Title

Advances in Optimization on Riemannian Manifolds

Author

Li, Jiaxiang

Publication Date

2023

Peer reviewed|Thesis/dissertation

Advances in Optimization on Riemannian Manifolds

By

JIAXIANG LI

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Krishnakumar Balasubramanian, Chair

Shiqian Ma

Thomas Strohmer

Committee in Charge

2023

To my parents

Contents

Advances in Optimization on Riemannian Manifolds

Abstract

Optimization on Riemannian manifolds is a topic that draws attention widely in the optimization community due to its applications in various fields. The problem differs from classical nonconvex optimization due to the loss of linearity. In this dissertation, we inspect the optimization problems on Riemannian manifolds with different aspects.

In the second chapter, we consider stochastic zeroth-order optimization over Riemannian submanifolds. We propose estimators of the Riemannian gradient and Hessians and use them to solve Riemannian optimization problems in the multiple settings and analyze their convergence theoretically. We also provide numerous numerical examples to verify the efficacy of the proposed method.

In the third chapter, we continue the topic of the second chapter by incorporating Riemannian moving-average stochastic gradient estimators. This improves the analysis of the previous chapter by achieving optimal sample complexities to get ϵ-approximation first-order stationary points with batch-free iteration. We also improve the algorithm's practicality by using retractions and vector transport which reduces per-iteration complexity.

In the fourth chapter, we consider the federated learning problem on Riemannian manifolds, with applications such as federated PCA and federated kPCA. We propose a Riemannian federated SVRG method and analyze its convergence rate under different scenarios. Numerical experiments are conducted to show that the advantages of the proposed method are significant.

In the last chapter, we consider a class of Riemannian optimization problems where the objective is the sum of a smooth function and a nonsmooth function. We propose a Riemannian alternating direction method of multipliers (ADMM) with easy computable steps in each iteration. The iteration complexity of the proposed algorithm for obtaining an ϵ-stationary point is analyzed under mild assumptions. Numerical experiments are conducted to demonstrate the advantage of the proposed method.

Acknowledgments

First and foremost, I'd like to show my heartfelt thanks to my esteemed advisors, Prof. Shiqian Ma and Prof. Krishnakumar Balasubramanian. Their patience, wisdom, and support have been instrumental in shaping my research and academic growth. Prof. Ma introduced me to the fascinating realm of optimization, providing me with invaluable insights and advice. Prof. Balasubramanian has been helping me from the very beginning of my Ph.D., offering unwavering support and mentorship. I consider myself immensely fortunate to have had the privilege of working with them for the past five years.

I am also deeply indebted to the distinguished members of my qualification and dissertation committee, Prof. Thomas Strohmer, Prof. Albert Fannjiang, and Prof. Lifeng Lai. Their expertise, suggestions, and valuable insights have played a crucial role in the development of my research. I appreciate their dedicated time and effort in evaluating my work, helping me refine it, and ultimately aiding me in achieving this academic milestone.

The Department of Mathematics at UC Davis has been an invaluable academic home, and I wish to express my gratitude to the outstanding faculty members who helped me in my Ph.D. journey. I am also profoundly thankful to the friendly and supportive staff in the math department. I'm also honored to be deeply connected with the Department of Statistics since Prof. Xiaodong Li served as my initial advisor and Prof. Balasubramanian became my advisor in my 5th year.

It has been a privilege to share this journey with exceptional peers and friends at UC Davis. Special thanks to Tesi Xiao, my friend since our undergraduate days and a priceless companion throughout our Ph.D. studies. Tesi's friendship and academic companionship have been a source of strength. I offer my heartfelt congratulations to him on his recent marriage and wish him a lifetime of love and happiness. I'd also like to express my appreciation to all my friends, including Shouwei Hui, Ye He, Xuxing Chen, and countless others. I feel greatly honored to have all of the people in my life during my Ph.D. journey.

Last, my deepest gratitude goes to my parents, who planted the seed of knowledge in my mind and nurtured it with constant love and support. There has been sorrow and sadness in my family during the pandemic yet they stood beside me, providing silent strength through the ups and downs of my Ph.D. studies. I thank them for being the real hero to support me to this accomplishment.

CHAPTER 1

Introduction

Optimization on Riemannian manifolds is a topic that draws attention widely in the optimization community due to its applications in various fields, including low-rank matrix completion [Boumal and Absil, 2011, Vandereycken, 2013], phase retrieval [Bendory et al., 2017, Sun et al., 2018], dictionary learning [Cherian and Sra, 2016, Sun et al., 2017b], dimensionality reduction [Harandi et al., 2017, Mishra et al., 2019, Tripuraneni et al., 2018] and manifold regression [Lin et al., 2017, 2020]. The problem can be formulated abstractly as

$$\min_{x\in\mathcal{M}} f(x) \tag{1.1}$$

where $\mathcal{M}$ is a d-dimensional Riemannian manifold and $f : \mathcal{M} \to \mathbb{R}$ is a function, usually assumed to be smooth. Note that f cannot satisfy the common notion of convexity due to the loss of linearity since we are now on a manifold. In practice one might just have the noisy estimate of the function/gradient, namely we have

$$\min_{x\in\mathcal{M}} f(x) := \mathbb{E}_\xi[F(x;\xi)] \tag{1.2}$$

where we only have access to the stochastic function $F(x;\xi)$. One concrete example is the so-called finite-sum structured problem:

$$\min_{x\in\mathcal{M}} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1.3}$$

where we only have access to f_i.

Another extension from (1.1) is the nonsmooth composite problem:

$$\min_{x\in\mathcal{M}\subset\mathbb{R}^D} f(x) + h(x) \tag{1.4}$$

where h is usually assumed to be a proper convex function in the ambient space $\mathbb{R}^D$.

Manifold optimization algorithms usually transform an manifold constrained problem into an unconstrained problem by viewing the manifold as the ambient space and using proper retraction to deal with the loss of linearity, thus achieve better convergence results. We refer to [Absil et al., 2008, Boumal, 2023] for a detailed discussion on general Riemannian optimization methods. In this dissertation, we mainly focus on solving the previously mentioned problems in the following aspects:

(1) We studied solving (1.1), (1.2) and (1.4) by zeroth-order (a.k.a. gradient-free) methods and analyzed it convergence behaviour. In particular, we solve (1.2) with a fully-online batch-free derivative-free algorithm by utilizing the moving average technique. Our proposed methods are the state-of-the-art methods for solving these three problems in gradient-free settings;

(2) We studied solving the stochastic problem (1.2) in federated learning setting to enable distributed machine learning on manifolds. The proposed algorithm utilizes a novel average-on-the-tangent-space technique which enables a much faster convergence in numerical experiments. We also provide theoretical convergence guarantee under certain algorithmic conditions;

(3) We proposed and studied the convergence of a Riemannian ADMM algorithm for solving (1.4). The proposed algorithm requires only mild conditions to converge to a KKT point for (1.4) and is the first algorithm with convergence guarantee in the line of operator-splitting algorithms for solving (1.4).

We provide a comprehensive review of literature before proceeding to the main body of this dissertation.

For smooth Riemannian optimization (1.1), it was shown that Riemannian gradient descent method require $\mathcal{O}(1/\epsilon^2)$ iterations to converge to an ϵ-stationary point [Boumal et al., 2018]. Stochastic algorithms for solving (1.2) were also studied [Bonnabel, 2013, Kasai et al., 2018, Weber and Sra, 2019, Zhang et al., 2016b, Zhou et al., 2019]. In particular, using the SPIDER variance reduction technique, Zhou et al. [2019] proved that $\mathcal{O}(1/\epsilon^3)$ oracle calls are required to obtain an ϵ-stationary point in expectation. When the function f takes a finite-sum structure as (1.3), the

Riemannian SVRG [Zhang et al., 2016b] achieves ϵ-stationary solution with $\mathcal{O}(k^{2/3}/\epsilon^2)$ oracle calls where k is number of summands.

Various works have studied the situation where one have no access to the gradient of f or F in (1.1), (1.2) and (1.4), which we refer to as gradient-free or zeroth-order optimization in this monograph. Most of these work consider the situation when $\mathcal{M}$ is simply the Euclidean space $\mathbb{R}^d$, and we refer the reader to Audet and Hare [2017], Conn et al. [2009], Larson et al. [2019] for more details. The oracle complexity of methods from the above works are at least linear in terms of their dependence on dimensionality. Recent works in this field have been focusing on stochastic zeroth-order optimization in high-dimensions Balasubramanian and Ghadimi [2021], Cai et al. [2022], Golovin et al. [2019], Wang et al. [2018]. Assuming a sparse structure (for example, the function being optimized depends only on s of the d coordinates), the above works have shown that the oracle complexity of zeroth-order optimization depends only poly-logarithmically on the dimension d, and it has a linear dependency only on the sparsity parameter s, which is typically small compared to d in several applications. Compared to these works, we assume a manifold structure on the function being optimized and obtain oracle complexities that depend only on the manifold dimension and independent of the ambient Euclidean dimension.

Apart from the above, *Bayesian optimization* is yet another popular class of methods for optimizing functions based on noisy function values. This approach aims at finding the global minimizer by enforcing a Gaussian process prior on the space of function being optimized and using Bayesian sampling techniques. We refer the reader to Frazier [2018], Mockus [1994, 2012], Shahriari et al. [2015] for an overview of such techniques in the Euclidean settings and their applications to a variety of fields including robotics, recommender systems, preference learning and hyperparameter tuning. A common limitation of the above algorithms is that they usually do not scale well to solve high-dimensional problems. Recent developments on Bayesian optimization for high-dimensional problems include Li et al. [2016], Mutny and Krause [2018], Rolland et al. [2018], Wang et al. [2020b, 2016] where people considered zeroth-order optimization with structured functions (for example, sparse or additive functions), and developed Bayesian optimization algorithms and related analysis.

Very recently, Jaquier and Rozo [2020], Jaquier et al. [2020], Oh et al. [2018] considered heuristic Bayesian optimization algorithms for function defined over non-Euclidean domains, including Riemannian domains, without any theoretical analysis.

For the finite-sum setting in (1.3), we called it a distributed optimization problem if each of the function F_i is stored in different devices. We call it a federated learning (FL) problem if there exists one central server that can access all of each F_i with certain communication concern. For the distributed optimization setting, minimizing the communication cost when solving (1.3) becomes another important concern. When $\mathcal{M}$ is simply the Euclidean space $\mathbb{R}^d$, perhaps the most natural idea for FL is the `FedAvg` algorithm [McMahan et al., 2017], which averages local gradient descent updates and yields a good empirical convergence. However in the data heterogeneous situation, `FedAvg` suffers from the client-drift effect that each local client will drift the solution towards the minimum of their own local loss function [Charles and Konečnỳ, 2021, Karimireddy et al., 2020, Li et al., 2019, Malinovskiy et al., 2020, Mitra et al., 2021, Pathak and Wainwright, 2020]. Many ideas were studied to resolve this issue. For example, Li et al. [2020] proposed the `FedProx` algorithm, which regularizes each of the local gradient descent update to ensure that the local iterates are not far from the previous consensus point. The `FedSplit` Pathak and Wainwright [2020] was proposed later to further mitigate the client-drift effect and convergence results were obtained for convex problems. `FedNova` Wang et al. [2020a] was also proposed to improve the performance of `FedAvg`, however it still suffers from a fundamental speed-accuracy conflict under objective heterogeneity Mitra et al. [2021]. Variance reduction techniques were also incorporated to FL leading to two new algorithms: federated SVRG (`FSVRG`) [Konečnỳ et al., 2016] and `FedLin` [Mitra et al., 2021]. These two algorithms require transmitting the full gradient from the central server to each local client for local gradient updates, therefore require more communication between clients and the central server. Nevertheless, `FedLin` achieves the theoretical lower bound for strongly convex objective functions [Mitra et al., 2021] with an acceptable amount of increase in the communication cost.

Decentralized distributed optimization on manifolds has also drawn attentions in recent years [Alimisis et al., 2021, Chen et al., 2021b, Shah, 2017]. Under this setting, each local agent solves a local problem and then the central server takes the consensus step. The consensus step is usually

done by calculating the Karcher mean on the manifold [Shah, 2017, Tron et al., 2012], or calculating the minimizer of the sum of the square of the Euclidean distances in the embedded submanifold case [Chen et al., 2021b]. Such consensus steps usually require solving an additional problem inexactly with no exact convergence rate guarantee [Chen et al., 2021c, Tron et al., 2012]. It is worth mentioning that the PCA problem under federated learning setting has been considered in the literature Grammenos et al. [2020]. The proposed method in Grammenos et al. [2020] relies on the SVD of data matrices and a subspace merging technique and the aim of the algorithm in Grammenos et al. [2020] is to achieve (ϵ, δ)-differential privacy. Note that above works are for decentralized distributed manifold optimization, where as federated learning manifold optimization is still largely empty in the literature.

When the nonsmooth function h presents as in (1.4), Riemannian sub-gradient methods (RSGM) are widely used [Borckmans et al., 2014, Li et al., 2021] and they require $\mathcal{O}(1/\epsilon^4)$ iterations. ADMM for solving (1.4) has also been studied [Kovnatsky et al., 2016, Lai and Osher, 2014], but they usually lack convergence guarantee, while the analysis presented in [Zhang et al., 2020] requires some strong assumptions. Our work of Riemannian ADMM [Li et al., 2022] proposed and analyzed an ADMM-type algorithm under mild conditions. On the other hand, proximal gradient type methods are also studied for solving (1.4). For example, the manifold proximal gradient method (ManPG) [Chen et al., 2020] for solving (1.4) requires $\mathcal{O}(1/\epsilon^2)$ number of iterations to find an ϵ-stationary solution. Variants of ManPG such as ManPPA [Chen et al., 2021a], ManPL [Wang et al., 2022b] and stochastic ManPG [Wang et al., 2022a] have also been studied. Note that none of these works considers the zeroth-order setting. Recently, there are some attempts on stochastic zeroth-order Riemannian optimization [Chattopadhyay et al., 2015, Fong and Tino, 2022], but they are mostly heuristics and do not have any rigorous convergence guarantees. We refer to Chapter 5 for a detailed review on primal-dual based manifold optimization.

1.1. Basics on numerical optimization

In this section we briefly review basic concepts for numerical optimization, which we refer to Beck [2017], Bubeck et al. [2015], Nesterov [2018], Nocedal and Wright [1999] for a detailed study.

Note that we don't consider the manifold constraint $\mathcal{M}$ here and only deal with Euclidean functions in this section, and $\langle x, y\rangle = x^\top y$ is the common Euclidean inner product.

We first review the concept of (Lipschitz) smoothness of a function.

DEFINITION 1.1.1 ((Lipschitz) smoothness). *A continuous differentiable function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ is called L-Lipschitz smooth if we have: $\forall x, y \in \Omega$*

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|x - y\| \tag{1.5}$$

Further, we have (see Beck [2017])

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|^2. \tag{1.6}$$

The next concept is the (strongly) convexity which is the central notion of convex optimization.

DEFINITION 1.1.2 ((Strong) convexity). *Consider a convex set $\Omega \subset \mathbb{R}^d$. A function $f : \Omega \to \mathbb{R}$ is called convex if for any $x, y \in \Omega$, we have $f((1-t)x + ty) \leq (1-t)f(x) + tf(y)$. It's further called μ-strongly convex if we have $f((1-t)x + ty) \leq (1-t)f(x) + tf(y) - \frac{\mu t(1-t)}{2}\|x - y\|^2$.*

If f is a continuous differentiable function, then it is convex if and only if (see Beck [2017]) $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle$, and is μ-strongly convex if and only if $h(y) \geq h(x) + \langle \nabla h(x), y - x\rangle + \frac{\mu}{2}\|x - y\|^2$.

If f is a second differentiable function, then it is convex if and only if (see Beck [2017]) $\frac{d^2 f((1-t)x+ty)}{dt^2} \geq 0$, and is μ-strongly convex if and only if $\frac{d^2 f((1-t)x+ty)}{dt^2} \geq \mu$.

The importance of convexity is that it guarantees that every local minimum is global minimum, thus provides a tamed environment that allows gradient-based algorithm to converge nicely; see Bubeck et al. [2015] for a comprehensive study.

Now we discuss the notion of stationarity for both convex and nonconvex problems.

DEFINITION 1.1.3 (Global and local minimizer). *$f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$. x^* is the global minimizer of f if $\forall x \in \Omega$,*

$$f(x^*) \leq f(x).$$

x^* is a local minimizer of f if there exist a neighborhood $U \subset \Omega$, $x^* \in U$, we have $\forall x \in U$,

$$f(x^*) \leq f(x).$$

A famous result in convex optimization is that x^* is the global minimizer of a proper convex function f if and only if $0 \in \partial f(x^*)$, where ∂ denotes the set of subgradients. We refer to [Bubeck et al., 2015] for a detailed study on classical convex optimization results.

For nonconvex optimization one usually cannot achieve a global (or even local) minimizer, thus we have the following notion of stationary point

DEFINITION 1.1.4 (Stationary point). *$f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ is continuous differentiable. $\bar{x}$ is the stationary point of f if*

$$\nabla f(\bar{x}) = 0$$

also $\bar{x}$ is the ϵ-(approximate) stationary point of f if

$$\|\nabla f(\bar{x})\| \leq \epsilon$$

We will generalize all of these notions to their manifold counterparts in the next sections. Now we turn to some basic notions in stochastic optimization. Suppose a function f, which is the expectation of some stochastic function:

$$f(x) = \mathbb{E}_\xi[F(x;\xi)].$$

We have the following assumptions that we utilize in the following chapters to character the approximation error between f and F:

DEFINITION 1.1.5 (Unbiased and bounded-variance estimators). *We say $\nabla F(x;\xi)$ is an unbiased estimator of ∇f if*

$$\mathbb{E}_\xi[\nabla F(x;\xi)] = \nabla f(x)$$

also $\nabla F(x;\xi)$ is an estimator of ∇f with bounded variance σ^2 if

$$\mathbb{E}_\xi\|\nabla F(x;\xi) - \nabla f(x)\|^2 \leq \sigma^2.$$

Note that if $F(x;\xi) = f(x) + \xi$ where $\xi \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable then the above assumption is naturally satisfied.

We also have the following stationarity notion under stochastic setting

DEFINITION 1.1.6 (Stationary point for stochastic setting). *If f and F are continuous differentiable. $\bar{x}$ is the ϵ-(approximate) stationary point of f if*

$$\mathbb{E}_\xi \|\nabla F(\bar{x};\xi)\|^2 \leq \epsilon^2.$$

1.2. Basics on Riemannian manifolds

In this part, we briefly review the basic Riemannian manifold tools we use for optimization on Riemannian manifolds; see Boumal [2023], Do Carmo [1992], Lee [2006], Tu [2011] for the details. Suppose $\mathcal{M}$ is an m-dimensional differentiable manifold. The tangent space $\mathrm{T}_x\mathcal{M}$ at $x \in \mathcal{M}$ is a linear subspace that consists of the derivatives of all differentiable curves on $\mathcal{M}$ passing through x: $\mathrm{T}_x\mathcal{M} := \{\gamma'(0) : \gamma(0) = x, \gamma([-\delta, \delta]) \subset \mathcal{M} \text{ for some } \delta > 0, \gamma \text{ is differentiable}\}$. Notice that for every vector $\gamma'(0) \in \mathrm{T}_x\mathcal{M}$, it can be defined in a coordinate-free sense via the operation over smooth functions: $\forall f \in C^\infty(\mathcal{M})$, $\gamma'(0)(f) := \frac{df\circ\gamma(t)}{dt} \mid_{t=0}$. The notion of Riemannian manifold is defined as follows.

DEFINITION 1.2.1 (Riemannian manifold). *Manifold $\mathcal{M}$ is a Riemannian manifold if it is equipped with an **inner product** on the tangent space, $\langle\cdot,\cdot\rangle_x : \mathrm{T}_x\mathcal{M} \times \mathrm{T}_x\mathcal{M} \to \mathbb{R}$, that varies smoothly on $\mathcal{M}$. The $(0,2)$-tensor field $\langle\cdot,\cdot\rangle_x$ is usually referred to as Riemannian metric.*

As an example, consider the Stiefel manifold given by

$$\mathcal{M} = \mathrm{St}(n,p) := \{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}. \tag{1.7}$$

The tangent space of $\mathrm{St}(n,p)$ is given by $\mathrm{T}_X\mathcal{M} = \{\xi \in \mathbb{R}^{n\times p} : X^\top\xi + \xi^\top X = 0\}$. One could equip the tangent space with common inner product $\langle X, Y\rangle := \mathrm{tr}(X^\top Y)$ to form a Riemannian manifold. For additional examples, see Absil et al. [2008, Chapter 3] or Boumal [2023, Chapter 7] .

We also review the notion of the differential between manifolds here.

DEFINITION 1.2.2 (Differential and Riemannian gradients). *Let $F : \mathcal{M} \to \mathcal{N}$ be a C^∞ map between two differential manifolds. At each point $x \in \mathcal{M}$, the differential of F is a mapping (also known as the push-forward):*

$$\mathrm{D}F : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_{F(x)} \mathcal{N}$$

s.t. $\forall \xi \in \mathrm{T}_x \mathcal{M}$, $\mathrm{D}F(\xi) \in \mathrm{T}_x \mathcal{N}$ is given by

$$(\mathrm{D}F(\xi))(f) := \xi(f \circ F) \in \mathbb{R}, \ \forall f \in C^\infty_{F(x)}(\mathcal{M})$$

If $\mathcal{N} = \mathbb{R}$, i.e. $f \in C^\infty(\mathcal{M})$, the differential of f is usually denoted as $\mathrm{d}f$. For a Riemannian manifold with Riemannian metric $\langle \cdot, \cdot \rangle$, the Riemannian gradient for $f \in C^\infty(\mathcal{M})$ is the unique tangent vector $\mathsf{grad} f(x) \in \mathrm{T}_x \mathcal{M}$ s.t.

$$\mathrm{d}f(\xi) = \langle \mathsf{grad} f, \xi \rangle_x, \ \forall \xi \in \mathrm{T}_x \mathcal{M}.$$

If $\mathcal{M}$ is an embedded submanifold of a Euclidean space and ∇f is the common Euclidean gradient, then we have [Boumal, 2023]

$$\mathsf{grad} f(x) = \mathsf{proj}_{\mathrm{T}_x \mathcal{M}}(\nabla f(x)).$$

where proj is just the Euclidean orthogonal projection.

We also need the notion of exponential mapping and parallel transport for the next chapters. To this end, we need to first recall the definition of a geodesic.

DEFINITION 1.2.3 (Geodesic, exponential mapping and retractions). *Given $x \in \mathcal{M}$ and $\xi \in \mathrm{T}_x \mathcal{M}$, the geodesic is the curve $\gamma : I \to \mathcal{M}$, $0 \in I \subset \mathbb{R}$ is an open set, so that $\gamma(0) = x$, $\dot{\gamma}(0) = \xi$ and $\nabla_{\dot{\gamma}} \dot{\gamma} = 0$ where $\nabla : \mathrm{T}_x \mathcal{M} \times \mathrm{T}_x \mathcal{M} \to \mathrm{T}_x \mathcal{M}$ is the Levi-Civita connection defined by Riemannian metric tensor. In local coordinate sense, γ is the unique solution of the following second-order differential equations:*

$$\frac{d^2 \gamma^k}{dt^2} + \Gamma^k_{i,j} \frac{d\gamma^i}{dt} \frac{d\gamma^j}{dt} = 0$$

under Einstein summation convention, where $\Gamma^k_{i,j}$ are Christoffel symbols, again defined by metric tensor. The exponential mapping Exp_x is defined as a mapping from $\mathrm{T}_x \mathcal{M}$ to $\mathcal{M}$ s.t. $\mathsf{Exp}_x(\xi) :=$

$\gamma(1)$ *with* γ *being the geodesic with* $\gamma(0) = x$, $\dot{\gamma}(0) = \xi$. *A natural corollary is* $\mathsf{Exp}_x(t\xi) := \gamma(t)$ *for* $t \in [0, 1]$.

Given any curve $\gamma(t)$ *on* $\mathcal{M}$, *one could calculate the length of the curve and define the distance between the two points* $x, y \in \mathcal{M}$ *respectively by* $L(\gamma) := \int_a^b \|\gamma'(t)\|_{\gamma(t)} dt$ *and* $\operatorname{dist}(x, y) := \min_{\gamma, \gamma(a)=x, \gamma(b)=y} L(\gamma)$. *If the manifold is a complete Riemannian manifold, according to Do Carmo [1992, Corollary 3.9], there exists a unique minimal geodesic* γ *satisfying* $\gamma(a) = x, \gamma(b) = y$ *that minimizes* $L(\gamma)$. *Therefore, we can always calculate the distance with respect to the minimal geodesic as* $\operatorname{dist}(x, y) = \int_a^b \|\gamma'(t)\|_{\gamma(t)} dt, \nabla_{\gamma'}\gamma' = 0, \gamma(a) = x, \gamma(b) = y$.

A retraction mapping Retr_x *is a smooth mapping from* $\mathrm{T}_x \mathcal{M}$ *to* $\mathcal{M}$ *such that:* $\mathsf{Retr}_x(0) = x$, *where* 0 *is the zero element of* $\mathrm{T}_x \mathcal{M}$, *and the differential of* Retr_x *at* 0 *is an identity mapping, i.e.,* $\frac{d\mathsf{Retr}_x(t\eta)}{dt}\Big|_{t=0} = \eta$, $\forall \eta \in \mathrm{T}_x \mathcal{M}$. *In particular, the exponential mapping* Exp_x *is a special case of retraction. Notice that the retraction is not always injective from* $\mathrm{T}_x \mathcal{M}$ *to* $\mathcal{M}$ *for any point* $x \in \mathcal{M}$, *thus the existence of the inverse of the retraction function* Retr_x^{-1} *is not guaranteed. However, when* $\mathcal{M}$ *is complete, the exponential mapping* Exp_x *is always defined for every* $\xi \in \mathrm{T}_x \mathcal{M}$, *and the inverse of the exponential mapping* $\mathsf{Exp}_x^{-1}(y) \in \mathrm{T}_x \mathcal{M}$ *is always well-defined for any* $x, y \in \mathcal{M}$. *Also, since* $\mathsf{Exp}_x(t\xi)$ *generates geodesics, we have* $\operatorname{dist}(x, \mathsf{Exp}_x(t\xi)) = t\|\xi\|_x$.

Throughout this dissertation, we always assume that $\mathcal{M}$ is complete, so that Exp_x is always define for every $\xi \in \mathrm{T}_x \mathcal{M}$. For $\forall x, y \in \mathcal{M}$, the inverse of the exponential mapping $\mathsf{Exp}_x^{-1}(y) \in \mathrm{T}_x \mathcal{M}$ is called the logarithm mapping, and we have $\operatorname{dist}(x, y) = \|\mathsf{Exp}_x^{-1}(y)\|_x$, which derives directly from $\operatorname{dist}(x, \mathsf{Exp}_x(\xi)) = \|\xi\|_x$.

As an example, the retractions on Stiefel manifolds can be defined by the QR decomposition, $R_X(\xi) := Q$ where $X + \xi = QR$. It can also be defined through the Polar decomposition as $R_X(\xi) := UV^\top$, where $X + \xi = U\Sigma V^\top$ is the (thin) singular value decomposition of $X + \xi$. The geodesic on the Stiefel manifold is given by:

$$X(t) = \begin{bmatrix} X(0) & \dot{X}(0) \end{bmatrix} \exp\left(t \begin{bmatrix} A(0) & -S(0) \\ I & A(0) \end{bmatrix} \right) \begin{bmatrix} I \\ 0 \end{bmatrix} \exp(-A(0)t),$$

for $A(t) = X^\top(t)\dot{X}(t)$ and $S(t) = \dot{X}^\top(t)\dot{X}(t)$ with initial point $X(0)$ and initial speed $\dot{X}(0)$. The exponential mapping is thus given by $\mathsf{Exp}_{X(0)}(\dot{X}(0)) = X(1)$. The computation cost of the QR and

Polar decomposition retractions are of order $2nk^2 + \mathcal{O}(k^3)$ and $3nk^2 + \mathcal{O}(k^3)$, whereas as shown by Chen et al. [2020, Section 3] the exponential mapping takes $8nk^2 + \mathcal{O}(k^3)$, which illustrates the favorability of retractions in practical computations. We refer to Absil et al. [2008, Chapter 4] and Boumal [2023, Chapter 3] for additional examples and more discussions on retractions and exponential mappings.

With the notion of geodesic, we have the following definition of geodesic convexity and strong-convexity, which are the generalizations of their Euclidean counterparts:

DEFINITION 1.2.4 (Geodesic (strong) convexity). *A geodesic convex set $\Omega \subset \mathcal{M}$ is a set such that for any two points in the set, there exists a geodesic connecting them that lies entirely in Ω. A function $h : \Omega \to \mathbb{R}$ is called geodesic convex if for any $p, q \in \Omega$, we have $h(\gamma(t)) \leq (1-t)h(p)+th(q)$ where γ is a geodesic in Ω with $\gamma(0) = p$ and $\gamma(1) = q$. It's called μ-geodesic strongly convex if we have $h(\gamma(t)) \leq (1-t)h(p) + th(q) - \frac{\mu t(1-t)}{2} \operatorname{dist}(p, q)^2$.*

If h is a continuous differentiable function, then it is geodesic convex if and only if (see [Boumal, 2023, Chapter 11]) $h(q) \geq h(p)+\langle \mathsf{grad} h(p), \mathsf{Exp}_p^{-1}(q)\rangle_p$, and is geodesic strongly convex if and only if $h(q) \geq h(p) + \langle \mathsf{grad} h(p), \mathsf{Exp}_p^{-1}(q)\rangle_p + \frac{\mu}{2} \operatorname{dist}(p, q)^2$.

If h is a second differentiable function, then it is geodesic convex if and only if (see [Boumal, 2023, Chapter 11]) $\frac{d^2h(\gamma(t))}{dt^2} \geq 0$, and is geodesic strongly convex if and only if $\frac{d^2h(\gamma(t))}{dt^2} \geq \mu$.

We also present the definition of vector and parallel transport, which are also used later in our algorithm design and convergence analysis.

DEFINITION 1.2.5 (Vector and parallel transport). *A vector transport $\mathcal{T}$ on a smooth manifold $\mathcal{M}$ is a smooth mapping $\mathrm{T}\mathcal{M} \times \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \to \mathcal{T}_{\eta_x}(\xi_x) \in \mathrm{T}\mathcal{M}$, where the subscript x means that the vector is in $\mathrm{T}_x\mathcal{M}$, such that: (i) There exists a retraction R so that $\mathcal{T}_{\eta_x}(\xi_x) \in \mathrm{T}_{R_x(\eta_x)}\mathcal{M}$, (ii) $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $\xi_x \in \mathrm{T}_x\mathcal{M}$, and (iii) $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$, i.e., linearity. Particularly, for a complete Riemannian manifold $(\mathcal{M}, \langle\cdot,\cdot\rangle)$, we can construct a special vector transport, namely the parallel transport P, that can map vectors to another tangent space "parallelly", i.e., $\forall \eta, \xi \in \mathrm{T}_x\mathcal{M}$ and $y \in \mathcal{M}$,*

$$\langle P_{\mathsf{Exp}_x^{-1}(y)}(\eta), P_{\mathsf{Exp}_x^{-1}(y)}(\xi)\rangle_y = \langle \eta, \xi\rangle_x. \tag{1.8}$$

Notice that parallel transport is not the only transport that satisfies (1.8)*, and we call the vector transport an isometric vector transport if it satisfies* (1.8)*.*

We can equivalently view P as a mapping from the tangent space $\mathrm{T}_x\,\mathcal{M}$ to $\mathrm{T}_y\,\mathcal{M}$. We hence denote $P_x^y : \mathrm{T}_x\,\mathcal{M} \to \mathrm{T}_y\,\mathcal{M}$. Note that parallel transport depends on the curve along which the vectors are moving. If the curve is not specified, it refers to the case when we are considering the minimal geodesic connecting the two points, which exists due to completeness.

As an example, for the Stiefel manifold in (1.7), there is no closed-form expression for the parallel transport, whereas one can always utilize the projection onto the tangent space, given by $\mathsf{proj}_{\mathrm{T}_X\,\mathcal{M}}(\xi) = (I - XX^\top)\xi + X\,\mathrm{skew}(X^\top\xi)$, where $\mathrm{skew}(A) := (A - A^\top)/2$, to transport $\xi \in \mathrm{T}_{X_0}\,\mathrm{St}(n,p)$ to $\mathrm{T}_X\,\mathrm{St}(n,p)$. We refer to Absil et al. [2008, Chapter 8] and Boumal [2023, Chapter 10] for additional examples and more discussions on vector and parallel transports.

We also have the following definition of Lipschitz smoothness on the manifolds:

Definition 1.2.6 ((Geodesic) Lipschitz smoothness). *A function $f : \Omega \subset \mathcal{M} \to \mathbb{R}$ is called (Geodesic) L-Lipschitz smooth if we have: $\forall x, y \in \Omega$*

$$\|\mathsf{grad} f(y) - P_{x\to y}\mathsf{grad} f(x)\| \le L\,\mathrm{dist}(x,y) \tag{1.9}$$

Further, we have (see [Zhang et al., 2016b])

$$f(y) \le f(x) + \langle \mathsf{grad} f(x), \mathsf{Exp}_x^{-1}(y)\rangle_x + \frac{L}{2}\,\mathrm{dist}(x,y)^2. \tag{1.10}$$

We new present the notion of Riemannian Hessian.

Definition 1.2.7 (Riemannian Hessian). *For function $f : \mathcal{M} \to \mathbb{R}$, the Riemannian Hessian is a symmetric 2-form $\mathrm{Hess}(f) : \mathrm{T}\,\mathcal{M} \times \mathrm{T}\,\mathcal{M} \to \mathbb{R}$ is defined as: $\forall \xi, \eta \in \mathrm{T}\,\mathcal{M}$,*

$$\mathrm{Hess}(f)(\xi,\eta) = \langle \nabla_\xi \mathsf{grad} f, \eta\rangle$$

where ∇ here is the Levi-Civita connection. H can also be interpreted as a linear map $\mathrm{Hess}(f) : \mathrm{T}\,\mathcal{M} \to \mathrm{T}\,\mathcal{M}$, $\forall \xi \in \mathrm{T}_x\,\mathcal{M}$,

$$\mathrm{Hess}(f)(\xi) = \nabla_\xi \mathsf{grad} f.$$

We now discuss the notion of second fundamental form, which will be helpful in characterizing a geometric condition used in later to quantify the error of approximating parallel transports with vector transports. In general, the notion of second fundamental form can be studied for general isometric immersions and we restrict here to the embedding in Euclidean spaces only for brevity.

DEFINITION 1.2.8 (Second fundamental form). *Suppose $\mathcal{M} \subset \mathbb{R}^D$ is a complete Riemannian manifold equipped with the Euclidean metric. For any $\xi, \eta \in \mathrm{T}\mathcal{M}$, denote the extension of two vector fields to $\mathbb{R}^D$ as $\bar{\xi}, \bar{\eta} \in \mathbb{R}^D$, also the directional derivative of $\bar{\eta}$ along $\bar{\xi}$ as $\bar{\nabla}_{\bar{\xi}}\bar{\eta} \in \mathbb{R}^D$. The second fundamental form refers to the bilinear and symmetric vector, $B(\xi, \eta) = \bar{\nabla}_{\bar{\xi}}\bar{\eta} - \nabla_{\xi}\eta \in (\mathrm{T}\mathcal{M})^{\perp}$, which quantifies the deviation of the Riemannian directional derivatives (depicted by Levi-Civita connection ∇) from the Euclidean one (common directional derivative $\bar{\nabla}$).*

Finally, we remark that there are various definitions of second fundamental forms, among which the most common one is a quadratic form related to B; see Do Carmo [1992, Chapter 6, Definition 2.2]. Here we simply refer to B as the second fundamental form.

1.3. Basic Riemannian optimization scheme

In this section we study basic schemes for solving (1.1) and (1.2) and provide their convergence behavior. The aim is to provide a flavor of the research in this subject without digging into the detailed setting of the main bodies of this dissertation. We refer to Absil et al. [2008], Boumal [2023], Boumal et al. [2018] for the details.

We have the following basic Riemannian gradient descent scheme for solving the Riemannian optimization problem (1.1) (see Absil et al. [2008]):

(1) Compute the Riemannian gradient $\mathsf{grad} f(x^k)$ at the current point x^k, either by definition or by projection (embedded submanifold case).
(2) Use a retraction operator $\mathsf{Retr}_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M}$ to map the update $-\eta_k \mathsf{grad} f(x^k)$ back to $\mathcal{M}$, as a result we get the next iteration by:

$$x^{k+1} = \mathsf{Retr}_{x^k}(-\eta_k \mathsf{grad} f(x^k)) \tag{1.11}$$

For (1.2), we have to change the update (1.11) into:

$$x^{k+1} = \mathsf{Retr}_{x^k}(-\eta_k \mathsf{grad} F(x^k; \xi_k)) \tag{1.12}$$

since in (1.2) we only have the access to the stochastic function $F(x;\xi)$. Here ξ_k is an independent and identically distributed (i.i.d.) sample. In practice one may employ the mini-batch sampling technique, which means that at iteration k, we sample multiple i.i.d. ξ to further boost the performance:

$$x^{k+1} \leftarrow \mathsf{Retr}_{x^k}(-\eta_k G^k), \text{ with } G^k = \frac{1}{m_k}\sum_{i=1}^{m_k} \mathsf{grad} F(x^k; \xi_{k,i}) \tag{1.13}$$

Note that if $\mathcal{M} = \mathbb{R}^d$, the above scheme reduces to the common gradient descent method since we can always take $\mathsf{Retr}_x(y) = x + y$ in Euclidean spaces.

Now to proceed to the convergence analysis, we need the following assumptions over our function f and F:

Assumption 1.3.1. *Function f in* (1.1) *and* (1.2) *is L-geodesic Lipschitz smooth, i.e. satisfies Definition 1.2.6.*

Assumption 1.3.2. *For the function F* (1.2)*, stochastic gradients of F are unbiased and have bounded-variance, i.e. we have $\mathbb{E}_\xi[\mathsf{grad} F(x;\xi)] = \mathsf{grad} f(x)$ and $\mathbb{E}_\xi[\|\mathsf{grad} F(x;\xi) - \mathsf{grad} f(x)\|_x^2] \le \sigma^2$.*

We have the following theorems for the convergence of RSGD (1.11):

Theorem 1.3.1 (Convergence of RSGD (1.11)). *Assume the inverse exponential map is well-defined on $\mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}$. Suppose Assumption 1.3.1 holds, then the RSGD algorithm in* (1.11) *with $\eta_k \equiv \eta = 1/L$ satisfies:*

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\|\mathsf{grad} f(x^k)\|^2 \le \frac{2L(f(x^0) - f^*)}{T} \tag{1.14}$$

Proof. Since

$$\begin{aligned} f(x^{k+1}) \le & f(x^k) + \langle \mathsf{grad} f(x^k), \mathsf{Exp}_{x^k}^{-1}(x^{k+1})\rangle + \frac{L}{2}\|\mathsf{Exp}_{x^k}^{-1}(x^{k+1})\|^2 \\ = & f(x^k) - (\eta - \frac{\eta^2 L}{2})\|\mathsf{grad} f(x^k)\|^2 \end{aligned}$$

if we take $\eta = \frac{1}{L}$ we get:

$$\frac{1}{2L}\|\mathsf{grad} f(x^k)\|^2 \leq f(x^k) - f(x^{k+1})$$

and the result follows by summing up above inequality from $k = 0$ to $k = T - 1$ (which is usually refer to as "telescoping" trick). □

We have the following theorems for the convergence of (mini-batch) stochastic RSGD (1.12):

THEOREM 1.3.2 (Convergence of stochastic RSGD (1.13)). *Assume the inverse exponential map is well-defined on $\mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}$. Suppose Assumption 1.3.1 and 1.3.2 hold, then the mini-batch RSGD algorithm in* (1.13) *with $\eta_k \equiv \eta = 1/L$ and $m_k \equiv m$ satisfies:*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathsf{grad} f(x^k)\|^2 \leq \frac{L(f(x^0) - f^*)}{2T} + \frac{\sigma^2}{4m} \tag{1.15}$$

PROOF. Denote

$$G^k := \frac{1}{m_k}\sum_{i=1}^{m_k}\mathsf{grad} F(x^k, \xi_{k,i})$$

which is the mini-batch stochastic gradient. Since

$$\begin{aligned} f(x^{k+1}) \leq & f(x^k) + \langle \mathsf{grad} f(x^k), \mathsf{Exp}_{x^k}^{-1}(x^{k+1})\rangle + \frac{L}{2}\|\mathsf{Exp}_{x^k}^{-1}(x^{k+1})\|^2 \\ = & f(x^k) - \eta\langle \mathsf{grad} f(x^k), G^k\rangle + \frac{\eta^2 L}{2}\|G^k\|^2 \\ = & f(x^k) + \frac{\eta^2 L}{2}\|G^k - \frac{1}{\eta L}\mathsf{grad} f(x^k)\|^2 - \frac{2}{L}\|\mathsf{grad} f(x^k)\|^2, \end{aligned}$$

if we take $\eta = \frac{1}{L}$ and take the expectation, we have

$$\mathbb{E} f(x^{k+1}) \leq \mathbb{E} f(x^k) + \frac{\eta^2 L}{2}\frac{\sigma^2}{m} - \frac{2}{L}\mathbb{E}\|\mathsf{grad} f(x^k)\|^2,$$

and we obtain the following inequality by telescoping:

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\mathsf{grad} f(x^k)\|^2 \leq \frac{L(f(x^0) - f^*)}{2T} + \frac{\sigma^2}{4m}.$$

□

To get rid of the batch size m, we have the follow theorem:

THEOREM 1.3.3 (Theorem 5 in Zhang et al. [2016a]). *Suppose the same setting as above theorem, then the RSGD algorithm in* (1.12) *(i.e.* (1.13) *with* $m_k = 1$*) with* $\eta = c/\sqrt{T}$, $c = \sqrt{\frac{2(f(x^0)-f(x^*))}{L\sigma^2}}$ *satisfies:*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}[\|\mathsf{grad} f(x^k)\|^2] \leq \sqrt{\frac{2(f(x^0)-f(x^*))L}{T}}\sigma$$

These fundamental results shows the things we can do: we are able to bound the norm of the Riemannian gradient to a reasonable level, i.e. we are able to approach a stationary point. The goals in the next chapters would still be around this central concern.

CHAPTER 2

Stochastic Gradient-free Algorithms for Riemannian Optimization

2.1. Introduction

In this chapter we consider the Riemannian optimization problem in (1.4), which we restate here:

$$\min_{x \in \mathcal{M} \subset \mathbb{R}^D} f(x) + h(x) \tag{2.1}$$

where $\mathcal{M}$ is a Riemannian submanifold embedded in $\mathbb{R}^D$, $f : \mathcal{M} \to \mathbb{R}$ is a smooth and possibly non-convex function, and $h : \mathbb{R}^D \to \mathbb{R}$ is a convex and nonsmooth function. Here, convexity and smoothness are interpreted as the function is being considered in the ambient Euclidean space. Unless stated otherwise, we consider the stochastic setting for f, i.e., $f(x) = \mathbb{E}_\xi[F(x, \xi)] = \int_\xi F(x, \xi) dP(\xi)$ where P refers to the distribution of random vector ξ. Iterative algorithms for solving (2.1) usually require the gradient and Hessian information of the objective function. However, in many applications, the analytical form of the function f (or h) and its gradient are not available, and we can only obtain noisy function evaluations via a zeroth-order oracle. This setting, termed as *stochastic zeroth-order Riemannian optimization*, generalizes stochastic zeroth-order Euclidean optimization (i.e., when $\mathcal{M} \equiv \mathbb{R}^d$ in (2.1)), a topic which goes back to the early works of Matyas [1965], Nelder and Mead [1965], Nemirovski and Yudin [1983] in the 1960's; see also Audet and Hare [2017], Conn et al. [2009], Larson et al. [2019] for recent books and surveys.

In the Euclidean setting, two popular techniques for estimating the gradient from (noisy) function queries include the finite-differences method [Spall, 2005] and the Gaussian smoothing techniques [Nemirovski and Yudin, 1983]. Earlier works in this setting focused on using the estimated gradient to obtain asymptotic convergence rates of iterative optimization algorithms. Recently, obtaining non-asymptotic guarantees on the oracle complexity of stochastic zeroth-order optimization has been of great interest. Towards that, Nesterov [2011], Nesterov and Spokoiny [2017] analyzed

the Gaussian smoothing technique for estimating the Euclidean gradient from noisy function evaluations and proved that for unconstrained convex minimization, one needs $O(d^2/\epsilon^2)$ noisy function evaluations to obtain an ϵ-optimal solution.

