, R-SPIDER with vector transport enjoys the IFO complexity of $\mathcal{O}\big(\min\big((n+\tau\Theta\sqrt{n})\log\big(\frac{1}{\epsilon}\big), \frac{\tau\Theta\sigma}{\epsilon}\big)\big)$. So compared with R-SRG with complexity bound $\mathcal{O}\big((n+\tau^2\Theta^2)\log\big(\frac{1}{\epsilon}\big)\big)$, R-SPIDER is more efficient in large-sample-moderate-accuracy settings, e.g., in cases when n dominates $1/\epsilon$. This conclusion also holds for parallel transport. Compared with R-SVRG, R-SPIDER improves the complexity bound by a factor of $\mathcal{O}\big(n^{1/6}\big)$.

For the online version of problem (1), we respectively establish the IFO complexity bounds $\mathcal{O}\big(\frac{\kappa\sigma}{\epsilon^3}\big)$ and $\mathcal{O}\left(\frac{\tau\kappa\sigma}{\epsilon}\right)$ for generic non-convex and gradient dominated problems, where $\kappa=\Theta$ under vector transport and $\kappa=L$ for parallel transport. To our best knowledge, these non-asymptotic convergence results are novel to non-convex online Riemannian optimization. Comparatively, Bonnabel *et al.* [17] only provided asymptotic convergence analysis of R-SGD: the iterating sequence generated by R-SGD converges to a critical point when the iteration number approaches to infinity.

Finally, our analysis reveals as a byproduct that R-SPIDER provably benefits from mini-batching. Specifically, our theoretic results imply linear speedups in parallel computing setting for large mini-batch sizes. We are not aware of any similar linear speedup results in the prior Riemannian stochastic algorithms.

This paper is an extension of our previous work [26] which analyzes the convergence behavior of R-SPIDER under the parallel transport for solving problem (1). Compared with its short version, this paper makes the following changes. 1) For both non-convex and gradient dominated problems, it extends R-SPIDER from exponential mapping and parallel transport to the general retraction and vector transport, and thus endows R-SPIDER applicable to more general kinds of manifolds. 2) For gradient dominated problems, improved theoretical results are obtained by allowing to use larger constant learning rate and constant mini-batch sizes, in contrast to the small optimization-accuracy-dependent step size and the algorithm-iteration-dependent mini-batch sizes in the previous work. 3) Experimental results under different retraction and vector

TABLE 1: Comparison of IFO complexity for different Riemannian first-order stochastic optimization algorithms on the noncovnex problem (1) under finite-sum and online settings. The ϵ-accuracy solution is measured by the expected gradient norm $\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|\right] \leq \epsilon$. Suppose $\Theta = \max(L_R, \sqrt{L_H^2 + G^2\theta^2})$. Here L_R bounds the second derivative of $f(\mathrm{R}_{\boldsymbol{x}}(t\boldsymbol{\xi}))$ w.r.t. t where $\|\boldsymbol{\xi}\| = 1$. L_H is the upper bound of the spectral norm $\|\nabla^2 f_i(\boldsymbol{x})\|$. L is the gradient Lipschitz constant of $f_i(\boldsymbol{x})$ under exponential mapping. θ is the difference constant between vector transport and parallel transport. G, σ and ζ respectively denote the upper bound of gradient norm, the gradient variance and the curvature parameter of the Riemannian manifold. See details of these parameters in Section 2.

		Non-convex Problem (retraction and vector transport)		Non-convex Problem (exponential mapping and parallel transport)	
		general non-convex	τ-gradient dominated	general non-convex	τ-gradient dominated
Finite-sum	R-SRG [18]	$\mathcal{O}\left(\min\left(n+\frac{\Theta^2}{\epsilon^4}\right)\right)$	$\mathcal{O}\left((n+\tau^2\Theta^2)\log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(n+\frac{L^2}{\epsilon^4}\right)$	$\mathcal{O}\left((n+\tau^2L^2)\log\left(\frac{1}{\epsilon}\right)\right)$
	R-SVRG [14]	—	—	$\mathcal{O}\left(n+\frac{\zeta n^{\frac{2}{3}}}{\epsilon^2}\right)$	$\mathcal{O}\left((n+\tau L\zeta^{\frac{1}{2}}n^{\frac{2}{3}})\log\left(\frac{1}{\epsilon}\right)\right)$
	this work	$\mathcal{O}\left(\min\left(n+\frac{\Theta\sqrt{n}}{\epsilon^2},\frac{\Theta\sigma}{\epsilon^3}\right)\right)$	$\mathcal{O}\left(\min\left((n+\tau\Theta\sqrt{n})\log\left(\frac{1}{\epsilon}\right),\frac{\tau\Theta\sigma}{\epsilon}\right)\right)$	$\mathcal{O}\left(\min\left(n+\frac{L\sqrt{n}}{\epsilon^2},\frac{L\sigma}{\epsilon^3}\right)\right)$	$\mathcal{O}\left(\min\left((n+\tau L\sqrt{n})\log\left(\frac{1}{\epsilon}\right),\frac{\tau L\sigma}{\epsilon}\right)\right)$
Online	this work	$\mathcal{O}\left(\frac{\Theta\sigma}{\epsilon^3}\right)$	$\mathcal{O}\left(\frac{\tau\Theta\sigma}{\epsilon}\right)$	$\mathcal{O}\left(\frac{L\sigma}{\epsilon^3}\right)$	$\mathcal{O}\left(\frac{\tau L\sigma}{\epsilon}\right)$

transport settings are provided to better testify the computational efficiency of R-SPIDER.

1.3 Related work

Riemannian optimization can be traced back to the work [27] which first provided comprehensive background and concepts of this topic. Then based on [27], Absil *et al.* [28] further detailed the Riemannian concepts and developed many Riemannian algorithms, e.g. Riemannian trust region approach. Recently, Zhang *et al.* [16] first established the sublinear rate of convergence of R-GD on geodesically convex problems. Later the Nesterov momentum methods [29] were applied to accelerate the convergence rate of R-GD for geodesically convex functions [15], [30]. To boost the efficiency of R-GD by leveraging the decomposable structure of problem, Bonnabel *et al.* [17] proposed R-SGD and showed the first asymptotic convergence analysis for Riemannian optimization. Though with good efficiency for each iteration, R-SGD converges slowly as it uses decaying learning rate for convergence guarantee due to its gradient variance. Then to tackle this issue, R-SVRG algorithm [14] was developed as an extension of SVRG [21] to Riemannian optimization. Benefiting from the variance-reduced technique, R-SVRG converges more stably and faster than R-SGD. Inspired by the variance-reduced SRG approach [22], [23], R-SRG [18] applies a similar recursion form as in (2) for full Riemannian gradient estimation, and the core difference between R-SRG and ours lies in that R-SPIDER is equipped with gradient normalization while R-SRG is not. There is also a rich body of algorithms customized for specific Riemannian optimization problems, e.g. dictionary learning [1], [2], low-rank matrix/tensor completion [3], [6], [8], Gaussian mixture models [7].

While the conference version [26] of this paper was under review, we were informed the concurrent work by Zhang *et al.* [31] which also generalizes SPIDER to non-convex Riemannian stochastic optimization. Despite sharing similar ideas, our work has the following advantages in algorithm and theory over [31]:

- Our algorithms and analysis are applicable to general retraction and vector transport, while [31] only analyzes the convergence behavior under more restrictive exponential mapping and parallel transport which could be inefficient on some kinds of manifolds, such as Stiefel and fixed-rank manifolds [18].
- Our theoretical computational complexity for general non-convex problem is $\mathcal{O}\big(\min\big(n+\frac{L\sqrt{n}}{\epsilon^2},\frac{L\sigma}{\epsilon^3}\big)\big)$ and is superior to the complexity $\mathcal{O}\big(n+\frac{L\sqrt{n}}{\epsilon^2}\big)$ in [31] for large-scale problem, namely $n > \mathcal{O}(\frac{1}{\epsilon^2})$. For gradient-dominated problems, our work enjoys similar advantages.
- Our algorithm uses a constant mini-batch size $\mathcal{S}_2$ and can select it from 1 to $\mathcal{O}(\min(\sqrt{n},\frac{1}{\epsilon}))$, while Zhang *et. al.* [31] adopt an algorithm-iteration-dependent mini-batch size $\mathcal{S}_2$ which may be very large.

In the meanwhile, the superiority of our proposed R-SPIDER algorithms to the prior state-of-the-arts is not only supported by strong theoretical guarantees but also confirmed by extensive numerical evidences.

2 Preliminaries

Throughout this paper, we assume that the Riemannian manifold $(\mathcal{M}, \mathfrak{g})$ is a real smooth manifold $\mathcal{M}$ equipped with a Riemannian metric $\mathfrak{g}$. We denote the induced inner product $\langle \boldsymbol{y}, \boldsymbol{z}\rangle$ of any two vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ in the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ at the point $\boldsymbol{x}$ as $\langle \boldsymbol{y}, \boldsymbol{z}\rangle = \mathfrak{g}(\boldsymbol{y}, \boldsymbol{z})$, and denote the norm $\|\boldsymbol{y}\|$ as $\|\boldsymbol{y}\| = \sqrt{\mathfrak{g}(\boldsymbol{y}, \boldsymbol{y})}$. Let $\nabla f_i(\boldsymbol{x})$ be the stochastic Riemannian gradient of $f_i(\boldsymbol{x})$ and also be a unbiased estimate to the full Riemannian gradient $\nabla f(\boldsymbol{x})$, i.e. $\mathbb{E}_i[\nabla f_i(\boldsymbol{x})] = \nabla f(\boldsymbol{x})$.

To retract the variable $\boldsymbol{x}$ into the manifold $\mathcal{M}$, we need to define the retraction $\mathrm{R}_{\boldsymbol{x}}$. The retraction $\mathrm{R}_{\boldsymbol{x}} : \mathrm{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$ maps $\boldsymbol{y} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ to $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{z}) \in \mathcal{M}$ with a local rigidity condition that preserves the gradients at $\boldsymbol{x}$ [28], [32]. Namely, for all $\boldsymbol{x} \in \mathcal{M}$ and $\boldsymbol{z} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$, the curve $t \to \mathrm{R}_{\boldsymbol{x}}(t\boldsymbol{z})$ is tangent to $\boldsymbol{z}$ at $t = 0$. $\mathrm{R}_{\boldsymbol{x}}^{-1}$ denotes the inverse operation of the retraction $\mathrm{R}_{\boldsymbol{x}}$ and satisfies that if $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{z}) = \boldsymbol{y}$, then $\mathrm{R}_{\boldsymbol{x}}^{-1}(\boldsymbol{y}) \simeq \boldsymbol{z}$. Exponentially mapping, denoted as $\mathrm{Exp}_{\boldsymbol{x}}$, is a classical instance of retraction. The exponential mapping $\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{y})$ maps $\boldsymbol{y} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ to $\boldsymbol{z} \in \mathcal{M}$ such that there is a geodesic $\gamma(t)$ with $\gamma(0) = \boldsymbol{x}$, $\gamma(1) = \boldsymbol{z}$ and $\dot{\gamma}(0) = \frac{d}{dt}\gamma(t) = \boldsymbol{y}$. Here the geodesic $\gamma(t)$ is a constant speed curve $\gamma : [0, 1] \to \mathcal{M}$ which is locally distance minimized. If there exists a unique geodesic between any two points on $\mathcal{M}$, then the exponential map has an inverse mapping $\mathrm{Exp}_{\boldsymbol{x}}^{-1} : \mathcal{M} \to \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ and the geodesic is the unique shortest path with the geodesic distance $\mathrm{d}(\boldsymbol{x}, \boldsymbol{z}) = \|\mathrm{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{z})\| = \|\mathrm{Exp}_{\boldsymbol{z}}^{-1}(\boldsymbol{x})\|$ between $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{M}$. See more specifical ways of constructing retractions in [28], [32]–[34].