This complexity was improved by Ghadimi and Lan [2013] to $O(d/\epsilon^2)$ when the objective function is further assumed to be gradient-smooth. Note that this oracle complexity depends linearly on the problem dimension n and it was proved that the linear dependency on d is unavoidable Duchi et al. [2015], Jamieson et al. [2012]. Nonconvex and smooth setting was also considered in Ghadimi and Lan [2013]. In particular, now assuming $h \equiv 0$ and $\mathcal{M} \equiv \mathbb{R}^d$ in (2.1), it was shown that the number of function evaluations for obtaining an ϵ-stationary point $\bar{x}$ (i.e., $\mathbb{E}\|\nabla f(\bar{x})\|^2 \leq \epsilon^2$), is $O(d/\epsilon^4)$.

2.1.1. Motivating Applications. Our motivation for developing a theoretical framework for stochastic zeroth-order Riemannian optimization is due to several important emerging applications; see, e.g., Chattopadhyay et al. [2015], Jaquier et al. [2020], Kachan [2020], Marco et al. [2017], Yuan et al. [2019]. Below, we discuss two concrete examples, which we will revisit in Section 2.4.2, to illustrate the applicability of the methods developed in this work. We also briefly discuss a third application in topological data analysis, and numerical experiments on this application will be conducted in a future work, as it is more involved and beyond the scope of this paper.

2.1.1.1. *Black-box Stiffness Control for Robotics.* Our first motivating application is from the field of robotics. It has become increasingly common to use zeroth-order optimization techniques to optimize control parameter and policies in robotics Drieß et al. [2017], Marco et al. [2016], Yuan et al. [2019]. This is because that the cost functions being optimized in robotics are not available in a closed form as a function of the control parameter. Invariably for a given choice of control parameter, the cost function needs to be evaluated through a real-world experiment on a given robot or through simulation. Recently, domain knowledge has been used as constraints on the control parameter space, among which a common choice is the geometry-aware constraint. For example, control parameters like stiffness, inertia and manipulability lie on the positive semidefinite manifold, orthogonal group and unit sphere, respectively. Hence, there is a need to develop zeroth-order optimization methods over the manifolds to optimize the above mentioned control parameters Jaquier et al. [2020].

2.1.1.2. *Zeroth-order Attacks on Deep Neural Networks (DNNs).* Our second motivating application is based on developing black-box attacks to DNNs. Despite the recent success of DNNs, studies have shown that they are vulnerable to adversarial attacks: even a well-trained DNN could completely misclassify a slightly perturbed version of the original image (which is undetectable to the human eyes); see, e.g., Goodfellow et al. [2014], Szegedy et al. [2013]. As a result, it is extremely important on the one hand to come up with methods to train DNNs that are robust to adversarial attacks, and on the other hand to develop efficient attacks on DNNs with the goal being to make them misclassify. In practice, as the architecture of the DNN is not known to the attacker, several works, for example, Chen et al. [2017], Cheng et al. [2018], Tu et al. [2019] use zeroth-order optimization algorithms for designing adversarial attacks. However, existing works have an inherent drawback– the perturbed testing example designed to fool the DNN is not in the same domain as the original training data. For example, despite the fact that natural images typically lie on a manifold Weinberger and Saul [2004], the perturbations are not constrained to lie on the same manifold. This naturally motivates us to use zeroth-order Riemannian optimization methods to design adversarial examples to fool DNNs, which at the same time preserves the manifold structures in the dataset. As demonstrated by our numerical experiments, the Riemannian black-box attack succeeds more often than the Euclidean black-box attack when the attack region is small; See Section 2.4.2 for more details.

2.1.1.3. *Black-box Methods for Topological Dimension Reduction.* The third motivating example is from the field of dimension reduction, a popular class of techniques for reducing the dimension of high-dimensional unstructured data for feature extraction and visualization. There exists a variety of methods for this task; we refer the interested reader to Burges [2010], Lee and Verleysen [2007] for more details. However, a majority of the existing techniques are based on *geometric* motivations. Recently, there has been a growing literature on using *topological* information for performing data analysis [Chazal and Michel, 2021, McInnes et al., 2018, Rabadán and Blumberg, 2019]. One such method is a dimension reduction technique called Persistent Homology-Based Projection Pursuit [Kachan, 2020]. Roughly speaking, given a point-cloud data set with cardinality n and dimension m (i.e., a matrix $X \in \mathbb{R}^{m\times n}$), persistence homology refers to developing a multiscale characterization of topologically invariant features available in the data. Such information is

summarized in terms of the so-called persistence diagram, $D(X)$, which is a multiset of points in a two-dimensional plane. The idea in Kachan [2020] is to obtain a transformation $P^\top \in \mathbb{R}^{p\times m}$, with $p \ll m$, such that the topological summaries of the original dataset X and the reduced dimensional dataset $P^\top X$ are close to each other; that is, the persistence diagram $D(X)$ and $D(P^\top X)$ are close in the 2-Wasserstein distance. The problem is then formulated as (informally speaking),

$$\min_{\{P\in\mathbb{R}^{m\times p}:P^\top P=I\}} W_2(D(X), D(P^\top X)),$$

which is an optimization problem over the Stiefel manifold. It turns out that calculating the gradient of the above objective function is highly non-trivial and computationally expensive Leygonie et al. [2021]. However, evaluating the objecting function for various value of the matrix P is relatively less expensive. Hence, this serves as yet another problem in which the methodology developed in this work could be applied naturally.

2.1.2. Main Contributions. We now summarize our main contributions.

(1) In Section 2.1.3, we propose the (stochastic) zeroth-order Riemannian gradient (2.2) and Hessian (2.12) estimators, which addresses the infeasibility issue of the sampling for the case of derivative-free optimization over manifolds.

(2) In Section 2.3, we demonstrate the applicability of the developed estimators for stochastic zeroth-order Riemannian optimization, as listed below. A summary of these results is given in Table 2.1. To the best of our knowledge, our results are the first complexity results for stochastic zeroth-order Riemannian optimization.

- When $h(x) \equiv 0$, and $F(x, \xi)$ satisfies certain Riemannian gradient smoothness assumption, we propose a zeroth-order Riemannian stochastic gradient descent method (`ZO-RSGD`) and analyze its oracle complexity under two different settings (see Theorem 2.3.1).
- When $h(x)$ is convex and nonsmooth, we propose a zeroth-order stochastic Riemannian proximal gradient method (`ZO-SManPG`) and provide its oracle complexity for obtaining an ϵ-stationary point of (2.1) (see Theorem 2.3.2).

Algorithm	Structure	Iteration Complexity	Oracle Complexity
ZO-RSGD	smooth, stochastic	$\mathcal{O}\left(1/\epsilon^2\right)$	$\mathcal{O}(d/\epsilon^4)$
ZO-RSGD	smooth, stochastic, Geo-convex	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}\left(d/\epsilon^2\right)$
ZO-SManPG	nonsmooth stochastic	$\mathcal{O}\left(1/\epsilon^2\right)$	$\mathcal{O}\left(d/\epsilon^4\right)$
ZO-RSCRN	Lipschitz Hessian stochastic	$\mathcal{O}(1/\epsilon^{1.5})$	$\mathcal{O}\left(d/\epsilon^{3.5} + d^4/\epsilon^{2.5}\right)$

Table 2.1. Summary of the convergence results proved in this paper. For all but the ZO-RSCRN algorithm, the reported complexities correspond to ϵ-stationary solution; for the ZO-RSCRN algorithm the complexities correspond to ϵ-local minimizers. Here, d is the intrinsic dimension of the manifold $\mathcal{M}$. Furthermore, Iteration complexity refers to the number of iterations and oracle complexity refers to the number of calls to the (stochastic) zeroth-order oracle.

- When $h(x) \equiv 0$ and $F(x, \xi)$ satisfies certain Lipschitz Riemannian Hessian property, we propose a zeroth-order Riemannian stochastic cubic regularized Newton's method (ZO-RSCRN) that provably converges to an ϵ-approximate local minimizer (see Theorem 2.3.3).

(3) In Section 2.4, we provide experimental results on simulated data to quantify the performance of our methods. We then demonstrate the applicability of our methods to the problem of black-box attacks to deep neural networks and robotics.

2.1.3. Zeroth-order Riemannian Gradient Estimator. Recall that in the Euclidean setting, Nesterov and Spokoiny Nesterov and Spokoiny [2017] analyzed the Gaussian smoothing based zeroth-order gradient estimator. However, as that estimator requires function evaluations outside of the manifold to be well-defined, it is not directly applicable for the Riemannian setting. To address this issue, we introduce our stochastic zeroth-order Riemannian gradient estimator below.

Definition 2.1.1 (Zeroth-Order Riemannian Gradient). *Generate $u = Pu_0 \in \mathrm{T}_x\mathcal{M}$, where $u_0 \sim \mathcal{N}(0, I_n)$ in $\mathbb{R}^n$, and $P \in \mathbb{R}^{n\times n}$ is the orthogonal projection matrix onto $\mathrm{T}_x\mathcal{M}$. Therefore u follows the standard normal distribution $\mathcal{N}(0, PP^\top)$ on the tangent space in the sense that, all the eigenvalues of the covariance matrix $PP^\top$ are either 0 (eigenvectors orthogonal to the tangent plane) or 1 (eigenvectors embedded in the tangent plane). The zeroth-order Riemannian gradient*

estimator is defined as

$$g_\mu(x) = \frac{f(\mathsf{Retr}_x(\mu u)) - f(x)}{\mu} u = \frac{f(\mathsf{Retr}_x(\mu P u_0)) - f(x)}{\mu} P u_0. \tag{2.2}$$

Note that the projection P is easy to compute for commonly used manifolds. For example, for the Stiefel manifold $\mathcal{M}$, the projection is given by $\mathsf{proj}_{T_x\mathcal{M}}(Y) = (I - XX^\top)Y + X\,\mathrm{skew}(X^\top Y)$, where $\mathrm{skew}(A) := (A - A^\top)/2$ (see Absil et al. [2008]).

REMARK 2.1.1. *In this work, we assume that the function f is defined on submanifolds embedded in Euclidean space, so that it is efficient to sample from the associated tangent space, as discussed above; see also Diaconis et al. [2013]. We remark that the above gradient estimation methodology is more generally applicable to other manifolds. However, the generality comes at the cost of practical applicability as it is not an easy task to efficiently sample Gaussian random objects on the tangent space of general manifolds; see Hsu [2002] for more details.*

We now discuss some differences between the zeroth-order gradient estimators in the Euclidean setting [Nesterov and Spokoiny, 2017] and the Riemannian setting (2.2). In the Euclidean case, the zeroth-order gradient estimator can be viewed as estimating the gradient of the Gaussian smoothed function, $f_\mu(x) = \frac{1}{\kappa}\int_{\mathbb{R}^n} f(x+\mu u)e^{-\frac{1}{2}\|u\|^2}du$, because $\nabla f_\mu(x) = \mathbb{E}_u(g_\mu(x)) = \frac{1}{\kappa}\int_{\mathbb{R}^n}\frac{f(x+\mu u)-f(x)}{\mu}ue^{-\frac{1}{2}\|u\|^2}du$, where κ is the normalization constant for Gaussian. This was also observed as an instantiation of Gaussian Stein's identity Balasubramanian and Ghadimi [2021]. However, this observation is no longer true in the Riemannian setting, as we incorporate the retraction operator when evaluating g_μ, and this forces us to seek for a direct evaluation of $\mathbb{E}_u(g_\mu(x))$, instead of utilizing properties of the smoothed function f_μ. We also remark that, $g_\mu(x)$ is a biased estimator of $\mathsf{grad} f(x)$. The difference between them can be bounded as in Proposition 2.1.1. Some intermediate results for this purpose are as follows.

LEMMA 2.1.1. *Suppose $\mathcal{X}$ is a d-dimensional subspace of $\mathbb{R}^n$, with orthogonal projection matrix $P \in \mathbb{R}^{n\times n}$. u_0 follows a standard normal distribution $\mathcal{N}(0, I_n)$, and $u = Pu_0$ is the orthogonal projection of u_0 onto the subspace $\mathcal{X}$. Then $\forall x \in \mathcal{X}$, we have*

$$x = \frac{1}{\kappa}\int_{\mathbb{R}^n}\langle x, u\rangle u e^{-\frac{1}{2}\|u_0\|^2}du_0, \qquad \textit{and} \qquad \|x\|^2 = \frac{1}{\kappa}\int_{\mathbb{R}^n}\langle x, u\rangle^2 e^{-\frac{1}{2}\|u_0\|^2}du_0, \tag{2.3}$$

where κ is the constant for normal density function: $\kappa := \int_{\mathbb{R}^n} e^{-\frac{1}{2}\|u\|^2} du = (2\pi)^{n/2}$.

PROOF. Proof of Lemma 2.1.1 By the definition of covariance matrix, we have $\frac{1}{\kappa}\int_{\mathbb{R}^n} u_0 u_0^\top e^{-\frac{1}{2}\|u_0\|^2} du_0 = I_n$. Since $\langle x, u\rangle = \langle x, u_0\rangle$, $\forall x \in \mathcal{X}$, we have

$$\frac{1}{\kappa}\int_{\mathbb{R}^n} \langle x, u\rangle u_0 e^{-\frac{1}{2}\|u_0\|^2} du_0 = x, \tag{2.4}$$

which implies $\frac{1}{\kappa}\int_{\mathbb{R}^n} \langle x, u\rangle u e^{-\frac{1}{2}\|u_0\|^2} du_0 = Px = x$. Similarly, taking inner product with x on both sides of Eq. (2.4), we have $\|x\|^2 = \frac{1}{\kappa}\int_{\mathbb{R}^n} \langle x, u\rangle^2 e^{-\frac{1}{2}\|u_0\|^2} du_0$. □

The following bound for the moments of normal distribution is restated without proof.

LEMMA 2.1.2. *[Nesterov and Spokoiny, 2017] Suppose $u \sim \mathcal{N}(0, I_n)$ is a standard normal distribution. Then for all integers $p \geq 2$, we have $M_p := \mathbb{E}_u(\|u\|^p) \leq (n+p)^{p/2}$.*

COROLLARY 2.1.1. *For $u_0 \sim \mathcal{N}(0, I_n)$ and $u = Pu_0$, where $P \in \mathbb{R}^{n\times n}$ is the orthogonal projection matrix onto a d dimensional subspace $\mathcal{X}$ of $\mathbb{R}^n$, we have $\mathbb{E}_{u_0}(\|u\|^p) \leq (d+p)^{p/2}$.*

PROOF. Proof of Corollary 2.1.1 Assume the eigen-decomposition of P is $P = Q^\top \Lambda Q$, where Q is an unitary matrix and Λ is a diagonal matrix with the leading d diagonal entries being 1 and other diagonal entries being 0. Denote $\tilde{u} = Qu_0 \sim \mathcal{N}(0, I_n)$, then $\Lambda\tilde{u} = (\tilde{u}_1, ..., \tilde{u}_d, 0, ..., 0)$. Since $u = Q^\top \Lambda \tilde{u}$ has the same distribution as $\Lambda\tilde{u}$, we have $\mathbb{E}\|u\|^p = \mathbb{E}\|(\tilde{u}_1, ..., \tilde{u}_d, 0, ..., 0)\|^p \leq (d+p)^{p/2}$, by Lemma 2.1.2. □

Now we provide the bounds on the error of our gradient estimator $g_\mu(x)$ in (2.2). To proceed we need the following assumption:

ASSUMPTION 2.1.1 (L-retraction-smoothness). *There exists $L_g \geq 0$ such that the following inequality holds for function f in* (2.1)*:*

$$|f(\mathsf{Retr}_x(\eta)) - f(x) - \langle \mathsf{grad} f(x), \eta\rangle_x| \leq \frac{L_g}{2}\|\eta\|^2, \forall x \in \mathcal{M}, \eta \in \mathrm{T}_x\mathcal{M}. \tag{2.5}$$

Assumption 2.1.1 is also known as the restricted Lipschitz-type gradient for pullback function $\hat{f}_x(\eta) := f(\mathsf{Retr}_x(\eta))$ Boumal et al. [2018]. The condition required in Boumal et al. [2018] is weaker because it only requires Eq. (2.5) to hold for $\|\eta\|_x \leq \rho_x$, where constant $\rho_x > 0$. In our convergence

analysis, we need this assumption to be held for all $\eta \in \mathrm{T}_x \mathcal{M}$, i.e., $\rho_x = \infty$. This assumption is satisfied when the manifold $\mathcal{M}$ is a compact submanifold of $\mathbb{R}^n$, the retraction Retr_x is globally defined[1] and function f is L-smooth in the Euclidean sense; we refer the reader to Boumal et al. [2018] for more details. We also emphasize that Assumption 2.1.1 is weaker than the geodesic smoothness assumption defined in Zhang and Sra [2016]. The geodesic smoothness states that, $\forall \eta \in \mathcal{M}$, $f(\mathsf{Exp}_x(\eta)) \leq f(x) + \langle g_x, \eta \rangle_x + L_g d^2(x, \mathsf{Exp}_x(\eta))/2$, where g_x is a subgradient of f, $d(\cdot, \cdot)$ represents the geodesic distance. Such a condition is stronger than our Assumption 2.1.1, in the sense that, if the retraction is the exponential mapping, then geodesic smoothness implies the L-retraction-smoothness with the same parameter L_g Bento et al. [2017].

Recall that d denotes the dimension of the manifold $\mathcal{M}$, we have the following error bounds:

PROPOSITION 2.1.1. *Under Assumption 2.1.1, we have*

(a) $\|\mathbb{E}_{u_0}(g_\mu(x)) - \mathsf{grad} f(x)\| \leq \frac{\mu L_g}{2}(d+3)^{3/2}$,

(b) $\|\mathsf{grad} f(x)\|^2 \leq 2\|\mathbb{E}_{u_0}(g_\mu(x))\|^2 + \frac{\mu^2}{2} L_g (d+6)^3$,

(c) $\mathbb{E}_{u_0}(\|g_\mu(x)\|^2) \leq \frac{\mu^2}{2} L_g^2 (d+6)^3 + 2(d+4)\|\mathsf{grad} f(x)\|^2$.

PROOF. Proof of Proposition 2.1.1 For part (a), since

$$\mathbb{E}(g_\mu(x)) - \mathsf{grad} f(x) = \frac{1}{\kappa} \int_{\mathbb{R}^n} \left(\frac{f(\mathsf{Retr}_x(\mu u)) - f(x)}{\mu} - \langle \mathsf{grad} f(x), u \rangle \right) u e^{-\frac{1}{2}\|u_0\|^2} du_0,$$

we have (by Lemma 2.1.1)

$$\begin{aligned} & \|\mathbb{E}(g_\mu(x)) - \mathsf{grad} f(x)\| \\ = \; & \|\frac{1}{\mu\kappa} \int_{\mathbb{R}^n} (f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u \rangle)\, u e^{-\frac{1}{2}\|u_0\|^2} du_0 \| \\ \leq \; & \frac{1}{\mu\kappa} \int_{\mathbb{R}^n} \frac{L_g}{2} \|\mu u\|^2 \|u\| e^{-\frac{1}{2}\|u_0\|^2} du_0 = \frac{\mu L_g}{2\kappa} \int_{\mathbb{R}^n} \|u\|^3 e^{-\frac{1}{2}\|u_0\|^2} du_0 \leq \frac{\mu L_g}{2}(d+3)^{3/2}, \end{aligned}$$

where the first inequality is by due to (2.5), and the last inequality is from Corollary 2.1.1. This completes the proof of part (a).

[1] If the manifold is compact, then the exponential mapping Exp_x is already globally defined. This is known as the Hopf-Rinow theorem Do Carmo [1992].

To prove part (b), note that

$$
\begin{aligned}
&\|\mathsf{grad} f(x)\|^2 = \left\| \frac{1}{\kappa} \int_{\mathbb{R}^n} \langle \mathsf{grad} f(x), u\rangle u e^{-\frac{1}{2}\|u_0\|^2} du_0 \right\|^2 \\
=&\left\| \frac{1}{\mu\kappa} \int_{\mathbb{R}^n} ([f(\mathsf{Retr}_x(\mu u)) - f(x)] - [f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u\rangle]) u e^{-\frac{1}{2}\|u_0\|^2} du_0 \right\|^2 \\
\leq& 2\|\mathbb{E}(g_\mu(x))\|^2 + \frac{2}{\mu^2} \left\| \int_{\mathbb{R}^n} (f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u\rangle) u e^{-\frac{1}{2}\|u_0\|^2} du_0 \right\|^2 \\
\leq& 2\|\mathbb{E}(g_\mu(x))\|^2 + \frac{2}{\mu^2} \int_{\mathbb{R}^n} (f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u\rangle)^2 \|u\|^2 e^{-\frac{1}{2}\|u_0\|^2} du_0 \\
\leq& 2\|\mathbb{E}(g_\mu(x))\|^2 + \frac{\mu^2}{2} L_g (d+6)^3,
\end{aligned}
$$

where the last inequality is from the same trick as in part (a). This completes the proof of part (b).

Finally, we prove part (c). Since $\mathbb{E}(\|g_\mu(x)\|^2) = \frac{1}{\mu^2}\mathbb{E}_{u_0}\left[(f(\mathsf{Retr}_x(\mu u)) - f(x))^2 \|u\|^2\right]$, and $(f(\mathsf{Retr}_x(\mu u)) - f(x))^2 = (f(\mathsf{Retr}_x(\mu u)) - f(x) - \mu\langle \mathsf{grad} f(x), u\rangle + \mu\langle \mathsf{grad} f(x), u\rangle)^2 \leq 2(\frac{L_g}{2}\mu^2\|u\|^2)^2 + 2\mu^2\langle \mathsf{grad} f(x), u\rangle^2$, we have

(2.6)

$$
\mathbb{E}(\|g_\mu(x)\|^2) \leq \frac{\mu^2}{2} L_g^2 \mathbb{E}(\|u\|^6) + 2\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^2) \leq \frac{\mu^2}{2} L_g^2 (d+6)^3 + 2\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^2).
$$

Now we bound the term $\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^2)$ using the same trick as in Nesterov and Spokoiny [2017]. Without loss of generality, assume $\mathcal{X}$ is the d-dimensional subspace generated by the first d coordinates, i.e., $\forall x \in \mathcal{X}$, the last $n-d$ elements of x are zeros. Also for brevity, denote $g = \mathsf{grad} f(x)$. We have that

$$
\begin{aligned}
\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^2) &= \frac{1}{\kappa} \int_{\mathbb{R}^n} \langle \mathsf{grad} f(x), u\rangle^2 \|u\|^2 e^{-\frac{1}{2}\|u_0\|^2} du_0 \\
&= \frac{1}{\kappa(d)} \int_{\mathbb{R}^d} \left(\sum_{i=1}^d g_i x_i \right)^2 \left(\sum_{i=1}^d x_i^2 \right) e^{-\frac{1}{2}\sum_{i=1}^d x_i^2} dx_1 \cdots dx_d,
\end{aligned}
$$

where x_i denotes the i-th coordinate of u_0, the last $n-d$ dimensions are integrated to be one, and $\kappa(d)$ is the normalization constant for d-dimensional Gaussian distribution. For simplicity, denote

$x = (x_1, ..., x_d)$, then

$$
\begin{aligned}
&\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^2) = \frac{1}{\kappa(d)} \int_{\mathbb{R}^d} \langle g, x\rangle^2 \|x\|^2 e^{-\frac{1}{2}\|x\|^2} dx \\
\leq &\frac{1}{\kappa(d)} \int_{\mathbb{R}^d} \|x\|^2 e^{-\frac{\tau}{2}\|x\|^2} \langle g, x\rangle^2 e^{-\frac{1-\tau}{2}\|x\|^2} dx \leq \frac{2}{\kappa(d)\tau e} \int_{\mathbb{R}^d} \langle g, x\rangle^2 e^{-\frac{1-\tau}{2}\|x\|^2} dx \\
= &\frac{2}{\kappa(d)\tau(1-\tau)^{1+d/2} e} \int_{\mathbb{R}^d} \langle g, x\rangle^2 e^{-\frac{1}{2}\|x\|^2} dx = \frac{2}{\tau(1-\tau)^{1+d/2} e} \|g\|^2,
\end{aligned} \tag{2.7}
$$

where the second inequality is due to the following fact: $x^p e^{-\frac{\tau}{2}x^2} \leq (\frac{2}{\tau e})^{p/2}$. Taking $\tau = \frac{2}{(d+4)}$ gives the desired result. □

2.1.4. Zeroth-order Riemannian Hessian Estimator. We now extend the above methodology and propose estimators for the Riemannian Hessian in the stochastic zeroth-order setting. We restrict our discussion to compact submanifolds embedded in Euclidean space, so that the definition of Riemannian Hessian in Definition 1.2.7 is applied. We assume the following assumption of $F(x, \xi)$:

ASSUMPTION 2.1.2. *Given any point $x \in \mathcal{M}$ and $\eta \in \mathrm{T}_x \mathcal{M}$, we have*

$$
\|P_\eta^{-1} \circ \mathrm{Hess} F(\mathsf{Retr}_x(\eta), \xi) \circ P_\eta - \mathrm{Hess} F(x, \xi)\|_{\mathrm{op}} \leq L_H \|\eta\|, \tag{2.8}
$$

almost everywhere for ξ, where $P_\eta : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_{\mathsf{Retr}_x(\eta)} \mathcal{M}$ denotes the parallel transport and $\circ$ is the function composition. Here $\|\cdot\|_{\mathrm{op}}$ is the operator norm in the ambient Euclidean space.

Assumption 2.1.2 is the analogue of the Lipschitz Hessian type assumption from the Euclidean setting, and induces the following equivalent conditions (see, also Agarwal et al. [2021]):

$$
\begin{aligned}
&\|P_\eta^{-1} \mathsf{grad} F(\mathsf{Retr}_x(\eta), \xi) - \mathsf{grad} f(x) - \mathrm{Hess} F(x, \xi)[\eta]\| \leq \frac{L_H}{2} \|\eta\|^2 \\
&\left| F(\mathsf{Retr}_x(\eta), \xi) - \left[F(x, \xi) + \langle \eta, \mathsf{grad} F(x, \xi)\rangle + \frac{1}{2} \langle \eta, \mathrm{Hess} F(x, \xi)[\eta]\rangle \right] \right| \leq \frac{L_H}{6} \|\eta\|^3.
\end{aligned} \tag{2.9}
$$

In the Euclidean setting, P_η reduces to the identity mapping. Throughout this section, we also assume that $F(\cdot, \xi)$ satisfies Assumption 2.1.1 and the following assumption, which is used frequently in zeroth-order stochastic optimization [Balasubramanian and Ghadimi, 2021, Ghadimi and Lan, 2013, Zhou et al., 2019].

ASSUMPTION 2.1.3. *We have (with $\mathbb{E} = \mathbb{E}_\xi$) that, $\mathbb{E}[F(x,\xi)] = f(x)$, $\mathbb{E}[\mathsf{grad}F(x,\xi)] = \mathsf{grad}f(x)$ and $\mathbb{E}\left[\|\mathsf{grad}F(x,\xi) - \mathsf{grad}f(x)\|^2\right] \leq \sigma^2$, $\forall x \in \mathcal{M}$.*

We first introduce the following identity which follows immediately from the second-order Stein's identity for Gaussian distribution [Stein, 1972].

LEMMA 2.1.3. *Suppose $\mathcal{X}$ is a d-dimensional subspace of $\mathbb{R}^n$, with orthogonal projection matrix $P \in \mathbb{R}^{n\times n}$, $P = P^2 = P^\top$, and $u_0 \sim \mathcal{N}(0, I_n)$ is a standard normal distribution and $u = Pu_0$ is the orthogonal projection of u_0 onto the subspace. Then $\forall H \in \mathbb{R}^{n\times n}$, $H^\top = H$, and $H = PHP$ (which means that the eigenvectors of H lies all in $\mathcal{X}$), we have*

$$PHP = \frac{1}{2\kappa}\int_{\mathbb{R}^n} \langle u, Hu\rangle(uu^\top - P)e^{-\frac{1}{2}\|u_0\|^2}du_0 = \mathbb{E}\left[\frac{1}{2}\langle u, Hu\rangle(uu^\top - P)\right], \tag{2.10}$$

where $\|\cdot\|$ here is the Euclidean norm on $\mathbb{R}^n$, and κ is the constant for normal density function given by $\kappa := \int_{\mathbb{R}^n} e^{-\frac{1}{2}\|u\|^2}du = (2\pi)^{n/2}$.

The identity in (2.10) simply follows by applying the second-order Stein's identity, $\mathbb{E}[(xx^\top - I_n)g(x)] = \mathbb{E}[\nabla^2 g(x)]$, directly to the function $g(x) = \frac{1}{2}\langle x, Hx\rangle$ and multiplying the resulting identity by P on both sides.

LEMMA 2.1.4. *[Balasubramanian and Ghadimi, 2021] Suppose $\mathcal{X}$ is a d-dimensional subspace of $\mathbb{R}^n$, with orthogonal projection matrix $P \in \mathbb{R}^{n\times n}$, $P = P^2 = P^\top$, and $u_0 \sim \mathcal{N}(0, I_n)$ is a standard normal distribution and $u = Pu_0$ is the orthogonal projection of u_0 onto the subspace. Then*

$$\mathbb{E}[\|u_0u_0^\top - I_n\|_F^8] \leq 2(n+16)^8 \text{ and } \mathbb{E}[\|uu^\top - P\|_F^8] \leq 2(d+16)^8. \tag{2.11}$$

PROOF. Proof of Lemma 2.1.4 See Balasubramanian and Ghadimi [2021] for the proof of the first inequality in Eq. (2.11). We now show how to get the right part from the left. Similar to the proof of Corollary 2.1.1, we use an eigen-decomposition of $P = Q^T\Lambda Q$ and get (again $\tilde{u} = Qu$):

$$\mathbb{E}\|uu^\top - P\|_F^8 = \mathbb{E}\|(\tilde{u}_1, ..., \tilde{u}_d)^\top(\tilde{u}_1, ..., \tilde{u}_d) - I_d\|_F^8 \leq 2(d+16)^8,$$

which completes the proof. □

We now propose our zeroth-order Riemannian Hessian estimator, motivated by the zeroth-order Hessian estimator in the Euclidean setting proposed by Balasubramanian and Ghadimi [2021].

DEFINITION 2.1.2 (Zeroth-Order Riemannian Hessian). *Generate $u \in \mathrm{T}_x\,\mathcal{M}$ following a standard normal distribution on the tangent space $\mathrm{T}_x\,\mathcal{M}$, by projection $u = P_x u_0$ as described in Section 2.1.3. Then, the zeroth-order Riemannian Hessian estimator of a function f at the point x is given by*

$$H_\mu(x) = \frac{1}{2\mu^2}(uu^\top - P)[F(\mathsf{Retr}_x(\mu u), \xi) + F(\mathsf{Retr}_x(-\mu u), \xi) - 2F(x, \xi)]. \tag{2.12}$$

Note that our Riemannian Hessian estimator is actually the Hessian estimator of the pullback function $\hat{F}_x(\eta, \xi) = F(\mathsf{Retr}_x(\eta), \xi)$, $\forall x \in \mathcal{M}$ and $\eta \in \mathrm{T}_x\,\mathcal{M}$ projected onto the tangent space $T_x\mathcal{M}$.

We immediately have the following bound on the variance of $H_\mu(x)$.

LEMMA 2.1.5. *Under Assumption 2.1.1, the Riemannian Hessian estimator given in Eq.* (2.12) *satisfies*

$$\mathbb{E}_{\mathcal{U},\Xi}\|H_\mu(x)\|_F^4 \leq \frac{(d+16)^8}{8} L_g^2. \tag{2.13}$$

PROOF. Proof of Lemma 2.1.5 From Assumption 2.1.1 and Corollary 2.1.1 we have

(2.14)

$$\begin{aligned}
&\mathbb{E}|F(\mathsf{Retr}_x(\mu u), \xi) + F(\mathsf{Retr}_x(-\mu u), \xi) - 2F(x, \xi)|^8 \\
=&\mathbb{E}|F(\mathsf{Retr}_x(\mu u), \xi) - F(x, \xi) - \langle \mathsf{grad} F(x, \xi), \mu u\rangle + F(\mathsf{Retr}_x(-\mu u), \xi) - F(x, \xi) - \langle \mathsf{grad} F(x, \xi), -\mu u\rangle|^8 \\
\leq&\mathbb{E}[\frac{\mu^2 L_g}{2}\|u\|^2 + \frac{\mu^2 L_g}{2}\|u\|^2]^8 = \mathbb{E}[\mu^{16} L_g^8 \|u\|^{16}] \leq \mu^{16} L_g^8 (d+16)^8.
\end{aligned}$$

Moreover, we have

$$\begin{aligned}
\mathbb{E}\|H_\mu(x)\|_F^4 =&\mathbb{E}\left\|\frac{1}{2\mu^2}(uu^\top - P)[F(\mathsf{Retr}_x(\mu u), \xi) + F(\mathsf{Retr}_x(-\mu u), \xi) - 2F(x, \xi)]\right\|_F^4 \\
\leq&\frac{1}{16\mu^8}\left(\mathbb{E}|F(\mathsf{Retr}_x(\mu u), \xi) + F(\mathsf{Retr}_x(-\mu u), \xi) - 2F(x, \xi)|^8 \mathbb{E}\|uu^\top - P\|^8\right)^{1/2} \\
\leq&\frac{(d+16)^4}{8\mu^8}\left(\mathbb{E}|F(\mathsf{Retr}_x(\mu u), \xi) + F(\mathsf{Retr}_x(-\mu u), \xi) - 2F(x, \xi)|^8\right)^{1/2},
\end{aligned} \tag{2.15}$$

where the first inequality is by Hölder's inequality and the second one is by Lemma 2.1.4. Combining (2.14) and (2.15) yields the desired result (2.13). □

We will also use the mini-batch multi-sampling technique. For $i = 1, ..., b$, denote each Hessian estimator as

$$H_{\mu,i}(x) = \frac{1}{2\mu^2}(u_i u_i^\top - P)[F(\mathsf{Retr}_x(\mu u_i), \xi_i) + F(\mathsf{Retr}_x(-\mu u_i), \xi_i) - 2F(x, \xi_i)]. \tag{2.16}$$

The averaged Hessian estimator is given by

$$\bar{H}_{\mu,\xi}(x) = \frac{1}{b}\sum_{i=1}^{b} H_{\mu,i}(x). \tag{2.17}$$

We now have the following bound of $\bar{H}_{\mu,\xi}(x)$ and $\mathrm{Hess} f(x)$.

LEMMA 2.1.6. *Under Assumption 2.1.1 and Assumption 2.1.2, let $\bar{H}_{\mu,\xi}(x)$ be calculated as in Eq.* (2.17)*, then we have that: $\forall x \in \mathcal{M}$ and $\forall \eta \in \mathrm{T}_x \mathcal{M}$,*

$$\mathbb{E}_{\mathcal{U},\Xi}\|\bar{H}_{\mu,\xi}(x) - \mathrm{Hess} f(x)\|_{\mathrm{op}}^2 \leq \frac{(d+16)^4}{\sqrt{2}b} L_g + \frac{\mu^2 L_H^2}{18}(d+6)^5, \tag{2.18}$$

$$\mathbb{E}_{\mathcal{U},\Xi}\|\bar{H}_{\mu,\xi}(x) - \mathrm{Hess} f(x)\|_{\mathrm{op}}^3 \leq \tilde{C}\frac{(d+16)^6}{b^{3/2}} L_g^{1.5} + \frac{1}{27}\mu^3 L_H^3 (d+6)^{7.5}, \tag{2.19}$$

where $\|\cdot\|_{op}$ denotes the operator norm and $\tilde{C}$ is some absolute constant.

PROOF. Proof of Lemma 2.1.6 Denote $\mathbb{E} = \mathbb{E}_{\mathcal{U},\Xi}$ as the expectation with respect to all previous random variables. We first show Eq. (2.18). Denote $X_i = H_{\mu,i} - \mathbb{E}H_{\mu,i}$, then X_i's are iid zero-mean random matrices. Since $\|\cdot\|_{\mathrm{op}} \leq \|\cdot\|_F$, we have

$$\begin{aligned}
&\mathbb{E}\|\bar{H}_{\mu,\xi}(x) - \mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^2 = \mathbb{E}\left\|\frac{1}{b}\sum_{i=1}^{b} X_i\right\|_{\mathrm{op}}^2 \leq \mathbb{E}\left\|\frac{1}{b}\sum_{i=1}^{b} X_i\right\|_F^2 \\
=&\mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\|X_i\|_F^2 + \frac{1}{b^2}\sum_{i\neq j}\langle X_i, X_j\rangle\right] = \mathbb{E}\left[\frac{1}{b^2}\sum_{i=1}^{b}\|X_i\|_F^2\right] \\
=&\mathbb{E}\frac{1}{b^2} b\|X_1\|_F^2 = \mathbb{E}\frac{1}{b}\|H_{\mu,1} - \mathbb{E}H_{\mu,1}\|_F^2 = \frac{1}{b}\mathbb{E}\left[\|H_{\mu,1}\|_F^2 - \|\mathbb{E}H_{\mu,1}\|_F^2\right] \\
\leq&\frac{1}{b}\mathbb{E}\|H_{\mu,1}\|_F^2 \leq \frac{1}{b}\sqrt{\mathbb{E}\|H_{\mu,1}(x)\|_F^4} \leq \frac{(d+16)^4}{2\sqrt{2}b} L_g,
\end{aligned} \tag{2.20}$$

where the third inequality is from the Jensen's inequality, and the last inequality is due to Eq. (2.13). Note that (2.20) immediately implies

$$\begin{aligned}\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^2 \leq & 2\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^2+2\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^2 \\ \leq & \frac{(d+16)^4}{\sqrt{2}b}L_g+2\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^2.\end{aligned} \tag{2.21}$$

Now we bound the term $\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^2$. Note that

$$\begin{aligned}&|\langle\eta,(\mathbb{E}H_{\mu,i}(x)-\mathrm{Hess}f(x))[\eta]\rangle| \\ =&\left|\langle\eta,\left(\mathbb{E}\left[\frac{1}{2\mu^2}(uu^\top-P)[f(\mathsf{Retr}_x(\mu u))+f(\mathsf{Retr}_x(-\mu u))-2f(x)]\right]-\mathrm{Hess}f(x)\right)[\eta]\rangle\right| \\ =&\left|\langle\eta,\left(\mathbb{E}\left[\frac{1}{2\mu^2}(uu^\top-P)[f(\mathsf{Retr}_x(\mu u))+f(\mathsf{Retr}_x(-\mu u))-2f(x)-\mu^2\langle u,\mathrm{Hess}f(x)[u]\rangle]\right]\right)[\eta]\rangle\right| \\ =&\frac{1}{2\mu^2}\left|\langle\eta,\left(\mathbb{E}\left[[f(\mathsf{Retr}_x(\mu u))-f(x)-\frac{\mu^2}{2}\langle u,\mathrm{Hess}f(x)[u]\rangle\right.\right.\right. \\ &\left.\left.\left.+f(\mathsf{Retr}_x(-\mu u))-f(x)-\frac{\mu^2}{2}\langle u,\mathrm{Hess}f(x)[u]\rangle](uu^\top-P)\right]\right)[\eta]\rangle\right|,\end{aligned}$$

which together with Assumption 2.1.2 yields

$$\begin{aligned}|\langle\eta,(\mathbb{E}H_{\mu,i}(x)-\mathrm{Hess}f(x))[\eta]\rangle| \leq & \frac{\mu L_H}{6}\mathbb{E}\left[\|u\|^3\|uu^\top-P\|_{\mathrm{op}}\right]\|\eta\|^2 \\ \overset{\text{Hölder}}{\leq} & \frac{\mu L_H}{6}\sqrt{\mathbb{E}\|u\|^6\mathbb{E}\|uu^\top-P\|_F^2}\|\eta\|^2 \leq \frac{\mu L_H}{6}(d+6)^{5/2}\|\eta\|^2,\end{aligned} \tag{2.22}$$

where the last inequality is by Corollary 2.1.1 and Lemma 2.1.4. (2.22) implies

$$\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}} \leq \frac{\mu L_H}{6}(d+6)^{5/2}. \tag{2.23}$$

Combining (2.21) and (2.23) gives Eq. (2.18).

Now we show Eq. (2.19). By a similar analysis we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^3\\
\leq&\mathbb{E}(\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}+\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}})^3\\
\leq&8\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^3+8\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^3\\
\overset{\text{Hölder}}{\leq}&8\sqrt{\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^2\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^4}\\
&+8\|\mathbb{E}\bar{H}_{\mu,\xi}(x)-\mathrm{Hess}f(x)\|_{\mathrm{op}}^3,
\end{aligned}
\tag{2.24}
$$

where the second inequality is by the following fact: when $a,b\geq 0$, $(a+b)^3\leq\max\{(2a)^3,(2b)^3\}\leq 8a^3+8b^3$. Moreover, since $\|\cdot\|_{\mathrm{op}}\leq\|\cdot\|_F$, and $X_i=H_{\mu,i}-\mathbb{E}H_{\mu,i}$ are iid zero-mean random matrices, we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{H}_{\mu,\xi}(x)-\mathbb{E}\bar{H}_{\mu,\xi}(x)\|_{\mathrm{op}}^4=\mathbb{E}\|\frac{1}{b}\sum_{i=1}^b X_i\|_{\mathrm{op}}^4\leq\frac{C}{b^4}\left(\mathbb{E}\|\sum_{i=1}^b X_i\|_{\mathrm{op}}+(b\mathbb{E}\|X_i\|_{\mathrm{op}}^4)^{1/4}\right)^4\\
\leq&\frac{C}{b^4}\left(\sqrt{\mathbb{E}\|\sum_{i=1}^b X_i\|_F^2}+(b\mathbb{E}\|X_i\|_F^4)^{1/4}\right)^4=\frac{C}{b^4}\left(\sqrt{\sum_{i=1}^b\mathbb{E}\|X_i\|_F^2}+(b\mathbb{E}\|X_i\|_F^4)^{1/4}\right)^4\\
=&\frac{C}{b^4}\left(\sqrt{b}\sqrt{\mathbb{E}\|X_1\|_F^2}+(b\mathbb{E}\|X_1\|_F^4)^{1/4}\right)^4\leq\frac{C}{b^4}\left(\sqrt{b}\sqrt[4]{\mathbb{E}\|X_1\|_F^4}+(b\mathbb{E}\|X_1\|_F^4)^{1/4}\right)^4\\
=&\frac{C}{b^4}(\sqrt{b}+\sqrt[4]{b})^4\mathbb{E}\|H_{\mu,1}-\mathbb{E}H_{\mu,1}\|_F^4\leq\frac{16C}{b^2}\mathbb{E}\|H_{\mu,1}-\mathbb{E}H_{\mu,1}\|_F^4\\
=&\frac{16C}{b^2}\mathbb{E}(\|H_{\mu,1}\|_F^2-2\langle H_{\mu,1},\mathbb{E}H_{\mu,1}\rangle+\|\mathbb{E}H_{\mu,1}\|_F^2)^2\\
\leq&\frac{16C}{b^2}\mathbb{E}(\|H_{\mu,1}\|_F^2+2\|H_{\mu,1}\|_F\|\mathbb{E}H_{\mu,1}\|_F+\|\mathbb{E}H_{\mu,1}\|_F^2)^2\\
\leq&\frac{16C}{b^2}\mathbb{E}(2\|H_{\mu,1}\|_F^2+2\|\mathbb{E}H_{\mu,1}\|_F^2)^2\leq\frac{16C}{b^2}\mathbb{E}(2\|H_{\mu,1}\|_F^2+2\mathbb{E}\|H_{\mu,1}\|_F^2)^2\\
\leq&\frac{64C}{b^2}(\mathbb{E}\|H_{\mu,1}\|_F^4+\mathbb{E}\|H_{\mu,1}\|_F^4)\leq\frac{128C}{b^2}(d+16)^8L_g^2,
\end{aligned}
\tag{2.25}
$$

where the first inequality is due to the Rosenthal inequality Rio [2009], C is an absolute constant, the fourth inequality is due to the fact $1\leq\sqrt[4]{b}\leq\sqrt{b}$. Plugging Eq. (2.20), Eq. (2.23) and Eq. (2.25) back to Eq. (2.24) gives the desired result (2.19). □

2.2. Zeroth-order Smooth (deterministic) Riemannian Optimization

For the sake of completeness, in this section, we focus on the case when the exact function evaluations of f are available and $h \equiv 0$. For this case, we propose `ZO-RGD`, the zeroth-order Riemannian gradient descent method and provide its complexity analysis. The algorithm is formally presented in Algorithm 1. The following theorem gives the iteration and oracle complexities of Algorithm 1 for obtaining an ϵ-stationary point of (2.1).