To utilize the historical and current Riemannian gradients, we need to transport the historical gradients into the tangent space of the current point such that these gradients can be linearly combined in one tangent space. For this purpose, we need to define vector transport $\Gamma_{\boldsymbol{x}}^{\boldsymbol{z}} : \mathrm{T}\mathcal{M} \oplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M}, (\boldsymbol{\xi}, \boldsymbol{y}) \mapsto \Gamma_{\boldsymbol{\xi}}(\boldsymbol{y})$ which is

associated with the retraction $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{\xi}) = \boldsymbol{z}$ with $\boldsymbol{\xi}, \boldsymbol{y} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$. It satisfies 1) $\Gamma_{\boldsymbol{\xi}}(\boldsymbol{y}) \in \mathrm{T}_{\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{\xi})}\mathcal{M}$, 2) $\Gamma_{0_{\boldsymbol{x}}}(\boldsymbol{\xi}) = \boldsymbol{\xi}$ for all $\boldsymbol{\xi} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ and 3) $\Gamma_{\boldsymbol{x}}^{\boldsymbol{z}}$ is a linear mapping, *i.e.* $\Gamma_{\boldsymbol{\xi}}(a\boldsymbol{y}_1 + b\boldsymbol{y}_2) = a\Gamma_{\boldsymbol{\xi}}(\boldsymbol{y}_1) + b\Gamma_{\boldsymbol{\xi}}(\boldsymbol{y}_2)$. Intuitively, both $\Gamma_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y})$ and $\Gamma_{\boldsymbol{\xi}}$ represent vector transport from $\boldsymbol{x}$ to $\boldsymbol{z}$ such that $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{\xi}) = \boldsymbol{z}$ and $\boldsymbol{\xi} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$. Parallel transport $\mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}$ is a kind of vector transport. $\mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}} : \mathrm{T}_{\boldsymbol{x}}\mathcal{M} \to \mathrm{T}_{\boldsymbol{z}}\mathcal{M}$ maps $\boldsymbol{y} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ to $\mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y}) \in \mathrm{T}_{\boldsymbol{z}}\mathcal{M}$ while preserving the inner product and norm, i.e., $\langle \boldsymbol{y}_1, \boldsymbol{y}_2 \rangle = \langle \mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y}_1), \mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y}_2) \rangle$ and $\|\boldsymbol{y}\| = \|\mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y})\|$ for $\forall \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$. Similar to [18], [25], [35], we only analyze the isometric vector transport, namely $\|\Gamma_{\boldsymbol{x}}^{\boldsymbol{z}}(\boldsymbol{y})\| = \|\boldsymbol{y}\|$.

Before imposing necessary assumptions on the objective $f(\boldsymbol{x})$, we first define upper-bounded Hessian, based on which smooth property on $f(\boldsymbol{x})$ can be defined.

Definition 1 (Upper-Bounded Hessian [18]). *A function $f(\boldsymbol{x})$ is said to have upper-bounded Hessian in $\mathcal{U} \subset \mathcal{M}$ associated with retraction $R_{\boldsymbol{x}}$, if there exists a constant L such that $\frac{d^2 f(R_{\boldsymbol{x}}(t\boldsymbol{\xi}))}{dt^2} \leq L$ for all $\boldsymbol{x} \in \mathcal{U}$, $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathcal{M}$ with $\|\boldsymbol{\xi}\| = 1$ and t such that $R_{\boldsymbol{x}}(s\boldsymbol{\xi}) \in \mathcal{U}$ for $\forall s \in [0, t]$.*

Then we impose on the loss components $f_i(\boldsymbol{x})$ the assumption of upper-bounded Hessian which is also required in [18], [25], [35] for analyzing vector transport.

Assumption 1 (Upper-bounded Hessian). *Assume each loss $f_i(\boldsymbol{x})$ are twice continuously differentiable. For retraction $R_{\boldsymbol{x}}$, each $f_i(\boldsymbol{x})$ has upper-bounded Hessian with parameter L_R in Definition 1. We also assume each individual Hessian $\nabla^2 f_i(\boldsymbol{x})$ is directly bounded as $\|\nabla^2 f_i(\boldsymbol{x})\| \leq L_H$.*

The variation of Assumption 1 is Lipschitz property on gradient $\nabla f_i(\boldsymbol{x})$ (or smooth condition on the individual objective $f_i(\boldsymbol{x})$) and is also conventionally assumed in analyzing Riemannian stochastic gradient algorithms with parallel transport [14], [31].

Assumption 2 (Geodesically L-gradient-Lipschitz under Exponential Mapping). *Each loss $f_i(\boldsymbol{x})$ is geodesically L-gradient Lipschitz such that $\mathbb{E}_i\|\nabla f_i(\boldsymbol{x}) - P_{\boldsymbol{y}}^{\boldsymbol{x}}(\nabla f_i(\boldsymbol{y}))\|^2 \leq L^2\|Exp_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\|^2$.*

Next, we avoid a bad case in Riemannian optimization. Namely, the sequence $\{\boldsymbol{x}_k\}$ may converge to an optimum $\boldsymbol{x}_*$, while the connecting retraction $\{\mathrm{R}_{\boldsymbol{x}_k}(\boldsymbol{\xi}_k)\}$ does not converge where $\boldsymbol{x}_{k+1} = \mathrm{R}_{\boldsymbol{x}_k}(\boldsymbol{\xi}_k)$ [25], [35]. To see this, in the unit sphere with the exponential retraction, we can have $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ with $\|\boldsymbol{\xi}_k\| = 2\pi$. To resolve this issue, following [18], [25], [35], we assume the neighborhood $\mathcal{U} \subset \mathcal{M}$ of an optimum $\boldsymbol{x}_*$ is a totally retractive neighborhood, namely $\{\mathrm{R}_{\boldsymbol{x}_k}(\boldsymbol{\xi}_k)\} \in \mathcal{U}$, formally stated in Assumption 3.

Assumption 3 (Totally Retractive Neighborhood). *Suppose the sequence $\{\boldsymbol{x}_k\}$ generated by algorithm stay continuously in a small totally retractive neighborhood $\mathcal{U} \subset \mathcal{M}$ of an optimum $\boldsymbol{x}_*$, namely $\{R_{\boldsymbol{x}_k}(\boldsymbol{\xi}_k)\} \in \mathcal{U}$ with $\boldsymbol{x}_{k+1} = R_{\boldsymbol{x}_k}(\boldsymbol{\xi}_k)$.*

For analysis, we also impose certain assumptions on the retraction and vector transport commonly used in [18], [25], [35].

Assumption 4 (Retraction and Transport Properties). *For retraction $R_{\boldsymbol{x}}$, suppose there are two constants c_R and c_E such that 1) $\|R_{\boldsymbol{x}}^{-1}(\boldsymbol{y}) - Exp_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\|^2 \leq c_R\|R_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\|^2$ and 2) $\|R_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\| \leq c_E\|\boldsymbol{z}\|$ if $R_{\boldsymbol{x}}(\boldsymbol{z}) = \boldsymbol{y}$.*
For vector transport Γ, assume it satisfies $\|\Gamma_{\boldsymbol{\xi}} - \Gamma_{R_{\boldsymbol{x}}(\boldsymbol{\xi})}\| \leq c_0\|\boldsymbol{\xi}\|$ and $\|\Gamma_{\boldsymbol{\xi}}^{-1} - \Gamma_{R_{\boldsymbol{x}}(\boldsymbol{\xi})}^{-1}\| \leq c_0\|\boldsymbol{\xi}\|$ for all $\boldsymbol{x}$ and $\boldsymbol{z}$ belonging in a neighborhood $\mathcal{U}$ of a point $\boldsymbol{x}$, where $R_{\boldsymbol{x}}(\boldsymbol{\xi}) = \boldsymbol{z}$, c_0 is a constant, and $\Gamma_{R_{\boldsymbol{x}}}$ denotes the differentiated retraction, i.e. *$\Gamma_{R_{\boldsymbol{x}}(\boldsymbol{\xi})}(\boldsymbol{y}) = DR_{\boldsymbol{x}}(\boldsymbol{\xi})[\boldsymbol{y}]$ for all $\boldsymbol{\xi}, \boldsymbol{y} \in T_{\boldsymbol{x}}\mathcal{M}$.*

Algorithm 1 R-SPIDER ($\boldsymbol{x}_0$, ϵ, η, p, $|\mathcal{S}_1|$, $|\mathcal{S}_2|$)

Input: initialization $\boldsymbol{x}_0$, accuracy ϵ, learning rate η, iteration interval p, mini-batch sizes $|\mathcal{S}_1|$ and $|\mathcal{S}_2|$.
for $k = 0$ to $K - 1$ **do**
  if mod$(k, p) = 0$ **then**
    Draw mini-batch $\mathcal{S}_1$ and compute $\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_1}(\boldsymbol{x}_k)$;
  else
    Draw mini-batch $\mathcal{S}_2$ and compute $\nabla f_{\mathcal{S}_2}(\boldsymbol{x}_k)$;
    $\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_2}(\boldsymbol{x}_k) - \Gamma_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k}(\nabla f_{\mathcal{S}_2}(\boldsymbol{x}_{k-1}) - \widetilde{\boldsymbol{v}}_{k-1})$;
  end if
  $\boldsymbol{x}_{k+1} = \mathrm{R}_{\boldsymbol{x}_k}\left(-\eta_k \frac{\widetilde{\boldsymbol{v}}_k}{\|\widetilde{\boldsymbol{v}}_k\|}\right)$;
end for
Output: $\widetilde{\boldsymbol{x}}$ which is chosen uniformly at random from $\{\boldsymbol{x}_k\}_{k=0}^{K-1}$.

Since the retraction $\mathrm{R}_{\boldsymbol{x}}$ and its inverse operation $\mathrm{R}_{\boldsymbol{x}}^{-1}(\boldsymbol{z})$ are usually first-order approximations to the exponential mapping $\mathrm{Exp}_{\boldsymbol{x}}$ and its inverse optation $\mathrm{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{z})$ respectively, the required assumption can well characterize such relation [18], [28]. Besides, the property of Γ is satisfied for $\Gamma \in C^0$, as derived from the Taylor expansion [18], [25]. Then we assume the stochastic Riemannian gradient and its variance can be bounded as in [14], [18], [31].

Assumption 5 (Bounded Stochastic Riemannian Gradient and Variance). *The gradient of each loss $f_i(\boldsymbol{x})$ is bounded, namely $\|\nabla f_i(\boldsymbol{x})\| \leq G$, and its variance is also upper bounded as $\mathbb{E}_i\|\nabla f_i(x) - \nabla f(x)\|_2^2 \leq \sigma^2$.*

We further introduce the following concept of τ-gradient dominated function [36], [37] which will also be investigated in this paper.

Definition 2 (τ-Gradient Dominated Functions). *$f(\boldsymbol{x})$ is said to be a τ-gradient dominated function if it satisfies $f(\boldsymbol{x}) - f(\boldsymbol{x}_*) \leq \tau\|\nabla f(\boldsymbol{x})\|^2$ for any $\boldsymbol{x} \in \mathcal{M}$, where τ is a universal constant and $\boldsymbol{x}_* = \mathrm{argmin}_{\boldsymbol{x} \in \mathcal{M}} f(\boldsymbol{x})$ is the global minimizer of $f(\boldsymbol{x})$ on the manifold $\mathcal{M}$.*

The following incremental first order oracle (IFO) complexity is usually adopted as the computational complexity measurement for evaluating stochastic optimization algorithms [14], [18]–[20].

Definition 3 (IFO Complexity). *For $f(\boldsymbol{x})$ in problem (1), an IFO takes in an index $i \in [n]$ and a point $\boldsymbol{x}$, and returns the pair $(f_i(\boldsymbol{x}), \nabla f_i(\boldsymbol{x}))$.*

The IFO complexity can well reflect the overall computational performance of a first-order Riemannian algorithm, since objective value and gradient evaluation usually dominate the per-iteration computation.

3 Riemannian SPIDER Algorithm

We first elaborate on the Riemannian SPIDER algorithm, and then analyze its convergence performance for general non-convex problems. For gradient dominated problems, we further develop a variant of R-SPIDER with a linear convergence rate.

3.1 Algorithm

The R-SPIDER method is outlined in Algorithm 1. At its core, R-SPIDER customizes SPIDER to recursively estimate/track the full

Riemannian gradient in a computationally economic way. For each cycle of p iterations, R-SPIDER first samples a large data batch $\mathcal{S}_1$ by with-replacement sampling and views the gradient estimate $\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_1}(\boldsymbol{x}_k) = \frac{1}{|\mathcal{S}_1|}\sum_{i\in\mathcal{S}_1} f_i(\boldsymbol{x}_k)$ as the snapshot gradient. For the next forthcoming $p-1$ iterations, R-SPIDER only samples a smaller mini-batch $\mathcal{S}_2$ and estimates the full Riemannian gradient $\nabla f(\boldsymbol{x}_k)$ as $\widetilde{\boldsymbol{v}}_k = \nabla f_{\mathcal{S}_2}(\boldsymbol{x}_k) - \Gamma_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k}\left(\nabla f_{\mathcal{S}_2}(\boldsymbol{x}_{k-1}) - \widetilde{\boldsymbol{v}}_{k-1}\right)$. Here the vector transport $\Gamma_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k}$ is applied to ensure that the Riemannian gradients can be linearly combined in a common tangent space. Then R-SPIDER performs normalized gradient descent to update $\boldsymbol{x}_{k+1} = \mathrm{Exp}_{\boldsymbol{x}_k}\left(-\eta_k \frac{\widetilde{\boldsymbol{v}}_k}{\|\widetilde{\boldsymbol{v}}_k\|}\right)$ until the termination of the algorithm.

The idea of recursive Riemannian gradient estimation has also been exploited by R-SRG [18]. Although sharing a similar spirit in full gradient approximation, R-SPIDER departs notably from R-SRG: at each iteration, R-SPIDER normalizes the gradient $\widetilde{\boldsymbol{v}}_k$ and thus is able to well control the distance between $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ by properly controlling the stepsize η_k, while R-SRG directly updates the variable without gradient normalization. It turns out that this normalization step is key to achieving faster convergence speed for non-convex problem in R-SPIDER, since it helps reduce the variance of stochastic gradient estimation by properly controlling the distance between $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ (see Lemma 1). As a consequence, at each iteration, R-SPIDER only needs to sample a necessary number of data points to estimate Riemannian gradient and decrease the objective sufficiently (see Theorems 1 and 2). In this way, R-SPIDER achieves lower overall computational complexity for solving problem (1).

3.2 Analysis for General Non-convex Problem

Here we first introduce a key lemma which is a basis for the following analysis and then focus on analyzing the computational complexity of Algorithm 1 on general non-convex problem (1).

3.2.1 Bounded Gradient Estimation Error

The vanilla SPIDER is known to achieve nearly optimal iteration complexity bounds for stochastic non-convex optimization in Euclidean space [24]. We here show that R-SPIDER generalizes such an appealing property of SPIDER to Riemannian manifolds. We first present the following key lemma which guarantees sufficiently accurate Riemannian gradient estimation for R-SPIDER. We denote $\mathbb{I}_{\{\mathcal{E}\}}$ as the indicator function: if the event $\mathcal{E}$ is true, then $\mathbb{I}_{\{\mathcal{E}\}} = 1$; otherwise, $\mathbb{I}_{\{\mathcal{E}\}} = 0$.

Lemma 1 (Bounded Gradient Estimation Error for General Retraction and Vector Transport)**.** *Suppose Assumptions 1 and 3 $\sim$ 5 hold. Let $k_0 = \lfloor k/p \rfloor$ and $\widetilde{k}_0 = k_0 p$. The estimation error between the full Riemannian gradient $\nabla f(\boldsymbol{x}_k)$ and its estimate $\widetilde{\boldsymbol{v}}_k$ in Algorithm 1 with general retraction and vector transport is bounded as*

$$\mathbb{E}\left[\|\widetilde{\boldsymbol{v}}_k - \nabla f(\boldsymbol{x}_k)\|^2 \mid \boldsymbol{x}_{\widetilde{k}_0}, \cdots, \boldsymbol{x}_{\widetilde{k}_0+p-1}\right] \leq \mathbb{I}_{\{|\mathcal{S}_1|<n\}}\frac{\sigma^2}{|\mathcal{S}_1|} + \frac{\Lambda^2}{|\mathcal{S}_2|}\sum_{i=\widetilde{k}_0}^{\widetilde{k}_0+p-1}\|R_{\boldsymbol{x}_i}^{-1}(\boldsymbol{x}_{i+1})\|^2, \tag{3}$$

where $\Lambda = \sqrt{2(\theta^2 G^2 + 2(1+c_R)L_H^2)}$ with the parameters L_H, c_R and G in Assumptions 1 and 3 $\sim$ 5 and a positive constant θ.

The proof of Lemma 1 can be found in Section B.1 in the supplementary material. Lemma 1 tells that by properly selecting the mini-batch sizes $|\mathcal{S}_1|$ and $|\mathcal{S}_2|$, the accuracy of gradient estimate $\widetilde{\boldsymbol{v}}_k$ can be controlled. Benefiting from the normalization step, we have $\|\mathrm{R}_{\boldsymbol{x}_k}^{-1}(\boldsymbol{x}_{k+1})\| \leq c_E\|\mathrm{Exp}_{\boldsymbol{x}_k}^{-1}(\boldsymbol{x}_{k+1})\| = c_E\eta_k$. As a result, the gradient estimation error can be bounded as $\mathbb{E}\left[\|\widetilde{\boldsymbol{v}}_k - \nabla f(\boldsymbol{x}_k)\|^2 \mid \boldsymbol{x}_{\widetilde{k}_0}, \cdots, \boldsymbol{x}_{\widetilde{k}_0+p-1}\right] \leq \mathbb{I}_{\{|\mathcal{S}_1|<n\}}\frac{\sigma^2}{|\mathcal{S}_1|} + \frac{c_E^2\Lambda^2}{|\mathcal{S}_2|}\sum_{i=\widetilde{k}_0}^{\widetilde{k}_0+p-1}\eta_i^2$, which is key to analyze the rate-of-convergence of R-SPIDER with retraction and vector transport. When using the exponential mapping and parallel transport, based on Lemma 1 we can derive a similar gradient estimation error bound in Corollary 1.

Corollary 1 (Bounded Gradient Estimation Error for Exponential Mapping and Parallel Transport)**.** *Suppose that each component loss $f_i(\boldsymbol{x})$ is geodesically L-gradient-Lipschitz under exponential mapping in Assumptions 2 and the stochastic gradient has bounded variance in Assumption 5. The estimation error between the full Riemannian gradient $\nabla f(\boldsymbol{x}_k)$ and its estimate $\widetilde{\boldsymbol{v}}_k$ in Algorithm 1 with exponential mapping and parallel transport is bounded as in Eqn. (3) with $\|R_{\boldsymbol{x}_i}^{-1}(\boldsymbol{x}_{i+1})\|^2$ and Λ replaced by $\|Exp_{\boldsymbol{x}_i}^{-1}(\boldsymbol{x}_{i+1})\|^2$ and L, respectively.*

Please refer to the proof of Corollary 1 in Sec. B.2 in the supplementary material. For exponential mapping and parallel transport, bounding the gradient estimation error requires the L-smoothness property of each individual loss $f_i(\boldsymbol{x})$ and the bounded-variance assumption of stochastic Riemannian gradient, and relax the conditions required in Lemma 1. This is because compared with general retraction and vector transport, exponential mapping and parallel transport are more specifical and enjoys many good properties, e.g. their isometric properties and the properties in Assumption 4, helping avoid many aforementioned assumptions.

3.2.2 Complexity Analysis for Finite-sum Setting

We first consider problem (1) under finite-sum setting. By properly selecting parameters, we prove that at each iteration, the sequence $\{\boldsymbol{x}_k\}$ produced by Algorithm 1 can lead to sufficient decrease of the objective loss $f(\boldsymbol{x})$ when $\|\widetilde{\boldsymbol{v}}_k\|$ is large. Based on this results, we further derive the iteration number of Algorithm 1 for computing an ϵ-accuracy solution. The result is formally summarized in Theorem 1.

Theorem 1. *Suppose Assumptions 1 and 3 $\sim$ 5 hold. Let $s = \min\left(n, \frac{16\sigma^2}{\epsilon^2}\right)$, $\Lambda = \sqrt{2(\theta^2G^2 + 2(1+c_R)L_H^2)}$, $\Theta = \max(L_R, \Lambda)$, $p = n_0 s^{\frac{1}{2}}$, $\eta_k = \min\left(\frac{\epsilon}{2\Theta n_0}, \frac{\|\widetilde{\boldsymbol{v}}_k\|}{4\Theta n_0}\right)$, $|\mathcal{S}_1| = s$, $|\mathcal{S}_2| = \frac{4c_E^2 s^{\frac{1}{2}}}{n_0}$ and $n_0 \in [1, 4c_E^2 s^{\frac{1}{2}}]$. Then for finite-sum problem (1), the sequence $\{\boldsymbol{x}_k\}$ produced by Algorithm 1 with retraction and vector transport satisfies*

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\right] \leq -\frac{\epsilon}{64\Theta n_0}\left(12\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|] - 7\epsilon\right).$$

Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}})\|] \leq \epsilon$, Algorithm 1 will terminate at most $\left(\frac{14\Theta n_0 \Delta}{\epsilon^2}\right)$ iterations in expectation and the IFO complexity of Algorithm 1 is $\mathcal{O}\big(\min\left(n + \frac{\Theta\Delta\sqrt{n}}{\epsilon^2}, \frac{\Theta\Delta\sigma}{\epsilon^3}\right)\big)$, where $\Delta = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \mathrm{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.