Algorithm 1: Zeroth-Order Riemannian Gradient Descent (`ZO-RGD`)

1: **Input:** Initial point $x_0 \in \mathcal{M}$, smoothing parameter μ, step size η_k, fixed number of iteration N.
2: **for** $k = 0, 1, 2, ...$ **do**
3: Sample a standard Gaussian random vector $u_k \in T_{x_k}\mathcal{M}$ by orthogonal projection in Definition 2.1.1.
4: Compute the zeroth-order gradient $g_\mu(x_k)$ by Eq. (2.2).
5: Update $x_{k+1} = \mathsf{Retr}_{x_k}(-\eta_k g_\mu(x_k))$.
6: **end for**

THEOREM 2.2.1. *Let f satisfy Assumption 2.1.1 and suppose $\{x_k\}$ is the sequence generated by Algorithm 1 with the stepsize $\eta_k = \hat{\eta} = \frac{1}{2(d+4)L_g}$. Then, we have*

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}_{\mathcal{U}_k}\|\mathsf{grad}f(x_k)\|^2 \leq \frac{4}{\hat{\eta}}\left(\frac{f(x_0)-f(x^*)}{N+1} + C(\mu)\right), \tag{2.26}$$

where $\mathcal{U}_k$ denotes the set of all Gaussian random vectors we drew for the first k iterations [2]*, and $C(\mu) = \frac{\mu^2 L_g}{16}\frac{(d+3)^3}{(d+4)} + \frac{\mu^2}{16}\frac{(d+6)^3}{(d+4)} + \frac{\mu^2 L_g}{16}\frac{(d+6)^3}{(d+4)^2}$. In order to have*

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}_{\mathcal{U}_k}\|\mathsf{grad}f(x_k)\|^2 \leq \epsilon^2, \tag{2.27}$$

we need the smoothing parameter μ and number of iteration N (which is also the number of calls to the zeroth-order oracle) to be set as $\mu = \mathcal{O}\left(\epsilon/d^{3/2}\right)$, $N = \mathcal{O}\left(d/\epsilon^2\right)$.

PROOF. Proof of Theorem 2.2.1 From Assumption 2.1.1 we have

$$f(x_{k+1}) \leq f(x_k) - \eta_k\langle g_\mu(x_k), \mathsf{grad}f(x_k)\rangle + \frac{\eta_k^2 L_g}{2}\|g_\mu(x_k)\|^2.$$

[2]The notation of taking the expectation w.r.t. a set, is to take the expectation for each of the elements in the set.

Taking the expectation w.r.t. u_k on both sides, we have

$$\begin{aligned}\mathbb{E}_{u_k}\left[f(x_{k+1})\right] \leq & f(x_k) - \eta_k \langle \mathbb{E}_{u_k}(g_\mu(x_k)), \mathsf{grad} f(x_k)\rangle + \frac{\eta_k^2 L_g}{2}\mathbb{E}_{u_k}(\|g_\mu(x_k)\|^2) \\ \leq & f(x_k) - \eta_k \langle \mathbb{E}_{u_k}(g_\mu(x_k)), \mathsf{grad} f(x_k)\rangle + \frac{\eta_k^2 L_g}{2}\left(\frac{\mu^2}{2}L_g^2(d+6)^3 + 2(d+4)\|\mathsf{grad} f(x_k)\|^2\right),\end{aligned}$$

where the last inequality is by Proposition 2.1.1. Now Take $\eta_k = \hat{\eta} = \frac{1}{2(d+4)L_g}$, we have

$$\begin{aligned}& \mathbb{E}_{u_k}\left[f(x_{k+1})\right] \\ \leq & f(x_k) + \frac{\hat{\eta}}{2}(\|\mathsf{grad} f(x_k)\|^2 - 2\langle \mathbb{E}_{u_k}(g_\mu(x_k)), \mathsf{grad} f(x_k)\rangle) + \frac{\mu^2 L_g}{16}\frac{(d+6)^3}{(d+4)^2} \\ = & f(x_k) + \frac{\hat{\eta}}{2}(\|\mathsf{grad} f(x_k) - \mathbb{E}_{u_k}(g_\mu(x_k))\|^2 - \|\mathbb{E}_{u_k}(g_\mu(x_k))\|^2) + \frac{\mu^2 L_g}{16}\frac{(d+6)^3}{(d+4)^2} \\ \leq & f(x_k) + \frac{\hat{\eta}}{2}\left(\frac{\mu^2 L_g^2}{4}(d+3)^3 - \frac{1}{2}\|\mathsf{grad} f(x_k)\|^2 + \frac{\mu^2}{4}L_g(d+6)^3\right) + \frac{\mu^2 L_g}{16}\frac{(d+6)^3}{(d+4)^2} \\ = & f(x_k) - \frac{\hat{\eta}}{4}\|\mathsf{grad} f(x_k)\|^2 + C(\mu),\end{aligned}$$

where the second inequality is from Proposition 2.1.1. Define $\phi_k := f(x_k) - f(x^*)$. Now take the expectation w.r.t. $\mathcal{U}_k = \{u_0, u_1, \ldots, u_{k-1}\}$, we have

$$\phi_{k+1} \leq \phi_k - \frac{\hat{\eta}}{4}\mathbb{E}_{\mathcal{U}_k}\|\mathsf{grad} f(x_k)\|^2 + C(\mu).$$

Summing the above inequality over $k = 0, \ldots, N$ yields (2.26).

Therefore with $\mu = \mathcal{O}(\epsilon/d^{3/2})$ we have $C(\mu) \leq \hat{\eta}\epsilon^2/4$. Taking $N \geq 8(d+4)L_g(f(x_0) - f(x^*))/\epsilon^2$ yields (2.27). In summary, the number of iterations for obtaining an ϵ-stationary solution is $\mathcal{O}(d/\epsilon^2)$, and hence the total zeroth-order oracle complexity is also $\mathcal{O}(d/\epsilon^2)$. □

REMARK 2.2.1. *Note that in Algorithm 1, we only sample one Gaussian vector in each iteration of the algorithm. In practice, one can also sample multiple Gaussian random vectors in each iteration and obtain an averaged gradient estimator. Suppose we sample m i.i.d. Gaussian random vectors in each iteration and use the average $\bar{g}_\mu(x) = \frac{1}{m}\sum_{i=1}^m g_{\mu,i}(x)$, then the bound for our zeroth-order estimator becomes*

$$\mathbb{E}(\|\bar{g}_\mu(x) - \mathsf{grad} f(x)\|^2) \leq \mu^2 L_g^2(d+6)^3 + \frac{2(d+4)}{m}\|\mathsf{grad} f(x)\|^2. \tag{2.28}$$

Hence, the final result in Theorem 2.2.1 can be improved to

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}_{\mathcal{U}_k}\|\mathsf{grad} f(x_k)\|^2 \leq 4L_g\frac{f(x_0)-f(x^*)}{N+1}+\mu^2L_g^2(d+6)^3, \tag{2.29}$$

with $\hat{\eta}=1/L_g$ *and* $C(\mu)=\mu^2L_g(d+6)^3/2$. *Therefore the number of iterations required is improved to* $N=\mathcal{O}(1/\epsilon^2)$ *when we set* $\mu=\mathcal{O}(\epsilon/d^{3/2})$ *and* $m=\mathcal{O}(d)$. *However, the zeroth-order oracle complexity is still* $\mathcal{O}(d/\epsilon^2)$. *The proof of* (2.28) *and* (2.29) *is given in Appendix C of our published version [Li et al., 2023]. This multi-sampling technique played a key role in our stochastic and non-smooth case analyses.*

2.3. Stochastic Zeroth-order Riemannian Optimization Algorithms

We now demonstrate the applicability of the developed Riemannian derivative estimation methodology in previous section, for various classes of stochastic zeroth-order Riemannian optimization algorithms.

2.3.1. Zeroth-Order Stochastic Riemannian Optimization for Nonconvex Problem.

Recall that our task is to solve (2.1). In this section, we focus on the following smooth problem

$$\min_{x\in\mathcal{M}} f(x) := \int_{\xi} F(x,\xi)dP(\xi), \tag{2.30}$$

where P is a random distribution, F is a function satisfying Assumption 2.1.1, in variable x, almost surely. Note that f automatically satisfies Assumption 2.1.1 by the Jensen's inequality. In the stochastic case, our zeroth-order Riemannian gradient estimator is given by

$$\bar{g}_{\mu,\xi}(x)=\frac{1}{m}\sum_{i=1}^{m}g_{\mu,\xi_i}(x),\ \text{where}\ g_{\mu,\xi_i}(x)=\frac{F(\mathsf{Retr}_x(\mu u_i),\xi_i)-F(x,\xi_i)}{\mu}u_i, \tag{2.31}$$

and u_i is a standard normal random vector on $\mathrm{T}_x\mathcal{M}$. We also immediately have that

$$\mathbb{E}_{\xi_i}g_{\mu,\xi_i}(x)=\frac{f(\mathsf{Retr}_x(\mu u))-f(x)}{\mu}u=g_\mu(x). \tag{2.32}$$

The mini-batch approach above enables us to obtain the following bound on $\mathbb{E}\|\bar{g}_{\mu,\xi}(x)-\mathsf{grad} f(x)\|^2$, the proof of which is given in the appendix of our published version Li et al. [2023].

LEMMA 2.3.1. *For the Riemannian gradient estimator in* (2.31)*, under Assumptions 2.1.1 and 2.1.3, we have*

$$\mathbb{E}\|\bar{g}_{\mu,\xi}(x) - \mathsf{grad} f(x)\|^2 \leq \mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 + \frac{8(d+4)}{m}\|\mathsf{grad} f(x)\|^2, \tag{2.33}$$

where the expectation $\mathbb{E}$ *is taken for both Gaussian vectors* $\mathcal{U} = \{u_1, ..., u_m\}$ *and* ξ*.*

Our zeroth-order Riemannian stochastic gradient descent algorithm (`ZO-RSGD`) for solving (2.30), is presented in Algorithm 2.

Algorithm 2: Zeroth-order Riemannian Stochastic Gradient Descent (`ZO-RSGD`)

1: **Input:** Initial point $x_0 \in \mathcal{M}$, smoothing parameter μ, multi-sample constant m, step size η_k.
2: **for** $k = 0, 1, 2, ...$ **do**
3: Sample the standard Gaussian random vectors u_i^k on $T_{x_k}\mathcal{M}$ by orthogonal projection in Definition 2.1.1, and sample ξ_i^k, $i = 1, ..., m$.
4: Compute the zeroth-order gradient $\bar{g}_{\mu,\xi}(x_k)$ by Eq. (2.31).
5: Update $x_{k+1} = \mathsf{Retr}_{x_k}(-\eta_k \bar{g}_{\mu,\xi}(x_k))$.
6: **end for**

Now we present convergence analysis for obtaining an ϵ-stationary point of (2.30).

THEOREM 2.3.1. *Let* F *satisfy Assumption 2.1.1, w.r.t. variable* x *almost surely. Suppose* $\{x_k\}$ *is the sequence generated by Algorithm 2 with the stepsize* $\eta_k = \hat{\eta} = \frac{1}{L_g}$*. Under Assumption 2.1.3, we have*

$$\frac{1}{N+1}\sum_{k=0}^{N} \mathbb{E}_{\mathcal{U}_k, \Xi_k}\|\mathsf{grad} f(x_k)\|^2 \leq 4L_g \frac{f(x_0) - f(x^*)}{N+1} + C(\mu), \tag{2.34}$$

where $C(\mu) = 2\mu^2 L_g^2 (d+6)^3 + \frac{16(d+4)}{m}\sigma^2$*,* $\mathcal{U}_k$ *denotes the set of all Gaussian random vectors and* Ξ_k *denotes the set of all random variable* ξ_k *in the first* k *iterations. In order to have* $\frac{1}{N+1}\sum_{k=0}^{N} \mathbb{E}_{\mathcal{U}_k, \Xi_k}\|\mathsf{grad} f(x_k)\|^2 \leq \epsilon^2$*, we need the smoothing parameter* μ*, number of sampling* m *in each iteration and number of iterations* N *to be*

$$\mu = \mathcal{O}\left(\epsilon/d^{3/2}\right),\ m = \mathcal{O}\left(d\sigma^2/\epsilon^2\right),\ N = \mathcal{O}\left(1/\epsilon^2\right). \tag{2.35}$$

Hence, the number of calls to the zeroth-order oracle is $mN = \mathcal{O}(d/\epsilon^4)$*.*

PROOF. Proof of Theorem 2.3.1 From Assumption 2.1.1, we have:

$$f(x_{k+1}) \leq f(x_k) - \eta_k \langle \bar{g}_{\mu,\xi}(x_k), \mathsf{grad} f(x_k) \rangle + \frac{\eta_k^2 L_g}{2} \|\bar{g}_{\mu,\xi}(x_k)\|^2$$

Take $\eta_k = \hat{\eta} = \frac{1}{L_g}$, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \eta_k \langle \bar{g}_{\mu,\xi}(x_k), \mathsf{grad} f(x_k) \rangle + \frac{\eta_k^2 L_g}{2} \|\bar{g}_{\mu,\xi}(x_k)\|^2 \\ &= f(x_k) + \frac{1}{2L_g} \left(\|\bar{g}_{\mu,\xi}(x_k) - \mathsf{grad} f(x_k)\|^2 - \|\mathsf{grad} f(x_k)\|^2 \right). \end{aligned}$$

Take the expectation for the random variables at iteration k on both sides, we have

$$\begin{aligned} \mathbb{E}_k f(x_{k+1}) &\leq f(x_k) + \frac{1}{2L_g} \left(\mathbb{E}_k \|\bar{g}_{\mu,\xi}(x_k) - \mathsf{grad} f(x_k)\|^2 - \|\mathsf{grad} f(x_k)\|^2 \right) \\ &\overset{Eq.\ (2.33)}{\leq} f(x_k) + \frac{1}{2L_g} \left(\mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m} \sigma^2 + \left(\frac{8(d+4)}{m} - 1 \right) \|\mathsf{grad} f(x_k)\|^2 \right). \end{aligned}$$

Summing up over $k = 0, ..., N$ (assuming that $m \geq 16(d+4)$) yields (2.34).

In summary, the total number of iterations for obtaining an ϵ-stationary solution of (2.30) is $\mathcal{O}(1/\epsilon^2)$, and the stochastic zeroth-order oracle complexity is $\mathcal{O}(d/\epsilon^4)$. □

REMARK 2.3.1. *For the Euclidean case, in the unconstrained setting, it was shown in Ghadimi and Lan [2013] that if we assume prior knowledge on the total number of iterations N, one could prove a similar oracle complexity (with the Euclidean dimension) even with $m = 1$. However for the Riemannian case, selecting m as in* (2.35) *seems to be required for our current theoretical analysis. In particular, properties of (Euclidean) Gaussian smoothed function $f_\mu(x) = \frac{1}{\kappa} \int_x f(x + \mu u) e^{-\frac{1}{2}\|u\|^2} du$, such as $\nabla f_\mu(x) = \mathbb{E}(g_\mu(x))$ used in Ghadimi and Lan [2013], cannot be naturally extended to the Riemannian case. Furthermore, it is not very obvious how to calculate and interpret the gradient of the Riemannian counterpart $f_\mu(x) = \frac{1}{\kappa} \int_{x \in T_x \mathcal{M}} f(\mathsf{Retr}_x(\mu u)) e^{-\frac{1}{2}\|u\|^2} du$.*

2.3.2. Zeroth-order Stochastic Riemannian Proximal Gradient Method. We now consider the general optimization problem of the form in (2.1). For convenience, we define $p(x) := f(x) + h(x)$. We assume that $\mathcal{M}$ is a compact submanifold, h is convex in the embedded space $\mathbb{R}^n$ and is also Lipschitz continuous with parameter L_h, and f satisfies Assumption 2.1.3.

The non-differentiability of h prohibits Riemannian gradient methods to be applied directly. In Chen et al. [2020], by assuming that the exact gradient of f is available, a manifold proximal gradient method (ManPG) is proposed for solving (2.1). One typical iteration of ManPG is as follows:

$$\begin{aligned} v_k &:= \operatorname{argmin} \langle \mathsf{grad} f(x_k), v\rangle + \frac{1}{2t}\|v\|^2 + h(x_k+v),\ \text{s.t.},\ v \in T_{x_k}\mathcal{M} \\ x_{k+1} &:= \mathsf{Retr}_{x_k}(\eta_k v_k), \end{aligned} \tag{2.36}$$

where $t > 0$ and $\eta_k > 0$ are step sizes. In this section, we develop a zeroth-order counterpart of ManPG (`ZO-SManPG`), where we assume that only noisy function evaluations of f are available. The following lemma from Chen et al. [2020] provides a notion of stationary point that is useful for our analysis.

LEMMA 2.3.2. *Let $\bar{v}_k$ be the minimizer of the v-subproblem in* (2.36)*. If $\bar{v}_k = 0$, then x_k is a stationary point of problem* (2.1)*. We say x_k is an ϵ-stationary point of* (2.1) *with $t = \frac{1}{L_g}$, if $\|\bar{v}_k\| \le \epsilon/L_g$.*

Our `ZO-SManPG` iterates as:

$$\begin{aligned} v_k &:= \operatorname{argmin} \langle \bar{g}_{\mu,\xi}(x_k), v\rangle + \frac{1}{2t}\|v\|^2 + h(x_k+v),\ \text{s.t.},\ v \in T_{x_k}\mathcal{M}, \\ x_{k+1} &:= \mathsf{Retr}_{x_k}(\eta_k v_k), \end{aligned} \tag{2.37}$$

where $\bar{g}_{\mu,\xi}(x_k)$ is defined in Eq. (2.31). Note that the only difference between `ZO-SManPG` (2.37) and ManPG (2.36) is that in (2.37) we use $\bar{g}_{\mu,\xi}(x)$ to replace the Riemannian gradient $\mathsf{grad} f$ in (2.36). A more complete description of the algorithm is given in Algorithm 3. Now we provide some useful lemmas for analyzing the iteration complexity of Algorithm 3.

LEMMA 2.3.3. *(Non-expansiveness) Suppose $v := \arg\min_{v\in T_x\mathcal{M}} \langle g_1, v\rangle + \frac{1}{2t}\|v\|^2 + h(x+v)$ and $w := \arg\min_{w\in T_x\mathcal{M}} \langle g_2, w\rangle + \frac{1}{2t}\|w\|^2 + h(x+w)$. Then we have*

$$\|v - w\| \le t\|g_1 - g_2\|. \tag{2.38}$$

PROOF. Proof of Lemma 2.3.3 By the first order optimality condition Yang et al. [2014], we have $0 \in \frac{1}{t}v + g_1 + \mathsf{proj}_{T_x\mathcal{M}}\partial h(x+v)$ and $0 \in \frac{1}{t}w + g_2 + \mathsf{proj}_{T_x\mathcal{M}}\partial h(x+w)$, i.e. $\exists p_1 \in \partial h(x+v)$

Algorithm 3: Zeroth-Order Stochastic Riemannian Proximal Gradient Descent (ZO-SManPG)

1: **Input:** Initial point x_0 on $\mathcal{M}$, smoothing parameter μ, number of multi-sample m, step size η_k.
2: **for** $k = 0, 1, 2, ...$ **do**
3: Sample m standard Gaussian random vector u_i on $T_{x_k}\mathcal{M}$ by by orthogonal projection in Definition 2.1.1, $i = 1, ..., m$.
4: Compute the zeroth-order gradient the random oracle $\bar{g}_\mu(x_k)$ by Eq. (2.31).
5: Solve v_k from Eq. (2.37).
6: Update $x_{k+1} = \mathsf{Retr}_{x_k}(\eta_k v_k)$.
7: **end for**

and $p_2 \in \partial h(x + w)$ such that $v = -t(g_1 + \mathsf{proj}_{T_x\mathcal{M}}(p_1))$ and $w = -t(g_2 + \mathsf{proj}_{T_x\mathcal{M}}(p_2))$. Therefore we have

$$\begin{aligned} \langle v, w - v\rangle &= t\langle g_1 + \mathsf{proj}_{T_x\mathcal{M}}(p_1), v - w\rangle \\ \langle w, v - w\rangle &= t\langle g_2 + \mathsf{proj}_{T_x\mathcal{M}}(p_2), w - v\rangle. \end{aligned} \tag{2.39}$$

Now since $v, w \in T_x\mathcal{M}$, and using the convxity of h, we have

$$\langle \mathsf{proj}_{T_x\mathcal{M}}(p_1), v - w\rangle = \langle p_1, v - w\rangle = \langle p_1, (v + x) - (w + x)\rangle \geq h(v + x) - h(w + x). \tag{2.40}$$

Substituting Eq. (2.39) and into (2.40) yields,

$$\langle v, w - v\rangle \geq t\langle g_1, v - w\rangle + h(v + x) - h(w + x)$$

$$\langle w, v - w\rangle \geq t\langle g_2, w - v\rangle + h(w + x) - h(v + x).$$

Summing these two inequalities gives $\langle v - w, v - w\rangle \leq t\langle g_2 - g_1, v - w\rangle$, and Eq. (2.38) follows by applying the Cauchy-Schwarz inequality. □

Corollary 2.3.1. *Suppose v_k is given by* (2.37)*, and $\bar{v}_k$ is solution of the v-subproblem in Eq.* (2.36)*, then we have*

$$\mathbb{E}_{\mathcal{U}_k, \Xi_k}\|v_k - \bar{v}_k\|_F^2 \leq t^2\left(\mu^2 L_g^2(d + 6)^3 + \frac{8(d + 4)}{m}\sigma^2 + \frac{8(d + 4)}{m}\|\mathsf{grad} f(x_k)\|^2\right).$$

Proof. Proof of Corollary 2.3.1 By Lemma 2.3.3, we have

$$\mathbb{E}_{\mathcal{U}_k, \Xi_k}\|v_k - \bar{v}_k\|_F^2 \leq t^2\mathbb{E}_{\mathcal{U}_k, \Xi_k}\|\bar{g}_{\mu,\xi}(x_k) - \mathsf{grad} f(x_k)\|_F^2.$$

From Lemma 2.3.1,

$$\begin{aligned}&\mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\bar{g}_{\mu,\xi}(x_k)-\mathsf{grad}f(x_k)\|_F^2\\ \leq&\mu^2L_g^2(d+6)^3+\frac{8(d+4)}{m}\sigma^2+\frac{8(d+4)}{m}\|\mathsf{grad}f(x_k)\|^2.\end{aligned}$$

The desired result hence follows by combining these two inequalities. □

The following lemma shows the sufficient decrease property for one iteration of `ZO-SManPG`.

LEMMA 2.3.4. *For any $t>0$, there exists a constant $\bar{\eta}>0$ such that for any $0\leq\eta_k\leq\min\{1,\bar{\eta}\}$, the (x_k,v_k) generated by Algorithm 3 satisfies*

$$p(x_{k+1})-p(x_k)\leq-\left(\frac{\eta_k}{2t}-\tilde{C}\right)\|v_k\|^2, \tag{2.41}$$

where $\tilde{C}=\mu^2L_g^2(d+6)^3+\frac{8(d+4)}{m}\sigma^2+\frac{8(d+4)}{m}G^2$ and G is the upper bound of the Riemannian gradient $\mathsf{grad}f(x)$ (existence by the compactness of $\mathcal{M}$).

PROOF. Proof of Lemma 2.3.4 Notice that

$$\begin{aligned}&f(x_{k+1})-f(x_k)\leq\langle\mathsf{grad}f(x_k),\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\rangle+\frac{L_g}{2}\|\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\|^2\\ &=\langle\mathsf{grad}f(x_k)-\bar{g}_{\mu,\xi}(x),\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\rangle+\langle\bar{g}_{\mu,\xi}(x),\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\rangle+\frac{L_g}{2}\|\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\|^2,\end{aligned}$$

where the inequality follows from Assumption 2.1.1. Moreover, by Lemma 2.3.1 and the Fact 3.6 of Chen et al. [2020], we have

$$\begin{aligned}&\langle\mathsf{grad}f(x_k)-\bar{g}_{\mu,\xi}(x),\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\rangle\leq\|\mathsf{grad}f(x_k)-\bar{g}_{\mu,\xi}(x)\|\|\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\|\\ &\leq M_1^2\eta_k^2\left[\mu^2L_g^2(d+6)^3+\frac{8(d+4)}{m}\sigma^2+\frac{8(d+4)}{m}\|\mathsf{grad}f(x)\|^2\right]\|v_k\|^2.\end{aligned}$$

The rest of the proof of bounding $\langle\bar{g}_{\mu,\xi}(x),\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\rangle+\frac{L_g}{2}\|\mathsf{Retr}_{x_k}(\eta_kv_k)-x_k\|^2$ follows from exactly the same process as in (Chen et al. [2020], Lemma 5.2). We omit the details for brevity. □

THEOREM 2.3.2. *Under Assumption 2.1.3 and Assumption 2.1.1, the sequence generated by Algorithm 3, with $\eta_k=\hat{\eta}<\min\{1,\bar{\eta}\}$ and $t=1/L_g$, satisfies:*

$$\frac{1}{N}\sum_{k=0}^{N-1}\mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\bar{v}_k\|^2\leq\frac{4t(p(x_0)-p(x^*))}{(\hat{\eta}-8\tilde{C})tN}+\frac{\hat{\eta}Nt^2}{\hat{\eta}-8\tilde{C}t}\tilde{C}+\frac{8t^3}{\hat{\eta}-8\tilde{C}t}\tilde{C}^2, \tag{2.42}$$

where $\tilde{C} = \mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 + \frac{8(d+4)}{m}G^2$ *and* G *is the upper bound of the Riemannian gradient* $\mathsf{grad} f(x)$ *over the manifold* $\mathcal{M}$*. To guarantee*

$$\min_{k=0,\ldots,N-1} \mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\bar{v}_k\|_F^2 \leq \epsilon^2/L_g^2,$$

the parameters need to be set as: $\mu = \mathcal{O}\left(\epsilon/d^{3/2}\right)$, $m = \mathcal{O}\left(dG^2/\epsilon^2\right)$, $N = \mathcal{O}\left(1/\epsilon^2\right)$*. Hence, the number of calls to the stochastic zeroth-order oracle is* $\mathcal{O}(d/\epsilon^4)$*.*

Proof. Proof of Theorem 2.3.2 Summing up (2.41) over $k = 0, \ldots, N-1$ and using Corollary 2.3.1, we have:

$$\begin{aligned}
&p(x_0) - \mathbb{E}_{\mathcal{U}_k,\Xi_k} p(x_k) \geq \sum_{k=0}^{N-1} [\frac{\eta_k}{2t} - \tilde{C}]\mathbb{E}_{\mathcal{U}_k}\|v_k\|_F^2 \geq [\frac{\hat{\eta}}{4t} - 2\tilde{C}] \sum_{k=0}^{N-1} 2\mathbb{E}_{\mathcal{U}_k,\Xi_k}\|v_k\|_F^2 \\
&\geq [\frac{\hat{\eta}}{4t} - 2\tilde{C}] \sum_{k=0}^{N-1} \left[\mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\bar{v}_k\|_F^2 - t^2 \left(\mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 \right.\right. \\
&\left.\left. + \frac{8(d+4)}{m}\|\mathsf{grad} f(x_k)\|^2\right)\right] \\
&\geq [\frac{\hat{\eta}}{4t} - 2\tilde{C}] \sum_{k=0}^{N-1} \mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\bar{v}_k\|_F^2 - \frac{\hat{\eta} N t}{4}\left(\mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 + \frac{8(d+4)}{m}G^2\right) \\
&+ 2t^2 \left(\mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 + \frac{8(d+4)}{m}G^2\right)^2,
\end{aligned}$$

which immediately implies the desired result (2.42). □

Remark 2.3.2. *The subproblem Eq.* (2.37) *is the main computational effort in Algorithm 3. Fortunately, this subproblem can be efficiently solved by a regularized semi-smooth Newton's method when* $\mathcal{M}$ *takes certain forms. We refer the reader to Chen et al. [2020], Xiao et al. [2018] for more details.*

2.3.3. Escaping saddle points: Zeroth-order stochastic cubic regularized Newton's method over Riemannian manifolds. In this section, we consider the problem of escaping saddle-points and converging to local minimizers in a stochastic zeroth-order Riemannian setting. Towards that, we leverage the Hessian estimator methodology developed in Section 2.1.4 and analyze a zeroth-order Riemannian stochastic cubic regularized Newton's method (`ZO-RSCRN`)

Algorithm 4: Zeroth-Order Riemannian Stochastic Cubic Regularized Newton's Method (`ZO-RSCRN`)

1: **Input:** Initial point x_0 on $\mathcal{M}$, smoothing parameter μ, multi-sample parameter m and b, cubic regularization parameter α.
2: **for** $k = 0, 1, 2, ...$ **do**
3: Compute $\bar{g}_{\mu,\xi}(x_k)$ and $\bar{H}_{\mu,\xi}(x_k)$ based on (2.31) and (2.17) respectively.
4: Solve $\eta_k = \operatorname{argmin}_\eta \hat{m}_{x_k,\alpha}(\eta)$, where $\hat{m}_{x,\alpha}(\eta)$ is defined in (2.43).
5: Update $x_{k+1} = \mathsf{Retr}_{x_k}(P_x(\eta_k))$.
6: **end for**

for solving (2.30), which provably escapes the saddle points. Our approach is motivated by Zhang and Zhang [2018], where the authors proposed the minimization of function $m_{x,\sigma}(\eta) = f(x) + \langle \mathsf{grad} f(x), \eta\rangle + \frac{1}{2}\langle P_x \circ \mathrm{Hess} f(x) \circ P_x[\eta], \eta\rangle + \frac{\alpha_k}{6}\|\eta\|^3$ at each iteration. The zeroth-order counterpart replaces the Riemannian gradient and Hessian with the corresponding zeroth-order estimators. The proposed `ZO-RSCRN` algorithm is described in Algorithm 4. In `ZO-RSCRN`, the function in the cubic regularized subproblem is

$$\hat{m}_{x,\alpha}(\eta) = f(x) + \langle \bar{g}_{\mu,\xi}(x), \eta\rangle + \frac{1}{2}\langle \bar{H}_{\mu,\xi}(x)[\eta], \eta\rangle + \frac{\alpha}{6}\|\eta\|^3. \tag{2.43}$$

Note that if $\hat{\eta} = \operatorname{argmin}_\eta \hat{m}_{x,\alpha}(\eta)$, then the projection $P_x(\hat{\eta})$ is also a minimizer, because $\bar{g}_{\mu,\xi}(x)$ and $\bar{H}_{\mu,\xi}(x)$ only take effect on the component that is in $\mathrm{T}_x\mathcal{M}$.

THEOREM 2.3.3. *For manifold $\mathcal{M}$ and function $f : \mathcal{M} \to \mathbb{R}$ under Assumptions 2.1.1, 2.1.2 and 2.1.3, define $k_{\min} := \operatorname{argmin}_k \mathbb{E}_{\mathcal{U}_k,\Xi_k}\|\eta_k\|$, then the update in Algorithm 4 with $\alpha \geq L_H$ satisfies:*

$$\mathbb{E}\|g_{k_{\min}+1}\| \leq \mathcal{O}(\epsilon), \text{ and } \mathbb{E}[\lambda_{\min}(\mathrm{Hess} f_{k_{\min}+1})] \geq -\mathcal{O}(\sqrt{\epsilon}), \tag{2.44}$$

given that the parameters satisfy:

$$N = \mathcal{O}\left(1/\epsilon^{3/2}\right),\ \mu = \mathcal{O}\left(\min\left\{\frac{\epsilon}{d^{3/2}}, \sqrt{\frac{\epsilon}{d^5}}\right\}\right),\ m = \mathcal{O}(d/\epsilon^2),\ b = \mathcal{O}(d^4/\epsilon), \tag{2.45}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue. Hence, the zeroth-order oracle complexity is $\mathcal{O}(d/\epsilon^{7/2} + d^4/\epsilon^{5/2})$.

PROOF. Proof of Theorem 2.3.3 Denote $f_k = f(x_k)$, $g_k = \mathsf{grad} f(x_k)$ and $\mathbb{E} = \mathbb{E}_{\mathcal{U}_k,\Xi_k}$ for ease of notation. We first provide the global optimality conditions of subproblem Eq. (2.43) following

Nesterov and Polyak [2006]:

$$(\bar{H}_{\mu,\xi}(x)+\lambda^* I)\eta+\bar{g}_{\mu,\xi}(x)=0,\ \lambda^*=\frac{\alpha}{2}\|\eta\|,\ \bar{H}_{\mu,\xi}(x)+\lambda^* I\succeq 0. \tag{2.46}$$

Since the parallel transport P_η is an isometry, we have

$$\begin{aligned}
&\|g_{k+1}\| = \|P_{\eta_k}^{-1} g_{k+1}\| \\
=&\|(P_{\eta_k}^{-1} g_{k+1} - g_k - \mathrm{Hess} f_k[\eta_k]) + (g_k - \bar{g}_{\mu,\xi}(x_k)) \\
&+ (\mathrm{Hess} f_k[\eta_k] - \bar{H}_{\mu,\xi}(x_k)[\eta_k]) + (\bar{g}_{\mu,\xi}(x_k) + \bar{H}_{\mu,\xi}(x_k)[\eta_k])\| \\
\le&\|P_{\eta_k}^{-1} g_{k+1} - g_k - \mathrm{Hess} f_k[\eta_k]\| + \|g_k - \bar{g}_{\mu,\xi}(x_k)\| \\
&+ \|\mathrm{Hess} f_k[\eta_k] - \bar{H}_{\mu,\xi}(x_k)[\eta_k]\| + \|\bar{g}_{\mu,\xi}(x_k) + \bar{H}_{\mu,\xi}(x_k)[\eta_k]\| \\
\overset{Eq.\ (2.9)}{\le}& \frac{L_H}{2}\|\eta_k\|^2 + \|g_k - \bar{g}_{\mu,\xi}(x_k)\| \\
&+ \|\mathrm{Hess} f_k[\eta_k] - \bar{H}_{\mu,\xi}(x_k)[\eta_k]\| + \|\bar{g}_{\mu,\xi}(x_k) + \bar{H}_{\mu,\xi}(x_k)[\eta_k]\| \\
\overset{Eq.\ (2.46)}{=}& \frac{L_H}{2}\|\eta_k\|^2 + \|g_k - \bar{g}_{\mu,\xi}(x_k)\| + \|\mathrm{Hess} f_k[\eta_k] - \bar{H}_{\mu,\xi}(x_k)[\eta_k]\| + \lambda^*\|\eta_k\| \\
\overset{Eq.\ (2.46)}{\le}& \frac{L_H}{2}\|\eta_k\|^2 + \|g_k - \bar{g}_{\mu,\xi}(x_k)\| + \|\mathrm{Hess} f_k[\eta_k] - \bar{H}_{\mu,\xi}(x_k)\|_{\mathrm{op}}\|\eta_k\| + \frac{\alpha}{2}\|\eta_k\|^2 \\
\le&\frac{L_H}{2}\|\eta_k\|^2 + \|g_k - \bar{g}_{\mu,\xi}(x_k)\| + \frac{1}{2}\|\mathrm{Hess} f_k - \bar{H}_{\mu,\xi}(x_k)\|_{\mathrm{op}}^2 + \frac{1}{2}\|\eta_k\|^2 + \frac{\alpha}{2}\|\eta_k\|^2.
\end{aligned}$$

Taking expectation on both sides of the above inequality gives (by Eq. (2.33) and Eq. (2.18))

$$\mathbb{E}\|g_{k+1}\| - \sqrt{\delta_g} - \delta_H \le \frac{1}{2}(L_H + \alpha + 1 + 2L_2\|g_k\|)\mathbb{E}\|\eta_k\|^2, \tag{2.47}$$

where $\delta_g = \mu^2 L_g^2 (d+6)^3 + \frac{8(d+4)}{m}(G^2+\sigma^2)$, G is the upper bound of $\|\mathsf{grad} f\|$ over $\mathcal{M}$, and $\delta_H = \frac{(d+16)^4}{b} L_g + \frac{\mu^2 L_H^2}{18}(d+6)^5$.

Since $P_{\eta_k}^{-1}$ is an isometry, we have:

$$\begin{aligned}
&\lambda_{\min}(\mathrm{Hess} f_{k+1}) = \lambda_{\min}(P_{\eta_k}^{-1} \circ \mathrm{Hess} f_{k+1} \circ P_{\eta_k}) \\
\geq &\lambda_{\min}(P_{\eta_k}^{-1} \circ \mathrm{Hess} f_{k+1} \circ P_{\eta_k} - \mathrm{Hess} f_k) \\
&+ \lambda_{\min}(\mathrm{Hess} f_k - \bar{H}_{\mu,\xi}(x_k)) + \lambda_{\min}(\bar{H}_{\mu,\xi}(x_k)) \\
\overset{Eq.\ (2.8)}{\geq} &- L_H \|\eta_k\| + \lambda_{\min}(\mathrm{Hess} f_k - \bar{H}_{\mu,\xi}(x_k)) + \lambda_{\min}(\bar{H}_{\mu,\xi}(x_k)) \\
= &\lambda_{\min}(\mathrm{Hess} f_k - \bar{H}_{\mu,\xi}(x_k)) + \lambda_{\min}(\bar{H}_{\mu,\xi}(x_k) - L_H \|\eta_k\| I) \\
\overset{Eq.\ (2.46)}{\geq} &\lambda_{\min}(\mathrm{Hess} f_k - \bar{H}_{\mu,\xi}(x_k)) - \frac{\alpha + 2L_H}{2} \|\eta_k\|.
\end{aligned}$$

Taking expectation, we obtain (by Eq. (2.18))

$$\frac{\alpha + 2L_H}{2} \mathbb{E}\|\eta_k\| \geq -(\sqrt{\delta_H} + \mathbb{E}\lambda_{\min}(\mathrm{Hess} f_{k+1})). \tag{2.48}$$

Now we will upper bound $\mathbb{E}\|\eta_k\|$. From Assumption 2.1.2, we have

$$\begin{aligned}
\hat{f}_{x_k}(\eta_k) &\leq f(x_k) + g_k^\top \eta_k + \frac{1}{2} \eta_k^\top H_k \eta_k + \frac{L_H}{6} \|\eta_k\|^3 \\
&= \left(f(x_k) + \bar{g}_\mu(x_k)^\top \eta_k + \frac{1}{2} \eta_k^\top \bar{H}_\mu(x_k) \eta_k + \frac{L_H}{6} \|\eta_k\|^3 \right) \\
&\quad + \left((g_k - \bar{g}_\mu(x_k))^\top \eta_k + \frac{1}{2} \eta_k^\top (H_k - \bar{H}_\mu(x_k)) \eta_k \right).
\end{aligned} \tag{2.49}$$

Using Eq. (2.46) we have

$$\begin{aligned}
&f(x_k) + \bar{g}_\mu(x_k)^\top \eta_k + \frac{1}{2} \eta_k^\top \bar{H}_\mu(x_k) \eta_k + \frac{L_H}{6} \|\eta_k\|^3 \\
= &f(x_k) - \frac{1}{2} \eta_k^\top \bar{H}_\mu(x_k) \eta_k + (\frac{L_H}{6} - \frac{\alpha}{2}) \|\eta_k\|^3 \\
= &f(x_k) - \frac{1}{2} \eta_k^\top (\bar{H}_\mu(x_k) + \frac{\alpha}{2} \|\eta_k\| I) \eta_k - (\frac{\alpha}{4} - \frac{L_H}{6}) \|\eta_k\|^3 \\
\leq &f(x_k) - (\frac{\alpha}{4} - \frac{L_H}{6}) \|\eta_k\|^3 \leq f(x_k) - \frac{\alpha}{12} \|\eta_k\|^3,
\end{aligned} \tag{2.50}$$

where the last inequality is due to $\alpha \geq L_H$. Moreover, by Cauchy-Schwarz inequality and Young's inequality, we have

$$\begin{aligned}
&\mathbb{E}\left[(g_k - \bar{g}_\mu(x_k))^\top \eta_k + \frac{1}{2}\eta_k^\top (H_k - \bar{H}_\mu(x_k))\eta_k\right] \\
\leq &\mathbb{E}\|g_k - \bar{g}_\mu(x_k)\|\|\eta_k\| + \frac{1}{2}\mathbb{E}\|H_k - \bar{H}_\mu(x_k)\|_{\mathrm{op}}\|\eta_k\|^2 \\
\leq &\frac{32}{3\alpha}\mathbb{E}\|g_k - \bar{g}_\mu(x_k)\|^{3/2} + \frac{12}{\alpha}\mathbb{E}\|H_k - \bar{H}_\mu(x_k)\|_{\mathrm{op}}^3 + \frac{\alpha}{24}\mathbb{E}\|\eta_k\|^3.
\end{aligned} \tag{2.51}$$

Plugging (2.50) and (2.51) to Eq. (2.49), we have

$$\mathbb{E}f_{k+1} \leq f_k - \frac{\alpha}{24}\mathbb{E}\|\eta_k\|^3 + \frac{32}{3L_H}\delta_g^{3/4} + \frac{12}{L_H}\tilde{\delta}_H, \tag{2.52}$$

where $\tilde{\delta}_H = \tilde{C}\frac{(d+16)^6}{b^{3/2}}L_g^{1.5} + \frac{1}{27}\mu^3 L_H^3(d+6)^{7.5}$. Taking the sum for (2.52) over $k = 0, \ldots, N-1$, we have

$$\frac{1}{N}\sum_{k=0}^{N}\mathbb{E}\|\eta_k\|^3 \leq \frac{24}{L_H}\left(\frac{f_0 - f^*}{N} + \frac{32}{3L_H}\delta_g^{3/4} + \frac{12}{L_H}\tilde{\delta}_H\right),$$

which together with (2.45) yields

$$\mathbb{E}\|\eta_{k_{\min}}\|^3 \leq \mathcal{O}(\epsilon^{3/2}), \text{ and } \mathbb{E}\|\eta_{k_{\min}}\|^2 \leq \mathcal{O}(\epsilon). \tag{2.53}$$

Combining Eq. (2.53), Eq. (2.47) and Eq. (2.48) yields (2.44). □

REMARK 2.3.3. *To solve the subproblem, we implement the same Krylov subspace method as in Agarwal et al. [2021], where the Riemannian Hessian and vector multiplication is approximated by Lanczos iterations. Note also that in our setting, we only require vector-vector multiplications due to the structure of our Hessian estimator in Eq.* (2.12)*. For the purpose of brevity, we refer to Agarwal et al. [2021], Carmon and Duchi [2018] for a comprehensive study of this method.*

2.4. Numerical Experiments and Applications

We now explore the performance of the proposed algorithms on various simulation experiments. Finally, we demonstrate the applicability of stochastic zeroth-order Riemannian optimization for the problems of zeroth-order attacks on deep neural networks and controlling stiffness matrix in

Dimension	ϵ	Stepsize	No. iter. ZO-RGD	Aver. No. iter. RGD
15×5	10^{-3}	10^{-2}	460 ± 137	442
25×15	10^{-3}	10^{-2}	892 ± 99	852
50×20	10^{-2}	5×10^{-3}	255 ± 26	236

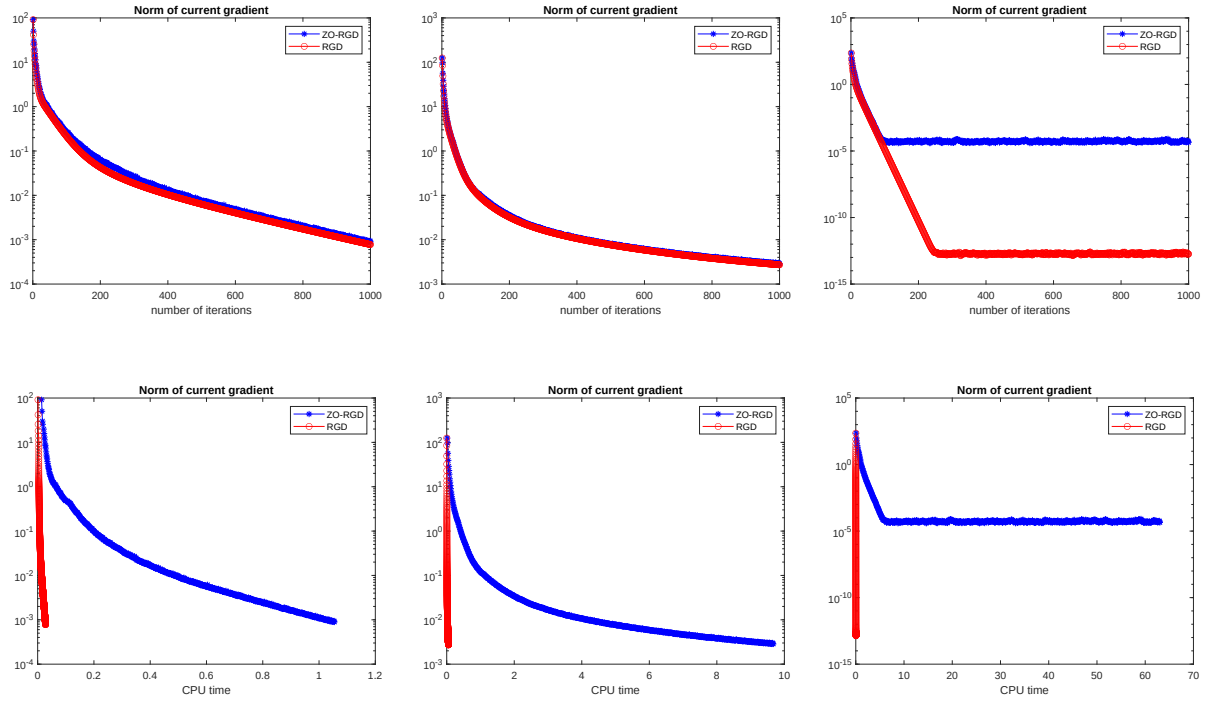
Table 2.2. Comparison of ZO-RGD and RGD on the Procrustes problem. Notice here we take a larger ϵ when the dimension is large. The variance introduced by the Gaussian random vector in the zeroth-order algorithms prevents us from taking smaller values in practice. The last column of Fig. 2.1 shows a similar phenomenon.

robotics. We conducted our experiments on a desktop with Intel Core 9600K CPU and NVIDIA GeForce RTX 2070 GPU.