See Appendix C.1 for a complete proof. We discuss the theoretical implications of Theorem 1. First, Theorem 1 shows that by one iteration loop of Algorithm 1, the objective value $f(\boldsymbol{x}_k)$ monotonously decreases in expectation when $\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|]$ is large, *e.g.* $\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|] \geq \frac{7\epsilon}{12}$. By comparison, Kasai *et al.* [18] only proved the sublinear convergence rate of the gradient norm $\mathbb{E}[\|\nabla f(\boldsymbol{x})\|^2]$

in R-SRG and did not reveal any convergence behavior of the objective $f(\boldsymbol{x})$.

Second, Theorem 1 also indicates that Algorithm 1 only needs to run at most $\big(\frac{14\Theta n_0\Delta}{\epsilon^2}\big)$ iteration to compute an ϵ-accuracy solution $\widetilde{\boldsymbol{x}}$, *i.e.* $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}})\|] \leq \epsilon$. This means the convergence rate of R-SPIDER is at the order of $\mathcal{O}\big(\frac{\Theta n_0\Delta}{\epsilon^2}\big)$. By observing this convergence rate, Theorem 1 further yields as a byproduct the benefits of mini-batching to R-SPIDER. Indeed, by controlling the parameter n_0 in R-SPIDER, the mini-batch size $|\mathcal{S}_2|$ at each iteration can range from 1 to $\mathcal{O}\big(\min\big(4\sqrt{n}, \frac{16\sigma}{\epsilon}\big)\big)$. Also, it can be seen from Theorem 1 that larger mini-batch size allows more aggressive step size η_k and thus leads to less necessary iterations to achieve an ϵ-accuracy solution. More specifically, the convergence rate bound $\mathcal{O}\big(\frac{\Theta n_0\Delta}{\epsilon^2}\big)$ indicates that at least in theory, increasing the mini-batch sizes in R-SPIDER provides linear speedups in parallel computing environment. In contrast, these important benefits of mini-batching are not explicitly analyzed in the existing Riemannian stochastic gradient algorithms [14], [18].

Thirdly, the IFO complexity of R-SPIDER for non-convex finite-sum problems is at the order of $\mathcal{O}\big(\min\big(n + \frac{\Theta\sqrt{n}}{\epsilon^2}, \frac{\Theta\sigma}{\epsilon^3}\big)\big)$. Kasai *et al.* [18] proved that the IFO complexity of R-SRG is at the order of $\mathcal{O}\big(n + \frac{\Theta^2}{\epsilon^4}\big)$ to obtain an ϵ-accuracy solution. By comparison, one can observe that R-SPIDER is at least faster than R-SRG by a factor of $\frac{1}{\epsilon}$. This is because the normalization step in R-SPIDER allows us to well control the gradient estimation error and thus avoids sampling too many redundant samples at each iteration, resulting in sharper IFO complexity.

Then we consider the special cases of retraction and vector transport, namely exponential mapping and parallel transport. Based on Theorem 1, we can derive similar results as stated in Corollary 2. Appendix C.2 provides its detailed proof.

Corollary 2. *Suppose that the each component loss $f_i(\boldsymbol{x})$ is geodesically L-gradient-Lipschitz under exponential mapping in Assumptions 2 and the stochastic gradient has bounded variance in Assumption 5. With the parameter setting $s = \min\big(n, \frac{16\sigma^2}{\epsilon^2}\big)$, $p = n_0 s^{\frac{1}{2}}$, $\eta_k = \min\big(\frac{\epsilon}{2Ln_0}, \frac{\|\widetilde{\boldsymbol{v}}_k\|}{4Ln_0}\big)$, $|\mathcal{S}_1| = s$, $|\mathcal{S}_2| = \frac{4s^{\frac{1}{2}}}{n_0}$ and $n_0 \in [1, 4s^{\frac{1}{2}}]$, for finite-sum problem (1) the sequence $\{\boldsymbol{x}_k\}$ produced by Algorithm 1 with exponential mapping and parallel transport satisfies*

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\right] \leq -\frac{\epsilon}{64Ln_0}\left(12\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|] - 7\epsilon\right).$$

Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}})\|] \leq \epsilon$, Algorithm 1 will terminate at most $\big(\frac{14Ln_0\Delta}{\epsilon^2}\big)$ iterations in expectation and the IFO complexity of Algorithm 1 is $\mathcal{O}\big(\min\big(n + \frac{L\Delta\sqrt{n}}{\epsilon^2}, \frac{L\Delta\sigma}{\epsilon^3}\big)\big)$, where $\Delta = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.

From Corollary 2, one can observe that similar to retraction and vector transport, Algorithm 1 equipped with exponential mapping and parallel transport can reduce the objective sufficiently in each iteration when the gradient $\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|]$ is large. But compared with Theorem 1, Corollary 2 requires milder conditions due to the good properties of exponential mapping and vector transport. See detailed discussion below Corollary 1. Besides, Corollary 2 also shows the benefits of mini-batching to R-SPIDER under the parallel computation setting.

For the IFO complexity $\mathcal{O}\big(\min\big(n + \frac{L\sqrt{n}}{\epsilon^2}, \frac{L\sigma}{\epsilon^3}\big)\big)$ of R-SPIDER, it matches the state-of-the-art complexity bounds for general non-convex optimization problems in Euclidean space [24], [38]. Indeed, under the L-gradient-Lipschitz assumption on each component loss $f_i(\boldsymbol{x})$, Fang *et al.* [24] proved that the lower IFO complexity bound for finite-sum problem (1) in Euclidean space is $\mathcal{O}\big(n + \frac{L\sqrt{n}}{\epsilon^2}\big)$ when the number n of the component function obeys $n \leq \mathcal{O}\big(\frac{L^2}{\epsilon^4}\big)$. In the sense that Euclidean space is a special case of Riemannian manifold, our IFO complexity $\mathcal{O}\big(n + \frac{L\Delta\sqrt{n}}{\epsilon^2}\big)$ for finite-sum problem (1) under the gradient-Lipschitz assumption is nearly optimal. If we further assume the gradient variance is bounded by σ^2 as in Assumption 5, we can establish tighter IFO complexity $\mathcal{O}\left(\frac{1}{\epsilon^2}\min\left(\sqrt{n}, \frac{1}{\epsilon}\right)\right)$. This is because when the sample number n satisfies $n \geq \frac{16\sigma^2}{\epsilon^2}$, by sampling $|\mathcal{S}_1| = \frac{16\sigma^2}{\epsilon^2}$ and $|\mathcal{S}_2| = \frac{16\sigma}{n_0\epsilon}$, the gradient estimation error already satisfies $\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k - \nabla f(\boldsymbol{x}_k)\|^2] \leq \frac{\epsilon^2}{8}$. Accordingly, if $\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\widetilde{\boldsymbol{v}}_k\| \leq 0.5\epsilon$ which is actually achieved after K iterations, then $\mathbb{E}[\|f(\widetilde{\boldsymbol{x}})\|] = \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_k)\| \leq \frac{1}{K}\sum_{k=0}^{K-1}[\mathbb{E}\|\nabla f(\boldsymbol{x}_k) - \widetilde{\boldsymbol{v}}_k\| + \mathbb{E}\|\widetilde{\boldsymbol{v}}_k\|] \leq \epsilon$. So here it is only necessary to sample $|\mathcal{S}_1| = \frac{16\sigma^2}{\epsilon^2}$ data points instead of the entire set of n samples.

Compared with R-SRG [18] having the IFO complexity $\mathcal{O}\big(n + \frac{L^2}{\epsilon^4}\big)$, R-SPIDER is more computationally efficient by an improved factor $\mathcal{O}\big(\frac{1}{\epsilon}\big)$. Zhang *et al.* [14] showed that R-SVRG has the IFO complexity $\mathcal{O}\big(n + \frac{\zeta^{1/2}n^{2/3}}{\epsilon^2}\big)$, where $\zeta \geq 1$ denotes the curvature parameter. Therefore, R-SPIDER improves over R-SVRG by a factor at least $n^{1/6}$ in terms of IFO complexity. Note, the curvature parameter ζ does not appear in our bounds, as we have avoided using the trigonometry inequality which characterizes the trigonometric geometric in Riemannian manifold [14], [16], [17].

3.2.3 Complexity Analysis for Online Setting

Next we consider the online setting of problem (1). Similar to finite-sum setting, we prove in Theorem 2 that the objective $f(\boldsymbol{x})$ can be sufficiently decreased when the gradient norm is not too small.

Theorem 2. *Suppose Assumptions 1 and 3 $\sim$ 5 hold. Let $\Lambda = \sqrt{2(\theta^2G^2 + 2(1+c_R)L_H^2)}$, $\Theta = \max(L_R, \Lambda)$, $p = \frac{\sigma n_0}{\epsilon}$, $\eta_k = \min\big(\frac{\epsilon}{2\Theta n_0}, \frac{\|\widetilde{\boldsymbol{v}}_k\|}{4\Theta n_0}\big)$, $|\mathcal{S}_1| = \frac{16\sigma^2}{\epsilon^2}$, $\mathcal{S}_2 = \frac{4c_E^2\sigma}{\epsilon n_0}$ and $n_0 \in [1, 4c_E^2\sigma/\epsilon]$. Then for problem (1) under online setting, the sequence $\{\boldsymbol{x}_k\}$ produced by Algorithm 1 using retraction and vector transport satisfies*

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\right] \leq -\frac{\epsilon}{64\Theta n_0}\left(12\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|] - 7\epsilon\right).$$

Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}})\|] \leq \epsilon$, Algorithm 1 will terminate at most $\big(\frac{14\Theta n_0\Delta}{\epsilon^2}\big)$ iterations in expectation and the IFO complexity is $\mathcal{O}\big(\frac{\Theta\sigma\Delta}{\epsilon^3}\big)$, where $\Delta = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.

See Appendix D.1 for a proof of this result. Bonnabel *et al.* [17] have also analyzed R-SGD under online setting, but only with asymptotic convergence guarantee obtained. By comparison, we for the first time establish non-asymptotic complexity bounds for Riemannian online non-convex optimization. Then for exponential mapping and parallel transport, we can also derive similar results as in Theorem 2. We defer the proof of Corollary 3 to Appendix D.2.

Corollary 3. *Suppose that the each component loss $f_i(\boldsymbol{x})$ is geodesically L-gradient-Lipschitz under exponential mapping in Assumptions 2 and the stochastic gradient has bounded variance in Assumption 5. Let $p = \frac{\sigma n_0}{\epsilon}$, $\eta_k = \min\big(\frac{\epsilon}{2Ln_0}, \frac{\|\widetilde{\boldsymbol{v}}_k\|}{4Ln_0}\big)$, $|\mathcal{S}_1| = \frac{16\sigma^2}{\epsilon^2}$, $\mathcal{S}_2 = \frac{4\sigma}{\epsilon n_0}$ and $n_0 \in [1, 4\sigma/\epsilon]$. Then for problem (1) under*

online setting, the sequence $\{\boldsymbol{x}_k\}$ produced by Algorithm 1 with exponential mapping and parallel transport satisfies

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\right] \leq -\frac{\epsilon}{64Ln_0}\left(12\mathbb{E}[\|\widetilde{\boldsymbol{v}}_k\|] - 7\epsilon\right).$$

Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}})\|] \leq \epsilon$, Algorithm 1 will terminate at most $\left(\frac{14Ln_0\Delta}{\epsilon^2}\right)$ iterations in expectation and the IFO complexity is $\mathcal{O}\left(\frac{L\sigma\Delta}{\epsilon^3}\right)$, where $\Delta = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.

Algorithm 2 Riemannian Gradient Dominated SPIDER (R-GD-SPIDER)

1: **Input:** initial point $\widetilde{\boldsymbol{x}}_0$, initial accuracy ϵ_0, learning rate η^0, mini-batch sizes $|\mathcal{S}_1^0|$ and $|\mathcal{S}_2^0|$, iteration interval p^0, final accuracy ϵ
2: **for** $t = 1$ to T **do**
3: $\quad\widetilde{\boldsymbol{x}}_t = \text{R-SPIDER}(\widetilde{\boldsymbol{x}}_{t-1}, \epsilon_{t-1}, \eta_t, p_t, |\mathcal{S}_1^t|, |\mathcal{S}_2^t|)$.
4: $\quad$Set $\epsilon_t = 0.5\epsilon_{t-1}$, and $\eta_t, p_t, |\mathcal{S}_1^t|, |\mathcal{S}_2^t|$ properly.
5: **end for**
6: **Output:** $\widetilde{\boldsymbol{x}}_t$

3.3 On gradient dominated functions

We now turn to a special case of problem (1) with gradient dominated loss function as defined in Definition 2. For instance, the strongly geodesically convex (SGC) functions[1] are gradient dominated. Some non-strongly convex problems, *e.g.* ill-conditioned linear prediction and logistic regression [39], and Riemannian non-convex problems, *e.g.* PCA [14], also belong to gradient dominated functions. Please refer to [37], [39] for more instances of gradient dominated functions. To better fit gradient dominated functions, we develop the Riemannian gradient dominated SPIDER (R-GD-SPIDER) as a multi-stage variant of R-SPIDER. A high-level description of R-GD-SPIDER is outlined in Algorithm 2. The basic idea is to use more aggressive learning rates in early stage of processing and gradually shrink the learning rate in later stage. With the help of such a simulated annealing process, R-GD-SPIDER exhibits linear convergence behavior for btoh finite-sum and online problems.

3.3.1 Complexity Analysis for Finite-Sum Setting

Here we show that R-GD-SPIDER enjoys linear convergence rate, as formally stated in Theorem 3.

Theorem 3. *Suppose that function $f(\boldsymbol{x})$ is τ-gradient dominated, the retraction and vector transport used in Algorithm 2 satisfy Assumptions 1 and 3 $\sim$ 5. For finite-sum setting, at the t-th iteration, set $\epsilon_0 = \frac{\sqrt{\Delta}}{2\sqrt{\tau}}$, $\epsilon_t = \frac{\epsilon_0}{2^t}$, $\Theta = \max(\sqrt{2(\theta^2G^2 + 2(1+c_R)L_H^2)}, L_R)$, $\eta_{t,k} = \frac{\|\widetilde{\boldsymbol{v}}_{t,k}\|}{2n_0\Theta}$, $s_t = \min\left(n, \frac{22\sigma^2}{\epsilon_t^2}\right)$, $p_t = n_0 s_t^{\frac{1}{2}}$, $|\mathcal{S}_1^t| = s_t$ and $|\mathcal{S}_2^t| = \frac{2c_E^2 s_t^{\frac{1}{2}}}{n_0}$, $K_t = \frac{72n_0\Theta\Delta_t}{\epsilon_t^2}$, where $n_0 \in [1, 2c_E^2 s_t^{\frac{1}{2}}]$.*
***(1)** The sequence $\{\widetilde{\boldsymbol{x}}_t\}$ produced by Algorithm 2 satisfies*

$$\mathbb{E}\left[f(\widetilde{\boldsymbol{x}}_t) - f(\boldsymbol{x}_*)\right] \leq \frac{\Delta}{4^t} \quad \text{and} \quad \mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_t)\|] \leq \frac{1}{2^t}\sqrt{\frac{\Delta}{\tau}},$$

where $\Delta = f(\widetilde{\boldsymbol{x}}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.
***(2)** To achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_t)\|] \leq \epsilon$, the IFO complexity is $\mathcal{O}\left(\min\left((n + \tau\Theta\sqrt{n})\log\left(\frac{1}{\epsilon}\right), \frac{\tau\Theta\sigma}{\epsilon}\right)\right)$.*

Please refer to Appendix E.1 for the proof of Theorem 3. The main message conveyed by Theorem 3 is that R-GD-SPIDER enjoys a linear rate of convergence and its IFO complexity is at the order of $\mathcal{O}\left(\min\left((n + \tau\Theta\sqrt{n})\log\left(\frac{1}{\epsilon}\right), \frac{\tau\Theta\sigma}{\epsilon}\right)\right)$. For R-SRG [18], its IFO complexity is $\mathcal{O}\left((n + \tau^2\Theta^2)\log\left(\frac{1}{\epsilon}\right)\right)$. Therefore, in terms of IFO complexity, R-GD-SPIDER is superior to R-SRG when the optimization accuracy ϵ is moderately small at a huge data size n. Next, based on Theorem 3, we derive similar results on exponential mapping and parallel transport as stated in Corollary 4 with proof in Appendix E.2.

Corollary 4. *Suppose that function $f(\boldsymbol{x})$ is τ-gradient dominated, the each component loss $f_i(\boldsymbol{x})$ is geodesically L-gradient-Lipschitz under exponential mapping in Assumptions 2 and the stochastic gradient has bounded variance in Assumption 5. Then for exponential mapping and parallel transport, by using the same parameter setting in Theorem 3 but with Θ replaced by L, the linear convergence results of the objective $f(\widetilde{\boldsymbol{x}}_t)$ and its gradient $\nabla f(\widetilde{\boldsymbol{x}}_t)$ in Theorem 3 still holds. Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_t)\|] \leq \epsilon$, the IFO complexity is $\mathcal{O}\left(\min\left((n + \tau L\sqrt{n})\log\left(\frac{1}{\epsilon}\right), \frac{\tau L\sigma}{\epsilon}\right)\right)$.*

For R-SVRG with τ-gradient dominated functions, Zhang *et al.* [14] also established a linear convergence rate and an IFO complexity bound $\mathcal{O}\left((n + \tau L\zeta^{\frac{1}{2}}n^{\frac{2}{3}})\log\left(\frac{1}{\epsilon}\right)\right)$. As a comparison, our R-GD-SPIDER makes an improvement over R-SVRG in IFO complexity by a factor of $n^{\frac{1}{6}}$. For R-SRG [18], when the optimization accuracy ϵ is moderately small at a huge data size n R-GD-SPIDER enjoys similar advantages as discussed above.

3.3.2 Complexity Analysis for Online Setting

Turning to the online setting, R-GD-SPIDER also converges linearly, as formally stated in Theorem 4. See its proof in Appendix F.1.

Theorem 4. *Suppose that $f(\boldsymbol{x})$ is τ-gradient dominated, the retraction and vector transport used in Algorithm 2 satisfy Assumptions 1 and 3 $\sim$ 5. For online setting, at the t-th iteration, let $\epsilon_0 = \frac{\sqrt{\Delta}}{2\sqrt{\tau}}$, $\epsilon_t = \frac{\epsilon_0}{2^t}$, $\Theta = \max(\sqrt{2(\theta^2G^2 + 2(1+c_R)L_H^2)}, L_R)$, $\eta_{t,k} = \frac{\|\widetilde{\boldsymbol{v}}_{t,k}\|}{2n_0\Theta}$, $s_t = \frac{22\sigma^2}{\epsilon_t^2}$, $p_t = n_0 s_t^{\frac{1}{2}}$, $|\mathcal{S}_1^t| = s_t$ and $|\mathcal{S}_2^t| = \frac{2c_E^2 s_t^{\frac{1}{2}}}{n_0}$, $K_t = \frac{72n_0\Theta\Delta_t}{\epsilon_t^2}$, where $n_0 \in [1, 2c_E^2 s_t^{\frac{1}{2}}]$.*
***(1)** The sequence $\{\widetilde{\boldsymbol{x}}_t\}$ produced by Algorithm 2 satisfies*

$$\mathbb{E}\left[f(\widetilde{\boldsymbol{x}}_t) - f(\boldsymbol{x}_*)\right] \leq \frac{\Delta}{4^t} \quad \text{and} \quad \mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_t)\|] \leq \frac{1}{2^t}\sqrt{\frac{\Delta}{\tau}},$$

where $\Delta = f(\widetilde{\boldsymbol{x}}_0) - f(\boldsymbol{x}_*)$ with $\boldsymbol{x}_* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{M}} f(\boldsymbol{x})$.
***(2)** To achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_t)\|] \leq \epsilon$, the IFO complexity is $\mathcal{O}\left(\frac{\tau\Theta\sigma}{\epsilon}\right)$.*

Then we also show the results on exponential mapping and parallel transport in Corollary 5. The proof of can be found in Appendix F.2.

Corollary 5. *Suppose that function $f(\boldsymbol{x})$ is τ-gradient dominated, the each component loss $f_i(\boldsymbol{x})$ is geodesically L-gradient-Lipschitz under exponential mapping in Assumptions 2 and the stochastic gradient has bounded variance in Assumption 5. Then for exponential mapping and parallel transport, by using the same parameter setting in Theorem 4 but with Θ replaced by L, the linear convergence results of the objective $f(\widetilde{\boldsymbol{x}}_t)$ and its gradient $\nabla f(\widetilde{\boldsymbol{x}}_t)$*

1. A strongly geodesically convex function satisfies $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle\nabla f(\boldsymbol{x}), \operatorname{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\rangle + \frac{\mu}{2}\|\operatorname{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{y})\|^2, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}$, for som $\mu > 0$, which immediately implies $f(\boldsymbol{x}) - f(\boldsymbol{x}_*) \leq \frac{1}{2\mu}\|\nabla f(\boldsymbol{x})\|^2$ by Cauchy-Schwarz inequality.

in Theorem 4 still holds. Moreover, to achieve $\mathbb{E}[\|\nabla f(\widetilde{\boldsymbol{x}}_T)\|] \leq \epsilon$, *the IFO complexity is* $\mathcal{O}\left(\frac{\tau L\sigma}{\epsilon}\right)$.

The non-asymptotic convergence result in Theorem 4 and Corollary 5 are new to online Riemannian gradient dominated optimization.

4 Experiments

In this section, we first introduce the testing problems and then develop a learning-rate-adaptive R-SPIDER algorithm. Next, we compare R-SPIDER with several state-of-the-art Riemannian stochastic gradient algorithms, including R-SGD [17], R-SVRG [14], [19], R-SRG [18] and R-SRG+ [18]. For all the considered algorithms, we tune their hyper-parameters optimally. Finally, we investigate the computational efficiency of the proposed R-SPIDER algorithm when it is equipped with 1) exponential mapping and parallel transport and 2) retraction and vector transport. We run simulations on ten datasets, including six datasets from LibSVM[1] (a9a, satimage, covtype, protein, ijcnn1 and epsilon), three face datasets (YaleB [40], AR [41] and PIE [42]) and one recommendation dataset (MovieLens-1M[2]). The statistics of these datasets are summarized in Table 2. From it we can observe that these datasets are different from each other due to their feature dimension, training samples, and class numbers, etc. Thus, those testing datasets can well investigate the performance of the proposed algorithms.