2.4.1. Simulation Experiments. For all the simulation experiments listed below, we plot the average result over 100 runs. We set $\mu = 10^{-8}$ if not otherwise specified. In theory, the smaller μ is, the better the results are. However in practice we have to compromise the machine precision, and 10^{-8} comes up as an appropriate choice. We set the stepsize $\eta_k = 1/L_g$ as specified in the theory. Moreover we choose the retraction which is based on the QR decomposition for Stiefel manifold and the exponential mapping on positive definite manifold as defined in Chapter 11 of Boumal [2023].

Experiment 1: Procrustes problem [Absil et al., 2008]. This is a matrix linear regression problem on a given manifold: $\min_{X \in \mathcal{M}} \|AX - B\|_F^2$, where $X \in \mathbb{R}^{n \times p}$, $A \in \mathbb{R}^{l \times n}$ and $B \in \mathbb{R}^{l \times p}$. The manifold we use is the Stiefel manifold $\mathcal{M} = \mathrm{St}(n, p)$. In our experiment, we pick up different dimension n and p and record the time cost to achieve prescribed precision ϵ. The entries of matrix A are generated by standard Gaussian distribution. We use the retraction based on Polar decomposition. We compare our ZO-RGD (Algorithm 1) with the first-order Riemannian gradient method (RGD) on this problem. The results are shown in Table 2.2. For each run, we sample $m = np - \frac{1}{2}p(p+1)$ Gaussian samples for each iteration. The multi-sample version of ZO-RGD closely resembles the convergence rate of RGD, as shown in Fig. 2.1. These results indicate our zeroth-order method ZO-RGD is comparable with its first-order counterpart RGD, though the former one only uses zeroth-order information.

Remark 2.4.1. *Fig. 2.2 shows the effect of choosing m that are smaller than the values suggested by our theory. We note that a small value of m, for example, $m = 15$ is already good enough for*

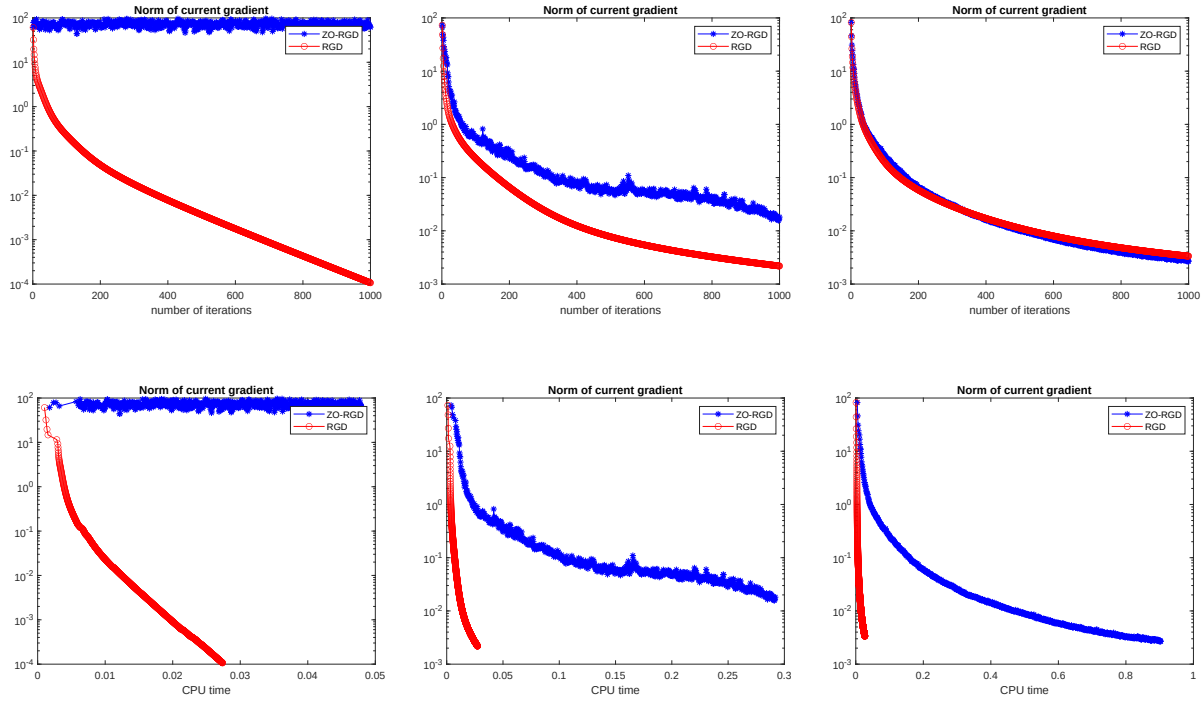


(a) $(n, p) = (15, 5)$ (b) $(n, p) = (25, 15)$ (c) $(n, p) = (50, 20)$

FIGURE 2.1. The convergence curve of ZO-RGD v.s. RGD. Above: The x-axis corresponds to the number of iterations and y-axis is the norm of Riemannian gradient at corresponding points; Below: The x-axis corresponds to the CPU time (in seconds). Here for the `ZO-RGD` we take the mini-batch m as suggested by our theory. However, in Figure 2.2 we notice that setting m sufficiently large suffices for obtaining the same accuracy. Correspondingly for this case, the CPU time is also reduced, although we do not plot it explicitly.

problem sizes $(n, p) = (15, 5)$. *Proving this observation theoretically seems to be non-trivial and we plan to examine this in a future work.*

Experiment 2: k-PCA [Tripuraneni et al., 2018, Zhang et al., 2016b, Zhou et al., 2019]. k-PCA on Grassmann manifold is a Rayleigh quotient minimization problem. Given a normalized data matrix $A \in \mathbb{R}^{d\times n}$ and $H = A^\top A$, we want to solve $\min_{X\in \mathrm{Gr}(n,p)} -\frac{1}{2}\mathrm{Tr}(X^\top H X)$. The Grassmann manifold $\mathrm{Gr}(n, p)$ is the set of p-dimensional subspaces in $\mathbb{R}^n$. We refer the reader to Absil et al. [2008] for more details about the Grassmann quotient manifold. This problem can be written as a finite sum problem: $\min_{X\in \mathrm{Gr}(n,p)} \sum_{i=1}^n -\frac{1}{2}\mathrm{Tr}(X^\top a_i^\top a_i X)$, where $a_i \in \mathbb{R}^n$ is the i-th row of A. We compare our `ZO-RSGD` algorithm (Algorithm 2) and its first-order counterpart RSGD



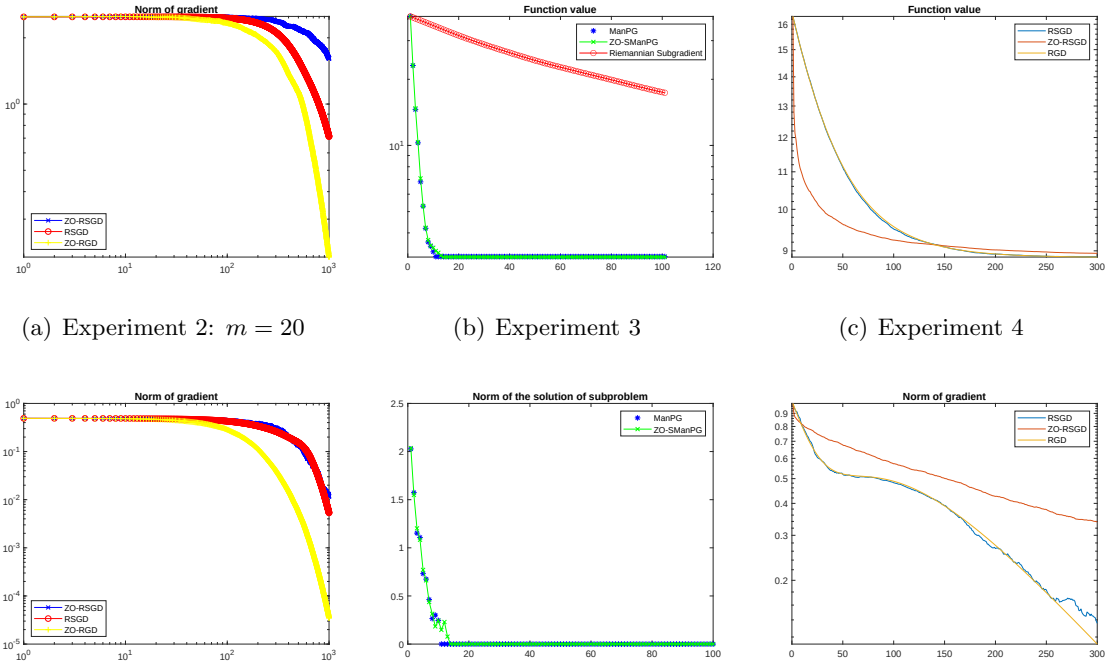
(a) $m = 1$ (b) $m = 15$ (c) $m = 50$

FIGURE 2.2. The convergence curve of ZO-RGD v.s. RGD with different choice of m. Here the dimension is fixed to be $(n, p) = (15, 5)$. The CPU time is in seconds.

on this problem. The results are shown in Fig. 2.3 (a) and (d). In our experiment, we set $n = 100$, $p = 50$, and the matrix $A \in \mathbb{R}^{n\times p}$ is a normalized randomly Gaussian data matrix. The mini-batch sample number m is taken as 40, a number which achieves reasonable results in experiment within a short time (taking $m = np - \frac{1}{2}p(p+1)$ is too time consuming in this case). We use the retraction based on the QR decomposition. From Fig. 2.3 (a) and (d), we see that the performance of `ZO-RSGD` is similar to its first-order counterpart RSGD.

Experiment 3: Sparse PCA [Jolliffe et al., 2003, Zou and Xue, 2018, Zou et al., 2006]. The sparse PCA problem, arising in statistics, is a Riemannian optimization problem over the Stiefel manifold with nonsmooth objective:

$$\min_{X\in \mathrm{St}(n,p)} -\frac{1}{2}\mathrm{Tr}(X^\top A^\top AX) + \lambda\|X\|_1,$$



(a) Experiment 2: $m = 20$ (b) Experiment 3 (c) Experiment 4

(d) Experiment 2: $m = 40$ (e) Experiment 3 (f) Experiment 4

FIGURE 2.3. The convergence of three numerical experiments. The x-axis always denotes the number of iterations. Figures (a) and (d) are results for k-PCA (Experiment 2). Here three algorithms are compared: `ZO-RSGD` (Algorithm 2), RSGD, and `ZO-RGD` (Algorithm 1). Figures (b) and (e) are results for sparse PCA (Experiment 3) in which the y-axis of Figure (e) denotes the norm of v_k in (2.36) (for ManPG) and (2.37) (for `ZO-SManPG`), which actually measures the optimality of the problem. Here three algorithms are compared: `ZO-SManPG` (Algorithm 3), ManPG and Riemannian subgradient method. Figures (c) and (f) are results for Karcher mean of PSD matrices problem (Experiment 4). Here three algorithms are compared: RSGD, `ZO-RSGD` (Algorithm 2), and RGD.

similar to a LASSO version of k-PCA problem. Here, $A \in \mathbb{R}^{d\times n}$ is again the normalized data matrix. We compare our `ZO-SManPG` (Algorithm 3) with ManPG Chen et al. [2020] and Riemannian subgradient method Li et al. [2021]. In our numerical experiments, we chose $(d, n, p) = (50, 100, 10)$, and entries of A are drawn from Gaussian distribution and rows of A are then normalized. Again the mini-batch m is taken as $np - \frac{1}{2}p(p+1)$, the dimension of the manifold. We use the retraction based on the Polar decomposition. The comparison results are shown in Fig. 2.3 (b) and (e). These results show that our `ZO-SManPG` is comparable to its first-order counterpart ManPG and they are both much better than the Riemannian subgradient method.

Experiment 4: Karcher mean of given PSD matrices [Bini and Iannazzo, 2013, Kasai et al., 2018, Zhang and Sra, 2016]. Given a set of positive semidefinite (PSD) matrices $\{A_i\}_{i=1}^n$ where $A_i \in \mathbb{R}^{d\times d}$ and $A_i \succeq 0$, we want to calculate their Karcher mean: $\min_{X\in\mathcal{S}_{++}^d} \frac{1}{2n}\sum_{i=1}^n (\operatorname{dist}(X, A_i))^2$, where $\operatorname{dist}(X,Y) = \|\operatorname{logm}(X^{-1/2}YX^{-1/2})\|_F$ (logm stands for matrix logarithm) represents the distance along the corresponding geodesic between the two points $X, Y \in \mathcal{S}_{++}^d$. This experiment serves as an example of optimizing geodesically convex functions over Hadamard manifolds, with `ZO-RSGD` (Algorithm 2). In our numerical experiment, we take $d = 3$, $n = 500$ and mini-batch number $m = 20$. We use the retraction that is based a second-order Taylor's expansion of the exponential mapping. We compare our `ZO-RSGD` algorithm with its first-order counterpart RSGD and RGD. The results are shown in Fig. 2.3 (c) and (f), and from these results we see that ZO-RSGD is comparable to its first-order counterpart RSGD in terms of function value, though it is inferior to RSGD and RGD in terms of the size of the gradient.

Experiment 5: Procrustes problem with `ZO-RSCRN`. Here, we consider the Procrustes problem in Experiment 1 and use the `ZO-RSCRN` with both estimated gradients and Hessians. Following Agarwal et al. [2021], we use the gradient norm as a performance measure (although the algorithm converges to local-minimzers). We use the Lanczos method (specifically Algorithm 2 from Agarwal et al. [2021]) for solving the sub-problem in Step 4. Furthermore, as we are estimating the second order information, we set $n = 12$ and $p = 8$ and consider $\epsilon = 10^{-3}$. We use the retraction based on the Polar decomposition. In Figure 2.4, (a), we plot the gradient norm versus iterations for `RSCRN` in the zeroth order and second-order setting. We notice that the zeroth-order method compares favourably to the second-order counterpart in terms of iteration complexity. Scaling up `ZO-RSCRN` to even higher-dimension seems more challenging. One plausible approach is to apply variance reduction techniques – we leave it as an interesting and important problem that we plan to tackle in a future work.

2.4.2. Real world applications.

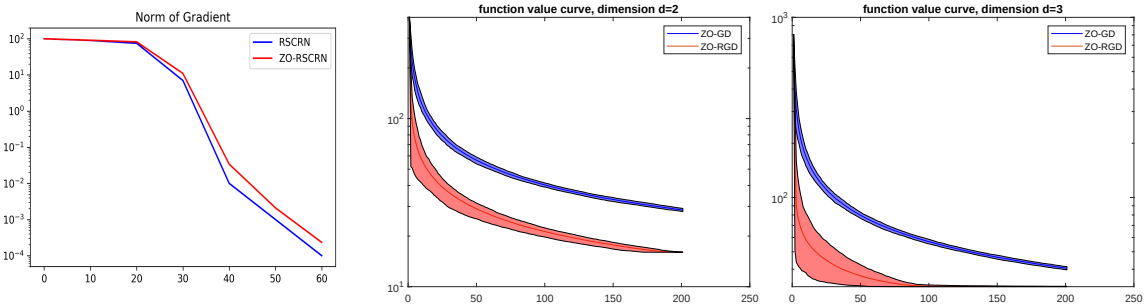
2.4.2.1. *Black-box stiffness control for robotics.* We now study the first motivating example discussed in Section 2.1.1.1 on the control of robotics with the policy parameter being the stiffness matrix $\boldsymbol{K}^{\mathcal{P}} \in \mathcal{S}_{++}^d$, see Jaquier et al. [2020] for more engineering details. Mathematically, given

the current position of robot $\hat{\boldsymbol{p}}$ and current speed $\dot{\boldsymbol{p}}$, the task is to minimize

$$f(\boldsymbol{K}^{\mathcal{P}}) = w_{\boldsymbol{p}}\|\hat{\boldsymbol{p}} - \boldsymbol{p}\|^2 + w_d \det(\boldsymbol{K}^{\mathcal{P}}) + w_c \operatorname{cond}(\boldsymbol{K}^{\mathcal{P}}) \tag{2.54}$$

with $\boldsymbol{p}$ being the new position, and cond is the condition number. With a constant external force $\boldsymbol{f}^e$ applied to the system, we have the following identity which solves $\boldsymbol{p}$ by $\boldsymbol{K}^{\mathcal{P}}$: $\boldsymbol{f}^e = \boldsymbol{K}^{\mathcal{P}}(\hat{\boldsymbol{p}}-\boldsymbol{p})-\boldsymbol{K}^{\mathcal{D}}\dot{\boldsymbol{p}}$, where the damping matrix $\boldsymbol{K}^{\mathcal{D}} = \boldsymbol{K}^{\mathcal{P}}$ for critical damped case. As the stiffness matrix is a positive definite matrix, the above optimization problem is a Riemannian optimization problem over the positive definite manifold (where the manifold structure is the same as the Karcher mean problem). The function f is not known analytically and following Jaquier et al. [2020], we use a simulated setting for a robot (7-DOF Franka Emika Panda robot) to evaluate the function f for a given value of $\boldsymbol{K}^{\mathcal{P}}$, with the same parameters as in Jaquier et al. [2020]. We compare our `ZO-RGD` method with Euclidean Zeroth-order gradient descent (ZO-GD) method Balasubramanian and Ghadimi [2021]. We use the retraction that is based a second-order Taylor's expansion of the exponential mapping. We test the cases when $d = 2$ and $d = 3$ for minimizing function f w.r.t $\boldsymbol{K}^{\mathcal{P}}$, and the results are shown in Figure 2.4, (b) and (c). In our experiments, the stepsize of ZO-GD is 3×10^{-4} and `ZO-RGD` is 10^{-3}. Note that for ZO-GD method, one has to project the matrix back to the positive definite set, whereas the `ZO-RGD` method intrinsically guarantees that the iterates are positive definite, thus is much more stable. Also, due to the fact that `ZO-RGD` is more stable, the stepsize of `ZO-RGD` can be larger than ZO-GD, which results in faster convergence.

2.4.2.2. *Zeroth-order black-box attack on Deep Neural Networks (DNNs).* We now return to the motivating example described in Section 2.1.1.2 and propose our black-box attack algorithm, as stated in the Appendix E of our published version [Li et al., 2023]. For the sake of comparison, we also assume the architecture of the DNN is known and use "white-box" attacks based on first-order Riemannian optimization methods and compare against the PGD attack Madry et al. [2017], which is a white-box attack and does not explicitly enforce any constraints on the perturbed testing data. For simplicity, we assume the manifold is a sphere. That is, we assume that the perturbation set S is given by $S(r) = \{\delta : \|\delta\|_2 = r\}$, where r is the radius of the sphere. The main motivation of the sphere constraint is to guarantee that the perturbed image is always within a certain distance from the original image, which is consistent with the optimal ℓ_2-norm attack studied in the literature Lyu

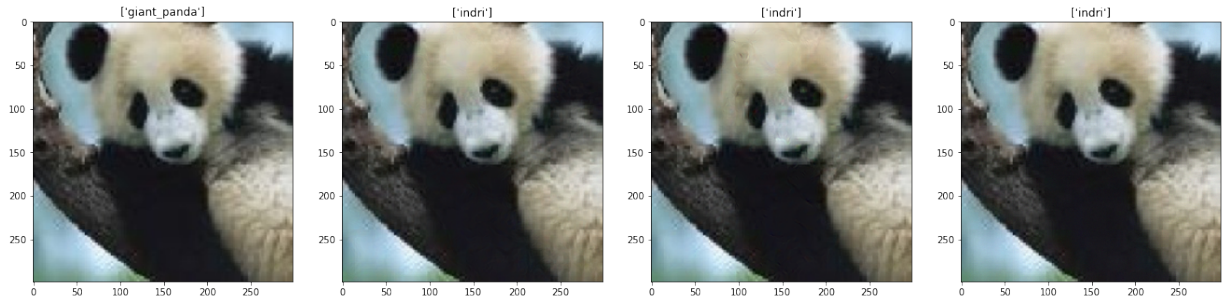


(a) RSCRN (b) 2d case (c) 3d case

FIGURE 2.4. Figure (a) corresponds to Experiment 5. Figures (b) and (c) correspond to the experiments on the robotic minimization function in (2.54). The x-axis in all figures correspond to iteration number.

et al. [2015]. We start our zeroth-order attack from a perturbation and maximize the loss function on the sphere. For the black-box method, to accelerate the convergence, we use Euclidean zeroth-order optimization to find an appropriate initial perturbation (the Appendix E of our published version [Li et al., 2023]). It is worth noting that the zeroth-order attack in Chen et al. [2017], Tu et al. [2019] has a non-smooth objective function, which has $\mathcal{O}(n^3/\epsilon^3)$ complexity to guarantee convergence Nesterov and Spokoiny [2017], whereas the complexity needed for our method is $\mathcal{O}(d/\epsilon^2)$.

We first tested our method on the giant panda picture in the Imagenet data set Deng et al. [2009], with the network structure the Inception v3 network structure Szegedy et al. [2016]. The attack radius in our algorithm is taken to be 0.05 times the ℓ_2 norm of the original image. We use the orthogonal projection onto the sphere as the retraction. Both white-box and black-box Riemannian attacks are successful, which means that they both create test images which the DNN misclassifies, see Figure 2.5. We also tested our first and zeroth-order manifold attack algorithms and the Euclidean black-box attack Chen et al. [2017] on the CIFAR10 dataset, and the network structure we used is the VGG net Simonyan and Zisserman [2014]. From the experiments, we note that the Riemannian black-box attack has a larger successful rate compared to the Euclidean black-box attack Chen et al. [2017] when the radius of the attack is relatively small. The detailed results are provided in Appendix E of our published version [Li et al., 2023].



(a) Original image (b) PGD attack (c) First-order attack on the sphere (d) Zeroth-order attack on the sphere

FIGURE 2.5. The attack on giant panda picture Deng et al. [2009]. (a): the original image; (b): the PGD attack with a small diameter; (c) Riemannian attack (see our published version [Li et al., 2023]) on the sphere with the same diameter; (d): Riemannian zeroth-order attack (see our published version [Li et al., 2023]). 'Indri' refers to the class to which the original image is misclassified to.

2.5. Conclusions

In this chapter, we proposed zeroth-order algorithms for solving Riemannian optimization over submanifolds embedded in Euclidean space in which only noisy function evaluations are available for the objective. These algorithms adopt new estimators of the Riemannian gradient and Hessian from noisy objective function evaluations, based on a Riemannian version of the Gaussian smoothing technique. The proposed estimators overcome the difficulty of the non-linearity of the manifold constraint and the issues that arise in using Euclidean Gaussian smoothing techniques when the function is defined only over the manifold. The iteration complexity and oracle complexity of the proposed algorithms are analyzed for obtaining an appropriately defined ϵ-stationary point or ϵ-approximate local minimum. The established complexities are independent of the dimension of the ambient Euclidean space and only depend on the intrinsic dimension of the manifold. Numerical experiments demonstrated that the proposed zeroth-order algorithms are comparable to their first-order counterparts.

CHAPTER 3

Zeroth-order Stochastic Averaging Algorithms for Riemannian Optimization

3.1. Introduction

Consider again zeroth-order algorithms for solving the following Riemannian optimization problem (1.2), which we restated here:

$$\min_{x\in\mathcal{M}} f(x) := \mathbb{E}_{\xi}[F(x,\xi)], \tag{3.1}$$

where $\mathcal{M}$ is a d-dimensional complete manifold, $f : \mathcal{M} \to \mathbb{R}$ is a smooth function, and we can access only the noisy function evaluations $F(x,\xi)$. A natural zeroth-order algorithm is to estimate the gradients of f and use them in the context of Riemannian stochastic gradient descent. The main difficulty in doing so is the construction of the zeroth-order gradient estimation. Assuming that we have independent samples u_i that are standard normal random vectors supported on $\mathrm{T}_x\mathcal{M}$, the tangent space at $x \in \mathcal{M}$, our previous chapter proposed to construct the zeroth-order gradient estimator as

$$G_{\mu}^{\mathsf{Exp}}(x) = \frac{1}{m}\sum_{i=1}^{m} \frac{F(\mathsf{Exp}_x(\mu u_i),\xi_i) - F(x,\xi_i)}{\mu} u_i \tag{3.2}$$

where $\mu > 0$ is a smoothing parameter. Note here that if a retraction is available, then one could also replace the exponential mapping with a retraction based estimator,

$$G_{\mu}^{\mathsf{Retr}}(x) = \frac{1}{m}\sum_{i=1}^{m} \frac{F(\mathsf{Retr}_x(\mu u_i),\xi_i) - F(x,\xi_i)}{\mu} u_i. \tag{3.3}$$

The merit of having a Gaussian distribution on the tangent space is that the variance of the constructed estimator $G_{\mu}(x)$ will only depend on the intrinsic dimension d of the manifold, and is

Result	Objective	Manifold	Operations	m	N
`Zo-RSGD` Algorithm 2 in Chapter 2	Smooth, 2MB	general	Retr	$\mathcal{O}(d/\epsilon^2)$	$\Omega(1)$
`Zo-RASA`, Alg 5, Thm 3.2.1	Smooth, 2MB	General	Exp map, PT	$\mathcal{O}(d)$	$\Omega(1)$
				$\mathcal{O}(1)$	$\Omega(d)$
`Zo-RASA`, Alg 6, Thm 3.3.2	Smooth, 4MB	Compact, 2nd FF bound	SO-Retr, VT	$\mathcal{O}(d)$	$\Omega(1)$
				$\mathcal{O}(1)$	$\Omega(1)$

Table 3.1. **Conditions required to establish a sample complexity of $\mathcal{O}(d/\epsilon^4)$ for various algorithms for convergence to stationarity in the sense of Definition 3.2.1**. For instance, to obtain the $\mathcal{O}(d/\epsilon^4)$ sample complexity for Alg 5, we need to require $m = \mathcal{O}(d)$ and $N = \Omega(1)$, or $m = \mathcal{O}(1)$ and $N = \Omega(d)$. Here, 2MB and 4MB stand for bounded second central moment (i.e., variance) (Assumption 3.2.2) and fourth central moment (Assumption 3.3.3) respectively. 2nd FF stands for second fundamental form (Theorem 3.3.1). SO-RETR stands for second-order retraction (Assumption 3.3.4). PT and VT stand for parallel and vector transport respectively (see, Definition 1.2.5). The parameter d is the intrinsic dimension of the manifold $\mathcal{M}$, m is the batch-size, N is the total number of iterations required, and ϵ is the desired precision. Oracle complexity refers to the number of calls to the stochastic zeroth-order oracle. We also remark here that although our previous chapter uses retraction, its convergence analysis also assumes retraction-based smoothness. For Zo-RASA, we need the initial batch-size $m_0 = \mathcal{O}(d)$.

independent of the dimension n of the ambient Euclidean space. See also Wang [2023], Wang et al. [2021] for additional follow-up works.

According to the previous chapter, to obtain an ϵ-approximate stationary solution of (3.1) (as in Definition 3.2.1) using the above approach, we established a sample complexity of $\mathcal{O}(d/\epsilon^4)$, with $\mathcal{O}(1/\epsilon^2)$ iteration complexity and $m = \mathcal{O}(d/\epsilon^2)$ per-iteration batch size. Even considering $d = 1$ for simplicity, this suggests for example that to get an accuracy of $\epsilon \approx 10^{-3}$, one needs batch-sizes of order $m \approx 10^6$ resulting in a highly impractical per-iteration complexity. Intriguingly, when implementing these algorithms in practice, favorable results are obtained even when the batch-size is simply set between ten and fifty. Thus, there exists a discrepancy between the current theory and practice of stochastic zeroth-order Riemannian optimization. Furthermore, in online Riemannian optimization problems [Maass et al., 2022, Wang et al., 2023] where the data sequence is observed in a streaming fashion, waiting for very long time-periods in each iteration in order to obtain the required order of batch-sizes is highly undesirable.

The main motivation of the current work stems from the above-mentioned undesirable issues associated with the use of mini-batches in stochastic Riemannian optimization algorithms by our

Chapter 2. We address the problem by getting rid of the use of mini-batches altogether, and by developing batch-free, fully-online algorithm, Zeroth-order Riemannian Averaging Stochastic Approximation (`Zo-RASA`) algorithm, for solving (3.1). We show that to obtain the sample complexity of $\mathcal{O}(d/\epsilon^4)$, `Zo-RASA` only requires $m = 1$ (see the remark after Theorem 3.2.1), which is a significant improvement compared to Li et al. [2023]. The first version of `Zo-RASA` in Algorithm 5 uses exponential mapping and parallel-transports. However, this version is not implementation-friendly. As a case-in-point, consider the Stiefel manifold (see (1.7)) for which we highlight that there is no closed-form expression for the parallel transport $P_{x^k}^{x^{k+1}}$. Indeed, they are only available as solutions to certain ordinary differential equation, which increases the per-iteration complexity of implementing Algorithm 5. To overcome this issue and to develop a practical version of the RASA framework, we replace the exponential mapping and parallel transport by retraction and vector transport respectively, resulting in the practical version of `Zo-RASA` method in Algorithm 6. As we discussed in Section 1.2, in the case of Stiefel manifolds, retractions cost only 1/4 the time of an exponential mapping. Also, while there is no closed-form for parallel transport on Stiefel manifolds, vector transport has an easy closed-form implementation. We establish that Algorithm 6 has the same sample complexity as Algorithm 5, with significantly improved per-iteration complexity. We now highlight two specific novelties that we introduce in this work to establish the above result.

- **Moving-average gradient estimators and Lifting-based Riemannian-Lyapunov analysis.** We introduce a Riemannian moving-average technique (see, Line 4 in Algorithm 5 and Algorithm 6) and a corresponding novel Riemannian-Lyapunov technique for analyzing zeroth-order stochastic Riemannian optimization problems, which works in the lifted space by tracking both the optimization trajectory and the gradient along the trajectory (see (3.7)). For Euclidean problems, these techniques were introduced and extended in Balasubramanian et al. [2022], Ghadimi et al. [2020], Ruszczyński [1987], Ruszczynski [2021], Ruszczynski and Syski [1983]. However, those works rely heavily on the Euclidean structure. Non-trivial adaptions are needed to extend such methodology and analyses to the Riemannian settings; see Theorem 3.2.1 and Theorem 3.3.2.
- **Approximation error between parallel and vector transports.** A major challenge in analyzing Algorithm 6 is to handle the additional errors introduced by the use of retractions and

Algorithm 5: Zo-RASA

Input: Initial point $x^0 \in \mathcal{M}$, $g^0 = G_\mu^{\mathsf{Exp}}(x^0)$, total number of iterations N, parameters $\beta > 0$, $\tau_0 = 1$, $\tau_k = 1/\sqrt{N}$ or $\tau_k = 1/\sqrt{dN}$ when $k \geq 1$, and stepsize $t_k = \tau_k/\beta$.
for $k = 0, 1, 2, \ldots, N-1$ **do**
  $x^{k+1} \leftarrow \mathsf{Exp}_{x^k}(-t_k g^k)$
  $g^{k+1} \leftarrow (1-\tau_k) P_{x^k}^{x^{k+1}} g^k + \tau_k P_{x^k}^{x^{k+1}} G_\mu^k$ where $G_\mu^k = G_\mu^{\mathsf{Exp}}(x^k)$ is given by (3.2) with batch-size $m = m_k$
end for

vector transports. We identify a novel geometric condition on the manifolds under consideration (see Assumption 3.3.1) under which we provide novel error bounds between parallel and vector transports (see Theorem 3.3.1). We further show that the proposed condition, which plays a crucial role in our subsequent convergence analysis, is naturally satisfied if the *second fundamental form* of the manifold is bounded. We remark that the obtained error bounds, between parallel and vector transport, are of independent interest and are potentially applicable to a variety of other Riemannian optimization problems.

In Table 3.1, we summarize the sample complexities of stochastic zeroth-order Riemannian optimization algorithms.

3.2. Zeroth-order RASA for smooth manifold optimization

We now introduce the Zeroth-order Riemannian Average Stochastic Approximation (`Zo-RASA`) algorithm for solving (3.1). The formal procedure is stated in Algorithm 5, where P_x^y is the parallel transport from $\mathrm{T}_x\,\mathcal{M}$ to $\mathrm{T}_y\,\mathcal{M}$ along the minimum geodesic connecting x and y. To establish the sample complexity of Algorithm 5, we extend the analysis of Ghadimi et al. [2020], which is in-turn motivated by the lifting-technique introduced in Ruszczyński [1987], Ruszczynski and Syski [1983], to the Riemannian setting. As such works heavily rely on the Euclidean structure, our proofs involve a non-trivial adaption of such techniques.

We first recalled from Chapter 1 that we have the following notion of stationarity:

Definition 3.2.1 (ϵ-approximate first-order stationary solution for (3.1)). *We call a point $\bar{x}$ an ϵ-approximate first-order stationary solution for* (3.1) *if it satisfies $\mathbb{E}[\|\mathsf{grad} f(\bar{x})\|_{\bar{x}}^2] \leq \epsilon^2$, where the expectation is with respect to both the problem and algorithm-based randomness.*

In our convergence analysis, we always choose $\tau_0 = 1$, and we consider two choices of τ_k when $k \geq 1$:

$$\tau_k = 1/\sqrt{N} \text{ or } \tau_k = 1/\sqrt{dN}, k \geq 1, \tag{3.4}$$

which corresponds to large or single batch, respectively. Moreover, we always choose $t_k = \tau_k/\beta$, where β is a positive constant determined by the smoothness constant in Assumption 3.2.1 (see Theorem 3.2.1), so that the step-size and the averaging weights are in the same order. Furthermore, we define

$$\Gamma_0 = \Gamma_1 = 1, \text{ and } \Gamma_k = \Gamma_1 \prod_{i=1}^{k-1}(1 - \tau_i^2). \tag{3.5}$$

This leads to the following inequalities which will be used frequently in our convergence analysis:

$$\sum_{i=k+1}^{N} \tau_i \Gamma_i \leq \Gamma_{k+1} \text{ and } \sum_{i=k+1}^{N} \tau_i^2 \Gamma_i \leq \tau_k \Gamma_{k+1}. \tag{3.6}$$

To proceed, we construct the following potential function

$$W(x, g) \coloneqq (f(x) - f^*) - \eta(x, g), \quad \text{where} \quad \eta(x, g) \coloneqq -\frac{1}{2\beta}\|g\|_x^2, \ g \in \mathrm{T}_x \mathcal{M}, \tag{3.7}$$

where $f^* = \min_{x \in \mathcal{M}} f(x)$ and $\beta > 0$ is a constant to be determined later. Note that the potential function in (3.7) has the component of both function value and the norm of the (estimated) gradients, also that W is always non-negative. In our analysis, we proceed by bounding the difference of potential function between successive iterates. More specifically, using the convexity of the norm, for any pair (x, g), we have $\|\mathsf{grad} f(x)\|_x^2 \leq -2\beta\,\eta(x, g) + 2\|g - \mathsf{grad} f(x)\|_x^2$. This observation will be leveraged in the proof of Theorem 3.2.1 to obtain the sample complexity of Algorithm 5 for obtaining an ϵ-approximate stationary solution.

We also highlight that our convergence analysis extensively utilizes the isometry property of parallel transport, stated in (1.8), i.e., $\langle P_x^y(\eta), P_x^y(\xi)\rangle_y = \langle \eta, \xi \rangle_x$. This result is a generalization of the isometry in the Euclidean spaces, since the inner product in Euclidean spaces is unchanged if one moves the beginning point of the vectors together. A direct result of this identity is that the length of the vectors is unchanged, namely $\|P_x^y(\xi)\|_y = \|\xi\|_x$, which we will also use extensively.

We now introduce the assumptions needed for our analysis.

ASSUMPTION 3.2.1. *The function $f : \mathcal{M} \to \mathbb{R}$ is L-smooth on $\mathcal{M}$, i.e., $\forall x, y \in \mathcal{M}$, we have $\|P_x^y \mathsf{grad} f(x) - \mathsf{grad} f(y)\|_y \leq L\, \mathsf{d}(x, y)$. An immediate consequence (see, for example, Boumal [2023, Proposition 10.53]) of this condition is that we have $|f(y) - f(x) - \langle \mathsf{grad} f(x), \mathsf{Exp}_x^{-1}(y)\rangle_x| \leq \frac{L}{2}\|\mathsf{Exp}_x^{-1}(y)\|_x^2$.*

Assumption 3.2.1 is a generalization of the standard gradient-Lipschitz assumption in Euclidean optimization [Lan, 2020, Nesterov, 2018] to the Riemannian setting, and is made in several works [Boumal, 2023]. To generalize it to the Riemannian setting, due to the fact that $\mathsf{grad} f(x)$ and $\mathsf{grad} f(y)$ are not in the same tangent space, we need to utilize parallel transports P_x^y to match the two vectors in the same tangent space.

Throughout the paper, we define $\mathcal{F}_k$ as the σ-algebra generated by all the randomness till iteration k of the algorithms. Namely, for Algorithm 5, we have $\mathcal{F}_k = \sigma(\xi_0, \ldots, \xi_k, x_0, \ldots, x_k, g_0, \ldots, g_k)$.

ASSUMPTION 3.2.2. *Along the trajectory of the algorithm, the stochastic gradients are unbiased and have bounded-variance, i.e., for $k \in \{1, \ldots, N\}$, we have $\mathbb{E}_\xi[\mathsf{grad} F(x^k; \xi_k)|\mathcal{F}_{k-1}] = \mathsf{grad} f(x^k)$ and $\mathbb{E}_\xi[\|\mathsf{grad} F(x^k; \xi_k) - \mathsf{grad} f(x^k)\|_{x^k}^2 |\mathcal{F}_{k-1}] \leq \sigma^2$.*

The above assumption is widely used in stochastic Riemannian optimization literature; see, for example, Boumal [2023], Li et al. [2023], Zhang et al. [2016a], and generalizes the standard assumption used in Euclidean stochastic optimization [Lan, 2020, Nesterov, 2018].

Now we proceed to the convergence analysis of Algorithm 5. We first state the following standard result characterizing the approximation error of G_μ^{Exp} (given by (3.2)) to the true Riemannian gradient.

LEMMA 3.2.1 (Proposition 1 in Li et al. [2023] with exponential mapping). *Under Assumptions 3.2.1, 3.2.2 we have $\|\mathbb{E} G_\mu^{\mathsf{Exp}}(x) - \mathsf{grad} f(x)\|_x^2 \leq \frac{\mu^2 L^2}{4}(d+3)^3$, $\mathbb{E}\|G_\mu^{\mathsf{Exp}}(x)\|_x^2 \leq \mu^2 L^2 (d+6)^3 + 2(d+4)\|\mathsf{grad} f(x)\|_x^2$ and $\mathbb{E}\|G_\mu^{\mathsf{Exp}}(x) - \mathsf{grad} f(x)\|_x^2 \leq \mu^2 L^2 (d+6)^3 + \frac{8(d+4)}{m}\sigma^2 + \frac{8(d+4)}{m}\|\mathsf{grad} f(x)\|_x^2$, where the expectation is taken toward all the Gaussian vectors in G_μ and the random variable ξ.*

Based on the above result, we have the following Lemma 3.2.2 which bounds the difference of g^k to the true Riemannian gradient $\mathsf{grad} f(x^k)$, and Lemma 3.2.3 bounds the difference of two

consecutive g^k, where we use parallel transport to make g^k and g^{k+1} in the same tangent space, i.e., $\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2$.

LEMMA 3.2.2. *Suppose the Assumptions 3.2.1 and 3.2.2 hold, and $\{x^k, g^k\}$ is generated by Algorithm 5. We have*

$$\begin{aligned}(3.8)\qquad &\mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \\ \leq &\Gamma_k \tilde{\sigma}_0^2 + \Gamma_k \sum_{i=1}^{k} \Big(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i} \frac{L^2 \|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i} \tilde{\sigma}_{i-1}^2 + \tau_k \hat{\sigma}^2 \Big),\end{aligned}$$

where the expectation $\mathbb{E}$ is taken with respect to all random variables up to iteration k, including the random variables $\{u_i\}_{i=1}^k$ used to construct the zeroth-order estimator as in (3.2)*. Here the notations are defined as:*

$$\begin{aligned}(3.9)\qquad &\hat{\sigma}^2 \coloneqq \frac{\mu^2 L^2}{4}(d+3)^3 \\ &\tilde{\sigma}_k^2 \coloneqq \sigma_k^2 + \frac{8(d+4)}{m_k} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \text{ where } \sigma_k^2 \coloneqq \mu^2 L^2 (d+6)^3 + \frac{8(d+4)}{m_k}\sigma^2.\end{aligned}$$

Moreover, from (3.6) *we have*

$$\sum_{k=1}^{N} \tau_k \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \sum_{k=0}^{N-1} \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \tilde{\sigma}_0^2,$$

$$\sum_{k=1}^{N} \tau_k^2 \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \sum_{k=0}^{N-1} \left((1+\tau_k)\tau_k^2 \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^3 \tilde{\sigma}_k^2 + \tau_k^2 \hat{\sigma}^2 \right) + \sum_{k=1}^{N} \tau_k^2 \tilde{\sigma}_0^2.$$

PROOF. Firstly, note that we have the following: $g^k - \mathsf{grad} f(x^k) = (1-\tau_{k-1}) P_{x^{k-1}}^{x^k} g^{k-1} + \tau_{k-1} P_{x^{k-1}}^{x^k} G_\mu^{k-1} - \mathsf{grad} f(x^k) = (1-\tau_{k-1}) P_{x^{k-1}}^{x^k} (g^{k-1} - \mathsf{grad} f(x^{k-1})) + (P_{x^{k-1}}^{x^k} \mathsf{grad} f(x^{k-1}) - \mathsf{grad} f(x^k)) + \tau_{k-1} P_{x^{k-1}}^{x^k} (G_\mu^{k-1} - \mathsf{grad} f(x^{k-1})) = (1-\tau_{k-1}) P_{x^{k-1}}^{x^k} (g^{k-1} - \mathsf{grad} f(x^{k-1})) + \tau_{k-1} e_{k-1} + \tau_{k-1} \Delta_{k-1}^f$. Hence, we have

$$\begin{aligned}(3.10)\qquad &\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \\ \leq &(1-\tau_{k-1}) \|g^{k-1} - \mathsf{grad} f(x^{k-1})\|_{x^{k-1}}^2 + \tau_{k-1} \|e_{k-1}\|_{x^k}^2 + \tau_{k-1}^2 \|\Delta_{k-1}^f\|_{x^k}^2 \\ &+ 2\tau_{k-1} \langle (1-\tau_{k-1}) P_{x^{k-1}}^{x^k} (g^{k-1} - \mathsf{grad} f(x^{k-1})) + \tau_{k-1} e_{k-1}, \Delta_{k-1}^f \rangle_{x^k},\end{aligned}$$

where the notation is defined as $e_{k-1} \coloneqq \frac{1}{\tau_{k-1}}(P_{x^{k-1}}^{x^k}\mathsf{grad}f(x^{k-1}) - \mathsf{grad}f(x^k))$, and $\Delta_{k-1}^f \coloneqq P_{x^{k-1}}^{x^k}(G_\mu^{k-1} - \mathsf{grad}f(x^{k-1}))$. Denote $\delta_{k-1} = \langle(1-\tau_{k-1})P_{x^{k-1}}^{x^k}(g^{k-1} - \mathsf{grad}f(x^{k-1})) + \tau_{k-1}e_{k-1}, \Delta_{k-1}^f\rangle_{x^k}$. The main novelty in the proof of this lemma is that δ is no longer an unbiased estimator (which is true for the first-order situation). We have by Lemma 3.2.1 that

$$
\begin{aligned}
&2\mathbb{E}_{u^k}[\delta_{k-1}] = 2\langle(1-\tau_{k-1})P_{x^{k-1}}^{x^k}(g^{k-1} - \mathsf{grad}f(x^{k-1})) + \tau_{k-1}e_{k-1}, \mathbb{E}_{u^k}[\Delta_{k-1}^f|\mathcal{F}_{k-2}]\rangle_{x^k} \\
\leq &\|(1-\tau_{k-1})P_{x^{k-1}}^{x^k}(g^{k-1} - \mathsf{grad}f(x^{k-1})) + \tau_{k-1}e_{k-1}\|_{x^k}^2 + \|\mathbb{E}_{u^k}G_\mu^{k-1} - \mathsf{grad}f(x^{k-1})\|_{x^{k-1}}^2 \\
\leq &(1-\tau_{k-1})\|g^{k-1} - \mathsf{grad}f(x^{k-1})\|_{x^{k-1}}^2 + \tau_{k-1}\|e_{k-1}\|_{x^k}^2 + \hat{\sigma}^2.
\end{aligned}
$$

Notice that in the above computation, the expectation is only taken with respect to the Gaussian random variables that we used to construct $G_\mu(x^{k-1})$. Plugging this back to (3.10), we have $\mathbb{E}_{u^k}\|g^k - \mathsf{grad}f(x^k)\|_{x^k}^2 \leq \tau_{k-1}\hat{\sigma}^2 + (1-\tau_{k-1}^2)\|g^{k-1} - \mathsf{grad}f(x^{k-1})\|_{x^{k-1}}^2 + \tau_{k-1}(1+\tau_{k-1})\|e_{k-1}\|_{x^k}^2 + \tau_{k-1}^2\|\Delta_{k-1}^f\|_{x^k}^2$. Now dividing both sides of this inequality by our new definition of Γ_k, we get $\frac{1}{\Gamma_k}\mathbb{E}_{u^k}\|g^k - \mathsf{grad}f(x^k)\|_{x^k}^2 \leq \frac{1}{\Gamma_{k-1}}\|g^{k-1} - \mathsf{grad}f(x^{k-1})\|_{x^{k-1}}^2 + \frac{(1+\tau_{k-1})\tau_{k-1}}{\Gamma_k}\|e_{k-1}\|_{x^k}^2 + \frac{\tau_{k-1}^2}{\Gamma_k}\|\Delta_{k-1}^f\|_{x^k}^2 + \frac{\tau_{k-1}}{\Gamma_k}\hat{\sigma}^2$.

By Assumptions 3.2.1, 3.2.2 and Lemma 3.2.1, we have that $\|e_i\|_{x^{i+1}}^2 \leq \frac{L^2}{\tau_i^2}\mathsf{d}(x^i, x^{i+1})^2 \leq \frac{L^2 t_i^2\|g^i\|_{x^i}^2}{\tau_i^2} = \frac{L^2\|g^i\|_{x^i}^2}{\beta^2}$, and $\mathbb{E}[\|\Delta_i^f\|_{x^{i+1}}^2|\mathcal{F}_{i-1}] \leq \sigma_i^2 + \frac{8(d+4)}{m_i}\mathbb{E}[\|\mathsf{grad}f(x^i)\|_{x^i}^2|\mathcal{F}_{i-1}]$. Hence, by applying law of total expectation (to take the expectation over all random variables), we have $\frac{1}{\Gamma_k}\mathbb{E}\|g^k - \mathsf{grad}f(x^k)\|_{x^k}^2 \leq \frac{1}{\Gamma_{k-1}}\mathbb{E}\|g^{k-1} - \mathsf{grad}f(x^{k-1})\|_{x^{k-1}}^2 + \frac{(1+\tau_{k-1})\tau_{k-1}}{\Gamma_k}\frac{L^2\mathbb{E}\|g^{k-1}\|_{x^{k-1}}^2}{\beta^2} + \frac{\tau_{k-1}^2}{\Gamma_k}\tilde{\sigma}_{k-1}^2 + \frac{\tau_{k-1}}{\Gamma_k}\hat{\sigma}^2$. Now by telescoping the sum in the above equation, we get (note that we take $g^0 = G_\mu(x^0)$)

$$
\begin{aligned}
&\mathbb{E}\|g^k - \mathsf{grad}f(x^k)\|_{x^k}^2 \leq \Gamma_k\mathbb{E}\|G_\mu(x^0) - \mathsf{grad}f(x^0)\|_{x^0}^2 \\
&+ \Gamma_k\sum_{i=1}^k\left(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i}\frac{L^2\mathbb{E}\|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i}\tilde{\sigma}_{i-1}^2 + \frac{\tau_{i-1}}{\Gamma_i}\hat{\sigma}^2\right) \\
&\leq \Gamma_k\tilde{\sigma}_0^2 + \Gamma_k\sum_{i=1}^k\left(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i}\frac{L^2\mathbb{E}\|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i}\tilde{\sigma}_{i-1}^2 + \frac{\tau_{i-1}}{\Gamma_i}\hat{\sigma}^2\right).
\end{aligned}
$$

This proves (3.8). From (3.6) we have

$$\begin{aligned}
&\sum_{k=1}^{N} \tau_k \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \\
&\leq \sum_{k=1}^{N} \tau_k \Gamma_k \sum_{i=1}^{k} \left(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i} \frac{L^2 \mathbb{E}\|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i} \tilde{\sigma}_{i-1}^2 + \frac{\tau_{i-1}}{\Gamma_i} \hat{\sigma}^2 \right) + \tilde{\sigma}_0^2 \\
&= \sum_{k=0}^{N-1} \left(\sum_{i=k+1}^{N} \tau_i \Gamma_i \right) \frac{1}{\Gamma_{k+1}} \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \tilde{\sigma}_0^2 \\
&\leq \sum_{k=0}^{N-1} \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \tilde{\sigma}_0^2,
\end{aligned}$$

where we used $\sum_{k=1}^{N} \tau_k \Gamma_k \leq \Gamma_1 = 1$ due to (3.6), so that the last term is simply $\tilde{\sigma}_0^2$.

By using similar calculations, we have that

$$\begin{aligned}
&\sum_{k=1}^{N} \tau_k^2 \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \\
&\sum_{k=1}^{N} \tau_k^2 \Gamma_k \sum_{i=1}^{k} \left(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i} \frac{L^2 \mathbb{E}\|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i} \tilde{\sigma}_{i-1}^2 + \frac{\tau_{i-1}}{\Gamma_i} \hat{\sigma}^2 \right) + \sum_{k=1}^{N} \tau_k^2 \tilde{\sigma}_0^2 \\
&= \sum_{k=0}^{N-1} \left(\sum_{i=k+1}^{N} \tau_i^2 \Gamma_i \right) \frac{1}{\Gamma_{k+1}} \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \sum_{k=1}^{N} \tau_k^2 \tilde{\sigma}_0^2 \\
&\leq \sum_{k=0}^{N-1} \tau_k \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \sum_{k=1}^{N} \tau_k^2 \tilde{\sigma}_0^2,
\end{aligned}$$

which completes the proof. □

LEMMA 3.2.3. *Suppose Assumptions 3.2.1 and 3.2.2 hold. We have*

$$\begin{aligned}
\sum_{k=1}^{N} \mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 &\leq 2\sum_{k=0}^{N} \tau_k^2 \hat{\sigma}^2 + 2\sum_{k=0}^{N} \left(\tau_k^2 + \tau_k^3\right) \sigma_k^2 + 2\sum_{k=0}^{N} \tau_k^2 \tilde{\sigma}_0^2 \\
&+ 2\sum_{k=0}^{N} (1+\tau_k)\tau_k^2 \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + 2\sum_{k=0}^{N} \left(\tau_k^2 + \tau_k^3\right) \frac{8(d+4)}{m_k} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2
\end{aligned} \tag{3.11}$$

where the expectation $\mathbb{E}$ *is taken with respect to all random variables up to iteration* k*, which includes the Gaussian variables* u *in the zeroth-order estimator as in* (3.2)*.*

PROOF. First note that $\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 = \tau_k^2 \|G_\mu^k - g^k\|_{x^k}^2 = \tau_k^2 \|G_\mu^k - \mathsf{grad} f(x^k) + \mathsf{grad} f(x^k) - g^k\|_{x^k}^2 \leq 2\tau_k^2 \|G_\mu^k - \mathsf{grad} f(x^k)\|_{x^k}^2 + 2\tau_k^2 \|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2$. Taking the expectation conditioned on $\mathcal{F}_{k-1}$, we get

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}[\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 | \mathcal{F}_{k-1}] \\
\leq &\tau_k^2 \mathbb{E}[\|G_\mu^k - \mathsf{grad} f(x^k)\|_{x^k}^2 | \mathcal{F}_{k-1}] + \tau_k^2 \mathbb{E}[\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 | \mathcal{F}_{k-1}] \\
\leq &\tau_k^2 \left(\sigma_k^2 + \frac{8(d+4)}{m_k} \mathbb{E}[\|\mathsf{grad} f(x^k)\|_{x^k}^2 | \mathcal{F}_{k-1}] \right) + \tau_k^2 \mathbb{E}[\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 | \mathcal{F}_{k-1}],
\end{aligned}$$

where last inequality is by Lemma 3.2.1. Now using law of total expectation to take the expectation for all random variables and summing up over $k = 0, ..., N-1$, we have

$$\begin{aligned}
&\frac{1}{2}\sum_{k=1}^{N} \mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 \\
\leq &\sum_{k=1}^{N} \tau_k^2 \sigma_k^2 + \sum_{k=1}^{N} \tau_k^2 \frac{8(d+4)}{m_k} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 + \sum_{k=1}^{N} \tau_k^2 \mathbb{E}\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 \\
\leq &\sum_{k=0}^{N} \tau_k^2 \hat{\sigma}^2 + \sum_{k=0}^{N} \left(\tau_k^2 + \tau_k^3\right) \sigma_k^2 + \sum_{k=0}^{N} \tau_k^2 \tilde{\sigma}_0^2 \\
&+ \sum_{k=0}^{N} (1+\tau_k) \tau_k^2 \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \sum_{k=0}^{N} \left(\tau_k^2 + \tau_k^3\right) \frac{8(d+4)}{m_k} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2,
\end{aligned}$$

where the second inequality is by Lemma 3.2.2. □

Now we are ready to present our main result.

THEOREM 3.2.1. *Suppose Assumptions 3.2.1 and 3.2.2 hold. In Algorithm 5, we set* $\mu = \mathcal{O}\left(\frac{1}{Ld^{3/2}N^{1/4}}\right)$, *and* $\beta \geq 4L$. *Then the following holds.*

(i) If we choose $\tau_0 = 1$, $\tau_k = 1/\sqrt{N}$, $k \geq 1$ *and* $m_k \equiv 8(d+4)$, $k \geq 0$, *then we have* $\frac{1}{N+1}\sum_{k=0}^{N} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \mathcal{O}(1/\sqrt{N})$.

(ii) If we choose $\tau_0 = 1$, $\tau_k = 1/\sqrt{dN}$, $k \geq 1$, $m_0 = d$ *and* $m_k = 1$ *for* $k \geq 1$, *then we have* $\frac{1}{N+1}\sum_{k=0}^{N} \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \mathcal{O}(\sqrt{d/N})$, *for all* $N = \Omega(d)$.

Here the expectation $\mathbb{E}$ *is taken with respect to all random variables up to iteration* k, *which includes the random variables* u *in zeroth-order estimator* (3.2).

Before we proceed to the proof of Theorem 3.2.1, we have the following Lemma 3.2.4 which will be utilized in the proof.

LEMMA 3.2.4. *Suppose we take parameters the same as Theorem 3.2.1, then we have*

$$\frac{\tau_k}{2\beta} - \frac{\tau_k^2 L}{2\beta^2} - \frac{(1+\tau_k)\tau_k^2}{\beta}\frac{L^2}{\beta^2} \geq \frac{\tau_k}{4\beta}, \tag{3.12a}$$

$$\frac{\tau_k}{2} - \left(4\left(\frac{2L^2}{\beta^2}+1\right)(1+\tau_k)+1\right)\frac{8(d+4)}{m_k}\tau_k^2 \geq \frac{\tau_k}{4}. \tag{3.12b}$$

PROOF. To show (3.12a), using $\beta \geq 4L$, we just need to show that $\tau_k/8 + (1+\tau_k)\tau_k/16 \leq 1/4$, which holds naturally in both cases (i) and (ii).

As for (3.12b), again by $\beta \geq 4L$ we just need to show that $\big(4(1/8{+}1)(1{+}\tau_k){+}1\big)(8(d+4)/m_k)\tau_k \leq 1/4$. In case (i), this is equivalent to $18\tau_k^2 + 22\tau_k - 1 \leq 0$, which is guaranteed when $N \geq 520$. For case (ii), similar calculation shows that we need $\tau_k \leq (\sqrt{22^2 + 9/(d+4)} - 22)/36$, which is guaranteed when $N \geq 3.2 \cdot 10^4 \cdot (d+4)^2/d$. □

PROOF OF THEOREM 3.2.1. By the isometry property of parallel transport,

$$\begin{aligned}
&\eta(x^k, g^k) - \eta(x^{k+1}, g^{k+1}) = \frac{1}{2\beta}\|g^{k+1}\|^2_{x^{k+1}} - \frac{1}{2\beta}\|g^k\|^2_{x^k} \\
=&\frac{1}{2\beta}\|P^{x^k}_{x^{k+1}} g^{k+1}\|^2_{x^k} - \frac{1}{2\beta}\|g^k\|^2_{x^k} \\
=&-\langle -\frac{1}{\beta} g^k, P^{x^k}_{x^{k+1}} g^{k+1} - g^k\rangle_{x^k} + \frac{1}{2\beta}\|P^{x^k}_{x^{k+1}} g^{k+1} - g^k\|^2_{x^k}.
\end{aligned}$$

By combining this and Assumption 3.2.1, we have the following bound for the difference of the merit function (defined in (3.7)), evaluated at successive iterates:

$$\begin{aligned}
&W(x^{k+1}, g^{k+1}) - W(x^k, g^k) \\
\leq& -t_k\langle \mathsf{grad} f(x^k), g^k\rangle_{x^k} + \frac{t_k^2 L}{2}\|g^k\|^2_{x^k} + \frac{1}{\beta}\langle g^k, P^{x^k}_{x^{k+1}} g^{k+1} - g^k\rangle_{x^k} + \frac{1}{2\beta}\|P^{x^k}_{x^{k+1}} g^{k+1} - g^k\|^2_{x^k} \\
=&\left(\frac{t_k^2 L}{2} - t_k\right)\|g^k\|^2_{x^k} + t_k\langle g^k, G^k_\mu - \mathsf{grad} f(x^k)\rangle_{x^k} + \frac{1}{2\beta}\|P^{x^k}_{x^{k+1}} g^{k+1} - g^k\|^2_{x^k}.
\end{aligned}$$

Moreover, we have

$$\begin{aligned}&\mathbb{E}_{u^k}[\langle g^k, G_\mu(x^k) - \mathsf{grad} f(x^k)\rangle_{x^k}] = \langle g^k, \mathbb{E}_{u^k} G_\mu(x^k) - \mathsf{grad} f(x^k)\rangle_{x^k}\\ \leq &\frac{1}{2}\|g^k\|_{x^k}^2 + \frac{1}{2}\|\mathbb{E}_{u^k} G_\mu(x^k) - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \frac{1}{2}\|g^k\|_{x^k}^2 + \frac{1}{2}\hat{\sigma}^2,\end{aligned}$$

where the expectation is only taken with respect to the Gaussian random variables that we used to construct $G_\mu(x^k)$. Therefore, by using the law of total expectation, we have $\mathbb{E}W(x^{k+1}, g^{k+1}) - \mathbb{E}W(x^k, g^k) \leq \frac{1}{\beta}\left(\frac{\tau_k^2 L}{2\beta} - \frac{\tau_k}{2}\right)\mathbb{E}\|g^k\|_{x^k}^2 + \frac{\tau_k}{2\beta}\hat{\sigma}^2 + \frac{1}{2\beta}\mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2$, and we thus have (by summing up the above inequality over $k = 0, ..., N$):

$$\begin{aligned}&\sum_{k=0}^{N}\Big(\mathbb{E}W(x^{k+1}, g^{k+1}) - \mathbb{E}W(x^k, g^k)\Big)\\ \leq &\sum_{k=0}^{N}\frac{1}{2\beta}\left(\frac{\tau_k^2 L}{\beta} - \tau_k\right)\mathbb{E}\|g^k\|_{x^k}^2 + \sum_{k=0}^{N}\frac{\tau_k}{2\beta}\hat{\sigma}^2 + \frac{1}{2\beta}\sum_{k=1}^{N}\mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2,\end{aligned} \tag{3.13}$$

where the last term sums from 1 since $g^1 - P_{x^0}^{x^1} g^0 = \tau_0(G_\mu^0 - g^0) = 0$.

Utilizing (3.11) and (3.13), we have (note that $W \geq 0$)

$$\begin{aligned}&\sum_{k=0}^{N}\frac{1}{2\beta}\left(\tau_k - \frac{\tau_k^2 L}{\beta}\right)\mathbb{E}\|g^k\|_{x^k}^2 \leq W(x^0, g^0) + \sum_{k=0}^{N}\frac{\tau_k}{2\beta}\hat{\sigma}^2 + \frac{1}{2\beta}\sum_{k=1}^{N}\mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2\\ \leq &W(x^0, g^0) + \frac{1}{2\beta}\sum_{k=0}^{N}(\tau_k + 2\tau_k^2)\hat{\sigma}^2 + \frac{1}{\beta}\sum_{k=0}^{N}\left(\tau_k^2 + \tau_k^3\right)\sigma_k^2 + \frac{1}{\beta}\sum_{k=0}^{N}\tau_k^2\tilde{\sigma}_0^2\\ &+ \frac{1}{\beta}\sum_{k=0}^{N}(1+\tau_k)\tau_k^2\frac{L^2\mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \frac{1}{\beta}\sum_{k=0}^{N}\left(\tau_k^2 + \tau_k^3\right)\frac{8(d+4)}{m_k}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2.\end{aligned}$$

Combining this with (3.12a) we have

$$\begin{aligned}\sum_{k=0}^{N}\tau_k\mathbb{E}\|g^k\|_{x^k}^2 \leq &\,4\beta W(x^0, g^0) + 2\sum_{k=0}^{N}(\tau_k + 2\tau_k^2)\hat{\sigma}^2 + 4\sum_{k=0}^{N}\left(\tau_k^2 + \tau_k^3\right)\sigma_k^2\\ &+ 4\sum_{k=0}^{N}\tau_k^2\tilde{\sigma}_0^2 + 4\sum_{k=0}^{N}\left(\tau_k^2 + \tau_k^3\right)\frac{8(d+4)}{m_k}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2.\end{aligned} \tag{3.14}$$

By Lemma 3.2.2 and (3.14), we get (also by $\tau_k \leq 1$)

$$
\begin{aligned}
&\frac{1}{2}\sum_{k=0}^{N}\tau_k\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \sum_{k=0}^{N}\tau_k\mathbb{E}\|g^k-\mathsf{grad} f(x^k)\|_{x^k}^2+\sum_{k=0}^{N}\tau_k\mathbb{E}\|g^k\|_{x^k}^2\\
&\leq \sum_{k=0}^{N-1}\tau_k^2\tilde{\sigma}_k^2+\sum_{k=0}^{N-1}\tau_k\hat{\sigma}^2+\left(\frac{2L^2}{\beta^2}+1\right)\sum_{k=0}^{N}\tau_k\mathbb{E}\|g^k\|_{x^k}^2+2\tilde{\sigma}_0^2\\
&\leq \left(\frac{8L^2}{\beta}+4\beta\right)W(x^0,g^0)+\sum_{k=0}^{N}\left[\tau_k+2\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k+2\tau_k^2)\right]\hat{\sigma}^2 \qquad (3.15)\\
&\quad+\sum_{k=0}^{N}\left[\tau_k^2+4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)\right]\sigma_k^2+\left[4\left(\frac{2L^2}{\beta^2}+1\right)\sum_{k=0}^{N}\tau_k^2+2\right]\tilde{\sigma}_0^2\\
&\quad+\sum_{k=0}^{N}\left[4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)+\tau_k^2\right]\frac{8(d+4)}{m_k}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2,
\end{aligned}
$$

where $\tau_0\mathbb{E}\|g^0-\mathsf{grad} f(x^0)\|_{x^0}^2 \leq \tilde{\sigma}_0^2$ is used in the last term on the second line. By combining (3.15) and (3.12b) we get

$$
\begin{aligned}
&\sum_{k=0}^{N}\frac{\tau_k}{4}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2\\
&\leq \sum_{k=0}^{N}\left[\frac{\tau_k}{2}-\left(4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)+\tau_k^2\right)\frac{8(d+4)}{m_k}\right]\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2\\
&\leq \left(\frac{8L^2}{\beta}+4\beta\right)W(x^0,g^0)+\sum_{k=0}^{N}\left[\tau_k+2\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k+2\tau_k^2)\right]\hat{\sigma}^2 \qquad (3.16)\\
&\quad+\sum_{k=0}^{N}\left[\tau_k^2+4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)\right]\sigma_k^2+\left[4\left(\frac{2L^2}{\beta^2}+1\right)\sum_{k=0}^{N}\tau_k^2+2\right]\tilde{\sigma}_0^2.
\end{aligned}
$$

For case (i) in Theorem 3.2.1, (3.16) can be rewritten as

$$
\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \frac{c_1 W(x^0,g^0)}{\sqrt{N}}+c_2\hat{\sigma}^2+\frac{c_3\frac{1}{N}\sum_{k=0}^{N}\sigma_k^2}{\sqrt{N}}+\frac{c_4}{\sqrt{N}}\tilde{\sigma}_0^2,
$$

for some absolute positive constants c_1, c_2, c_3 and c_4. The proof for case (i) is completed by noting that (see (3.9)) $\hat{\sigma}^2=\mathcal{O}(1/\sqrt{N})$, $\frac{1}{N}\sum_{k=0}^{N}\sigma_k^2=\mathcal{O}(1)$ and $\tilde{\sigma}_0^2=\mathcal{O}(1)$.

For case (ii) in Theorem 3.2.1, (3.16) can be rewritten as

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad}f(x^k)\|_{x^k}^2 \leq c_1' W(x^0,g^0)\sqrt{\frac{d}{N}} + c_2'\hat{\sigma}^2 + \frac{c_3'\frac{1}{N}\sum_{k=0}^{N}\sigma_k^2}{\sqrt{dN}} + c_4'\sqrt{\frac{d}{N}}\tilde{\sigma}_0^2,$$

for some positive constants c_1', c_2', c_3' and c_4'. The proof of case (ii) is completed by noting that $\tilde{\sigma}_0^2 = \mathcal{O}(1)$, $\hat{\sigma}^2 = \mathcal{O}(1/\sqrt{N})$ and $\frac{1}{N}\sum_{k=0}^{N}\sigma_k^2 = \mathcal{O}(d)$. □

REMARK 3.2.1. *If we sample $R \in \{0,1,2,...,N\}$ with $\mathbb{P}(R=k) = \tau_k/(\sum_{k=0}^{N}\tau_k)$, then the left hand side of the inequalities in Theorem 3.2.1, i.e., $\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad}f(x^k)\|_{x^k}^2$, becomes $\mathbb{E}\|\mathsf{grad}f(x^R)\|_{x^R}^2$. If we use this sampling in case (i) of Theorem 3.2.1, then to get an ϵ-approximate stationary solution as in Definition 3.2.1, we require an iteration complexity of $N = \mathcal{O}(1/\epsilon^4)$ and so an oracle complexity of $Nm = \mathcal{O}(d/\epsilon^4)$. Case (i) requires $m = \mathcal{O}(d)$ per-iteration, which might be inconvenient in practice. Case (ii) of Theorem 3.2.1 avoids this, as in case (ii) both the iteration complexity and the oracle complexity are $N = \mathcal{O}(d/\epsilon^4)$, with batch size $m = \mathcal{O}(1)$. This makes case (ii) more convenient to use in practice, from a streaming or online perspective. For the simulations in Section 3.4, we thus choose $m = \mathcal{O}(1)$ and apply the result from case (ii). We also remark that the above results provide concrete solutions to the question raised by Scheinberg [2022], namely, on the need for mini-batches (and its order per-iteration) in zeroth-order stochastic optimization*[1].

REMARK 3.2.2. *Notice that to prove* (3.12b)*, we need $N = \Omega(d)$ for case (ii) in Theorem 3.2.1. We can remove this condition if in addition we have that $\mathsf{grad}f(x)$ is uniformly upper bounded: $\|\mathsf{grad}f(x)\|_x \leq G,\ \forall x \in \mathcal{M}$; see also Assumption 3.3.2 which we utilize in the next section. Under this condition,* (3.15) *directly gives:*

$$\begin{aligned}
\frac{1}{2}\sum_{k=0}^{N}\tau_k\mathbb{E}\|\mathsf{grad}f(x^k)\|_{x^k}^2 \leq{}& \sum_{k=0}^{N}\left[\tau_k + 2\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k+2\tau_k^2)\right]\hat{\sigma}^2 \\
&+ \sum_{k=0}^{N}\left[\tau_k^2 + 4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)\right]\sigma_k^2 + \left[4\left(\frac{2L^2}{\beta^2}+1\right)\sum_{k=0}^{N}\tau_k^2 + 2\right]\tilde{\sigma}_0^2 \\
&+ \sum_{k=0}^{N}\left[4\left(\frac{2L^2}{\beta^2}+1\right)(\tau_k^2+\tau_k^3)+\tau_k^2\right]\frac{8(d+4)}{m_k}G^2 + \left(\frac{8L^2}{\beta}+4\beta\right)W(x^0,g^0),
\end{aligned}$$

[1]Although Scheinberg [2022] focuses on the Euclidean case, the discussion there also holds in the Riemannian setting.

whose right hand side has the same order as (3.16)*. Therefore in this case we do not need* $N = \Omega(d)$ *for case (ii) to achieve the same rates of convergence as in Theorem 3.2.1.*

3.3. RASA with retractions and vector transports

Algorithm 5 is based on exponential mapping and parallel transport, which has a high per-iteration complexity for various manifold choices $\mathcal{M}$. In this section, we focus on reducing the per-iteration complexity of the `Zo-RASA` algorithm. The approach is based on replacing the exponential mapping and parallel transport with retractions and vector transports, respectively, which leads to practically efficient implementations and improved per-iteration complexity.

The convergence analysis of algorithms with retractions and vector transports are sharply different and much harder than the one we presented in Section 3.2. Recall that the analysis in Section 3.2 relied on the isometry property (1.8) of the parallel transports, which is no longer available for vector transports. We hence assume explicit global error bounds between the difference of retraction to exponential mapping, as well as vector transport to parallel transport in Assumption 3.3.1. In Section 3.3.1.2 we provide conditions on the manifold under which such assumptions are naturally satisfied and provide explicit examples. Based on this, we establish that under a bounded fourth (instead of the second) central moment condition, the same sample complexity result as in the previous section could be obtained for the practical versions of `Zo-RASA` algorithm on compact manifolds.

3.3.1. Approximation error of retractions and vector transports. We start with the following condition on the vector transport used; recall the notation from Definition 1.2.5.

Assumption 3.3.1. *If* $x^+ = \mathsf{Retr}_x(g)$*,* $g \in \mathrm{T}_x\mathcal{M}$*, then with* d *denoting the geodesic distance, the vector transport* $\mathcal{T}_g$ *satisfies the following inequalities:*

$$\|\mathcal{T}_g(v)\|_{x^+} \leq \|v\|_x,\ \mathsf{d}(x,x^+) \leq \|g\|_x,\ \|\mathcal{T}_g(v) - P_x^{x^+}(v)\|_{x^+} \leq C\|v\|_x \mathsf{d}(x,x^+) \tag{3.17}$$

for any vector $v \in \mathrm{T}_x\mathcal{M}$*.*

An intuitive explanation of the first inequality in (3.17) is that our retraction and vector transport are "conservative" so that their length/magnitude is not longer than the exact operation of

exponential mapping and parallel transport. As for the last inequality in (3.17), we are essentially positing that the vector transport would not "twist" the vector too much so that its difference from the parallel-transported vector is not large. In general, conditions in (3.17) require the vector transport not to be very far from the parallel-transported vectors on the new tangent space.

3.3.1.1. *Comparison to prior works.* We now provide a detailed comparison to similar type of conditions proposed in two prior works, Huang et al. [2015] and Sato [2022], and highlight the differences and advantages of our proposal. According to the definition of vector transport in Definition 1.2.5, we need to specify a retraction associated with the transport so that $\mathcal{T}_{\eta_x}(\xi_x) \in \mathrm{T}_{R_x(\eta_x)}\mathcal{M}$. In this section, we consider the projection retraction, denoted simply as R.

Given two transports, $\mathcal{T}_S$ and $\mathcal{T}_R$, Huang et al. [2015] propose certain conditions on approximating one with the other. First they require that $\mathcal{T}_S$ is isometric, i.e., $\langle \mathcal{T}_{S_\eta}(\xi), \mathcal{T}_{S_\eta}(\zeta)\rangle_{R_x(\eta)} = \langle \xi, \zeta\rangle_x$, hence we can basically regard $\mathcal{T}_S$ as parallel transport for comparison. Let $\mathcal{T}_R$ denote the differential of the retraction, given by $\mathcal{T}_{R_\eta}(\xi) = DR_x(\eta)[\xi] = \frac{d}{dt}R_x(\eta + t\xi) \in \mathrm{T}_{R_x(\eta)}\mathcal{M}$. Now the conditions stated in Equations (2.5) and (2.6) in Huang et al. [2015] are as follows: there exists a *neighborhood* $\mathcal{U}$ of x, such that $\forall y \in \mathcal{U}$ we have $\|\mathcal{T}_{S_\eta} - \mathcal{T}_{R_\eta}\|_{\mathrm{op}} \le c_0\|\eta\|_x$ and $\|\mathcal{T}_{S_\eta}^{-1} - \mathcal{T}_{R_\eta}^{-1}\|_{\mathrm{op}} \le c_0\|\eta\|_x$, where $\eta = R_x^{-1}(y)$ and $\|\cdot\|_{\mathrm{op}}$ is the operator norm. These assumptions are essentially local results, and as a result, Huang et al. [2015] needs to impose an additional stringent condition (see, their Assumption 3.2) that all the updates in their algorithms are already sufficiently close to the (local) optimal value to prove their convergence results. With the above conditions (in particular for a $\mathcal{T}_{1_\eta}$ satisfying their conditions in (2.5) and (2.6)), Huang et al. [2015] shows in Lemma 3.5 that *locally* we have $\|\mathcal{T}_{1_\eta}(\xi) - \mathcal{T}_{2_\eta}(\xi)\|_y \le c_0\|\eta\|_x\|\xi\|_x$. The proof of their Lemma 3.5 relies on the smoothness of the local coordinate form of the vector transports, which could hold only when we have a coordinate chart covering the local neighborhood we consider. Hence, the assumptions in Huang et al. [2015] are in a different flavor from ours. In particular, our assumptions are global, and we show in Theorem 3.3.1 that they are satisfied by a certain (global) assumption on the second fundamental form of the manifold $\mathcal{M}$.

The existing work Huang et al. [2015] also assumes the so-called locking condition $\mathcal{T}_{S_\eta}(\xi) = \beta\mathcal{T}_{R_\eta}(\xi)$, where $\beta = \|\xi\|_x/\|\mathcal{T}_{R_\xi}(\xi)\|_{R_\xi(x)}$, which means that the approximating transport keeps the same direction as the parallel transport $\mathcal{T}_S$. In our analysis, we avoid such a condition since we

are trying to transport two vectors g^k and G_μ^k (see Algorithm 6), and not just one previous gradient as in the Riemannian quasi-Newton method [Huang et al., 2015]. Another existing work Sato [2022] requires algorithm-specific conditions in their Assumption 3.1. To elaborate, we recall that the *deterministic* Riemannian conjugate gradient iterates (Algorithm 1 in Sato [2022]) are given by $x_{k+1} \leftarrow R_{x_k}(t_k\eta_k)$ and $\eta_{k+1} \leftarrow -\mathsf{grad} f(x_{k+1}) + \beta_{k+1} s_k \mathscr{T}^k(\eta_k)$, where t_k, β_k and s_k are parameters and $\mathscr{T}^k$ is a transport map from $\mathrm{T}_{x_k}\mathcal{M}$ to $\mathrm{T}_{x_{k+1}}\mathcal{M}$. Given this, their Assumption 3.1 requires that there exist $C \geq 0$ and index sets $K_1 \subset \mathbb{N}$ and $K_2 = \mathbb{N} - K_1$ such that $\left\| \mathscr{T}^{(k)}(\eta_k) - \mathrm{D}R_{x_k}(t_k\eta_k)[\eta_k] \right\|_{x_{k+1}} \leq C t_k \|\eta_k\|_{x_k}^2, k \in K_1$ and $\left\| \mathscr{T}^{(k)}(\eta_k) - \mathrm{D}R_{x_k}(t_k\eta_k)[\eta_k] \right\|_{x_{k+1}} \leq C\left(t_k + t_k^2\right)\|\eta_k\|_{x_k}^2, k \in K_2$.

Our assumption differs from the above in three aspects: (i) we do not make algorithm-specific assumptions, where each inequality depends on the iterate number k; (ii) we are not only comparing transporting η_k (which is the direction along which we update x^k), but also the zeroth-order estimator G_μ^k (see Algorithm 6), i.e., we assume a more general inequality by replacing $\mathrm{D}R_x(t_k\eta)[\eta]$ with $\mathrm{D}R_x(t_k\eta)[\xi]$, where ξ can be different from η; (iii) we *derive* the last inequality in (3.17) using global assumption of second fundamental form of the manifold $\mathcal{M}$ in Theorem 3.3.1, instead of *assuming* it.

3.3.1.2. *Illustrative Examples.* We now further inspect Assumption 3.3.1 by checking the conditions under which (3.17) holds in general, and also verifying it for various matrix-manifolds arising in applications.

We start with the first inequality in (3.17). It holds naturally if the manifold is a submanifold and the vector transport is the orthogonal projection, due to the non-expansiveness of orthogonal projections. The second inequality in (3.17) is much trickier. For the scope of this work, we show that the second equation in (3.17) holds for projectional retractions and projectional vector transports on Stiefel manifold, which also includes spheres and orthogonal groups as special cases. If the inverse of the retraction in Assumption 3.3.1 is well-defined, the second inequality in (3.17) could equivalently be stated as $\|\mathsf{Exp}_x^{-1}(x^+)\|_x \leq \|\mathsf{Retr}_x^{-1}(x^+)\|_x$, which may hold for a larger class of manifolds and retractions. We leave a detailed study of this as future work.

Stiefel manifold. Consider the Stiefel manifold $\mathrm{St}(d,p)$ defined in (1.7), with the tangent space $\mathrm{T}_X\,\mathrm{St}(d,p) = \{\xi | X^\top\xi + \xi^\top X = 0\}$ and Euclidean inner product $\langle X, Y\rangle := \mathrm{tr}(X^\top Y)$. We

consider the projectional retraction [Absil and Malick, 2012] given by $X^+ = R_X(\xi) := UV^\top$, where $X + \xi = U\Sigma V^\top$ is the (thin) singular value decomposition of $X + \xi$. Also, the projectional vector transport $\mathcal{T}$ is simply projecting a tangent vector $\xi \in \mathrm{T}_{X_0}\,\mathrm{St}(d,p)$ to $\mathrm{T}_X\,\mathrm{St}(d,p)$. It is clear that $\|\mathcal{T}(\xi)\| \le \|\xi\|$ due to the non-expansiveness of orthogonal projections (note that $\mathrm{T}_X\,\mathrm{St}(d,p)$ is simply a linear subspace). To show $\mathsf{d}(X, X^+) \le \|\xi\|$, denote $\gamma(t)$ the minimal geodesic connecting X and X^+ with $\gamma(0) = X$ and $\gamma(1) = X^+$, so that $\mathsf{d}(X, X^+) = \int_0^1 \|\gamma'(t)\|dt$. Notice that we can define another curve $c(t) = U(t)V^\top(t)$, where $X + t\xi = U(t)\Sigma(t)V^\top(t)$ is the singular value decomposition. The curve $c(t) = \mathsf{Retr}_X(t\xi)$ is the parameterized curve of projectional retraction. Now using the distance with respect to the minimal geodesic, we have $\mathsf{d}(X, X^+) = \int_0^1 \|\gamma'(t)\|dt \le \int_0^1 \|c'(t)\|dt \le \int_0^1 \|\xi\|dt = \|\xi\|$, where $\|c'(t)\| \le \|\xi\|$ is due to the non-expansiveness of orthogonal projections, namely, $\|c(t_1) - c(t_2)\| \le \|X + t_1\xi - (X + t_2\xi)\|$. Indeed, although $\mathrm{St}(d,p)$ is not a convex set, the non-expansiveness condition still holds [Gallivan and Absil, 2010], because $(X+\xi)^\top(X+\xi) = I_p + \xi^\top\xi \succeq I_p$, and the projection of $X + \xi$ onto the Stiefel manifold is the same as projection onto its convex hull $\{X \in \mathbb{R}^{d\times p} | \|X\|_2 \le 1\}$. Now we turn to the last inequality in (3.17). Given a complete embedded submanifold, we can show that the last inequality in (3.17) holds under the boundedness of the second fundamental form in Theorem 3.3.1, given that the vector transport is the orthogonal projection to the new tangent space.

THEOREM 3.3.1. *Suppose $\mathcal{M}$ is an embedded complete Riemannian submanifold of Euclidean space. Suppose for all unit vector $\xi, \eta \in \mathrm{T}\,\mathcal{M}$, $\|\xi\| = \|\eta\| = 1$, the norm of the second fundamental form $B(\xi,\eta)$ is bounded by constant C. Consider the parallel transport P_x^y along the minimal geodesic from $x \in \mathcal{M}$ to $y \in \mathcal{M}$, we have $\|\mathsf{proj}_{\mathrm{T}_y\,\mathcal{M}}(v) - P_x^y(v)\| \le C\|v\|\mathsf{d}(x,y)$, for any $v \in \mathrm{T}_x\,\mathcal{M}$. That is, the last inequality in* (3.17) *holds with constant C.*

PROOF. Without loss of generality, we assume $\|v\| = 1$, otherwise conduct the proof for $v/\|v\|$. Denote the minimum geodesic γ with unit speed connecting x and y, parameterized by variable t, also denote the parallel transported vector of v along γ as $v(t)$, i.e. $v(0) = v$. Now for the extrinsic geometry, we denote $v = v^\top(t) + v^\perp(t)$, where $v^\top(t) \in \mathrm{T}_{\gamma(t)}\,\mathcal{M}$ and $v^\perp(t)$ is orthogonal to $\mathrm{T}_{\gamma(t)}\,\mathcal{M}$. Note that the left-hand side of the inequality we want to prove is now parameterized as $\|v(t) - v^\top(t)\|$.

Now since $v(t)$ is a parallel transport of v, the tangent component must be zero, i.e., $(v'(t))^\top = 0$. Now consider any parallel unit vector $z(t) \in \mathrm{T}_{\gamma(t)}\mathcal{M}$ along γ, then $\langle (v^\perp)'(t), z(t)\rangle = -\langle v^\perp(t), z'(t)\rangle = -\langle v^\perp(t), B(\gamma'(t), z(t))\rangle$, where B is the second fundamental form. Along with the fact that $(v^\top)' = -(v^\perp)'$ we get $\langle (v^\top)'(t), z(t)\rangle = \langle v^\perp(t), B(\gamma'(t), z(t))\rangle$. Now the right-hand side has a uniform upper bound of C, and by the arbitrarily chosen $z(t) \in \mathrm{T}_{\gamma(t)}\mathcal{M}$, we get $\|((v^\top)'(t))^\top\| \leq C$.

We can now bound the derivative of $\|v(t) - v^\top(t)\|$ as $(\|v(t) - v^\top(t)\|^2)' = (1 - 2\langle v(t), v^\top(t)\rangle + \|v^\top(t)\|^2)' = -2\langle v(t), (v^\top(t))'\rangle + 2\langle v^\top(t), (v^\top(t))'\rangle = 2\langle v^\top(t) - v(t), ((v^\top(t))')^\top\rangle \leq 2C\|v^\top(t) - v(t)\|$. Therefore, we get $\|v(t) - v^\top(t)\|' \leq C$. Now integrating the above inequality from x to y along the minimal geodesic γ (i.e., with respect to t) and using the distance with respect to the minimal geodesic, we obtain $\|\mathsf{proj}_{\mathrm{T}_y\mathcal{M}}(v) - P_x^y(v)\| \leq C\mathsf{d}(x, y)$, which completes the proof. □

Theorem 3.3.1 connects extrinsic and intrinsic geometry by measuring the difference of orthogonal projection (extrinsic operation) and parallel transport (intrinsic operation), which might be of independent interest for studying embedded submanifolds. The condition in Theorem 3.3.1 is stronger than the bounded sectional curvature condition since if the second fundamental form is bounded, the sectional curvature is also bounded by the Gauss formula (see Chapter 6, Theorem 2.5 in Do Carmo [1992]). We point out that the condition of Theorem 3.3.1 is still satisfied by all the embedded submanifold applications we consider, namely the sphere, the orthogonal group and the Stiefel manifold. In particular, we have the following observation.

Proposition 3.3.1. *Suppose $\mathcal{M}$ is a compact complete embedded Riemannian submanifold of Euclidean space (i.e. satisfying Assumption 3.3.2), then the norm of the second fundamental form $\|B(\xi, \eta)\|$ is uniformly bounded for all unit vector $\xi, \eta \in \mathrm{T}\mathcal{M}$, $\|\xi\| = \|\eta\| = 1$.*

The proof is immediate, since for all unit vector $\xi, \eta \in \mathrm{T}\mathcal{M}$, $\|B(\xi, \eta)\| \in \mathbb{R}$ is a smooth function defined over a compact domain, and therefore it is upper bounded. As a result, Assumption 3.3.1 holds for all the embedded submanifold applications we consider, namely the sphere, the orthogonal group and the Stiefel manifold.

Remark 3.3.1. *We remind the readers that Theorem 3.3.1 requires the embedded submanifold assumption, yet Assumption 3.3.1 does not, as long as* (3.17) *hold. This is also the main reason why we summarize our assumption as in Assumption 3.3.1, and not present Theorem 3.3.1 directly.*

Example: Grassmann manifold. Above, we have shown that Assumption 3.3.1 holds for a class of embedded matrix submanifolds. Yet another setting is that of quotient manifolds (e.g., the Grassmann manifold) which arises in applications of Riemannian optimization. Such manifolds are not naturally embedded submanifolds of a Euclidean space. As a result, we can inspect Assumption 3.3.1 directly for such manifolds. Taking the Grassmann manifold as an example, we next verify Assumption 3.3.1. To proceed, we utilize the following result.

LEMMA 3.3.1. *Suppose $X \in \mathrm{St}(d,p)$, $G \in \mathbb{R}^{d\times p}$ with $X^\top G = 0$, and the QR decomposition of $X+G = QR$ where $Q \in \mathrm{St}(d,p)$ and $R \in \mathbb{R}^{p\times p}$ is upper triangular. The principal angle between the subspace spanned by X and Q is given by $\|\Theta\|_F$, where $\Theta := \arccos(\Sigma)$ where Σ is the singular value matrix of $X^\top Q$, i.e., $X^\top Q = U\Sigma V^\top$; see, for example Edelman et al. [1998, Section 4.3]. We have that $\|\Theta\|_F \leq \|G\|_F$.*

PROOF. Since $R^\top R = (X+G)^\top(X+G) = I_p + \|G\|_F^2$, we know that all the singular values of R are greater than or equal to 1. Denote $\Sigma = \mathrm{diag}([\sigma_1, ..., \sigma_p])$. Since $X^\top Q = X^\top(X+G)R^{-1} = R^{-1}$, we know that the singular value decomposition of $R = V\Sigma^{-1}U^\top$ (which implies that $\sigma_i \leq 1$, $\forall i = 1, 2, ..., p$) and $\|R\|_F^2 = \|\Sigma^{-1}\|_F^2 = \sum_{i=1}^p \frac{1}{\sigma_i^2}$. Also, as $\|R\|_F^2 = \|X+G\|_F^2 = \mathrm{tr}((X+G)^\top(X+G)) = p + \|G\|_F^2$, we get $\|G\|_F^2 = \sum_{i=1}^p \frac{1}{\sigma_i^2} - p$. Thus, $\|\Theta\|_F^2 = \|\arccos(\Sigma)\|_F^2 = \sum_{i=1}^p (\arccos(\sigma_i))^2 \leq \sum_{i=1}^p (\frac{1}{\sigma_i^2} - 1) = \|G\|_F^2$, where we use the fact that $(\arccos(t))^2 \leq \frac{1}{t^2} - 1$, $\forall t \in (0, 1]$. □

Now we can inspect the Grassmann manifold. The Grassmann manifold $\mathrm{Gr}(d,p)$ is the set of all p-dimensional subspace of $\mathbb{R}^d$; see, for example, [Absil et al., 2008, Section 2.1]. A quotient formulation writes $\mathrm{Gr}(d,p) = \mathrm{St}(d,p)/\mathcal{O}(p)$ with $\mathcal{O}(p) = \{Q \in \mathbb{R}^{p\times p} | Q^\top Q = I_p\}$ being the orthogonal group. The elements of the Grassmann manifold can be expressed as $[X] \in \mathrm{Gr}(d,p)$ with $[X] := \{XQ | Q \in \mathcal{O}(p)\}$ and $X \in \mathrm{St}(d,p)$. The element $\bar{\xi}$ on the tangent space $\mathrm{T}_{[X]}\,\mathrm{Gr}(d,p)$ can be shown with a one-to-one mapping (called the horizontal lift) to the set $[\xi]$ with $\xi \in \mathrm{T}_X\,\mathrm{St}(d,p)$ and $X^\top \xi = 0$.