4.1 Testing problems and experimental settings

We evaluate all the considered algorithms on two widely studied Riemannian manifold learning problems: the k-PCA problem and the low-rank matrix completion (LRMC) problem.

The k-PCA problem and experiment setting. Given n data points, k-PCA is formulated as the following problem of quadratic minimization over Stiefel manifold:

$$\min_{\boldsymbol{U}\in \mathsf{St}(k,d)} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{a}_i,$$

where $\boldsymbol{a}_i \in \mathbb{R}^d$ denotes the i-th sample vector and $\mathsf{St}(k,d) = \{\boldsymbol{U} \in \mathbb{R}^{d\times k} \mid \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}\}$ denotes the Stiefel manifold. For this problem, the columns of the ground truth $\boldsymbol{U}^*$ are known to be the top k eigenvectors of the data covariance matrix which can be estimated using singular value decomposition (SVD). We thus can use $f(\boldsymbol{U}^*)$ as optimal value f^* for sub-optimality estimation in Fig.2 and 3. In this group of experiments, we compute the first twenty leading eigenvectors on six datasets from LibSVM, including a9a, covtype, epsilon, ijcnn, protein and satimage.

Since the parallel transport has no closed-form solution on Stiefel manifold, here we only present the vector transport associated with QR based retraction [28], [43]. The retraction is defined as

$$\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{\xi}) = \mathrm{QR}(\boldsymbol{x}+\boldsymbol{\xi}),$$

where $\mathrm{QR}(\boldsymbol{x})$ denotes the $\boldsymbol{Q}$ factor in the QR decomposition $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{R}$ with $\boldsymbol{Q} \in \mathsf{St}(k,d)$ and an upper triangular matrix $\boldsymbol{R}$. Then its vector transport is defined as

$$\Gamma_{\boldsymbol{x}}^{\boldsymbol{y}}(\boldsymbol{z}) = \boldsymbol{z} - \boldsymbol{y}\mathsf{sym}(\boldsymbol{y}^T\boldsymbol{z}),$$

where $\mathsf{sym}(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}+\boldsymbol{x}^T)$.

1. https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
2. https://grouplens.org/datasets/movielens/1m/

TABLE 2: Descriptions of the ten testing datasets.

	#class	#sample	#feature		#class	#sample	#feature
a9a	2	32,561	123	satimage	6	4,435	36
covtype	2	581,012	54	YaleB	38	2,414	2,016
epsilon	2	40,000	2,000	AR	100	2,600	1,200
ijcnn	2	49,990	22	PIE	64	11,554	1,024
protein	3	14,895	357	MovieLens-1M	—	6,040	3,706

The LRMC problem and experiment setting. When given an incomplete observation of a low-rank matrix $\boldsymbol{A} \in \mathbb{R}^{d\times n}$, LRMC aims at exactly or approximately recovering the full matrix $\boldsymbol{A}$. The mathematical formulation is $\min_{\boldsymbol{U}\in\mathsf{Gr}(k,d),\boldsymbol{G}\in\mathbb{R}^{k\times n}} \|\mathcal{P}_\Omega(\boldsymbol{A}) - \mathcal{P}_\Omega(\boldsymbol{U}\boldsymbol{G})\|^2$, where the Grassmann manifold $\mathsf{Gr}(k,d)$ denotes the set of all k-dimensional linear subspaces of $\mathbb{R}^d$, and the set Ω of locations corresponds to the observed entries, namely $(i,j) \in \Omega$ if $\boldsymbol{A}_{ij}$ is observed. $\mathcal{P}_\Omega$ is a linear operator that extracts entries in Ω and fills the entries not in Ω with zeros. When each column $\boldsymbol{A}_i$ in the matrix $\boldsymbol{A}$ denotes a sample vector, the LRMC problem can be expressed equivalently as

$$\min_{\boldsymbol{U}\in\mathsf{Gr}(k,d),\boldsymbol{G}_i\in\mathbb{R}^k} \frac{1}{n}\sum_{i=1}^{n} \|\mathcal{P}_{\Omega_i}(\boldsymbol{A}_i) - \mathcal{P}_{\Omega_i}(\boldsymbol{U}\boldsymbol{G}_i)\|^2.$$

Since there is no ground truth for the optimum, we run Riemannian GD sufficiently long until the gradient satisfies $\|\nabla f(\boldsymbol{x})\|/\|\boldsymbol{x}\| \leq 10^{-8}$ with $\boldsymbol{x} = [\boldsymbol{U}, \boldsymbol{G}]$, and then use the output as an approximate optimal value f^* for sub-optimality estimation in Fig.1, 6, 4 and 5. We test the considered algorithms on three face datasets (YaleB [40], AR [41] and PIE [42]) and one recommendation dataset (MovieLens-1M), considering these data approximately lie on a union of low-rank subspaces [18], [44]. For face images, we randomly sample 30% pixels in each image as the observations and set $k = 30$. For MovieLens-1M, we use its one million ratings for 3,952 movies from 6,040 users as the observations and set $k = 100$.

Let us consider the Grassmann manifold in the LRMC problem. Its exponential mapping is defined as

$$\boldsymbol{z} = \mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{V}\cos(\boldsymbol{\Sigma})\boldsymbol{V}^T + \boldsymbol{U}\sin(\boldsymbol{\Sigma})\boldsymbol{V}^T,$$

where $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the skinny SVD of $\boldsymbol{y}$. Accordingly, the inverse exponential mapping is computed as $\boldsymbol{y} = \mathrm{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{z}) = \boldsymbol{U}\arctan(\boldsymbol{\Sigma})\boldsymbol{V}^T$ where $(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{z}(\boldsymbol{x}^T\boldsymbol{z}^T)^{-1} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Consequently, we can compute the parallel transport

$$\mathrm{P}_{\boldsymbol{x}}^{\boldsymbol{z}}(\widehat{\boldsymbol{y}}) = -\boldsymbol{x}\boldsymbol{V}\sin(\boldsymbol{\Sigma})\boldsymbol{U}^T\widehat{\boldsymbol{y}} + \boldsymbol{U}\cos(\boldsymbol{\Sigma})\boldsymbol{U}^T\widehat{\boldsymbol{y}} + (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\widehat{\boldsymbol{y}},$$

where $\mathrm{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{z}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ denotes the skinny SVD of $\mathrm{Exp}_{\boldsymbol{x}}^{-1}(\boldsymbol{z})$. For the retraction $\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{z})$, we adopt the polar retraction

$$\mathrm{R}_{\boldsymbol{x}}(\boldsymbol{z}) = \mathsf{polar}(\boldsymbol{x}+\boldsymbol{z}),$$

where $\mathsf{polar}(\boldsymbol{x}+\boldsymbol{z}) \in \mathsf{Gr}(k,d)$ denotes the $k \times d$ orthonormal factor of the polar decomposition of $\boldsymbol{x}+\boldsymbol{z}$. It can be computed by the skinny SVD, namely $\boldsymbol{x}+\boldsymbol{z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ and $\mathsf{polar}(\boldsymbol{x}+\boldsymbol{z}) = \boldsymbol{U}\boldsymbol{V}^T$. Then based on such a retraction, the vector transport is defined as

$$\Gamma_{\boldsymbol{x}}^{\boldsymbol{y}}(\boldsymbol{z}) = \mathsf{Proj}_{\boldsymbol{y}}(\boldsymbol{z}) = (\boldsymbol{I} - \boldsymbol{y}\boldsymbol{y}^T)\boldsymbol{z}$$

which denotes the orthogonal projection onto the orthogonal complement of $\boldsymbol{y}$. This polar retraction and vector transport is commonly used for the Grassmann manifold optimization (see Example 8.1.6 in [28] and Sec. 4.1 in [43]). By comparison, the polar retraction and its vector transport are respectively much

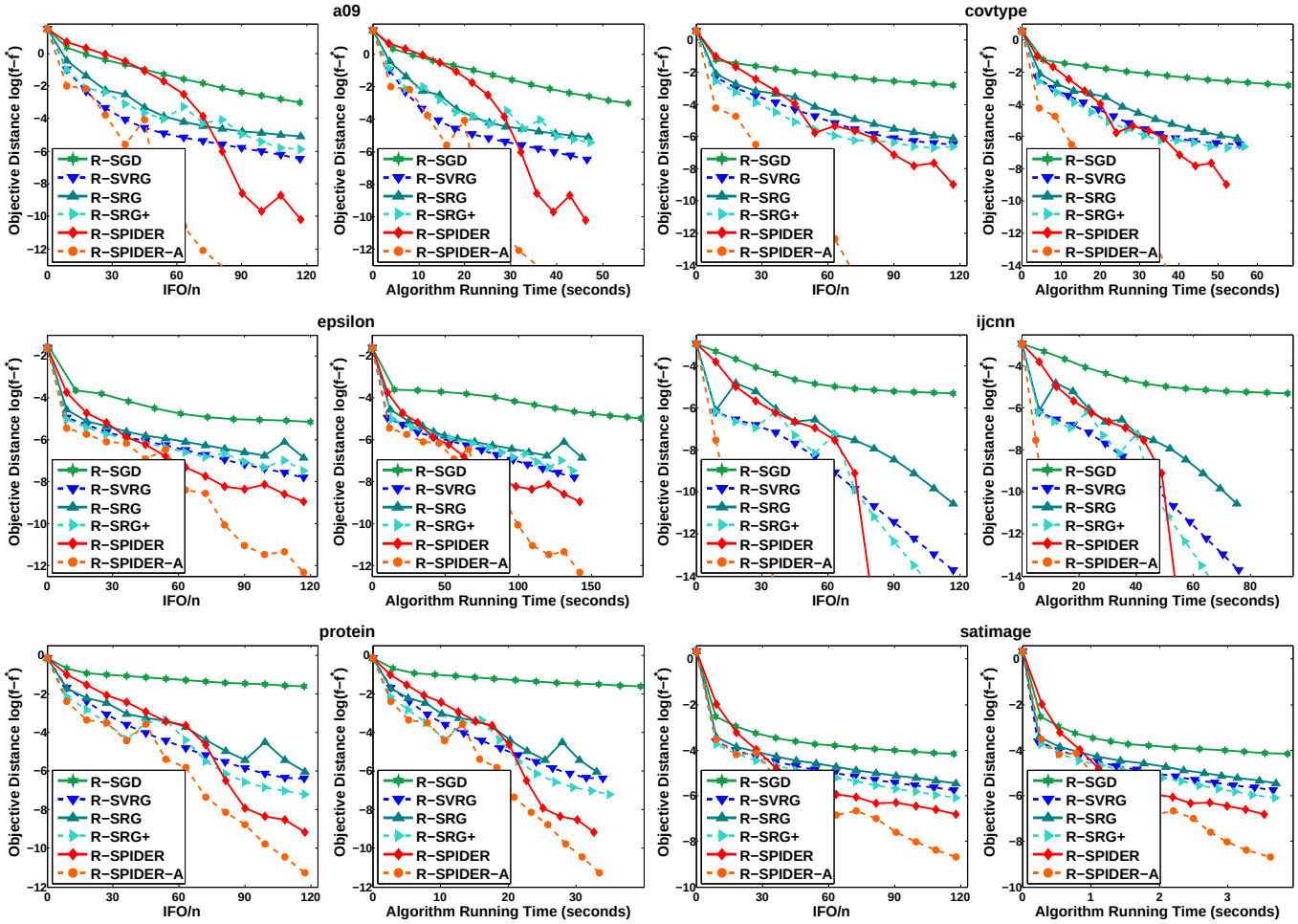


Fig. 2: Comparison among Riemannian stochastic gradient algorithms on the k-PCA problem.

simpler than the exponential mapping and the parallel transport, and thus enjoy more appealing computational efficiency.