Suppose we start from an element $[X] \in \mathrm{Gr}(d,p)$ with $X \in \mathrm{St}(d,p)$ and the initial speed $\bar{G} \in \mathrm{T}_{[X]}\,\mathrm{Gr}(d,p)$, where $G \in \mathrm{T}_X\,\mathrm{St}(d,p)$ and $X^\top G = 0$. We denote the singular value decomposition of $G = U\Sigma V^\top$ with $U \in \mathbb{R}^{d\times p}$ and $\Sigma, V \in \mathbb{R}^{p\times p}$. Then the exponential mapping is given by $Y := \mathsf{Exp}_{[X]}(\bar{G}) = [XV\cos(\Sigma) + U\sin(\Sigma)]$, where sin and cos are matrix trigonometric functions;

see [Absil et al., 2008, Example 5.4.3]. Also, the parallel transport is given by: $\bar{\xi_1} = P_{[X]}^{[Y]}(\bar{\xi})$ with $\xi_1 = -XV\sin(\Sigma)U^\top\xi + U\cos(\Sigma)U^\top\xi + (I - UU^\top)\xi$. See [Absil et al., 2008, Example 8.1.3]. Hence, the projectional retraction is given by $Y' := \mathsf{Retr}_{[X]}(\bar{G}) = [X + G] = [Q]$, where $X + G = QR$ is the QR decomposition of $X + G$; see [Absil et al., 2008, Example 4.1.5]. Furthermore, the projectional vector transport is given by $\bar{\xi_2} = \mathcal{T}_{\bar{G}}(\bar{\xi})$ with $\xi_2 = (I - YY^\top)\xi$. See [Absil et al., 2008, Example 8.1.10].

Now we show that (3.17) is satisfied. It is obvious that $\|\mathcal{T}_{\bar{G}}(\bar{\xi})\| = \|(I - YY^\top)\xi\| \leq \|\xi\|$. The geodesic distance of $[X]$ and the projectional retraction $[Q]$ is exactly the principal angle between the subspace spanned by X and Q, see [Edelman et al., 1998, Section 4.3]. Following Lemma 3.3.1, we can hence conclude that $\mathsf{d}([X], [Q]) = \|\Theta\|_F \leq \|G\|_F$. Now we inspect the last equation in (3.17). We can directly check that $\|\xi_1 - \xi_2\|_F = \|A\xi\|_F \leq \|A\|_F\|\xi\|_F$, with

$$\begin{aligned} A \coloneqq & - XV\sin(\Sigma)U^\top + U\cos(\Sigma)U^\top + YY^\top - UU^\top \\ = & - XV\sin(\Sigma)U^\top + U\cos(\Sigma)U^\top - U\cos^2(\Sigma)U^\top + XV\cos^2(\Sigma)V^\top X^\top \\ & + U\sin(\Sigma)\cos(\Sigma)V^\top X^\top + XV\cos(\Sigma)\sin(\Sigma)U^\top. \end{aligned}$$

Note also that we have the bound

$$\begin{aligned} \|A\| = & \| - XV\sin(\Sigma)U^\top + U\cos(\Sigma)U^\top - U\cos^2(\Sigma)U^\top + XV\cos^2(\Sigma)V^\top X^\top \\ & + U\sin(\Sigma)\cos(\Sigma)V^\top X^\top + XV\cos(\Sigma)\sin(\Sigma)U^\top\| \\ \leq & \|\sin(\Sigma)\| + \|\cos(\Sigma)(I - \cos(\Sigma))\| + 2\|\sin(\Sigma)\cos(\Sigma)\| \leq 4\|\sin(\Sigma)\| \leq 4\|G\|, \end{aligned}$$

where we use the fact that $X^\top X = U^\top U = V^\top V = I_p$ and all norms are the Frobenius norm. Therefore, we see that the last equation in (3.17) is satisfied with $C = 4$.

3.3.2. Convergence of retraction and vector transport based `Zo-RASA`. We now proceed to the convergence analysis of `Zo-RASA` algorithm with retraction and vector transports. Algorithm 6 is the analog of Algorithm 5, using retraction and vector transport. Notice that the zeroth-order estimator used in Algorithm 6 is as defined in (3.3), which is with respect to the retraction in contrast to (3.2). Also $\mathcal{T}$ is the vector transport where we write $\mathcal{T}^k := \mathcal{T}_{-t_k g^k}$ for

Algorithm 6: Zo-RASA with retraction and vector transport

1: Change the updates of x^{k+1} and g^{k+1} in Algorithm 5 respectively to

$$x^{k+1} \leftarrow \mathsf{Retr}_{x^k}(-t_k g^k) \quad \text{and} \quad g^{k+1} \leftarrow (1-\tau_k)\mathcal{T}^k(g^k) + \tau_k \mathcal{T}^k(G_\mu^k),$$

where $G_\mu^k = G_\mu^{\mathsf{Retr}}(x^k)$ is given by (3.3) with batch-size $m = m_k$.

brevity. The vector transport we use in experiments is simply the orthogonal projection onto the target tangent space.

For our analysis, apart from the smoothness condition in Assumption 3.2.1, we also need to assume that the manifold is compact.

Assumption 3.3.2. *The manifold $\mathcal{M}$ is compact with diameter D, and the Riemannian gradient satisfies $\|\mathsf{grad} f(x)\|_x \le G$.*

Here, G could potentially be a function of D and the constant L from Assumption 3.2.1, due to compactness and smoothness. We remark that this compactness assumption is satisfied by various matrix manifolds like the Stiefel manifold and the Grassmann manifold (see, for example, Lemma 5.1 in Milnor and Stasheff [1974]).

Turning to the stochastic gradient oracles, the bounded second moment condition in Assumption 3.2.2 is now replaced by the following condition of bounded fourth central moment. Such a condition is needed to conduct our convergence analysis. It is interesting to relax this assumption or show this condition is necessary and sufficient to design batch-free, fully-online algorithms with vector transports and retractions.

Assumption 3.3.3. *Along the trajectory of the algorithm, we have that the stochastic gradients are unbiased and have bounded fourth central moment, i.e., for each $k \in \{1, \dots, N\}$, we have $\mathbb{E}_\xi[\mathsf{grad} F(x^k; \xi_k) | \mathcal{F}_{k-1}] = \mathsf{grad} f(x^k)$ and $\mathbb{E}_\xi[\|\mathsf{grad} F(x^k; \xi_k) - \mathsf{grad} f(x^k)\|_{x^k}^4 | \mathcal{F}_{k-1}] \le \sigma^4$.*

Note that Assumption 3.3.3 implies Assumption 3.2.2. To proceed with the convergence analysis of Algorithm 6, we also need to assume that the retraction we use in Algorithm 6 is a second-order retraction, as in Assumption 3.3.4.

Assumption 3.3.4. *The retraction we use in Algorithm 6 is a second order retraction, i.e. $\forall \xi \in \mathrm{T}_x \mathcal{M}$, we have $\mathsf{d}(\mathsf{Retr}_x(\xi), \mathsf{Exp}_x(\xi)) \le C\|\xi\|_x^2$.*

Note that the notion of second order retraction is only a local property, i.e., the above inequality only holds when $\|\xi\|$ is not too large. We refer to second order retraction without this locality restriction, since we assume the compactness of $\mathcal{M}$ in Assumption 3.3.2 and thus the condition in Assumption 3.3.4 also holds for large $\|\xi\|$ and the constant C will globally depend on the curvature of the manifold. We also point out that the condition in Assumption 3.3.4 is satisfied by projectional retractions; see, for example, [Absil and Malick, 2012, Proposition 2.2]. The study of higher-order (better) approximation to the exponential mapping by the retractions is still an on-going research topic Gawlik and Leok [2018], while here we only need a second-order retraction.

The following result in Lemma 3.3.2, which is a standard comparison-type result, will be utilized in the subsequent proof.

LEMMA 3.3.2 (Theorem 6.5.6 in Burago et al. [2022]). *Suppose the sectional curvature of $\mathcal{M}$ is upper bounded, then $\forall \xi, \eta \in \mathrm{T}_x \mathcal{M}$, we have $\|\xi - \eta\|_x \leq C\, \mathsf{d}(\mathsf{Exp}_x(\xi), \mathsf{Exp}_x(\xi))$, without loss of generality we assume the constant to be $C = 1$ for the rest of the paper.*

The following result shows that with a second-order retraction, the smoothness with respect to exponential mapping implies the smoothness with respect to retractions.

LEMMA 3.3.3. *Suppose Assumption 3.2.1, 3.3.1 and 3.3.2 hold, if the retraction we use in Algorithm 6 and* (3.3) *satisfy Assumption 3.3.4, then there exists a parameter $L' > 0$, such that f is also L'-smooth with the retraction, i.e., $|f(\mathsf{Retr}_x(\eta)) - f(x) - \langle \mathsf{grad} f(x), \eta\rangle_x| \leq \frac{L'}{2}\|\eta\|_x^2,\ \forall \eta \in \mathrm{T}_x \mathcal{M}$. From now on, we denote L as the parameter that satisfies both Assumption 3.2.1 and Lemma 3.3.3 for brevity.*

PROOF. Denote $y = \mathsf{Retr}_x(\eta)$. Note that we have $|f(y) - f(x) - \langle \mathsf{grad} f(x), \eta\rangle_x| \leq |f(y) - f(x) - \langle \mathsf{grad} f(x), \mathsf{Exp}_x^{-1}(y)\rangle_x| + |\langle \mathsf{grad} f(x), \mathsf{Exp}_x^{-1}(y) - \eta\rangle_x| \leq L\|\mathsf{Exp}_x^{-1}(y)\|_x^2 + \|\mathsf{grad} f(x)\|_x \|\eta - \mathsf{Exp}_x^{-1}(y)\|_x \leq L\|\eta\|_x^2 + G\mathsf{d}(\mathsf{Exp}_x(\eta), y) \leq (L + GC)\|\eta\|_x^2 =: L'\|\eta\|_x^2$, where the first inequality is by Assumption 3.2.2, the second is by Assumption 3.3.1 and Lemma 3.3.2, and the last inequality is by Assumption 3.3.4. □

We remind the readers that Lemma 3.3.3 can guarantee that the retraction-based zeroth-order estimator (3.3) still satisfies Lemma 3.2.1. In addition, we have the following bound on the fourth moment of G_μ^{Retr}.

LEMMA 3.3.4. *Consider G_μ given by* (3.3). *Under Assumptions 3.2.1, 3.3.1, 3.3.2 and 3.3.3, we have* $\mathbb{E}\|G_\mu^{\mathsf{Retr}}(x)\|_x^4 \le \frac{\mu^4 L^4}{2}(d+12)^6 + 3d^2\|\mathsf{grad} f(x)\|_x^4$, *where the expectation is taken toward the Gaussian vectors when constructing G_μ and the random variable ξ.*

PROOF. Since $\mathbb{E}\|G_\mu^{\mathsf{Retr}}(x)\|_x^4 = \frac{1}{\mu^4}\mathbb{E}_u[(f(\mathsf{Retr}_x(\mu u)) - f(x))^4\|u\|_x^4]$ and

$$\begin{aligned}
&(f(\mathsf{Retr}_x(\mu u)) - f(x))^4 \\
=&(f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u\rangle_x + \langle \mathsf{grad} f(x), \mu u\rangle_x)^4 \\
\le& 8(f(\mathsf{Retr}_x(\mu u)) - f(x) - \langle \mathsf{grad} f(x), \mu u\rangle_x)^4 + 8(\langle \mathsf{grad} f(x), \mu u\rangle_x)^4 \\
\le& 8\left(\frac{L}{2}\|\mu u\|_x^2\right)^4 + 8(\langle \mathsf{grad} f(x), \mu u\rangle_x)^4,
\end{aligned}$$

where the last inequality is by Lemma 3.3.3. Therefore we have

$$\begin{aligned}
\mathbb{E}\|G_\mu^{\mathsf{Retr}}(x)\|_x^4 &\le \frac{\mu^4 L^4}{2}\mathbb{E}\|u\|_x^{12} + 8\mathbb{E}[\langle \mathsf{grad} f(x), u\rangle_x^4\|u\|_x^4] \\
&\le \frac{\mu^4 L^4}{2}(d+12)^6 + 8\mathbb{E}[\langle \mathsf{grad} f(x), u\rangle_x^4\|u\|_x^4],
\end{aligned}$$

where the last inequality is by Lemma 2 in Li et al. [2023]. It remains to bound the last term on the right hand side, and we apply the same trick as in Proposition 1 in Li et al. [2023] here. Since u is an Gaussian vector on the tangent space $\mathrm{T}_x\mathcal{M}$ (dimension is d), we can calculate the expectation using the integral directly (denote $g = \mathsf{grad} f(x)$ and omit the subscript x for simplicity):

$$\begin{aligned}
&\mathbb{E}(\|\langle \mathsf{grad} f(x), u\rangle u\|^4) = \frac{1}{\kappa(d)}\int_{\mathbb{R}^d}\langle g, x\rangle^4\|x\|^4 e^{-\frac{1}{2}\|x\|^2}dx \\
\le&\frac{1}{\kappa(d)}\int_{\mathbb{R}^d}\|x\|^4 e^{-\frac{\tau}{2}\|x\|^2}\langle g, x\rangle^4 e^{-\frac{1-\tau}{2}\|x\|^2}dx \le \frac{1}{\kappa(d)}\left(\frac{4}{\tau e}\right)^2\int_{\mathbb{R}^d}\langle g, x\rangle^4 e^{-\frac{1-\tau}{2}\|x\|^2}dx \\
=&\frac{1}{\kappa(d)}\left(\frac{4}{\tau e}\right)^2\left(\frac{1}{1-\tau}\right)^{d/2-2}\int_{\mathbb{R}^d}\langle g, x\rangle^4 e^{-\frac{1}{2}\|x\|^2}dx = 48\left(\frac{1}{\tau e}\right)^2\left(\frac{1}{1-\tau}\right)^{d/2-2}\|g\|^4,
\end{aligned}$$

where $\kappa(d) := \int_{\mathbb{R}^d} e^{-\frac{1}{2}\|x\|^2}dx$ is the constant that normalizes Gaussian distribution, the second inequality is by the following fact: $x^p e^{-\frac{\tau}{2}x^2} \le (\frac{p}{\tau e})^{p/2}$, the second equality is by change of variables and the last equality is by $\mathbb{E}_{x\sim\mathcal{N}(0,I_d)}\langle g, x\rangle^4 = 3\|g\|^4$. Taking $\tau = 4/d$ gives the desired result. □

We now provide the convergence result for `Zo-RASA` (Algorithm 6). We remind the readers that we assume $C = 1$ in both Assumptions 3.3.1 and 3.3.4. We would first need to utilize the following Lemma 3.3.5, which is an analog to Lemma 3.2.2.

LEMMA 3.3.5. *Suppose Assumptions 3.2.1, 3.3.1, 3.3.2, 3.3.3 and 3.3.4 hold, and $\{x^k, g^k\}$ is generated by Algorithm 5. We have*

$$\mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \Gamma_k \tilde{\sigma}_0^2 + \Gamma_k \sum_{i=1}^{k} \Big(\frac{(1+\tau_{i-1})\tau_{i-1}}{\Gamma_i} \frac{L^2 \|g^{i-1}\|_{x^{i-1}}^2}{\beta^2} + \frac{\tau_{i-1}^2}{\Gamma_i} \tilde{\sigma}_{i-1}^2 + \tau_k \hat{\sigma}^2 \Big),$$

where the expectation $\mathbb{E}$ is taken with respect to all random variables up to iteration k, including the Gaussian variables $\{u_i\}_{i=1}^k$ in the zeroth-order estimator (3.2), and $\tilde{\sigma}_k^2$ is defined in (3.9). Further, from the definition of τ_k in (3.4), we have

$$\sum_{k=1}^{N} \tau_k \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \sum_{k=0}^{N-1} \left((1+\tau_k)\tau_k \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^2 \tilde{\sigma}_k^2 + \tau_k \hat{\sigma}^2 \right) + \tilde{\sigma}_0^2,$$

$$\sum_{k=1}^{N} \tau_k^2 \mathbb{E}\|g^k - \mathsf{grad} f(x^k)\|_{x^k}^2 \leq \sum_{k=0}^{N-1} \left((1+\tau_k)\tau_k^2 \frac{L^2 \mathbb{E}\|g^k\|_{x^k}^2}{\beta^2} + \tau_k^3 \tilde{\sigma}_k^2 + \tau_k^2 \hat{\sigma}^2 \right) + \sum_{k=1}^{N} \tau_k^2 \tilde{\sigma}_0^2.$$

PROOF. The proof is almost identical to the proof of Lemma 3.2.2, and we thus omit the details. Note that here we need to utilize Assumption 3.3.1 to show $\mathsf{d}(x^i, x^{i+1})^2 \leq t_i^2 \|g^i\|_{x^i}^2$. □

To show the bound for the term $\mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2$, we further need to utilize the following bound for $\|g^k\|_{x^k}$ first.

LEMMA 3.3.6. *Consider g^k given by Algorithm 6. Suppose Assumption 3.2.1, 3.3.1, 3.3.2, 3.3.3 and 3.3.4 hold. Then, we have $\mathbb{E}\|g^k\|_{x^k}^2 \leq \mu^2 L^2 (d+6)^3 + 2(d+4)G^2$ and $\mathbb{E}\|g^k\|_{x^k}^4 \leq \frac{\mu^4 L^4}{2}(d+12)^6 + 3d^2 G^4$, where the expectation $\mathbb{E}$ is taken with respect to all random variables up to iteration k.*

PROOF. Note that we have

$$\begin{aligned} \|g^k\|_{x^k}^2 &= \|(1-\tau_{k-1})\mathcal{T}^{k-1}(g^{k-1}) + \tau_{k-1}\mathcal{T}^{k-1}(G_\mu^{k-1})\|_{x^k}^2 \\ &\leq (1-\tau_{k-1})\|g^{k-1}\|_{x^{k-1}}^2 + \tau_{k-1}\|G_\mu^{k-1}\|_{x^{k-1}}^2. \end{aligned}$$

Taking expectation conditioned on $\mathcal{F}_{k-1}$, we have by Lemma 3.2.1 that $\mathbb{E}[\|g^k\|_{x^k}^2 | \mathcal{F}_{k-1}] \leq (1-\tau_{k-1})\mathbb{E}\|g^{k-1}\|_{x^{k-1}}^2 + \tau_{k-1}(\mu^2 L^2 (d+6)^3 + 2(d+4)\|\mathsf{grad} f(x^{k-1})\|_{x^{k-1}}^2)$. We remove the conditional

expectation by law of total expectation, also by Assumption 3.3.2 we have that

$$\mathbb{E}\|g^k\|_{x^k}^2 \leq (1-\tau_{k-1})\mathbb{E}\|g^{k-1}\|_{x^{k-1}}^2 + \tau_{k-1}(\mu^2 L^2 (d+6)^3 + 2(d+4)G^2).$$

Denote $A_k = \mathbb{E}\|g^k\|_{x^k}^2$, note that we have $A_k \leq (1-\tau_{k-1})A_{k-1} + \tau_{k-1}(\mu^2 L^2 (d+6)^3 + 2(d+4)G^2)$. Again from Lemma 3.2.1 we have $A_0 \leq \mu^2 L^2 (d+6)^3 + 2(d+4)G^2$, from which and using induction, we conclude that $A_k = \mathbb{E}\|g^k\|_{x^k}^2 \leq \mu^2 L^2 (d+6)^3 + 2(d+4)G^2$. As for the fourth moment, note that

$$\begin{aligned}
&\mathbb{E}(\|g^k\|_{x^k}^2)^2 \leq \mathbb{E}\left((1-\tau_{k-1})\|g^{k-1}\|_{x^{k-1}}^2 + \tau_{k-1}\|G_\mu^{k-1}\|_{x^{k-1}}^2\right)^2 \\
\leq &(1-\tau_{k-1})\mathbb{E}\|g^{k-1}\|_{x^{k-1}}^4 + \tau_{k-1}\mathbb{E}\|G_\mu^{k-1}\|_{x^{k-1}}^4, \\
\leq &(1-\tau_{k-1})\mathbb{E}\|g^{k-1}\|_{x^{k-1}}^4 + \tau_{k-1}\left(\frac{\mu^4 L^4}{2}(d+12)^6 + 3d^2\|\mathsf{grad} f(x^k)\|_{x^k}^4\right)
\end{aligned}$$

where the last inequality is by Lemma 3.3.4. The final result follows similarly to the second moment case. □

Now we are ready to study the difference between g^k and g^{k+1}.

LEMMA 3.3.7. *Suppose Assumptions 3.2.1, 3.3.1, 3.3.2, 3.3.3 and 3.3.4 hold, and take τ_k as in* (3.4)*. Then, we have*

$$\begin{aligned}
\sum_{k=1}^N \mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 \leq \frac{4L^2}{\beta^2}\sum_{k=0}^{N-1}(1+\tau_k)\tau_k^2\mathbb{E}\|g^k\|_{x^k}^2 + 4\sum_{k=0}^N(\tau_k^2+\tau_k^3)\tilde{\sigma}_k^2 \\
+ \left[4\tilde{\sigma}_0^2 + 4\hat{\sigma}^2 + \frac{8}{\beta^2}\left(\frac{\mu^4 L^4}{2}(d+12)^6 + 3d^2 G^4\right)\right]\sum_{k=0}^N \tau_k^2,
\end{aligned} \tag{3.18}$$

where the expectation $\mathbb{E}$ is taken with respect to all random variables up to iteration k, which includes the random variables u in the zeroth-order estimator (3.3)*.*

PROOF. Since

$$\begin{aligned}
&\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 = \|g^{k+1} - P_{x^k}^{x^{k+1}} g^k\|_{x^{k+1}}^2 \\
\leq& 2\|g^{k+1} - \mathcal{T}^k g^k\|_{x^{k+1}}^2 + 2\|\mathcal{T}^k g^k - P_{x^k}^{x^{k+1}} g^k\|_{x^{k+1}}^2 \\
\leq& 2\tau_k^2 \|G_\mu^k - g^k\|_{x^k}^2 + 2\mathsf{d}(x^{k+1}, x^k)^2 \|g^k\|_{x^k}^2 \\
\leq& 4\tau_k^2 \|G_\mu^k - \mathsf{grad} f(x^k)\|_{x^k}^2 + 4\tau_k^2 \|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 + 2\frac{\tau_k^2}{\beta^2}\|g^k\|_{x^k}^4,
\end{aligned}$$

where the second inequality is by the update and Assumption 3.3.1, and the last inequality is by Assumption 3.3.1. Now taking the expectation conditioned on $\mathcal{F}_{k-1}$ we get:

$$\begin{aligned}
\mathbb{E}[\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 | \mathcal{F}_{k-1}] \leq\ & 4\tau_k^2 \mathbb{E}[\|G_\mu^k - \mathsf{grad} f(x^k)\|_{x^k}^2 | \mathcal{F}_{k-1}] \\
&+ 4\tau_k^2 \mathbb{E}[\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 | \mathcal{F}_{k-1}] + 2\frac{\tau_k^2}{\beta^2}\mathbb{E}[\|g^k\|_{x^k}^4 | \mathcal{F}_{k-1}].
\end{aligned}$$

Thus we have (by law of total expectation):

$$\begin{aligned}
&\sum_{k=1}^N \mathbb{E}\|P_{x^{k+1}}^{x^k} g^{k+1} - g^k\|_{x^k}^2 \\
\leq& 4\sum_{k=1}^N \tau_k^2 \mathbb{E}\|G_\mu^k - \mathsf{grad} f(x^k)\|_{x^k}^2 + 4\sum_{k=1}^N \tau_k^2 \mathbb{E}\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 + \frac{2}{\beta^2}\sum_{k=1}^N \tau_k^2 \mathbb{E}\|g^k\|_{x^k}^4 \\
\leq& 4\sum_{k=1}^N \tau_k^2 \tilde{\sigma}_k^2 + 4\sum_{k=1}^N \tau_k^2 \mathbb{E}\|\mathsf{grad} f(x^k) - g^k\|_{x^k}^2 + \frac{8}{\beta^2}\left(\frac{\mu^4 L^4}{2}(d+12)^6 + 3d^2 G^4\right)\sum_{k=1}^N \tau_k^2
\end{aligned}$$

where the second inequality is by Lemmas 3.2.1 and 3.3.6. The desired result follows by applying Lemma 3.3.5 to the above inequality. □

We now state the main result in Theorem 3.3.2, as an analog to Theorem 3.2.1. Notice that different from Theorem 3.2.1, we do not need $N = \Omega(d)$ in case (ii), in view of Remark 3.2.2 and Assumption 3.3.2.

THEOREM 3.3.2. *Suppose Assumptions 3.2.1, 3.3.1, 3.3.2, 3.3.3 and 3.3.4 hold. In Algorithm 6, we set* $\mu = \mathcal{O}\big(\frac{1}{Ld^{3/2}N^{1/4}}\big)$ *and* $\beta \geq \sqrt{d}L$. *Then the following holds.*

(i) If we choose $\tau_0 = 1$, $\tau_k = 1/\sqrt{N}$, $k \geq 1$ *and* $m_k \equiv 8(d+4)$, $k \geq 0$, *then we have* $\frac{1}{N+1}\sum_{k=0}^N \mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \mathcal{O}(1/\sqrt{N})$.

(ii) If we choose $\tau_0 = 1$, $\tau_k = 1/\sqrt{dN}$, $k \geq 1$, $m_0 = d$ *and* $m_k = 1$ *for* $k \geq 1$, *then we have* $\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \mathcal{O}(\sqrt{d/N})$.

Here the expectation $\mathbb{E}$ *is taken with respect to all random variables up to iteration* k, *which includes the random variables* u *in zeroth-order estimator* (3.3).

Proof of Theorem 3.3.2. The proof is very similar to the proof of Theorem 3.2.1. We first will have the following inequality analogue to (3.14):

$$\frac{1}{8\beta^2}\sum_{k=0}^{N}\tau_k\mathbb{E}\|g^k\|_{x^k}^2 \leq W^0 + \frac{1}{2\beta}\sum_{k=0}^{N}\tau_k\hat{\sigma}^2 + \frac{2}{\beta}\sum_{k=0}^{N}(\tau_k^2+\tau_k^3)\tilde{\sigma}_k^2 + \frac{1}{2\beta}[4\tilde{\sigma}_0^2 + 4\hat{\sigma}^2 + \frac{8}{\beta^2}(\frac{\mu^2L^2}{2}(d+12)^6 + 3d^2G^4)]\sum_{k=0}^{N}\tau_k^2$$

Note that we still need (3.12a) to show the above inequality.

We then directly provide the result corresponding to (3.16):

(3.19)
$$\sum_{k=1}^{N}\frac{\tau_k}{2}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq (8\beta^2+16L^2)\bigg(W^0 + \frac{1}{2\beta}\sum_{k=0}^{N}\tau_k\hat{\sigma}^2 + \frac{2}{\beta}\sum_{k=0}^{N}(\tau_k^2+\tau_k^3)\tilde{\sigma}_k^2 + \frac{1}{2\beta}[4\tilde{\sigma}_0^2 + 4\hat{\sigma}^2 + \frac{8}{\beta^2}(\frac{\mu^2L^2}{2}(d+12)^6 + 3d^2G^4)]\sum_{k=0}^{N}\tau_k^2\bigg) + \sum_{k=0}^{N-1}\tau_k^2\tilde{\sigma}_k^2 + \sum_{k=0}^{N-1}\tau_k^2\hat{\sigma}^2 + \tilde{\sigma}_0^2$$

Now by Assumption 3.3.2, we have $\tilde{\sigma}_k^2 \leq \sigma_k^2 + \frac{8(d+4)}{m_k}G^2$, which is exactly the reason we don't need to show an inequality similar to (3.12b).

For case (i) in Theorem 3.3.2, (3.19) can be rewritten as

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq \frac{c_1 W(x^0,g^0)}{\sqrt{N}} + c_2\hat{\sigma}^2 + \frac{c_3\frac{1}{N}\sum_{k=0}^{N}\tilde{\sigma}_k^2}{\sqrt{N}} + \frac{c_4}{\sqrt{N}}\tilde{\sigma}_0^2,$$

for some absolute positive constants c_1, c_2, c_3 and c_4. The proof for case (i) is completed by noting that (see (3.9)) $\hat{\sigma}^2 = \mathcal{O}(1/\sqrt{N})$, $\frac{1}{N}\sum_{k=0}^{N}\tilde{\sigma}_k^2 = \mathcal{O}(1)$ and $\tilde{\sigma}_0^2 = \mathcal{O}(1)$.

For case (ii) in Theorem 3.3.2, (3.19) can be rewritten as

$$\frac{1}{N+1}\sum_{k=0}^{N}\mathbb{E}\|\mathsf{grad} f(x^k)\|_{x^k}^2 \leq c_1' W(x^0,g^0)\sqrt{\frac{d}{N}} + c_2'\hat{\sigma}^2 + \frac{c_3'\frac{1}{N}\sum_{k=0}^{N}\tilde{\sigma}_k^2}{\sqrt{dN}} + c_4'\sqrt{\frac{d}{N}}\tilde{\sigma}_0^2,$$

for some positive constants c_1', c_2', c_3' and c_4'. The proof of case (ii) is completed by noting that $\tilde{\sigma}_0^2 = \mathcal{O}(1)$, $\hat{\sigma}^2 = \mathcal{O}(1/\sqrt{N})$ and $\frac{1}{N}\sum_{k=0}^{N} \tilde{\sigma}_k^2 = \mathcal{O}(d)$. □

REMARK 3.3.2. *By the technique discussed in Remark 3.2.1, to obtain an ϵ-approximate stationary point in Definition 3.2.1 we need an oracle complexity of $\mathcal{O}(d/\epsilon^4)$.*

3.4. Numerical experiments

3.4.1. k-PCA. We now provide numerical results on the k-PCA problem to demonstrate the effectiveness of the Zo-RASA algorithms. For a given centered random vector $\mathbf{z} \in \mathbb{R}^n$, the k-PCA problem corresponds to finding the subspace spanned by the top-k eigenvectors of its positive definite covariance matrix $\Sigma = \mathbb{E}[\mathbf{z}\mathbf{z}^\top]$. Formally, we have the following problem on the Stiefel manifold:

$$\min_{X \in \mathrm{St}(n,r)} f(X) := -\frac{1}{2}\operatorname{tr}(X^\top \mathbb{E}[\mathbf{z}\mathbf{z}^\top] X). \tag{3.20}$$

Note that the dimension of the Stiefel is given by $d = nr - r(r+1)/2$.

For any $Y = XQ$ where $Q \in \mathbb{R}^{r\times r}$, and $Q^\top Q = QQ^\top = I_r$, we have $f(X) = f(Y)$. Hence, we can equivalently view (3.20) as the following minimization problem on the Grassmann manifold:

$$\min_{[X] \in \mathrm{Gr}(n,r)} f([X]) := -\frac{1}{2}\operatorname{tr}(X^\top \mathbb{E}[\mathbf{z}\mathbf{z}^\top] X).$$

Note that the dimension of the Grassmannian is given by $d = r(n-r)$.

We solve (3.20) using Algorithm 6 and compare it with the zeroth-order Riemannian SGD method from Li et al. [2023]. In all the experiments, we used projecting vector transport rather than parallel transport for Stiefel manifolds, due to the aforementioned facts that parallel transport is time-consuming to numerically compute on Stiefel manifold, and has no closed form. In the stochastic zeroth-order setting, for each query point X_k, the stochastic oracle returns a noise estimate of $f(x)$ based on a single observation $\mathbf{z}_k$, i.e. $F(X^k; \mathbf{z}_k) = -1/2 \operatorname{tr}((X^k)^\top \mathbf{z}_k \mathbf{z}_k^\top X^k)$. For our experiments, we assume $\mathbf{z}_k$ is sampled from a centered Gaussian distribution with covariance matrix given by $\Sigma = \sum_{i=1}^{r} \lambda_i v_i v_i^\top + \sum_{i=r+1}^{n} \lambda_i v_i v_i^\top$, where $V = [v_1, ..., v_n]$ is an orthogonal matrix. The first r λ_is are uniform random numbers in $[100, 200]$ and the last $n-r$ are uniform random

numbers in $[1, 50]$. For our experiments, we fix r and try different n (reflected in different rows in Figure 3.1).

We set $N = 50000 \times n$ for `Zo-RASA` and one-batch Zo-RSGD (`Zo-RSGD-1`) algorithms, while $N = 50000$ for our mini-batch Zo-RSGD algorithm (`Zo-RSGD-m`). The reason here is that for `Zo-RSGD-m`, we take $m = n = \mathcal{O}(d)$ since we fix r and change n. While the theoretical result in Li et al. [2023] requires the batch-size m to be $\mathcal{O}(d/\epsilon^2)$, they empirically observed reasonable-order batch-sizes suffices. For `Zo-RASA`, according to our theory, we again take $\tau_k = 0.01/\sqrt{N}$ and $\beta = 100$. For `Zo-RSGD-1` and `Zo-RSGD-m`, we set t_k as $t_k = 10^{-4}/\sqrt{N}$ and $t_k = 5 \times 10^{-4}/\sqrt{N}$ respectively.

For all algorithms, we again compare the function value, norm of the Riemannian gradient and the principal angles between the current iterate and the optimal subspace. Figures 3.1 plots the results. The experimental results provide support for the proposed algorithms (and the established theory), demonstrating that the proposed Zo-RASA algorithm is more efficient in terms of decreasing the Riemannian gradient and principal angles compared to conventional zeroth-order Riemannian stochastic gradient descent methods that utilize mini-batches.

3.4.2. Identification of a fixed rank symmetric positive semi-definite matrix. We now provide another numerical example from Bonnabel [2013]. Consider a matrix-version linear model as in Tsuda et al. [2005]:

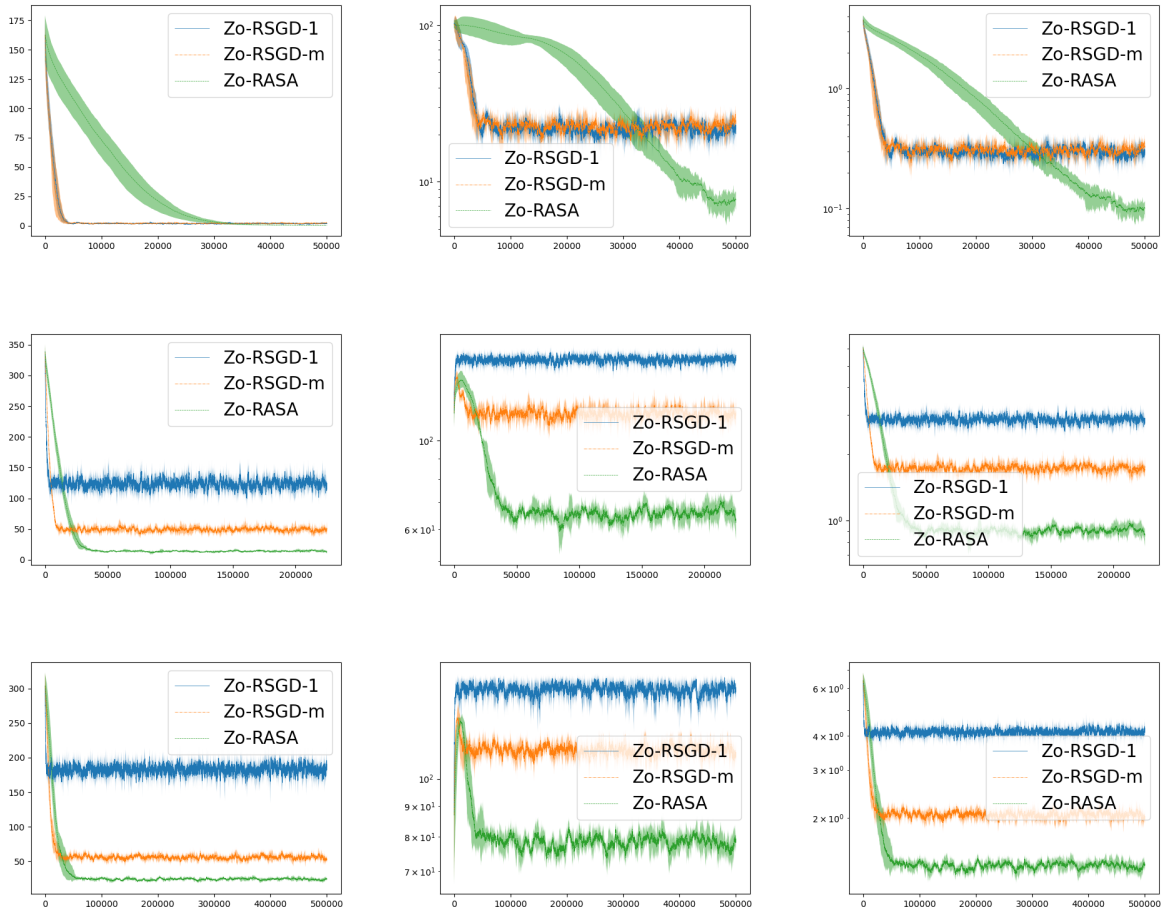
$$y_t = \mathrm{tr}(W\mathbf{x}_t\mathbf{x}_t^\top) = \mathbf{x}_t^\top W\mathbf{x}_t$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the input and $y_t \in \mathbb{R}$ is the output, and the unknown matrix $W \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix with a fixed rank r ($r \leq n$). Denote the set

$$S_+(n, r) = \{W \in \mathbb{R}^{n\times n} | W = W^\top, \mathrm{rank}(W) = r\} \tag{3.21}$$

which is the set of positive definite matrices with rank r. The problem is thus formulated as a matrix least square problem

$$\min_{W\in S_+(n,r)} f(W) := \frac{1}{2}\mathbb{E}_{\mathbf{x},y}(\mathbf{x}^\top W\mathbf{x} - y)^2 \tag{3.22}$$



(a) Optimality gap (b) $\|\mathsf{grad} f(X^t)\|$ (c) Principal angles

FIGURE 3.1. Results for kPCA (3.20) with $n \in \{10, 30, 50\}$ (corresponding to three rows) and $r = 5$. The resulting manifold (Stiefel) dimensions are $d = \{35, 135, 235\}$. The x-axis is the number of zeroth-order oracle calls (i.e. number of function value calls).

Notice that W can be represented as $W = GG^\top$ where $G \in \mathbb{R}^{n,r}$ is a matrix with full column rank. Also notice that for any orthogonal matrix $O \in \mathbb{R}^{r\times r}$ we have $W = GOO^\top G^\top = GG^\top$, we have the following quotient representation of the set of fixed rank positive definite matrices $S_+(n, r) \simeq \mathbb{R}_*^{n\times r}/\mathcal{O}(r)$, where the right hand side represents the set of equivalent classes:

$$[G] = \{GO | O \in \mathcal{O}(r)\}.$$

We could thus conduct our experiment on the quotient manifold $\mathbb{R}_*^{n\times r}/\mathcal{O}(r)$, with the following re-formulated problem:

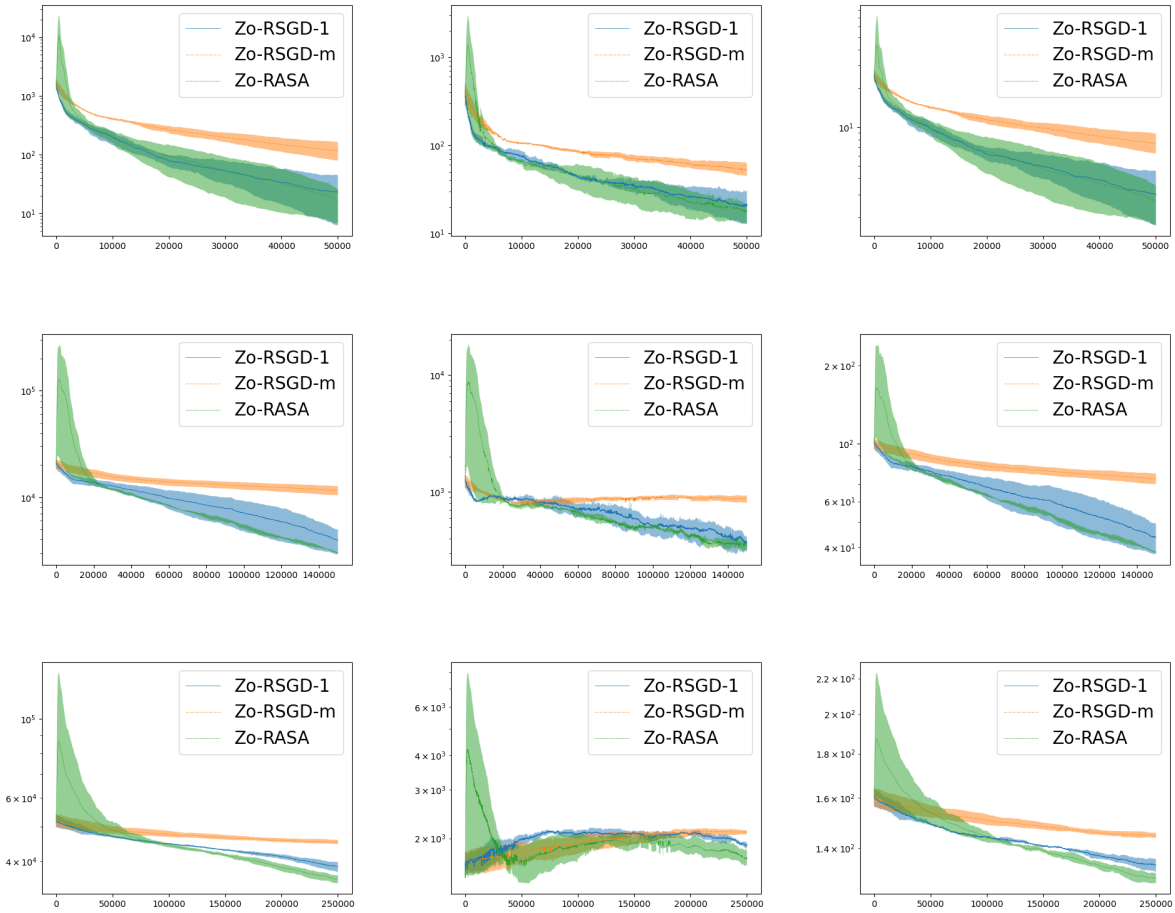
$$\min_{[G]\in\mathbb{R}_*^{n\times r}/\mathcal{O}(r)} f(G) := \frac{1}{2}\mathbb{E}_{\mathbf{x},y}(\mathbf{x}^\top GG^\top\mathbf{x} - y)^2 \tag{3.23}$$

The manifold $S_+(n,r)$ has dimension $d = nr - r(r-1)/2$ and is not a compact manifold. We test (3.23) to show the efficiency of our proposed algorithm even without the compactness assumption (Assumption 3.3.2) which we need to conduct our theoretical analysis.

We solve (3.23) using Algorithm 6 and compare it with the zeroth-order Riemannian SGD method from Li et al. [2023]. In all the experiments, we used again retraction and projecting vector transport rather than exponential mapping and parallel transport. The ground-truth $G^\star \in \mathbb{R}^{n\times r}$ is sampled randomly with standard Gaussian entries. For our experiments, we sample $\mathbf{x} \sim \mathcal{N}(0, I_d)$ and construct $y = \mathbf{x}^\top W\mathbf{x}$ noiselessly. Specifically, given a query point G^t and a Gaussian sample $\mathbf{x}_t$ with $y_t = \mathbf{x}_t^\top G^\star(G^\star)^\top\mathbf{x}_t$, the stochastic zeroth-order oracle gives the value $\frac{1}{2}(\mathbf{x}_t^\top G^t(G^t)^\top\mathbf{x}_t - y_t)^2$. For our experiments, we fix r and test with different n (reflected in different rows in Figure 3.2).

We set $N = 5000 \times n$ for `Zo-RASA` and one-batch Zo-RSGD (`Zo-RSGD-1`) algorithms, while $N = 5000$ for our mini-batch Zo-RSGD algorithm (`Zo-RSGD-m`) for the same reason as the kPCA experiments. For `Zo-RASA`, according again to our theory, we again take $\tau_k = 10^{-3}/\sqrt{N}$ and $\beta = 100$. For `Zo-RSGD-1` and `Zo-RSGD-m`, we set $t_k = 10^{-5}/\sqrt{N}$.

For all algorithms, we again compare the function value, norm of the Riemannian gradient and the quantity $\|G^t(G^t)^\top - G^\star(G^\star)^\top\|$ which measures the error to the ground truth positive semi-definite matrix. Figures 3.2 plots the results. It's worth noticing here that mini-batch Zo-RSGD seems to work the worst in the plots, which is due to the fact that we take the step sizes the same for `Zo-RSGD-1` and `Zo-RSGD-m`. The reason we cannot enlarge the step size for `Zo-RSGD-m` is that the projectional retraction and projectional vector transport requires solving a Sylvester equation which leads to numerical stability issues if the step sizes become large (see Boumal et al. [2014] for details). The experimental results provide support for the proposed algorithms (and the established theory), demonstrating that the proposed Zo-RASA algorithm is more efficient in terms of decreasing the Riemannian gradient and function values compared to conventional zeroth-order Riemannian stochastic gradient descent methods that utilize mini-batches.



(a) Optimality gap (b) $\|\mathsf{grad} f(G^t)\|$ (c) $\|G^t(G^t)^\top - G^\star(G^\star)^\top\|$

FIGURE 3.2. Results for (3.23) with $n \in \{10, 30, 50\}$ (corresponding to three rows) and $r = 5$. The resulting manifold as defined in (3.21) are $d = \{40, 140, 240\}$ dimensional, respectively. The x-axis is the number of zeroth-order oracle calls (i.e. number of function value calls).