4.2 A practical implementation of R-SPIDER

To achieve the IFO complexity in Theorem 1, it is suggested to set the learning rate as $\eta_k = \min\left(\frac{\epsilon}{2\Theta n_0}, \frac{\|\widetilde{\boldsymbol{v}}_k\|}{4\Theta n_0}\right)$ where ϵ is the desired optimization accuracy. However, since in the initial epochs the computed point is far from the optimum to problem (1), using a tiny learning rate could usually be conservative. In contrast, by using a more aggressive learning rate at the initial optimization stage, we can expect stable but faster convergence behavior. Here for R-SPIDER we design a decaying learning rate with formulation $\eta_k = \alpha^{\lfloor \frac{k}{p} \rfloor} \cdot \beta$ and call it "R-SPIDER-A", where α and β are two constants. In our experiments, α is selected from $\{0.8, 0.85, 0.9, 0.95, 0.99\}$ and β from $\{1 \times 10^{-2}, 5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, 10^{-3}\}$.

4.3 Efficiency comparison among stochastic Riemannian algorithms

We first compare R-SPIDER with state-of-the-art stochastic Riemannian algorithms, including R-SGD [17], R-SVRG [14], [19], R-SRG [18] and R-SRG+ [18], on the k-PCA and LRMC problems. All the algorithms respectively adopts the QR based retraction and the polar retraction in Section 4.1 for k-PCA and LRMC problems.

Results on the k-PCA problem. Fig. 2 shows the experimental results on the k-PCA problem. From this group of results one can observe that as the learning-rate-adaptive version of R-SPIDER, R-SPIDER-A shows much faster convergence rate in terms of both the IFO complexity and the algorithm running time. For R-SPIDER, it also reveals satisfactory convergence performance: it can quickly converge to a relatively high accuracy, e.g. 10^{-8}. R-SPIDER shows relatively flat convergence behavior in the initial epochs. This is because it uses very small learning rate and also normalizes the gradient, leading to very small steps towards to the optimum. Then along with more iterations, the computed solution becomes close to the optimum. As a result, the gradient begins to vanish and those considered algorithms without normalization tend to update the variable with small progress. By comparison, thanks to the normalization step, R-SPIDER moves more rapidly along the gradient descent direction and thus shows much faster convergence rate. For R-SPIDER-A, in the initial epochs it adopts a relatively more aggressive learning rate and then decreases the learning rate along with more iterations. Such a mechanism allows it to converge fast due to the large step size when the solution is far from the optimum in the initial stage and the relatively small step size when the solution is close to the optimum after sufficient iteration. As a result, it exhibits the sharpest convergence behavior. Note, R-SGD usually shows much slower convergence behaviors in terms of algorithm running time than the IFO complexity, since it needs to load the data from disk more frequently than other algorithms which

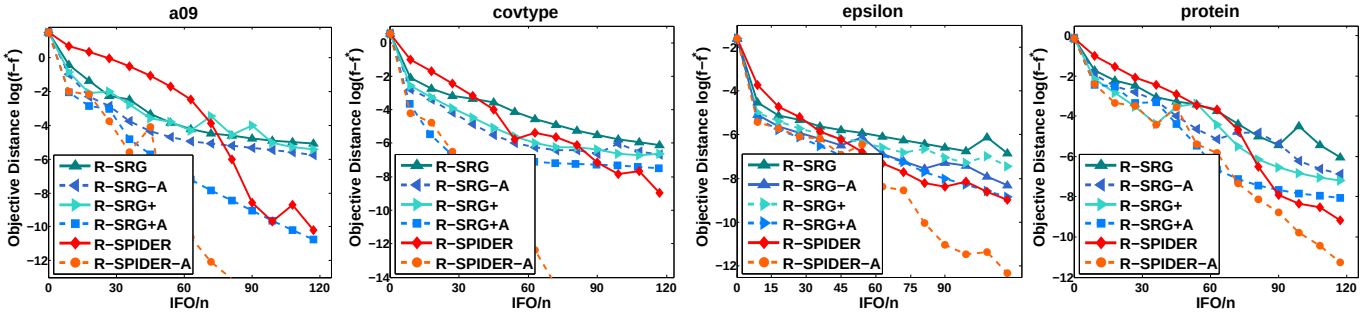


Fig. 3: Comparison between R-SPIDER and R-SRG with adaptive learning rates on the k-PCA problem.

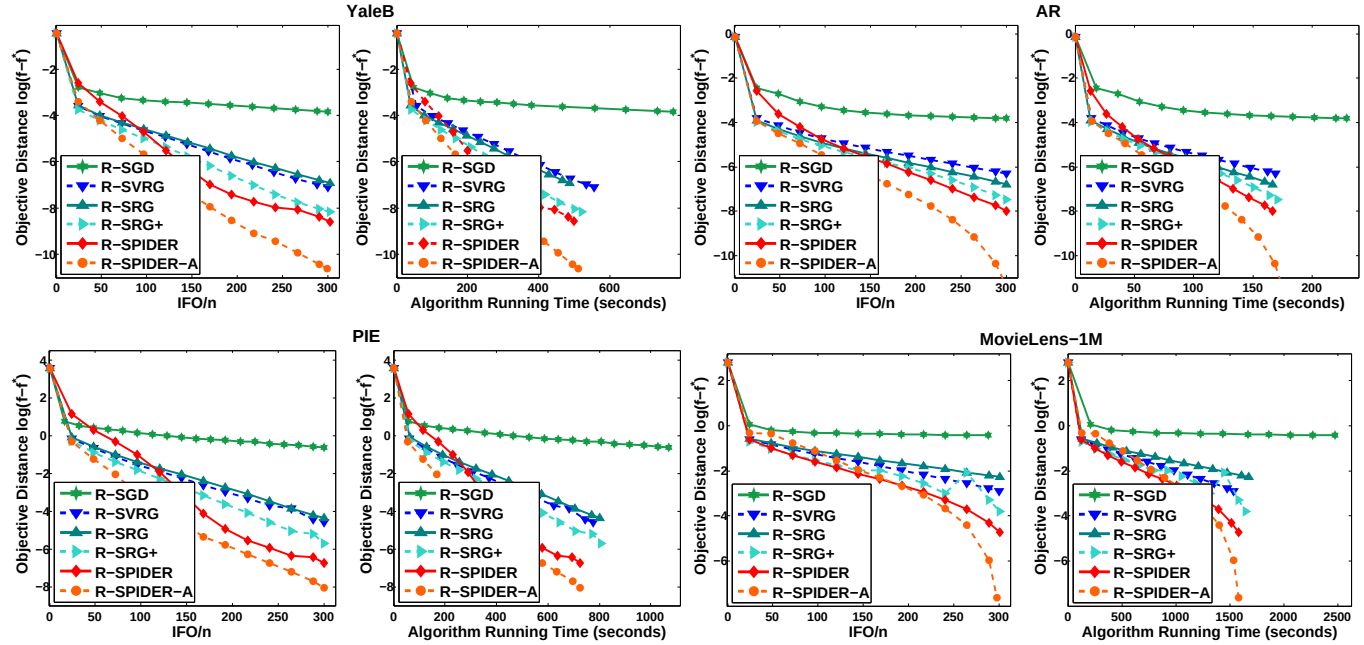


Fig. 4: Comparison among Riemannian stochastic gradient algorithms on low-rank matrix completion problem.

is actually time-consuming. All these results show the superior computational efficiency of our proposed algorithms, R-SPIDER and R-SPIDER-A, on both the IFO complexity and algorithm running time.

Then in Fig. 3, we compare R-SPIDER-A more closely with R-SRG-A and R-SRG+A which are respectively the counterparts of R-SRG and R-SRG+ with adaptive learning rate. In [18], both R-SRG-A and R-SRG+A tune their learning rate η_k as $\eta_k = \alpha(1+\alpha\lambda_\alpha\lfloor\frac{k}{p}\rfloor)$, where k is the number of inner iterations, α and λ_α are tunable hyper-parameters. By observing the experimental results in Fig. 3, we can find that the algorithms using adaptive learning rate usually outperform their vanilla counterparts, demonstrating the effectiveness of such an implementation trick. Besides, R-SPIDER-A consistently converges faster than R-SRG-A and R-SRG+A in both the IFO complexity and the algorithm running time, testifying the efficiency advantages of R-SPIDER-A.

Results on the LRMC problem. From Fig. 4, R-SPIDER-A and R-SPIDER show very similar convergence behavior to those in Fig. 2. More specifically, R-SPIDER-A achieves fastest convergence rate, and R-SPIDER has similar convergence speed as other algorithms in the initial epochs and then runs faster along with more epochs. All these results confirm the superiority of R-SPIDER and R-SPIDER-A. Moreover, we also compare R-SPIDER-A with R-SRG-A and R-SRG+A in Fig. 5. The comparison results also show the faster convergence rate of R-SPIDER-A over R-SRG-A and R-SRG+A.

4.4 Efficiency comparison between parallel and vector transports

We now turn to evaluate and compare the computational efficiency between 1) R-SPIDER using exponential mapping and parallel vector transport (R-SPIDER-Exp for short) and 2) R-SPIDER with polar retraction and vector transport (R-SPIDER-Rec) on the LRMC problem. Actually, we also compare the learning-rate-adaptive R-SPIDER algorithms, namely R-SPIDER-A, under those two transports. Fig. 1 and 6 summarize the experimental results, from which one can observe that R-SPIDER-Rec usually exhibits very similar convergence behavior to R-SPIDER-Exp in terms of the IFO complexity, but it shows faster convergence rate from the aspect of algorithm running time. The R-SPIDER-A algorithm reveals very similar comparison results when respectively using parallel and vector transports. This is because the polar retraction and its vector transport can respectively well approximate the exponential mapping and parallel transport, but the former ones are respectively much computational efficient than the latter ones which can be observed from their formulations in Section 4.1. Therefore,

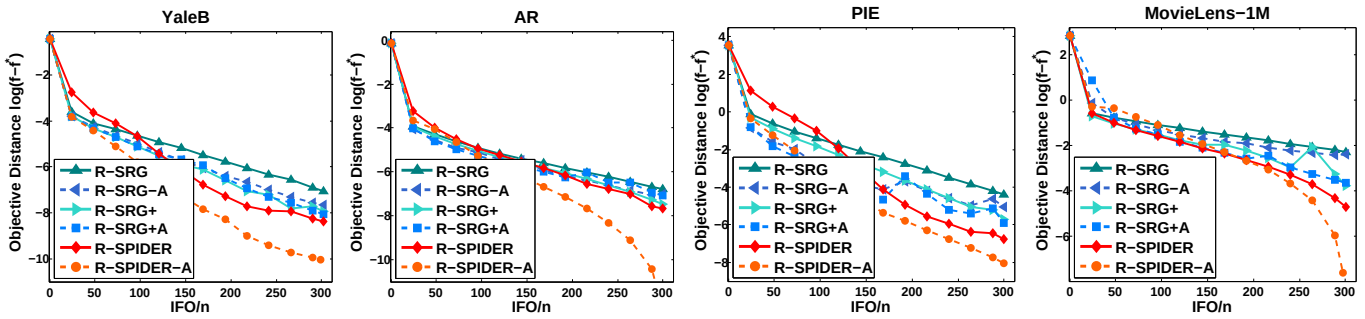


Fig. 5: Comparison among Riemannian stochastic gradient algorithms on low-rank matrix completion problem.

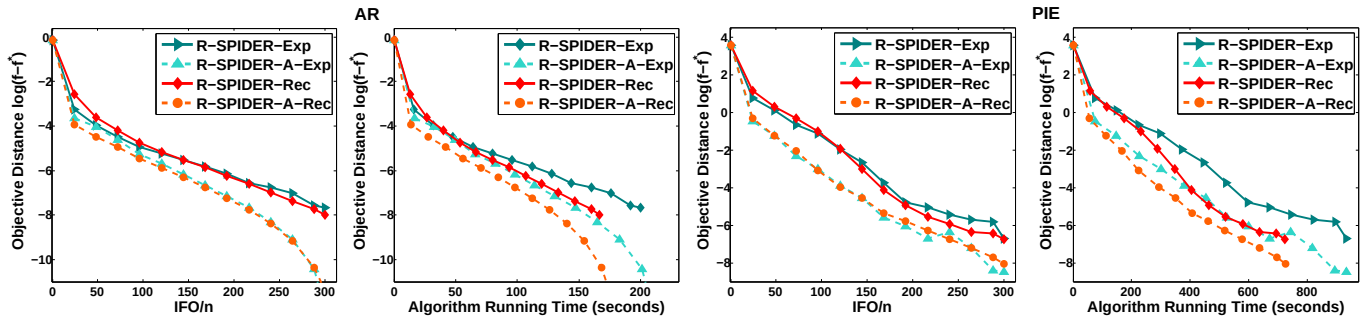


Fig. 6: Comparison between (1) R-SPIDER equipped with exponential mapping and parallel transport (R-SPIDER-Exp) and (2) R-SPIDER equipped with polar retraction and vector transport (R-SPIDER-Rec) on the LRMC problem.

the retraction and vector transport are usually more preferable than exponential mapping and parallel transport in Riemannian optimization.

5 Conclusions

We proposed R-SPIDER, which is an efficient stochastic Riemannian gradient method for non-convex optimization on Riemannian manifolds. Compared to existing first-order Riemannian algorithms, when using general retraction and vector transport or the particular exponential mapping and parallel transport, R-SPIDER provably enjoys lower computational complexity bounds for finite-sum minimization. For online optimization, similar non-asymptotic bounds are established for R-SPIDER, which to our best knowledge has not been addressed in previous study. For the special case of gradient dominated functions, we further developed a variant of R-SPIDER with improved linear rate of convergence. Finally, extensive experimental results well justify the computational superiority of R-SPIDER over the state-of-the-arts.

Acknowledgements

Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112, NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112. Xiao-Tong Yuan is supported by Natural Science Foundation of China (NSFC) under Grant 61876090.

References

[1] A. Cherian and S. Sra, "Riemannian dictionary learning and sparse coding for positive definite matrices," *IEEE trans. on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 2859–2871, 2017.

[2] J. Sun, Q. Qu, and J. Wright, "Complete dictionary recovery over the sphere ii: Recovery by Riemannian trust-region method," *IEEE Trans. on Information Theory*, vol. 63, no. 2, pp. 885–914, 2017.

[3] M. Tan, I. Tsang, L. Wang, B. Vandereycken, and S. Pan, "Riemannian pursuit for big matrix recovery," in *Proc. Int'l Conf. Machine Learning*, 2014, pp. 1539–1547.

[4] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013.

[5] G. Meyer, S. Bonnabel, and R. Sepulchre, "Linear regression under fixed-rank constraints: a Riemannian approach," in *Proc. Int'l Conf. Machine Learning*, 2011.

[6] B. Mishra and R. Sepulchre, "R3MC: A Riemannian three-factor algorithm for low-rank matrix completion," in *Proc. IEEE Conf. on Decision and Control*, 2014, pp. 1137–1142.

[7] R. Hosseini and S. Sra, "Matrix manifold optimization for Gaussian mixtures," in *Proc. Conf. Neutral Information Processing Systems*, 2015, pp. 910–918.

[8] H. Kasai and B. Mishra, "Low-rank tensor completion: a Riemannian manifold preconditioning approach," in *Proc. Int'l Conf. Machine Learning*, 2016, pp. 1012–1021.

[9] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[10] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[11] E. Oja, "Principal components, minor components, and linear neural networks," *Neural networks*, vol. 5, no. 6, pp. 927–935, 1992.

[12] J. da Cruz Neto, L. De Lima, and P. Oliveira, "Geodesic algorithms in Riemannian geometry," *Balkan Journal of Geometry and its Applications*, vol. 3, no. 2, pp. 89–100, 1998.

[13] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. on Signal Processing*, vol. 53, no. 8, pp. 2931–2941, 2005.

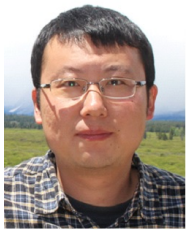
[14] H. Zhang, S. Reddi, and S. Sra, "Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds," in *Proc. Conf. Neutral Information Processing Systems*, 2016, pp. 4592–4600.

[15] H. Zhang and S. Sra, "An estimate sequence for geodesically convex optimization," in *Proc. Conf. on Learning Theory*, 2018, pp. 1703–1723.

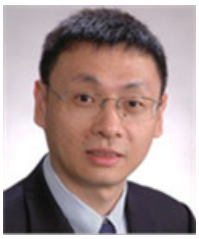
[16] H. Zhang and S. Sra, "First-order methods for geodesically convex optimization," in *Proc. Conf. on Learning Theory*, 2016, pp. 1617–1638.

[17] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds.," *IEEE Trans. Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.
[18] H. Kasai, H. Sato, and B. Mishra, "Riemannian stochastic recursive gradient algorithm with retraction and vector transport and its convergence analysis," in *Proc. Int'l Conf. Machine Learning*, 2018, pp. 2521–2529.
[19] H. Kasai, H. Sato, and B. Mishra, "Riemannian stochastic variance reduced gradient on Grassmann manifold," *arXiv preprint arXiv:1605.07367*, 2016.
[20] H. Kasai, H. Sato, and B. Mishra, "Riemannian stochastic quasi-Newton algorithm with variance reduction and its convergence analysis," *Prof. Int'l Conf. Artificial Intelligence and Statistics*, 2018.
[21] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Conf. Neutral Information Processing Systems*, 2013, pp. 315–323.
[22] L. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," *Proc. Int'l Conf. Machine Learning*, 2018.
[23] L. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "Stochastic recursive gradient algorithm for nonconvex optimization," *arXiv preprint arXiv:1705.07261*, 2017.
[24] C. Fang, C. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path integrated differential estimator," *arXiv preprint arXiv:1807.01695*, 2018.
[25] Wen Huang, P-A Absil, and Kyle A Gallivan, "A riemannian symmetric rank-one trust-region method," *Mathematical Programming*, vol. 150, no. 2, pp. 179–216, 2015.
[26] P. Zhou, X. Yuan, and J. Feng, "Faster first-order methods for stochastic non-convex optimization on riemannian manifolds," in *Prof. Int'l Conf. Artificial Intelligence and Statistics*, 2019.
[27] C. Udriste, *Convex functions and optimization methods on Riemannian manifolds*, vol. 297, Springer Science & Business Media, 1994.
[28] P. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.
[29] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Springer Science & Business Media, 2006.
[30] Y. Liu, F. Shang, J. Cheng, H. Cheng, and L. Jiao, "Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds," in *Proc. Conf. Neutral Information Processing Systems*, 2017, pp. 4868–4877.
[31] 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.
[32] R. Adler, J. Dedieu, J. Margulies, M. Martens, and M. Shub, "Newton's method on Riemannian manifolds and a geometric model for the human spine," *IMA Journal of Numerical Analysis*, vol. 22, no. 3, pp. 359–390, 2002.
[33] P. Absil and J. Malick, "Projection-like retractions on matrix manifolds," *SIAM Journal on Optimization*, vol. 22, no. 1, pp. 135–158, 2012.
[34] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.
[35] W. Huang, K. Gallivan, and P. Absil, "A broyden class of quasi-Newton methods for Riemannian optimization," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1660–1685, 2015.
[36] B. Polyak, "Gradient methods for the minimisation of functionals," *USSR Computational Mathematics and Mathematical Physics*, vol. 3, no. 4, pp. 864–878, 1963.
[37] Y. Nesterov and B. Polyak, "Cubic regularization of Newton method and its global performance," *Mathematical Programming*, vol. 108, no. 1, pp. 177–205, 2006.
[38] D. Zhou, P. Xu, and Q. Gu, "Stochastic nested variance reduction for nonconvex optimization," *arXiv preprint arXiv:1806.07811*, 2018.
[39] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.
[40] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 643–660, Jun. 2001.
[41] A. Martinez and R. Benavente, "The AR face database," *CVC Tech. Rep. 24*, Jun. 1998.
[42] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1615–1618, Dec. 2003.
[43] Nicolas Boumal, *Optimization and estimation on manifolds.*, Ph.D. thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2014.
[44] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 11, 2011.

Pan Zhou received Master Degree in computer science from Peking University in 2016. Now he is a Ph.D. candidate at the Department of Electrical and Computer Engineering (ECE), National University of Singapore, Singapore. His research interests include computer vision, machine learning, and optimization. He was the winner of the Microsoft Research Asia Fellowship 2018.



Xiao-Tong Yuan received the BA degree in computer science from Nanjing University of Posts and Telecommunications, in 2002, the ME degree in electrical engineering from Shanghai Jiao-Tong University, in 2005, and the PhD degree in pattern recognition from Chinese Academy of Sciences, in 2009. After graduation, he held various appointments as postdoctoral research associate working in the Department of Electrical and Computer Engineering, National University of Singapore, the Department of Statistics and Biostatistics, Rutgers University, and the Department of Statistical Science, Cornell University. In 2013, he joined Nanjing University of Information Science & Technology where, he is currently a professor of computer science. His main research interests include machine learning, data mining, and computer vision. He is a member of the IEEE.



Shuicheng Yan is chief scientist of Qihoo/360 company, and also the Dean's Chair Associate Professor at National University of Singapore. Dr. Yan's research areas include machine learning, computer vision and multimedia, and he has authored/co-authored hundreds of technical papers over a wide range of research topics, with Google Scholar citation over 20,000 times and H-index 66. He is ISI Highly-cited Researcher of 2014, 2015 and 2016. His team received 7 times winner or honorable-mention prizes in PASCAL VOC and ILSVRC competitions, along with more than 10 times best (student) paper prizes. He is also an IAPR Fellow and IEEE Fellow.

Jiashi Feng received the Ph.D. degree from the National University of Singapore (NUS) in 2014. He was a Post-Doctoral Research Follow with the University of California at Berkeley, Berkeley. He joined NUS as a Faculty Member, where he is currently an Assistant Professor with the Department of Electrical and Computer Engineering. His research areas include computer vision, machine learning, robust learning, and deep learning.