CHAPTER 4

Federated Learning Algorithms on Riemannian Manifolds

In this chapter, we consider the finite-sum FL problem over a Riemannian manifold $\mathcal{M}$ as in (1.3), which we restate here:

$$\min_{x\in\mathcal{M}} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{4.1}$$

where $f_i : \mathcal{M} \to \mathbb{R}$ are smooth but not necessarily (geodesically) convex. Each of the f_i (or the data associated with f_i) is stored in different client/agent that could have different physical locations and different hardware. This makes the mutual connection impossible Konečnỳ et al. [2016]. Therefore, there is a central server that can collect the information from different agents and output a consensus that minimizes the summation of the loss functions from all the clients. The aim of such a framework is to utilize the computation resources of different agents while still maintain the data privacy by not sharing data among all the local agents. Thus the communication is always between the central server and local servers. This setting is commonly observed in modern smart-phone-APP based machine learning applications Konečnỳ et al. [2016]. We emphasize that we always consider the heterogeneous data scenario where the functions f_i's might be different and have different optimal solutions. This problem is inherently hard to solve because each local minima will empirically diverge the update from the global optimum Li et al. [2020], Mitra et al. [2021].

It is noted that most FL algorithms are designed for the unconstrained setting and convex constraint setting [Charles and Konečnỳ, 2021, Karimireddy et al., 2020, Konečnỳ et al., 2016, Li et al., 2019, Malinovskiy et al., 2020, McMahan et al., 2017, Mitra et al., 2021, Pathak and Wainwright, 2020], and FL problems with nonconvex constraints such as (4.1) have not been considered. The main difficulty for solving (4.1) lies in aggregating points over a nonconvex set, which may lead to the situation where the averaging point is outside of the constraint set.

One motivating application of (4.1) is the federated kPCA problem

$$\min_{X\in \mathrm{St}(d,r)} f(X) := \frac{1}{n}\sum_{i=1}^{n} f_i(X), \quad \text{where } f_i(X) = -\frac{1}{2}\operatorname{tr}(X^\top A_i X), \tag{4.2}$$

where $\mathrm{St}(d,r) = \{X \in \mathbb{R}^{d\times r} | X^\top X = I_r\}$ denotes the Stiefel manifold, and A_i is the covariance matrix of the data stored in the i-th local agent. When $r = 1$, (4.2) reduces to classical PCA

$$\min_{\|x\|_2=1} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where } f_i(x) = -\frac{1}{2}x^\top A_i x. \tag{4.3}$$

Existing FL algorithms are not applicable to (4.2) and (4.3) due to the difficulty on aggregating points on nonconvex set.

4.0.1. Main Contributions. We focus on designing efficient federated algorithms for solving (4.1). Our main contributions are:

(1) We propose a Riemannian federated SVRG algorithm (`RFedSVRG`) for solving (4.1). We prove that the convergence rate of our RFedSVRG algorithm is $\mathcal{O}(1/\epsilon^2)$ for obtaining an ϵ-stationary point. This result matches that of its Euclidean counterparts Mitra et al. [2021]. To the best of our knowledge, this is the first algorithm for solving FL problems over Riemannian manifolds with convergence guarantees.

(2) The main novelty of our `RFedSVRG` algorithm is a consensus step on the tangent space of the manifold. We compare this new approach with the widely used Karcher mean approach. We show that our method achieves certain "regularization" property and performs very well in practice.

(3) We conduct extensive numerical experiments on our method for solving the PCA (4.3) and kPCA (4.2) problems with both synthetic and real data. The numerical results demonstrate that our `RFedSVRG` algorithm significantly outperforms the Riemannian counterparts of two widely used FL algorithms: `FedAvg` McMahan et al. [2017] and `FedProx` Li et al. [2020].

4.1. The RFedSVRG Algorithm

The most challenging task for FL on Riemannian manifolds is the consensus step. Suppose the central server receives $x^{(i)}$, $i \in S_t \subset [n]$ from each of the local clients at round t, the question is

how the central server aggregates the points to output a unique consensus. In Euclidean space, the most straightforward way is to take the average $\frac{1}{k}\sum_{i\in S_t} x^{(i)}$ with $k = |S_t|$. However, this approach does not apply to the Riemannian setting due to the loss of linearity: the arithmetic average of points can be outside of the manifold. A natural choice for the consensus step on the manifold is to take the Karcher mean of the points Tron et al. [2012]:

$$x_{t+1} \leftarrow \operatorname*{argmin}_x \frac{1}{k}\sum_{i\in S_t} d^2(x, x^{(i)}), \tag{4.4}$$

where x_{t+1} is the next iterate point on the central server. This is a natural generalization of the arithmetic average because $d^2(x, y) = \|x - y\|^2$ in Euclidean space. However, solving (4.4) can be time consuming in practice.

We propose the following tangent space consensus step:

$$x_{t+1} \leftarrow \mathsf{Exp}_{x_t}\left(\frac{1}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x^{(i)})\right), \tag{4.5}$$

where we project each of the point $x_t^{(i)}$ back to the tangent space $T_{x_t}\mathcal{M}$ and then take their average on the tangent space. The consensus step (4.5) has several advantages over the Karcher mean method (4.4). First, (4.5) is of closed-form and easy to compute. Second, (4.5) still coincides with the arithmetic mean when the manifold reduces to the Euclidean space. Third, the tangent space mean (4.5) can easily be extended to the following moving average mean:

$$\mathsf{Exp}_{x_t}\left(\frac{\beta}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x^{(i)})\right),$$

which corresponds to $(1-\beta)x_t + \frac{\beta}{k}\sum_{i\in S_t} x^{(i)}$ in the Euclidean space, while the Karcher mean cannot be easily extended in this scenario. Last, (4.5) has the following "regularization" property as the distance between two consensus points can be controlled, and the Karcher mean method (4.4) does not have this kind of property.

LEMMA 4.1.1. *For the update defined in* (4.5)*, it holds that* $d(x_{t+1}, x_t) \leq \frac{1}{k}\sum_{i\in S_t} d(x^{(i)}, x_t)$.

To further illustrate this "regularization" property of the tangent space mean (4.5), we consider an (extreme) example on the unit sphere $\mathcal{S}^2$ (see Figure 4.1) . Here we take x_t on the north pole

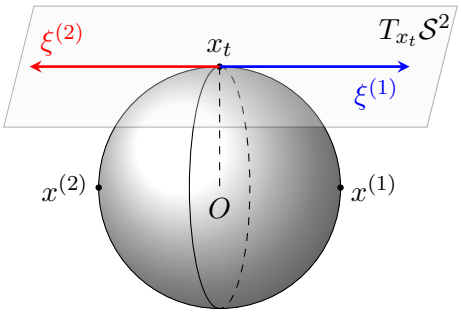


FIGURE 4.1. Comparison of two consensus methods on $\mathcal{S}^2$

and two point from the local server as $x^{(1)}$ and $x^{(2)}$, also $\xi^{(i)} = \mathsf{Exp}_{x_t}^{-1}(x^{(i)}) \in T_{x_t}\mathcal{M}$. Then the tangent space mean (4.5) would yield the original point x_t, whereas the Karcher mean could yield any point on the vertical great circle, depending on the starting point in solving the optimization problem (4.4).

`FedAvg` [McMahan et al., 2017] and `FedProx` [Li et al., 2020] are two widely used algorithms for FL problems in Euclidean space. We now discuss their plain extensions to the manifold optimization situation. As a review, at each iteration, `FedAvg` minimizes the local loss f_i for fixed steps using gradient descents:

$$x_{\ell+1}^{(i)} \leftarrow x_{\ell}^{(i)} - \eta^{(i)} \nabla f_i(x_{\ell+1}^{(i)}), \tag{4.6}$$

while `FedProx` solves a local proximal point subproblem:

$$x^{(i)} \leftarrow \operatorname*{argmin}_{x} f_i(x) + \frac{\mu}{2}\|x - x_t\|^2. \tag{4.7}$$

For `RFedAvg`, which is the Riemannian counterpart of `FedAvg`, (4.6) is replaced by

$$x_{\ell+1}^{(i)} \leftarrow \mathsf{Exp}_{x_{\ell}^{(i)}}\left(-\eta^{(i)} \mathsf{grad} f_i(x_{\ell}^{(i)})\right).$$

For `RFedProx`, which is the Riemannian counterpart of `FedProx`, (4.7) is replaced by

$$x_{t+1}^{(i)} \leftarrow \operatorname*{argmin}_{x \in \mathcal{M}} f_i(x) + \frac{\mu}{2} d^2(x, x_t), \tag{4.8}$$

where $d(x, y)$ is the geodesic distance between x and y. In the implementation of `RFedProx`, (4.8) is solved by Riemannian gradient descent:

$$x_{\ell+1}^{(i)} \leftarrow \mathsf{Exp}_{x_\ell^{(i)}}(-\eta^{(i)}\mathsf{grad}h_i(x_\ell^{(i)})),\ \ell = 0, ..., \tau_i - 1. \tag{4.9}$$

`RFedAvg` and `RFedProx` are described in Algorithms 7 and 8, respectively.

Algorithm 7: Riemannian FedAvg algorithm

input : n, k, T, $\{\eta^{(i)}\}$, $\{\tau_i\}$
output: x_T
for $t = 0, ..., T-1$ **do**
  Uniformly sample $S_t \subset [n]$ with $|S_t| = k$;
  for *each agent* i *in* S_t **do**
    Receive x_t from the central server;
    for $\ell = 0, ..., \tau_i - 1$ **do**
      $x_{\ell+1}^{(i)} \leftarrow \mathsf{Exp}_{x_\ell^{(i)}}\left(-\eta^{(i)}\mathsf{grad}f_i(x_\ell^{(i)})\right)$;
    end
    Send the obtained $x_{\tau_i}^{(i)}$ to the central server;
  end
  The central server aggregates the points by the tangent space mean (4.5);
end

Algorithm 8: Riemannian FedProx Algorithm

input : n, k, T, μ, γ
output: x_T
for $t = 0, ..., T-1$ **do**
  Uniformly sample $S_t \subset [n]$ with $|S_t| = k$;
  for *each agent* i *in* S_t **do**
    Receive x_t from the central server;
    Obtain $x^{(i)} \leftarrow \mathrm{argmin}_{x\in\mathcal{M}} f_i(x) + \frac{\mu}{2}d^2(x, x_t)$ upto a γ approximate solution;
    Send the obtained $x^{(i)}$ to the central server;
  end
  The central server aggregates the points by the tangent space mean (4.5);
end

Our `RFedSVRG` algorithm is presented in Algorithm 9, which is a non-trivial manifold extension of the FSVRG algorithm Konečnỳ et al. [2016].

For `RFedSVRG`, the local gradient update becomes

$$x_{\ell+1}^{(i)} \leftarrow \mathsf{Exp}_{x_\ell^{(i)}}\left[-\eta^{(i)}\left(\mathsf{grad} f_i(x_\ell^{(i)}) - P_{x_t \to x_\ell^{(i)}}(\mathsf{grad} f_i(x_t) - \mathsf{grad} f(x_t))\right)\right], \tag{4.10}$$

which matches the existing manifold SVRG work Zhang et al. [2016b]. The introduction of the parallel transport $P_{x_t \to x_\ell^{(i)}}$ is necessary because we need to "transport" all the vectors to the same tangent space to conduct addition and subtraction. The algorithm utilizes the gradient information at the previous iterate $\mathsf{grad} f(x_t)$, thus avoids the "client-drift" effect and correctly converges to the global stationary points. This is confirmed by both the theory and the numerical experiments.

Algorithm 9: Riemannian FedSVRG Algorithm (RFedSVRG)

input : n, k, T, $\{\eta^{(i)}\}$, $\{\tau_i\}$
output: Option 1: $\tilde{x} = x_T$; or **Option 2:** $\tilde{x}$ is uniformly sampled from $\{x_1, ..., x_T\}$
for $t = 0, ..., T-1$ **do**
  Uniformly sample $S_t \subset [n]$ with $|S_t| = k$;
  for *each agent* i *in* S_t **do**
    Receive $x_0^{(i)} = x_t$ from the central server;
    for $\ell = 0, ..., \tau_i - 1$ **do**
      Take the local gradient step (4.10).
    end
    Send $\hat{x}^{(i)}$ (obtained by one of the following options) to the central server
      - **Option 1:** $\hat{x}^{(i)} = x_{\tau_i}^{(i)}$;
      - **Option 2:** $\hat{x}^{(i)}$ is uniformly sampled from $\{x_1^{(i)}, ..., x_{\tau_i}^{(i)}\}$;
  end
  The central server aggregates the points by the tangent space mean (4.5);
end

4.2. Convergence analysis

In this section we analyze the convergence behaviour of the `RFedSVRG` algorithm (Algorithm 9). Before we proceed to the convergence results, we briefly review the necessary assumptions, which are standard assumptions for optimization on manifolds Boumal et al. [2018], Zhang and Sra [2016].

Assumption 4.2.1 (Smoothness). *Suppose f_i is L_i-smooth as defined in Definition 1.2.6. It implies that f is L-smooth with $L = \sum_{i=1}^n L_i$.*

Now we give the convergence rate results for Algorithm 9. Specifically, Theorem 4.2.1 gives the convergence rate of Algorithm 9 with $\tau_i = 1$, Theorem 4.2.2 gives the convergence rate of Algorithm 9 with $\tau_i > 1$, and Theorem 4.2.3 gives the convergence rate of Algorithm 9 when the objective function is geodescially convex.

THEOREM 4.2.1 (Nonconvex, Algorithm 9 with $\tau_i = 1$). *Suppose the problem* (4.1) *satisfies Assumption 4.2.1. If we run Algorithm 9 with* ***Option 1*** *in Line 8,* $\eta^{(i)} \leq \frac{1}{L}$ *and* $\tau_i = 1$ *(i.e. only one step of gradient update for each agent), then the* ***Option 1*** *of the output of Algorithm 9 satisfies:*

$$\min_{t=0,\ldots,T} \|\mathsf{grad} f(x_t)\|^2 \leq \mathcal{O}\left(\frac{L(f(x_0) - f(x^*))}{T}\right). \tag{4.11}$$

REMARK 4.2.1. *Our proof of Theorem 4.2.1 relies heavily on the choice of* $\tau_i = 1$ *and the consensus step* (4.5)*. When* $\tau_i > 1$*, we need to introduce multiple exponential mappings at multiple points for each iteration, which makes the convergence analysis much more challenging due to the loss of linearity. Moreover, the aggregation step makes the situation even worse. However, we are able to show the convergence of Algorithm 9 with* $\tau_i > 1$ *when* $k = 1$*. Our numerical experiments show the effectiveness of the* `RFedSVRG` *algorithm with both* $\tau_i = 1$ *and* $\tau_i > 1$*.*

To prove the convergence of Algorithm 9 with $\tau_i > 1$, we also need the following regularization assumption over the manifold $\mathcal{M}$ due to Zhang et al. [2016b].

ASSUMPTION 4.2.2 (Regularization over manifold). *The manifold is complete and there exists a compact set* $\mathcal{D} \subset \mathcal{M}$ *(diameter bounded by* D*) so that all the iterates of Algorithm 9 and the optimal points are contained in* $\mathcal{D}$*. The sectional curvature is bounded in* $[\kappa_{\min}, \kappa_{\max}]$*. Moreover, we denote the following key geometrical constant that captures the impact of manifold:*

$$\zeta = \begin{cases} \frac{\sqrt{|\kappa_{\min}|}D}{\tanh\left(\sqrt{|\kappa_{\min}|}D\right)}, & \text{if } \kappa_{\min} < 0 \\ 1, & \text{if } \kappa_{\min} \geq 0. \end{cases} \tag{4.12}$$

Notice that this assumption holds when the manifold is a sphere or a Stiefel manifold (since they are compact). Now we are ready to give the convergence rate result of Algorithm 9 with $\tau_i > 1$ and $k = 1$, the proof of which is inspired by Zhang et al. [2016b].

THEOREM 4.2.2 (Nonconvex, Algorithm 9 with $\tau_i > 1$ and $k = 1$). *Suppose the problem* (4.1) *satisfies Assumptions 4.2.1 and 4.2.2. If we run Algorithm 9 with* ***Option 2*** *in Line 8,* $k = 1$, $\tau_i = \tau > 1$, $\eta^{(i)} = \eta \leq \mathcal{O}(\frac{1}{nL\zeta^2})$, *then the* ***Option 2*** *of the output of Algorithm 9 satisfies:*

$$\mathbb{E}\|\mathsf{grad} f(\tilde{x})\|^2 \leq \mathcal{O}\left(\frac{\rho(f(x_0) - f(x^*))}{\tau T}\right),$$

where ρ *is an absolute constant specified in the proof and the expectation is taken with respect to the random index* i*, as well as the randomness introduced by the* ***Option 2****.*

Finally, we have the convergence result when the objective function of (4.1) is geodesically convex.

THEOREM 4.2.3 (Geodesic convex). *Suppose the problem* (4.1) *satisfies Assumption 4.2.1 and 4.2.2. Also the functions* f_i*'s are geodesically convex (see Definition 1.2.4) in* $\mathcal{D}$ *(as in Assumption 4.2.2). If we run Algorithm 9 with* ***Option 1*** *in Line 8,* $\tau_i = 1$, $S_t = [n]$ *(full parallel gradient), and* $\eta = \eta^{(1)} = \cdots = \eta^{(n)} \leq \frac{1}{2L}$, *then the* ***Option 1*** *of the output of Algorithm 9 satisfies:*

$$f(x_T) - f^* \leq \mathcal{O}\left(\frac{Ld^2(x_0, x^*)}{T}\right). \tag{4.13}$$

4.3. Proofs

In this section we provide the proofs of lemmas and theorems mentioned in the previous section. We first finish the proof of Lemma 4.1.1:

PROOF OF LEMMA 4.1.1. By Cauchy-Schwarz inequality we have

$$\begin{aligned} d(x_{t+1}, x_t) &= \|\mathsf{Exp}_{x_t}^{-1}(x_{t+1})\| \\ &= \|\frac{1}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x^{(i)})\| \leq \frac{1}{k}\sum_{i\in S_t} \|\mathsf{Exp}_{x_t}^{-1}(x^{(i)})\| = \frac{1}{k}\sum_{i\in S_t} d(x_t, x^{(i)}). \end{aligned}$$

□

Now we turn to the proof of Theorem 4.2.1. We would utilize the following lemma:

LEMMA 4.3.1. *Under the same settings as Theorem 4.2.1, we have*

$$f(x_{t+1}) - f(x_t) \leq -\eta_t^{(i)}\|\mathsf{grad} f(x_t)\|^2 + \frac{(\eta_t^{(i)})^2 L}{2}\|\mathsf{grad} f(x_t)\|^2.$$

PROOF OF LEMMA 4.3.1. From the update we know that

$$x_{\ell+1}^{(i)} \leftarrow \mathsf{Exp}_{x_\ell^{(i)}}\left[-\eta_t^{(i)}\left(\mathsf{grad} f_i(x_\ell^{(i)}) - P_{x_t \to x_\ell^{(i)}}(\mathsf{grad} f_i(x_t) - \mathsf{grad} f(x_t))\right)\right]$$

i.e.

$$\mathsf{Exp}_{x_\ell^{(i)}}^{-1}(x_{\ell+1}^{(i)}) \leftarrow -\eta_t^{(i)}\left(\mathsf{grad} f_i(x_\ell^{(i)}) - P_{x_t \to x_\ell^{(i)}}(\mathsf{grad} f_i(x_t) - \mathsf{grad} f(x_t))\right).$$

When $\tau_i = 1$, $x_0^{(i)} = x_t$ thus

$$\mathsf{Exp}_{x_t}^{-1}(x_1^{(i)}) \leftarrow -\eta_t^{(i)}\left(\mathsf{grad} f_i(x_t) - P_{x_t \to x_1^{(i)}}(\mathsf{grad} f_i(x_t) - \mathsf{grad} f(x_t))\right) = -\eta_t^{(i)} \mathsf{grad} f(x_t)$$

Using Lipschitz smooth of f_i again and the tangent space mean (4.5), we have

$$\begin{aligned} f(x_{t+1}) - f(x_t) \leq & \langle \mathsf{Exp}_{x_t}^{-1}(x_{t+1}), \mathsf{grad} f(x_t)\rangle + \frac{L}{2} d^2(x_{t+1}, x_t) \\ = & \langle \frac{1}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x_1^{(i)}), \mathsf{grad} f(x_t)\rangle + \frac{L}{2}\|\frac{1}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x_1^{(i)})\|^2 \\ = & -\eta_t^{(i)}\|\mathsf{grad} f(x_t)\|^2 + \frac{(\eta_t^{(i)})^2 L}{2}\|\mathsf{grad} f(x_t)\|^2, \end{aligned}$$

where we used the tangent space mean (4.5) for the first equality. □

Now we are ready to present the proof of Theorem 4.2.1.

PROOF OF THEOREM 4.2.1. By taking $\eta^{(i)} \leq \frac{1}{L}$, from Lemma 4.3.1 we have

$$f(x_{t+1}) - f(x_t) \leq -\frac{1}{2L}\|\mathsf{grad} f(x_t)\|^2.$$

Summing this inequality over $t = 0, 1, \ldots, T$, we obtain

$$\frac{1}{2L}\sum_{t=0}^{T}\|\mathsf{grad} f(x_t)\|^2 \leq f(x_0) - f(x_{T+1}) \leq f(x_0) - f(x^*),$$

which yields (4.11) immediately. □

Before we present the proof of Theorem 4.2.2, we need the following lemma, which is adopted from Zhang et al. [2016b].

LEMMA 4.3.2 (Lemma 2 in Zhang et al. [2016b]). *Consider Algorithm 9 with* ***Option*** *2. Suppose we run randomly chosen local agent i at the t-th outer iteration. If we run the local agent i for τ_i*

local gradient steps (4.10) with initial point x_t, then it holds:

$$\mathbb{E}\|\mathsf{grad} f(x_\ell^{(i)})\|^2 \leq \frac{R_\ell - R_{\ell+1}}{\delta_\ell},\ \ell = 0, ..., \tau_i - 1, \tag{4.14}$$

where the expectation is taken with respect to the randomly selected index i, $R_\ell := \mathbb{E}[f(x_\ell^{(i)}) + c_\ell \|\mathsf{Exp}_{x_t}^{-1}(x_\ell^{(i)})\|^2]$, $c_\ell = c_{\ell+1}(1 + \beta\eta + 2\zeta L^2\eta^2) + L^3\eta^2$ and $\delta_\ell = \eta - \frac{c_{\ell+1}\eta}{\beta} - L\eta^2 - 2c_{\ell+1}\zeta\eta^2$. Here β is a free constant to be determined and we take $c_{\tau_i} = 0$ in the recursive definition.

Now we turn to the proof of Theorem 4.2.2:

Proof of Theorem 4.2.2. Since $k = 1$, without loss of generality, we denote i as the agent that we choose at the t-th iteration. Moreover, we denote $\eta = \eta^{(i)}$ because there is only one agent.

From (4.14), we note that if we set $\eta < \frac{1}{L+2c_{\ell+1}\zeta}(1 - \frac{c_{\ell+1}}{\beta})$, then we have $\delta^{(i)} := \min_{\ell=0,...,\tau_i} \delta_\ell > 0$. In this case, summing (4.14) over $\ell = 0, 1, ..., \tau_i - 1$ yields

$$\frac{1}{\tau_i} \sum_{\ell=0,...,\tau_i-1} \mathbb{E}\|\mathsf{grad} f(x_\ell^{(i)})\|^2 \leq \frac{R_0 - R_{\tau_i}}{\tau_i \delta^{(i)}} \leq \mathbb{E}\left(\frac{f(x_t) - f(x_{\tau_i}^{(i)})}{\tau_i \delta^{(i)}}\right), \tag{4.15}$$

since $R_0 = f(x_t)$ and $R_{\tau_i} = \mathbb{E}[f(x_{\tau_i}^{(i)}) + c_\ell \|\mathsf{Exp}_{x_t}^{-1}(x_{\tau_i}^{(i)})\|^2] \geq \mathbb{E}[f(x_{\tau_i}^{(i)})]$. Now we take $\beta = L\zeta^{1/2}/n^{1/3}$ and $\eta = 1/(10Ln^{2/3}\zeta^{1/2})$[1]. From the recurrence $c_\ell = c_{\ell+1}(1 + \beta\eta + 2\zeta L^2\eta^2) + L^3\eta^2$ and $c_{\tau_i} = 0$ we have

$$c_0 = \frac{L}{100n^{4/3}\zeta} \frac{(1+\theta)^{\tau_i} - 1}{\theta},$$

where

$$\theta = \eta\beta + 2\zeta\eta^2 L^2 = \frac{1}{10n} + \frac{1}{50n^{4/3}} \in \left(\frac{1}{10n}, \frac{3}{10n}\right)$$

is a parameter. If we take $\tau_i = \lfloor 10n/3 \rfloor$ such that $(1+\theta)^{\tau_i} < (1 + \frac{3}{10n})^{\tau_i} < e$, then

$$c_0 \leq \frac{L}{10n^{1/3}\zeta}(e - 1),$$

[1]It is straightforward to verify that $\eta < \frac{1}{L+2c_{\ell+1}\zeta}(1 - \frac{c_{\ell+1}}{\beta})$ with this choice of η for $\ell = 0, ..., \tau_i$.

and $\delta^{(i)}$ is bounded by

$$\begin{aligned}\delta^{(i)} &\geq \left(\eta - \frac{c_0\eta}{\beta} - \eta^2 L - 2c_0\zeta\eta^2\right) \\ &\geq \eta\left(1 - \frac{e-1}{10\zeta^{3/2}} - \frac{1}{10n^{2/3}\zeta^{1/2}} - \frac{e-1}{50n\zeta^{1/2}}\right) \\ &\geq \frac{\eta}{2} = \frac{1}{20Ln^{2/3}\zeta^{1/2}},\end{aligned}$$

where the last inequality is by $\zeta, n \geq 1$. Note that this lower bound of $\delta^{(i)}$ is independent from the choice of local agent i.

Now summing (4.15) over $t = 0, ..., T-1$ with $\delta^{(i)} \geq \frac{\eta}{2}$ we get

$$\frac{1}{T}\sum_{t=0,\ldots,T-1}\frac{1}{\tau_i}\sum_{\ell=0,\ldots,\tau_i-1}\mathbb{E}\|\mathsf{grad} f(x_\ell^{(i)})\|^2 \leq \frac{2\Delta}{\tau\eta T}, \tag{4.16}$$

where $\Delta = f(x_0) - f^*$.

Now using the **Option 2** of the output of Algorithm 9, we get

$$\mathbb{E}\|\mathsf{grad} f(\tilde{x})\|^2 \leq \frac{\Delta\rho}{\tau T},$$

where $\rho = \frac{\eta}{2} = \frac{1}{20Ln^{2/3}\zeta^{1/2}}$. □

Before we present the proof of Theorem 4.2.3, we need the following lemma Zhang and Sra [2016].

LEMMA 4.3.3 (Corollary 8 in Zhang and Sra [2016]). *Suppose the sectional curvature of $\mathcal{M}$ is lower bounded by $\kappa_{\min}$ and we update $x_{t+1} \leftarrow \mathsf{Exp}_{x_t}(-\eta_t g_t)$. Suppose also that the update sequence $\{x_t\} \subset \mathcal{D}$ where $\mathcal{D}$ is a compact set with diameter D, then for any $x \in \mathcal{M}$ it holds:*

$$\langle -g_t, \mathsf{Exp}_{x_t}^{-1}(x)\rangle \leq \frac{1}{2\eta_t}(d^2(x_t, x) - d^2(x_{t+1}, x)) + \frac{\zeta\eta_t}{2}\|g_t\|^2. \tag{4.17}$$

where ζ is given in (4.12).

We now present the proof of Theorem 4.2.3.

PROOF OF THEOREM 4.2.3. From Lemma 4.3.3 we get

$$\langle \frac{1}{k}\sum_{i\in S_t} \mathsf{Exp}_{x_t}^{-1}(x^{(i)}), \mathsf{Exp}_{x_t}^{-1}(x)\rangle \leq \frac{1}{2}(d^2(x_t,x)-d^2(x_{t+1},x)) + \frac{\zeta}{2}\|\frac{1}{k}\sum_{i\in S_t}\mathsf{Exp}_{x_t}^{-1}(x^{(i)})\|^2, \tag{4.18}$$

which is equivalent to (since we assume $S_t=[n]$ and $\eta^{(i)}=\eta$):

$$-\eta\langle \frac{1}{n}\sum_{i=1,\dots,n} \mathsf{grad} f_i(x_t), \mathsf{Exp}_{x_t}^{-1}(x)\rangle \leq \frac{1}{2}(d^2(x_t,x)-d^2(x_{t+1},x)) + \frac{\zeta}{2}\|\frac{1}{n}\sum_{i=1,\dots,n}\mathsf{Exp}_{x_t}^{-1}(x^{(i)})\|^2. \tag{4.19}$$

Now use the geodesic convexity of f_i and (4.19), we have (denote $\Delta_t := f(x_t)-f(x^*)$ and $\Delta_t^i := f_i(x_t)-f_i(x^*)$)

$$\Delta_t^i \leq -\langle \mathsf{grad} f_i(x_t), \mathsf{Exp}_{x_t}^{-1}(x^*)\rangle.$$

Summing this inequality over $i=1,\dots,n$, we get

$$\begin{aligned}
\Delta_t \leq & -\langle \frac{1}{n}\sum_{i=1,\dots,n}\mathsf{grad} f_i(x_t), \mathsf{Exp}_{x_t}^{-1}(x^*)\rangle \\
\leq & \frac{1}{2\eta}(d^2(x_t,x^*)-d^2(x_{t+1},x^*)) + \frac{\zeta}{2\eta}\|\frac{1}{n}\sum_{i=1,\dots,n}\mathsf{Exp}_{x_t}^{-1}(x^{(i)})\|^2 \qquad (4.20)\\
\leq & \frac{1}{2\eta}(d^2(x_t,x^*)-d^2(x_{t+1},x^*)) + \frac{\zeta\eta}{2n}\|\mathsf{grad} f(x_t)\|^2.
\end{aligned}$$

Again from Lemma 4.3.1 we get

$$\Delta_{t+1}-\Delta_t \leq (-\eta_t^{(i)} + \frac{(\eta_t^{(i)})^2 L}{2})\|\mathsf{grad} f(x_t)\|^2. \tag{4.21}$$

Now multiply (4.21) by ζ and add it to (4.20), we get

$$\zeta\Delta_{t+1}-(\zeta-1)\Delta_t \leq \zeta\left(\frac{\eta}{2n}-\eta+\frac{\eta^2 L}{2}\right)\|\mathsf{grad} f(x_t)\|^2 + \frac{1}{2\eta}(d^2(x_t,x^*)-d^2(x_{t+1},x^*)). \tag{4.22}$$

Now take $\eta \leq \frac{1}{2L}$, we know that $\frac{\eta}{2n}-\eta+\frac{\eta^2 L}{2}\leq 0$, thus

$$\zeta\Delta_{t+1}-(\zeta-1)\Delta_t \leq \frac{1}{2\eta}(d^2(x_t,x^*)-d^2(x_{t+1},x^*)). \tag{4.23}$$

Summing this up over t from 0 to $T-1$ we get

$$\zeta\Delta_T + \sum_{t=0}^{T-1}\Delta_t \leq (\zeta-1)\Delta_1 + \frac{d^2(x_0,x^*)}{2\eta}. \tag{4.24}$$

Also by (4.21) we know $\Delta_{t+1} \leq \Delta_t$, thus

$$\Delta_T \leq \frac{\zeta D^2}{2\eta(\zeta + T - 2)}. \tag{4.25}$$

□

4.4. Numerical experiments

We now show the performance of `RFedSVRG` and compare it with two natural ideas for solving (4.1): Riemannian FedAvg (`RFedAvg`) and Riemannian FedProx (`RFedProx`), which are natural extensions of FedAvg McMahan et al. [2017] and FedProx Li et al. [2020] to the Riemannian setting. Algorithms `RFedAvg` and `RFedProx` are descried in the supplementary material of our published version [Li and Ma, 2023]. We conducted our experiments on a desktop with Intel Core 9600K CPU, 32GB RAM and NVIDIA GeForce RTX 2070 GPU. For the codes of operations on Riemannian manifolds we used the ones from the `Manopt` and `PyManopt` packages Boumal et al. [2014], Townsend et al. [2016]. Since the logarithm mapping (the inverse of the exponential mapping) on the Stiefel manifold is not easy to compute Zimmermann and Hüper [2021], we adopted the projection-like retraction Absil and Malick [2012] and the inverse of it Kaneko et al. [2012] to approximate the exponential and the logarithm mappings, respectively.

We tested the three algorithms on PCA (4.3), kPCA (4.2) and PSD Karcher mean (see the appendix of our published version [Li and Ma, 2023]) problems. For all problems, we measure the norm of the global Riemannian gradients. Additionally, we also measure the sum of principal angles Zhu and Knyazev [2013] for kPCA. [2]

4.4.1. Comparison of the two consensus methods (4.4) and (4.5). We first compare the two consensus methods (4.4) and (4.5). To this end, we randomly generate x_t and $k = 100$ points $x^{(i)}$ on the unit ball $\mathcal{S}^{d-1}$ with different dimensions d. We then compare the distances $\frac{1}{k}\sum_i d^2(x_t, x^{(i)})$, $\frac{1}{k}\sum_i d^2(x_{t+1}, x^{(i)})$ and $d^2(x_t, x_{t+1})$, as well as the CPU time for computing them. Note that the smaller these distances are, the better. To calculate the Karcher mean, we run the Riemannian gradient descent method starting at x_t until the norm of the Riemannian gradient is

[2]For the loss f in (4.2), note that $f(X) = f(XQ)$ for any orthogonal matrix $Q \in \mathbb{R}^{r\times r}$. As a result, the optimal solution of $f(X)$ only represents the eigen-space corresponds to the r-largest eigenvalues. Therefore we need the principal angles to measure the angles between the subspaces.

TABLE 4.1. Comparison of the two consensus methods (4.4) and (4.5). Here $h(x) := \frac{1}{k}\sum_i d^2(x^{(i)}, x)$, CPU time is in seconds and the experiments are repeated and averaged over 10 times.

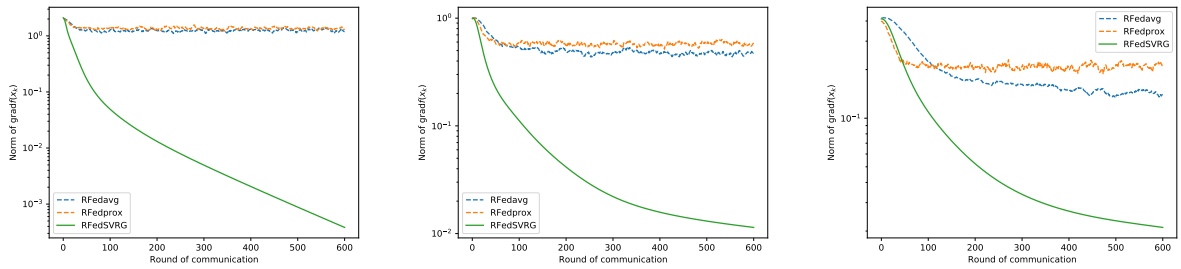
Dim d	$h(x_t)$	Karcher mean (4.4)			Tangent space mean (4.5)		
		$d^2(x_{t+1}, x_t)$	$h(x_{t+1})$	Time	$d^2(x_{t+1}, x_t)$	$h(x_{t+1})$	Time
100	2.478	2.469	2.813	0.706	0.025	2.427	0.004
200	2.472	2.484	2.804	0.641	0.025	2.422	0.004
500	2.469	2.469	2.795	0.725	0.024	2.421	0.005

smaller than $\epsilon = 10^{-6}$. The results are shown in Table 4.1. From Table 4.1 we see that the tangent space mean (4.5) is indeed better than Karcher mean (4.4) in terms of both quality and CPU time.

4.4.2. Experiments on synthetic data. In this section, we report the results of the three algorithms for solving PCA (4.3) and kPCA (4.2) on synthetic data. We first generate the data $X_i \in \mathbb{R}^{d\times p}$ whose entries are drawn from standard normal distribution. We then set $A_i := X_i X_i^\top$. Notice that under this experiment setting the data in different agents are homogeneous in distribution, which provides a mild environment for comparing the behaviour of the proposed algorithms. We test highly heterogeneous real data later.

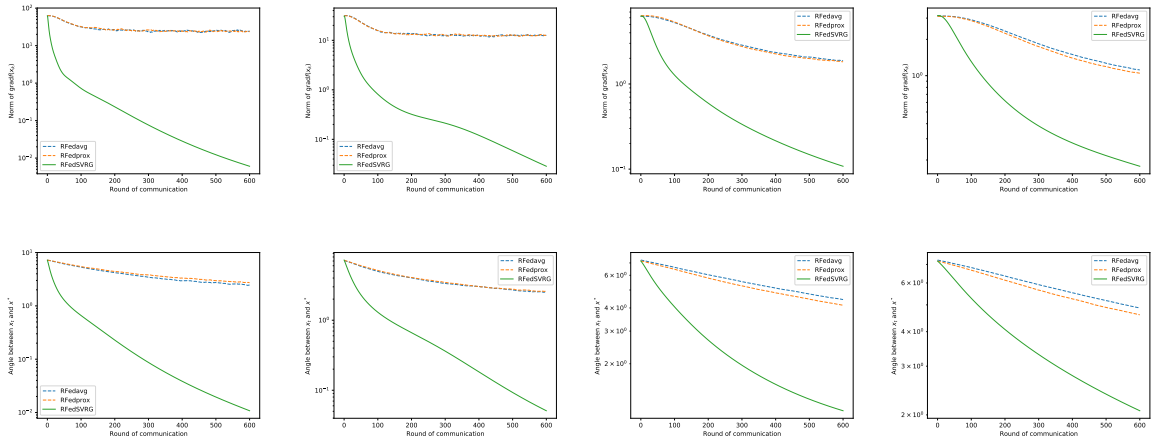
Experiments on PCA. We test the three algorithms on the standard PCA problem (4.3). The data generation process follows Section 4.4.2. We test our codes with different numbers of agents n and set $k = n/10$ as the number of clients we pick up for each round. We terminate the algorithms if the number of rounds of communication exceeds 600. We sample 10000 data points in $\mathbb{R}^{100}$ and partition them into n agents, each of which contains equal number of data. We test `RFedSVRG` with one iteration for each local agents, i.e. $\tau_i = 1$ and test `RFedAvg` and `RFedProx` with $\tau_i = 5$ iterations in (4.9). We use the constant stepsizes for all three algorithms, and take $\mu = n/10$ for each choice of n. The results are presented in Figure 4.2, from which we see that only `RFedSVRG` can efficiently decrease $\|\mathsf{grad} f(x_t)\|$ to an acceptable level.

Experiments on kPCA.. We now test the three algorithms on the kPCA problem (4.2). In the first experiment we sample 10000 data points in $\mathbb{R}^{200}$ and partition them into n agents, each of which contains equal number of data. We test our codes with different number of agents n, and again set $k = n/10$. Here we take $(d, r) = (200, 5)$. The results are given in Figure 4.3, where we see that `RFedSVRG` can efficiently decrease $\|\mathsf{grad} f(x_t)\|$ and the principal angle in all tested cases.



(a) $n = 500$ (b) $n = 1000$ (c) $n = 2500$

FIGURE 4.2. Results for PCA (4.3). The y-axis denotes $\|\mathsf{grad} f(x_t)\|$. For each figure, the experiments are repeated and averaged over 10 times.

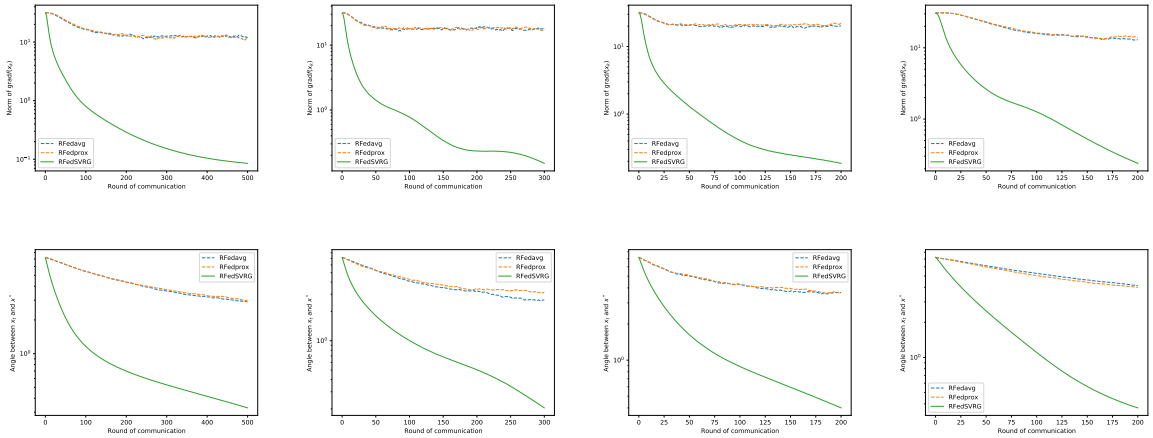


(a) $(n, k) = (50, 5)$ (b) $(n, k) = (100, 10)$ (c) $(n, k) = (500, 50)$ (d) $(n, k) = (1000, 100)$

FIGURE 4.3. Results for kPCA. The y-axis of the figures in the first row denotes $\|\mathsf{grad} f(x_t)\|$, and the y-axis of the figures in the second row denotes the principal angle between x_t and x^*. The experiments are repeated and averaged over 10 times.

In the second experiment we test the effect of the number of inner loops τ_i. We generate 10000 standard Gaussian vectors. We set $(d, r) = (200, 5)$, $k = 10$ and $n = 100$ so that $p = 100$. We choose $\tau = [1, 10, 50, 100]$ for the inner steps for all three algorithms. The results are presented in Figure 4.4. From this figure we again observe the great performance of `RFedSVRG`.

4.4.3. Experiments for kPCA on real data. We now show the numerical results of the three algorithms on real data. We focus on the kPCA problem (4.2) and three real data sets: the Iris dataset Forina et al. [1998], the wine dataset Forina et al. [1998] and the MNIST hand-written



(a) $\tau = 1$ (b) $\tau = 10$ (c) $\tau = 50$ (d) $\tau = 100$

FIGURE 4.4. Results for kPCA (4.2) with different number of inner loops $\tau = [1, 10, 50, 100]$. The y-axis of the figures in the first row denotes $\|\mathsf{grad} f(x_t)\|$, and the one in the second row denotes the principal angle between x_t and x^*. The experiments are repeated and averaged over 10 times.

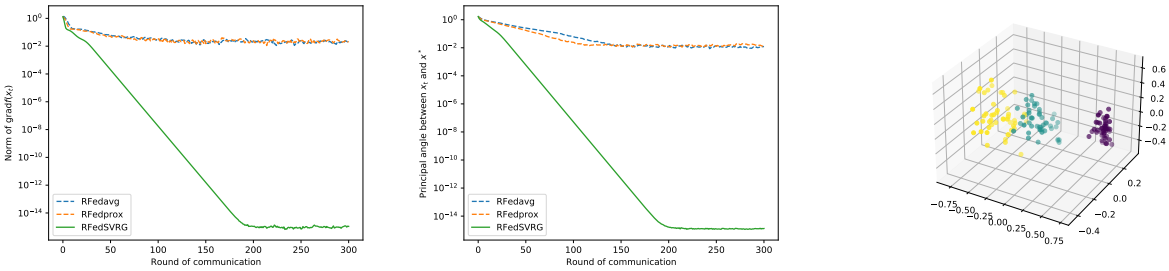
dataset LeCun et al. [1998]. For all three datasets, we calculate the first r principal directions and the true optimal loss value directly. We can thus compute the principal angles between the iterate and the ground truth. The experiments are repeated and averaged for 10 random initializations.

For the first two datasets, we randomly partition the datasets into 10 agents and at each iteration we take $k = 5$ agents. The Figures 4.5 and 4.6 show that `RFedSVRG` is able to effectively decrease the norm of Riemannian gradient and the principal angles while the other two are not as efficient. We also draw the scatter plots of the dataset toward the principal subspaces computed by `RFedSVRG`, which show that the algorithm indeed grasps the principal direction of the datasets.

For the MNIST hand-written dataset, the (training) dataset contains 60000 hand-written images of size 28×28, i.e. $d = 784$. This is a relatively large dataset and we test the proposed algorithms with different number of clients. The results are shown in Figure 4.7 where the efficiency of `RFedSVRG` is demonstrated again.

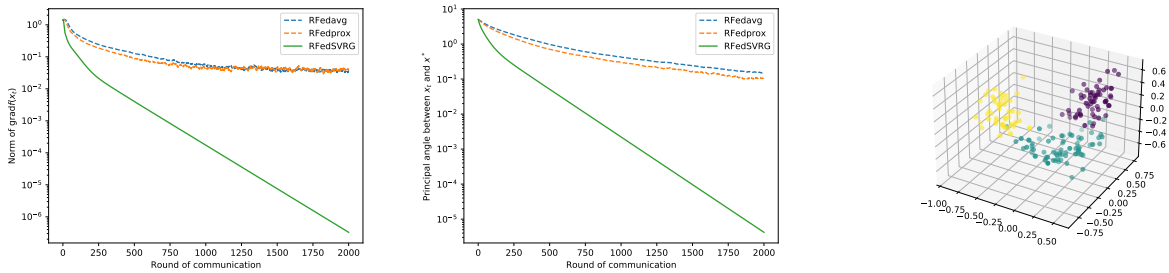
4.5. Conclusions

In this chapter, we studied the federated optimization over Riemannian manifolds. We proposed a Riemannian federated SVRG algorithm and analyzed its convergence rate to an ϵ-stationary point.



(a) Gradient norm (b) Principal angle (c) Scatter plot

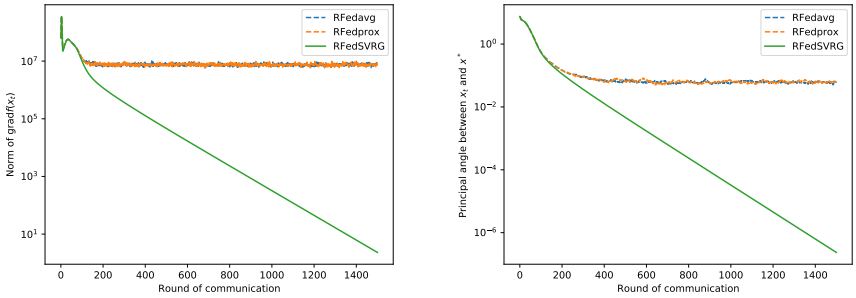
FIGURE 4.5. Results for kPCA (4.2) on Iris dataset. The data is in $\mathbb{R}^4$ ($d = 4$) and we take $r = 3$. The first figure is the norm of Riemannian gradient $\|\mathsf{grad} f(x_t)\|$ and the second is the principal angle between x_t and the true solution x^*, whereas the last figure is the scatter plot of projected data on to the subspace defined by the output of `RFedSVRG`.



(a) Gradient norm (b) Principal angle (c) Scatter plot

FIGURE 4.6. Results for kPCA (4.2) with wine dataset. The data is in $\mathbb{R}^{13}$ ($d = 13$) and we take $r = 3$. The first figure is the norm of Riemannian gradient $\|\mathsf{grad} f(x_t)\|$ and the second is the principal angle between x_t and the true solution x^*, whereas the last figure is still the scatter plot of projected data on to the subspace defined by the output of `RFedSVRG`.

To the best of our knowledge, this is the first federated algorithm over Riemannian manifolds with convergence guarantees. Numerical experiments on federated PCA and federated kPCA were conducted to demonstrate the efficiency of the proposed method. Developing algorithms with lower communication cost, better scalability and sparse solutions are some important topics for future research.



(a) Gradient norm (b) Principal angle

FIGURE 4.7. Results for kPCA (4.2) with MNIST dataset. The data is in $\mathbb{R}^{784}$ ($d = 784$) and we take $n = 200$ and $r = 5$. Fig (a) is the norm of Riemannian gradient $\mathsf{grad} f(x_t)$ and Fig (b) is the principal angle between x_t and the true solution x^*. We take $k = n/10$ and $\tau = 5$ for all algorithms.

CHAPTER 5

Riemannian Alternating Direction Method of Multipliers

5.1. Introduction

In this chapter we consider solving the nonsmooth manifold constrained problem in (1.4) which we generalize and restate here:

$$\begin{aligned} &\min_x F(x) := f(x) + g(Ax) \\ &\text{s.t. } x \in \mathcal{M}, \end{aligned} \tag{5.1}$$

where f is smooth and possibly nonconvex, g is nonsmooth but convex, $\mathcal{M}$ is an embedded submanifold in $\mathbb{R}^n$, and matrix $A \in \mathbb{R}^{m\times n}$. Throughout this paper, the smoothness, Lipschitz continuity, and convexity of functions are interpreted as the functions are being considered in the ambient Euclidean space. If $\mathcal{M} = \mathbb{R}^n$, then problem (5.1) reduces to the Euclidean case, and there exist efficient methods such as proximal gradient method, accelerated proximal gradient method, and ADMM for solving it. If the nonsmooth function vanishes, i.e., $g \equiv 0$, then problem (5.1) reduces to a smooth problem over manifold, and it can be solved by various methods for smooth Riemannian optimization. Therefore, the main challenge of solving (5.1) lies in the fact that there exist both manifold constraint and nonsmooth objective in the problem. As a result, a very natural idea to deal with this situation is to split the difficulty caused by the manifold constraint and nonsmooth objective. In particular, one can introduce an auxiliary variable y and rewrite (5.1) equivalently as

$$\begin{aligned} &\min_{x,y} f(x) + g(y) \\ &\text{s.t. } Ax = y,\ x \in \mathcal{M}. \end{aligned} \tag{5.2}$$

ADMM is a good candidate for solving (5.2), because it can deal with the nonsmooth objective and the manifold constraint separately and alternately. Here we point out that there exist many ADMM-like algorithms for problems with nonconvex objective [Jiang et al., 2019, Themelis and

Patrinos, 2020, Yang et al., 2017, Zhang et al., 2022]. However, these algorithms do not allow the constraint set to be nonconvex. Therefore, they do not apply to the case where manifold constraints are present. In the following, we give a brief literature review on ADMM-like algorithms that allow manifold constraint – a nonconvex constraint set.

The idea of splitting the nonsmooth objective and manifold constraint in (5.1) is not new. The first algorithm for this purpose is the SOC (splitting orthogonality constraints) algorithm proposed by Lai and Osher [2014]. SOC for solving (5.1) splits the problem in the following way:

$$\begin{aligned} &\min_{x,y} \ f(x) + g(Ax) \\ &\text{s.t. } x = y, \ y \in \mathcal{M}, \end{aligned} \tag{5.3}$$

and iterates as follows:

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x f(x) + g(Ax) + \langle \lambda^k, x - y^k \rangle + \frac{\rho}{2}\|x - y^k\|_2^2 \\ y^{k+1} &:= \operatorname*{argmin}_{y \in \mathcal{M}} \langle \lambda^k, x^{k+1} - y \rangle + \frac{\rho}{2}\|x^{k+1} - y\|_2^2 \\ \lambda^{k+1} &:= \lambda^k + \rho(x^{k+1} - y^{k+1}), \end{aligned} \tag{5.4}$$

where λ denotes the Lagrange multiplier and $\rho > 0$ is a penalty parameter. Note that the x-subproblem in (5.4) is an unconstrained problem, which can be solved by proximal gradient method and many others, and the y-subproblem corresponds to a projection onto the manifold $\mathcal{M}$. A closely related algorithm named MADMM (manifold ADMM), proposed in Kovnatsky et al. [2016] for solving (5.2), iterates as follows:

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_{x \in \mathcal{M}} f(x) + \langle \lambda^k, Ax - y^k \rangle + \frac{\rho}{2}\|Ax - y^k\|_2^2 \\ y^{k+1} &:= \operatorname*{argmin}_y g(y) + \langle \lambda^k, Ax^{k+1} - y \rangle + \frac{\rho}{2}\|Ax^{k+1} - y\|_2^2 \\ \lambda^{k+1} &:= \lambda^k + \rho(Ax^{k+1} - y^{k+1}). \end{aligned} \tag{5.5}$$

In (5.5), the x-subproblem is a Riemannian optimization with smooth objective which can be solved by Riemannian gradient method, and the y-subproblem corresponds to the proximal mapping of function g. However, there lacks convergence guarantees for both SOC and MADMM.

When the nonsmooth term in (5.1) vanishes, i.e., $g \equiv 0$, an ADMM for nonconvex optimization can be used to solve (5.1) as illustrated in Wang et al. [2019]. Since $g \equiv 0$, the problem (5.1) reduces to

$$\begin{aligned} &\min_{x,y} \ f(x) + I_{\mathcal{M}}(y) \\ &\text{s.t.,} \ x = y, \end{aligned} \tag{5.6}$$

where $I_{\mathcal{M}}$ is the indicator function of manifold $\mathcal{M}$. The ADMM for solving (5.6) iterates as follows:

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x f(x) + \langle \lambda^k, x - y^k \rangle + \frac{\rho}{2}\|x - y^k\|_2^2 \\ y^{k+1} &:= \operatorname*{argmin}_{y \in \mathcal{M}} \langle \lambda^k, x^{k+1} - y \rangle + \frac{\rho}{2}\|x^{k+1} - y\|_2^2 \\ \lambda^{k+1} &:= \lambda^k + \rho(x^{k+1} - y^{k+1}). \end{aligned} \tag{5.7}$$

The convergence of (5.7) is established in Wang et al. [2019] under the assumption that f is Lipschitz differentiable. Note that the convergence only applies when $g \equiv 0$. The ADMM studied in Wang et al. [2019] does not apply to (5.1) when the nonsmooth function g presents.

Another ADMM was proposed in Lu et al. [2018] for solving a particular smooth Riemannian optimization problem: the sparse spectral clustering. This problem can be cast below.

$$\begin{aligned} &\min_{P,U} \ \langle L, UU^\top \rangle + g(P), \\ &\text{s.t. } P = UU^\top, U^\top U = I, \end{aligned} \tag{5.8}$$

where L is a given matrix, g is a smooth function that promotes the sparsity of $UU^\top$. The ADMM for solving (5.8) iterates as follows.

$$\begin{aligned} U^{k+1} &:= \operatorname*{argmin}_{U^\top U = I} \langle L, UU^\top \rangle + \langle \Lambda^k, P^k - UU^\top \rangle + \frac{\rho}{2}\|P^k - UU^\top\|_F^2 \\ P^{k+1} &:= \operatorname*{argmin}_P g(P) + \langle \Lambda^k, P - U^{k+1}(U^{k+1})^\top \rangle + \frac{\rho}{2}\|P - U^{k+1}(U^{k+1})^\top\|_F^2 \\ \Lambda^{k+1} &:= \Lambda^k + \rho(P^{k+1} - U^{k+1}(U^{k+1})^\top). \end{aligned} \tag{5.9}$$

Note that the ADMM in Lu et al. [2018] requires the smoothness on the objective function as well, and it does not apply to the case where the objective function is nonsmooth. Zhang et al. [2020]

proposed a proximal ADMM which solves the following problem:

$$\begin{aligned} &\min\ f(x_1,\ldots,x_N)+\sum_{i=1}^{N-1} g_i(x_i) \\ &\text{s.t. } x_N = b-\sum_{i=1}^{N-1} A_i x_i \\ &x_i\in\mathcal{M}_i\cap\mathcal{X}_i, i=1,\ldots,N-1, \end{aligned} \tag{5.10}$$

where f is a smooth function, g_i is a nonsmooth function, $\mathcal{M}_i$ is a Riemannian manifold, and $\mathcal{X}_i$ is a convex set. The authors of Zhang et al. [2020] established the iteration complexity of the proposed proximal ADMM for obtaining an ϵ-stationary point of (5.10). A notable requirement in (5.10) is that the last block variable (i.e., x_N) must not appear in the nonsmooth part of the objective, nor be subject to manifold constraints. This is in sharp contrast to problem (5.2), where one block variable is associated with the manifold constraint, and the other block variable is associate with the nonsmooth part of the objective.

Other than ADMM-type algorithms, there also exist some other algorithms for solving (5.1). Here we briefly discuss two of them: Riemannian subgradient method and Riemannian proximal gradient method. Because the objective function of (5.1) is nonsmooth, it is a natural idea to use Riemannian subgradient method [Borckmans et al., 2014, Ferreira and Oliveira, 1998, Grohs and Hosseini, 2016a,b, Hosseini, 2015, Hosseini and Uschmajew, 2017, Hosseini et al., 2018, Li et al., 2021] to solve it. The Riemannian subgradient method for solving (5.1) updates the iterate by

$$x^{k+1}=\mathsf{Retr}_{x^k}(-\eta_k v^k),$$

where v^k is a Riemannian subgradient of F at $\mathcal{M}$, $\eta_k>0$ is a stepsize, and Retr denotes the retraction operation. Convergence of this method is established in Ferreira and Oliveira [1998] when F is geodesically convex, and iteration complexity is analyzed in Li et al. [2021] when F is weakly convex over the Stiefel manifold. Another representative algorithm for solving (5.1) is the manifold proximal gradient method (ManPG), which was proposed recently by Chen et al. [2020].

A typical iteration of ManPG is given below:

$$
\begin{aligned}
v^k &:= \operatorname*{argmin}_{v \in \mathrm{T}_{x^k}\mathcal{M}} \langle \mathsf{grad} f(x^k), v\rangle + \frac{1}{2t}\|v\|^2 + g(A(x^k+v)) \\
x^{k+1} &:= \mathsf{Retr}_{x^k}(\alpha v^k),
\end{aligned} \tag{5.11}
$$

where $t > 0$ and $\alpha > 0$ are stepsizes, $\mathrm{T}_x\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at x, and $\mathsf{grad} f$ denotes the Riemannian gradient of f. Chen et al. [2020] analyzed the iteration complexity of ManPG for obtaining an ϵ-stationary point of (5.1). Moreover, Chen et al. [2020] suggested to solve the subproblem for determining v_k in (5.11) by a semi-smooth Newton method [Chen et al., 2020, Xiao et al., 2018]. Huang and Wei [2022] extended ManPG to more general manifold and Huang and Wei [2019] also designed an accelerated ManPG that demonstrates superior numerical behaviour than the original ManPG. In a more recent work, Zhou et al. [2022] proposed an augmented Lagrangian method that solves the manifold constrained problems with nonsmooth objective. Note that similar to ManPG, the algorithms in Huang and Wei [2019, 2022], Zhou et al. [2022] all require solving a relatively difficult subproblem which needs to be solved by semi-smooth Newton algorithm. In this paper we do not need to deal with difficult subproblems – all steps of our algorithms are explicit and easy-to-compute.

Our contributions. In this paper, we propose a Riemannian ADMM (RADMM) for solving (5.2) based on a Moreau envelope smoothing technique. Our RADMM for solving (5.2) contains easily computable steps in each iteration. We analyze the iteration complexity of our RADMM for obtaining an ϵ-stationary point to (5.2) under mild assumptions. Existing ADMM for solving nonconvex problems either does not allow nonconvex constraint set, or does not allow nonsmooth objective function. In contrast, our complexity result is established for problems with simultaneous nonsmooth objective and manifold constraint. Numerical results of the proposed algorithm for solving sparse principal component analysis and dual principal component pursuit are reported, which demonstrate its superiority over existing methods.

Organizations. The rest of this paper is organized as follows. We propose our RADMM in Section 5.2, whose iteration complexity is analyzed in Section 5.3. Section 5.4 is devoted to applications and numerical experiments. We draw some concluding remarks in Section 5.5.

5.2. A Riemannian ADMM

We now introduce our RADMM algorithm. Our RADMM for solving (5.2) is based on the Moreau envelope smoothing technique. In particular, we consider to smooth the function g in (5.2) by adding a quadratic proximal term, which leads to:

$$\begin{aligned} &\min_{x,y,z} \ f(x) + g(y) + \frac{1}{2\gamma}\|y - z\|^2 \\ &\text{s.t. } Ax = z, \ x \in \mathcal{M}, \end{aligned} \tag{5.12}$$

where $\gamma > 0$ is a parameter. Equivalently, (5.12) can also be rewritten as

$$\begin{aligned} &\min_{x,z} \ f(x) + g_\gamma(z) \\ &\text{s.t. } Ax = z, \ x \in \mathcal{M}, \end{aligned} \tag{5.13}$$

where $g_\gamma(z) = \min_y \left\{ g(y) + \frac{1}{2\gamma}\|y - z\|^2 \right\}$ is the Moreau envelope of g Zeng et al. [2022], and it is known that g_γ is a smooth function when g is convex.

We need to point out that the idea of Moreau envelope smoothing has been proposed in Zeng et al. [2022] for solving the following problem in Euclidean space:

$$\min_x \ f(x) + g(x), \ \text{s.t.}, \ Ax = b, \tag{5.14}$$

where f is smooth and g is weakly convex with easily computable proximal mapping. In particular, the authors of Zeng et al. [2022] proposed an augmented Lagrangian method for solving the Mereau envelope smoothed problem of (5.14). We apply the same idea of Moreau envelope smoothing and design our RADMM algorithm.

We define the augmented Lagrangian function of (5.13) as:

$$\mathcal{L}_{\rho,\gamma}(x, z; \lambda) = f(x) + g_\gamma(z) + \langle \lambda, Ax - z \rangle + \frac{\rho}{2}\|Ax - z\|^2. \tag{5.15}$$

A direct application of ADMM for solving (5.13) yields the following updating scheme:

$$
\begin{aligned}
x^{k+1} &:= \operatorname*{argmin}_{x\in\mathcal{M}} \ \mathcal{L}_{\rho,\gamma}(x, z^k; \lambda^k) \\
z^{k+1} &:= \operatorname*{argmin}_{z} \ \mathcal{L}_{\rho,\gamma}(x^{k+1}, z; \lambda^k) \\
\lambda^{k+1} &:= \lambda^k + \rho(Ax^{k+1} - z^{k+1}).
\end{aligned} \tag{5.16}
$$

Now since the x-subproblem in (5.16) is usually not easy to solve, we propose to replace it with a Riemannian gradient step, and this leads to our RADMM, which iterates as follows:

$$
\begin{aligned}
x^{k+1} &:= \mathsf{Retr}_{x^k}(-\eta_k \mathsf{grad}_x \mathcal{L}_{\rho,\gamma}(x^k, z^k; \lambda^k)) \\
z^{k+1} &:= \operatorname*{argmin}_{z} \ \mathcal{L}_{\rho,\gamma}(x^{k+1}, z; \lambda^k) \\
\lambda^{k+1} &:= \lambda^k + \rho(Ax^{k+1} - z^{k+1}),
\end{aligned} \tag{5.17}
$$

where $\eta_k > 0$ is a stepsize, and $\mathsf{grad}_x \mathcal{L}_{\rho,\gamma}$ denotes the Riemannian gradient of $\mathcal{L}_{\rho,\gamma}$ with respect to x. The remaining thing is to discuss how to solve the z-subproblem in (5.17). It turns out that it is closely related to the proximal mapping of function g, and can be easily solved as long as the proximal mapping of g can be easily evaluated, as shown in the following lemma.

LEMMA 5.2.1. *The solution of the z-subproblem in* (5.17) *is given by*

$$
z^{k+1} := \frac{\gamma}{1+\gamma\rho}\left(\frac{1}{\gamma}y^{k+1} + \lambda^k + \rho A x^{k+1}\right), \tag{5.18}
$$

where

$$
y^{k+1} := \operatorname{prox}_{\frac{1+\rho\gamma}{\rho}g}\left(Ax^{k+1} + \frac{1}{\rho}\lambda^k\right), \tag{5.19}
$$

where prox_h *denotes the proximal mapping of function h, which is defined as*

$$
\operatorname{prox}_h(u) = \operatorname*{argmin}_{v} \ h(v) + \frac{1}{2}\|u - v\|_2^2.
$$

PROOF. The z-subproblem in (5.17) can be equivalently rewritten as

$$
(z^{k+1}, y^{k+1}) := \operatorname*{argmin}_{z,y} \ g(y) + \frac{1}{2\gamma}\|y - z\|^2 + \langle \lambda^k, Ax^{k+1} - z\rangle + \frac{\rho}{2}\|Ax^{k+1} - z\|^2. \tag{5.20}
$$

The optimality conditions of (5.20) are given by

$$0 = \frac{1}{\gamma}(z^{k+1} - y^{k+1}) - \lambda^k + \rho(z^{k+1} - Ax^{k+1}), \tag{5.21a}$$

$$0 \in \partial g(y^{k+1}) + \frac{1}{\gamma}(y^{k+1} - z^{k+1}). \tag{5.21b}$$

It is easy to see that (5.21a) immediately yields (5.18). Plugging (5.18) into (5.21b) gives

$$0 \in \frac{1+\gamma\rho}{\rho}\partial g(y^{k+1}) + y^{k+1} - \left(Ax^{k+1} + \frac{\lambda^k}{\rho}\right),$$

which implies

$$y^{k+1} = \operatorname*{argmin}_y \frac{1+\gamma\rho}{\rho} g(y) + \frac{1}{2}\left\| y - \left(Ax^{k+1} + \frac{\lambda^k}{\rho}\right)\right\|_2^2,$$

i.e., (5.19) holds.

□

Our RADMM for solving (5.2) can therefore be summarized as in Algorithm 10. We can see that all the steps can be easily computed and implemented.

Algorithm 10: A Riemannian ADMM

Given $(x^0, z^0; \lambda^0)$, stepsize $\eta_k > 0$, parameters $\rho > 0$ and $\gamma > 0$;

for $k = 0, 1, \ldots$ **do**

- Update $x^{k+1} := \mathsf{Retr}_{x^k}(-\eta_k \mathsf{grad}_x \mathcal{L}_{\rho,\gamma}(x^k, z^k; \lambda^k))$;
- Update $y^{k+1} := \operatorname{prox}_{\frac{1+\rho\gamma}{\rho} g}\left(Ax^{k+1} + \frac{1}{\rho}\lambda^k\right)$;
- Update $z^{k+1} := \frac{\gamma}{1+\gamma\rho}\left(\frac{1}{\gamma}y^{k+1} + \lambda^k + \rho A x^{k+1}\right)$;
- Update $\lambda^{k+1} := \lambda^k + \rho(Ax^{k+1} - z^{k+1})$.

5.3. Convergence Analysis

In this section, we analyze the iteration complexity of Algorithm 10 for obtaining an ϵ-stationary point of (5.2).

The following assumption is needed in the analysis.

ASSUMPTION 5.3.1. *We assume f, g and $\mathcal{M}$ in (5.2) satisfy the following conditions.*

(1) $\mathcal{M} \subset \mathbb{R}^n$ *is a compact and complete Riemannian manifold embedded in Euclidean space* $\mathbb{R}^n$ *with diameter* D*;*

(2) ∇f *is Lipschitz continuous with Lipschitz constant* L *in the ambient space* $\mathbb{R}^n$*;*

(3) g *is convex and Lipschitz continuous with Lipschitz constant* L_g *in the ambient space* $\mathbb{R}^m$*.*

Also since $\mathcal{M}$ is compact and ∇f is continuous, we can denote the maximum of the norm of f as a constant M, i.e.,

$$\|\nabla f(x)\| \leq M, \ \forall x \in \mathcal{M}. \tag{5.22}$$

Now we proceed to study the optimality of the problem (5.2). First, we note that the first-order optimality conditions of (5.2) are given by (see, e.g., Yang et al. [2014]):

$$\begin{aligned} 0 &= \mathsf{grad}_x \mathcal{L}(x^*, y^*, \lambda^*) = \mathsf{proj}_{\mathrm{T}_{x^*}\mathcal{M}} \left(\nabla f(x^*) + A^\top \lambda^* \right), \\ 0 &\in \partial_y \mathcal{L}(x^*, y^*, \lambda^*) = \partial g(y^*) - \lambda^*, \\ 0 &= Ax^* - y^*, \\ x^* &\in \mathcal{M}, \end{aligned} \tag{5.23}$$

where the Lagrangian function of (5.2) is defined as

$$\mathcal{L}(x, y, \lambda) := f(x) + g(y) + \langle \lambda, Ax - y \rangle.$$

Based on this, we can define the ϵ-stationary point of (5.2) as follows.

Definition 5.3.1. *For* (x, y, λ) *with* $x \in \mathcal{M}$*, denote*

$$\partial \mathcal{L}(x, y, \lambda) := \begin{bmatrix} \mathsf{proj}_{\mathrm{T}_x \mathcal{M}} \left(\nabla f(x) + A^\top \lambda \right) \\ \partial g(y) - \lambda \\ Ax - y \end{bmatrix}.$$

Then $(\bar{x}, \bar{y}, \bar{\lambda})$ *with* $\bar{x} \in \mathcal{M}$ *is called an* ϵ*-stationary point of* (5.2) *if there exists* $G \in \partial \mathcal{L}(\bar{x}, \bar{y}, \bar{\lambda})$ *such that* $\|G\|_2 \leq \epsilon$*.*

Before we present our main convergence results, we need the following lemmas. The first one is a brief recap of the properties of Moreau envelope (see e.g. Beck [2017] Chapter 6).

LEMMA 5.3.1 (Properties of Moreau envelope). *Suppose g is a L_g Lipschitz continuous and convex function. The Moreau envelope $g_\gamma(z) := \min_y g(y) + \frac{1}{2\gamma}\|z-y\|^2$ satisfies the following:*

(1) $0 \le g(z) - g_\gamma(z) \le \gamma L_g^2$;

(2) $\nabla g_\gamma(z) = \frac{1}{\gamma}(z - prox_{\gamma g}(z))$;

(3) $g_\gamma(z)$ *is* L_g *Lipschitz continuous;*

(4) $g_\gamma(z)$ *is* $1/\gamma$ *Lipschitz smooth, i.e.* $\nabla g_\gamma(z)$ *is Lipschitz continuous with parameter* $1/\gamma$.

Now we proceed to bound the difference of dual sequence by the primal sequence.

LEMMA 5.3.2 (Bound dual by primal). *For the updates of Algorithm 10, we have:*

$$\|\lambda^{k+1} - \lambda^k\| \le \frac{1}{\gamma}\|z^{k+1} - z^k\|. \tag{5.24}$$

PROOF. Note that the optimality conditions of the z-subproblem in (5.17) is given by:

$$\nabla g_\gamma(z^{k+1}) - \lambda^k + \rho(z^{k+1} - Ax^{k+1}) = 0, \tag{5.25}$$

which, together with the λ update in (5.17), yields

$$\nabla g_\gamma(z^{k+1}) = \lambda^{k+1}. \tag{5.26}$$

The desired result (5.24) follows from Lemma 5.3.1. □

We now provide the smoothness notion over manifolds, which is also known as Lipschitz-type gradient for pullbacks.

DEFINITION 5.3.2. *[Boumal et al., 2018] Function f is called L_1-geodesic smooth on complete Riemannian manifold $\mathcal{M}$ if $\forall x \in \mathcal{M}$ and $\forall v \in \mathrm{T}_x\,\mathcal{M}$, it holds that*

$$f(\mathsf{Retr}_x(v)) \le f(x) + \langle \mathsf{grad} f(x), v\rangle + \frac{L_1}{2}\|v\|^2. \tag{5.27}$$

The following lemma is from Boumal et al. [2018], which bridges the smoothness on the manifold with the smoothness in the ambient Euclidean space.

LEMMA 5.3.3. *[Boumal et al., 2018] Suppose $\mathcal{M} \in \mathbb{E}$ is a compact and complete Riemannian manifold embedded in Euclidean space $\mathbb{E}$ and f is L-Lipschitz smooth in $\mathbb{E}$, then f is also L_1-geodesic*

smooth, where L_1 is determined by the manifold $\mathcal{M}$ and f. Specifically, it can be shown (see Boumal et al. [2018]) that there exist positive constants α and β so that $\forall x \in \mathcal{M}$ and $\forall \eta \in \mathrm{T}_x \mathcal{M}$,

$$\|\mathsf{Retr}_x(\eta) - x\| \leq \alpha\|\eta\|, \text{ and } \|\mathsf{Retr}_x(\eta) - x - \eta\| \leq \beta\|\eta\|^2.$$

As a result, it can be shown that

$$L_1 = \frac{L}{2}\alpha^2 + M\beta,$$

where M is the upper bound of the gradient, which is defined in (5.22).

Now we are ready to present the smoothness of the augmented Lagrangian function (5.15).

LEMMA 5.3.4. *For any $\{(z^k, \lambda^k)\}$ generated in Algorithm 10, the augmented Lagrangian function $\mathcal{L}_{\rho,\gamma}(x, z^k, \lambda^k)$ defined in* (5.15) *is L_ρ-geodesic smooth with respect to $x \in \mathcal{M}$, where*

$$L_\rho = \frac{L + \rho\|A^\top A\|_2}{2}\alpha^2 + (M + \|A\|_2 L_g + \rho\|A^\top A\|_2 D + \|A\|_2(2L_g + \rho\|A\|_2 D))\beta, \tag{5.28}$$

and $\|B\|_2$ denotes the spectral norm of matrix B.

PROOF. We first show that $\{z^k\}$, $\{\lambda^k\}$, $k = 0, 1, \ldots$, generated in Algorithm 10 are uniformly bounded. Note that from (5.26), we have

$$\|\lambda^k\| = \|\nabla g_\gamma(z^k)\| \leq L_g, \tag{5.29}$$

where the inequality follows from the facts that g is L_g-Lipschitz continuous (Assumption 5.3.1) and Lemma 5.3.1.

From the update of λ^{k+1}, i.e., $\lambda^{k+1} := \lambda^k + \rho(Ax^{k+1} - z^{k+1})$, we have

$$z^{k+1} = (\lambda^k - \lambda^{k+1})/\rho + Ax^{k+1},$$

which, together with (5.29) and Assumption 5.3.1, immediately implies

$$\|z^{k+1}\| \leq \frac{2L_g}{\rho} + \|A\|_2 D. \tag{5.30}$$

We now show that the gradient of $\mathcal{L}_{\rho,\gamma}(x, z^k, \lambda^k)$, i.e., $\nabla_x \mathcal{L}_{\rho,\gamma}(x, z^k, \lambda^k) = \nabla f(x) + A^\top \lambda^k + \rho A^\top (Ax - z^k)$, is uniformly upper bounded $\forall x \in \mathcal{M}$. To this end, we note that

$$\begin{aligned} \|\nabla_x \mathcal{L}_{\rho,\gamma}(x, z^k, \lambda^k)\| \leq & \|\nabla f(x)\| + \|A^\top \lambda^k\| + \rho \|A^\top (Ax - z^k)\| \\ \leq & \|\nabla f(x)\| + \|A\|_2 \|\lambda^k\| + \rho \|A^\top A\|_2 \|x\| + \rho \|A\|_2 \|z^k\| \\ \leq & M + \|A\|_2 L_g + \rho \|A^\top A\|_2 D + \|A\|_2 (2L_g + \rho \|A\|_2 D), \end{aligned} \tag{5.31}$$

where the last inequality is due to (5.29), (5.30) and Assumption 5.3.1.

Moreover, we have

$$\begin{aligned} \|\nabla_x \mathcal{L}_{\rho,\gamma}(x_1, z^k, \lambda^k) - \nabla_x \mathcal{L}_{\rho,\gamma}(x_2, z^k, \lambda^k)\| \leq & \|\nabla f(x_1) - \nabla f(x_2)\| + \rho \|A^\top A(x_1 - x_2)\| \\ \leq & L\|x_1 - x_2\| + \rho \|A^\top A\|_2 \|x_1 - x_2\|. \end{aligned} \tag{5.32}$$

By applying Lemma 5.3.3 together with (5.31) and (5.32), we immediately obtain the desired result. □

Now we give the following lemma regarding the decrease of the augmented Lagrangian function $\mathcal{L}_{\rho,\gamma}$.

LEMMA 5.3.5. *For the sequence $\{(x^k, z^k, \lambda^k)\}$ generated in Algorithm 10, we have:*

$$\begin{aligned} & \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^{k+1}) - \mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k) \\ \leq & \left(\frac{1}{\rho\gamma^2} - \frac{\rho}{2}\right) \|z^{k+1} - z^k\|^2 - \left(\frac{1}{\eta_k} - \frac{L_\rho}{2}\right) \|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2, \end{aligned} \tag{5.33}$$

where L_ρ is defined in (5.28).

PROOF. First, we have

$$\begin{aligned} & \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^{k+1}) - \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^k) \\ = & \langle \lambda^{k+1} - \lambda^k, Ax^{k+1} - z^{k+1} \rangle \\ = & \frac{1}{\rho} \|\lambda^{k+1} - \lambda^k\|^2 \leq \frac{1}{\rho\gamma^2} \|z^{k+1} - z^k\|^2, \end{aligned} \tag{5.34}$$

where the inequality is from Lemma 5.3.2.

Second, we have,

$$
\begin{aligned}
&\mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^k) - \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^k, \lambda^k) \\
=&g_\gamma(z^{k+1}) - g_\gamma(z^k) + \langle \lambda^k, z^k - z^{k+1} \rangle + \frac{\rho}{2}(\|Ax^{k+1} - z^{k+1}\|^2 - \|Ax^{k+1} - z^k\|^2) \\
=&g_\gamma(z^{k+1}) - g_\gamma(z^k) + \langle \lambda^k + \rho(Ax^{k+1} - z^{k+1}), z^k - z^{k+1} \rangle - \frac{\rho}{2}\|z^{k+1} - z^k\|^2 \\
\leq& -\frac{\rho}{2}\|z^{k+1} - z^k\|^2,
\end{aligned} \tag{5.35}
$$

where the inequality is by convexity of g_γ and $\nabla g_\gamma(z^{k+1}) = \lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} - z^{k+1})$.

Third, by Lemma 5.3.4 and (5.27), we obtain

$$
\begin{aligned}
&\mathcal{L}_{\rho,\gamma}(x^{k+1}, z^k, \lambda^k) - \mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k) \\
\leq& \langle \mathsf{grad}_x \mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k), \mathsf{Retr}_{x^k}^{-1}(x^{k+1}) \rangle + \frac{L_\rho}{2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 \\
=& -\left(\frac{1}{\eta_k} - \frac{L_\rho}{2}\right)\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2,
\end{aligned} \tag{5.36}
$$

where the equality follows from the x-update in Algorithm 10.

Combining (5.34), (5.35), and (5.36) yields the desired result (5.33).

□

The following lemma shows that the augmented Lagrangian function $\mathcal{L}_{\rho,\gamma}$ is lower bounded.

LEMMA 5.3.6. *If $\rho\gamma \geq 1$, then the sequence $\{\mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k)\}$ is uniformly lower bounded by $F^* - \gamma L_g^2$, where F^* is the optimal value of* (5.1).

PROOF. By the $1/\gamma$ Lipschitz smoothness of g_γ (see Lemma 5.3.1) and $\nabla g_\gamma(z^k) = \lambda^k$, we get

$$
g_\gamma(Ax) \leq g_\gamma(z) + \langle \nabla g_\gamma(z), Ax - z \rangle + \frac{1}{2\gamma}\|Ax - z\|^2,
$$

which implies

$$\begin{aligned}\mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k) &= f(x^k) + g_\gamma(z^k) + \langle \lambda^k, Ax^k - z^k\rangle + \frac{\rho}{2}\|Ax^k - z^k\|^2 \\ &\geq f(x^k) + g_\gamma(Ax^k) + \left(\frac{\rho}{2} - \frac{1}{2\gamma}\right)\|Ax^k - z^k\|^2 \\ &\geq f(x^k) + g_\gamma(Ax^k) \\ &\geq f(x^k) + g(Ax^k) - \gamma L_g^2 \\ &\geq F^* - \gamma L_g^2,\end{aligned}$$

where the third inequality follows from Lemma 5.3.1. □

The following lemma gives an upper bound for $G^k \in \partial\mathcal{L}(x^k, y^k, \lambda^k)$.

LEMMA 5.3.7. *Denote the iterates of Algorithm 10 by* $\{(x^k, y^k, z^k, \lambda^k)\}$. *There exists* $G^k \in \partial\mathcal{L}(x^k, y^k, \lambda^k)$, $\forall k \geq 1$, *as defined in Definition 5.3.1, such that:*

$$\|G^k\|^2 \leq \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + \frac{2(\rho^2\|A\|_2^2 + 1)}{\rho^2\gamma^2}\|z^k - z^{k-1}\|^2 + 2\gamma^2 L_g^2.$$

PROOF. From (5.21a), (5.21b) and the update of λ^{k+1} in Algorithm 10, we know that $\lambda^k \in \partial g(y^k)$ for $k = 1, 2, \ldots$. Therefore, there exist $G^k \in \partial\mathcal{L}(x^k, y^k, \lambda^k)$ such that

$$\begin{aligned}\|G^k\|^2 =& \|\mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}\left(\nabla f(x^k) + A^\top\lambda^k\right)\|^2 + \|Ax^k - y^k\|^2 \\ \leq& \|\mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}\left(\nabla f(x^k) + A^\top\lambda^k\right)\|^2 + 2\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2.\end{aligned}$$

Now from the x update of Algorithm 10, we know that

$$\mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}\left(\nabla f(x^k) + A^\top\lambda^k\right) = -\frac{1}{\eta_k}\mathsf{Retr}_{x^k}^{-1}(x^{k+1}) - \mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}(\rho A^\top(Ax^k - z^k)).$$

Therefore, we have

$$\begin{aligned}\|G^k\|^2 \leq & \left\| \frac{1}{\eta_k} \mathsf{Retr}_{x^k}^{-1}(x^{k+1}) + \mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}(\rho A^\top (Ax^k - z^k)) \right\|^2 + 2\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2 \\ \leq & \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + 2\rho^2 \|\mathsf{proj}_{\mathrm{T}_{x^k}\mathcal{M}}(A^\top (Ax^k - z^k))\|^2 + 2\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2 \\ \leq & \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + 2\rho^2 \|A\|_2^2 \|Ax^k - z^k\|^2 + 2\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2 \\ = & \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + 2(\rho^2 \|A\|_2^2 + 1)\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2.\end{aligned}$$

Now by the update of λ^k in Algorithm 10 and (5.24) we have $\rho\|Ax^k - z^k\| = \|\lambda^k - \lambda^{k-1}\| \leq \frac{1}{\gamma}\|z^k - z^{k-1}\|$. By (5.21b) we have $z^k - y^k \in \gamma \partial g(y^k)$ so that $\|z^k - y^k\| \leq \gamma L_g$. Combining these results we get

$$\begin{aligned}\|G^k\|^2 \leq & \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + 2(\rho^2 \|A\|_2^2 + 1)\|Ax^k - z^k\|^2 + 2\|z^k - y^k\|^2 \\ \leq & \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\|^2 + \frac{2(\rho^2 \|A\|_2^2 + 1)}{\rho^2\gamma^2}\|z^k - z^{k-1}\|^2 + 2\gamma^2 L_g^2,\end{aligned}$$

which gives the desired result. □

Finally, we have the following convergence result for Algorithm 10.

THEOREM 5.3.1. *Denote the iterates of Algorithm 10 by* $\{(x^k, y^k, z^k, \lambda^k)\}$. *For a given tolerance* $\epsilon > 0$, *we set* $\rho = 1/\epsilon$, $\gamma = \sqrt{\frac{2}{\rho^2} + \frac{\rho^2\|A\|_2^2+1}{\rho^3 L_\rho}} = \mathcal{O}(\epsilon)$, *also* $\eta_k = \eta = \frac{1}{L_\rho}$. *Note that our choices of* ρ *and* γ *guarantees that* $\rho\gamma > 1$. *Then there exist* $G^k \in \partial\mathcal{L}(x^k, y^k, \lambda^k)$, $k = 1, 2, \ldots$, *such that*

$$\min_{k=1,\ldots,K} \|G^k\|^2 \leq \epsilon^2,$$

provided that

$$K = \mathcal{O}\left(\frac{1}{\epsilon^4}\right).$$

That is, Algorithm 10 generates an ϵ*-stationary point to Problem* (5.2) *in* $\mathcal{O}(\epsilon^{-4})$ *iterations.*

PROOF. From Lemma 5.3.7, there exist $G^k \in \partial\mathcal{L}(x^k, y^k, \lambda^k)$, $k = 1, 2, \ldots$ such that

$$\|G^k\|^2 \leq \frac{2}{\eta_k^2}\|\mathsf{Retr}_{x^k}^{-1}(x^{k+1})\| + \frac{2(\rho^2 \|A\|_2^2 + 1)}{\rho^2\gamma^2}\|z^k - z^{k-1}\|^2 + 2\gamma^2 L_g^2,$$

which, combining with (5.33) and $\eta_k = 1/L_\rho$, yields

$$\begin{aligned}\|G^k\|^2 \leq & \frac{4}{\eta_k}\left(\mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k) - \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^{k+1})\right) \\ & + \left(\frac{2(\rho^2\|A\|_2^2+1)}{\rho^2\gamma^2}\|z^k - z^{k-1}\|^2 - \frac{4}{\eta_k}\left(\frac{\rho}{2} - \frac{1}{\rho\gamma^2}\right)\|z^k - z^{k+1}\|^2\right) + 2\gamma^2 L_g^2.\end{aligned}$$

Now by taking γ, ρ and $\eta_k = \eta$ as described in the theorem, it is easy to verify that

$$\frac{2(\rho^2\|A\|_2^2+1)}{\rho^2\gamma^2} \leq \frac{4}{\eta_k}\left(\frac{\rho}{2} - \frac{1}{\rho\gamma^2}\right).$$

Therefore, we have

$$\begin{aligned}\|G^k\|^2 \leq & \frac{4}{\eta_k}\left(\mathcal{L}_{\rho,\gamma}(x^k, z^k, \lambda^k) - \mathcal{L}_{\rho,\gamma}(x^{k+1}, z^{k+1}, \lambda^{k+1})\right) \\ & + \left(\frac{4}{\eta_k}\left(\frac{\rho}{2} - \frac{1}{\rho\gamma^2}\right)\|z^k - z^{k-1}\|^2 - \frac{4}{\eta_k}\left(\frac{\rho}{2} - \frac{1}{\rho\gamma^2}\right)\|z^k - z^{k+1}\|^2\right) + 2\gamma^2 L_g^2.\end{aligned}$$

Now by summing this inequality over $k = 1, \ldots, K$ and using Lemma 5.3.6, we get

$$\frac{1}{K}\sum_{k=1}^{K}\|G^k\|^2 \leq \frac{4}{\eta K}(\mathcal{L}_{\rho,\gamma}(x^1, z^1, \lambda^1) - F^* + \gamma L_g^2) + \frac{2\rho}{\eta K}\|z^1 - z^0\|^2 + 2\gamma^2 L_g^2.$$

Since we take $\gamma = \mathcal{O}(\epsilon)$, $\rho = \frac{1}{\epsilon}$ and $\eta = \frac{1}{L_\rho} = \mathcal{O}(\epsilon)$, to ensure $\min_{k=1,\ldots,K}\|G^k\|^2 \leq \epsilon^2$, we need $K = \mathcal{O}(\frac{1}{\epsilon^4})$. □

5.4. Applications and Numerical Experiments

Problem (5.1) finds many applications in machine learning, statistics and signal processing. For example, K-means clustering [Carson et al., 2017], sparse spectral clustering [Lu et al., 2018, Park and Zhao, 2018], and orthogonal dictionary learning [Demanet and Hand, 2014, Qu et al., 2016, Spielman et al., 2012, Sun et al., 2017a,b] are all of the form of (5.1). In this section, we present two representative applications of (5.1) and then report the numerical results of our Algorithm 10 for solving them.

Example 1. Sparse Principal Component Analysis (PCA). Principal Component Analysis, proposed by Pearson [1901] and later developed by Hotelling [1933], is one of the most fundamental statistical tools in analyzing high-dimensional data. Sparse PCA seeks principal components with very few nonzero components. For given data matrix $A \in \mathbb{R}^{m\times n}$, the sparse PCA that seeks

the leading p ($p < \min\{m, n\}$) sparse loading vectors can be formulated as

$$\begin{aligned} \min_X \ & F(X) := -\frac{1}{2}\mathrm{Tr}(X^\top A^\top A X) + \mu\|X\|_1 \\ \text{s.t. } & X \in \mathrm{St}(n, p), \end{aligned} \tag{5.37}$$

where $\mathrm{Tr}(Y)$ denotes the trace of matrix Y, the ℓ_1 norm is defined as $\|X\|_1 = \sum_{ij} |X_{ij}|$, $\mu > 0$ is a weighting parameter. This is the original formulation of sparse PCA as proposed by Jolliffe et al. [2003], where the model is called SCoTLASS and imposes sparsity and orthogonality to the loading vectors simultaneously. When $\mu = 0$, (5.37) reduces to computing the leading p eigenvalues and the corresponding eigenvectors of $A^\top A$. When $\mu > 0$, the ℓ_1 norm $\|X\|_1$ can promote sparsity of the loading vectors. There are many numerical algorithms for solving (5.37) when $p = 1$. In this case, (5.37) is relatively easy to solve because X reduces to a vector and the constraint set reduces to a sphere. However, there has been very limited literature for the case $p > 1$. Existing works, including d'Aspremont et al. [2007], Journee et al. [2010], Ma [2013], Shen and Huang [2008], Zou et al. [2006], do not impose orthogonal loading directions. As discussed in Journee et al. [2010], "Simultaneously enforcing sparsity and orthogonality seems to be a hard (and perhaps questionable) task." We refer the interested reader to Zou and Xue [2018] for more details on existing algorithms for solving sparse PCA.

Example 2. Orthogonal Dictionary Learning (ODL) and Dual principal component pursuit (DPCP). In ODL, one is given a set of p ($p \gg n$) data points $\mathbf{y}_1, \ldots, \mathbf{y}_p \in \mathbb{R}^n$ and aims to find an orthonormal basis of $\mathbb{R}^n$ to represent them compactly. In other words, by letting $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_p] \in \mathbb{R}^{n\times p}$, we want to find an orthogonal matrix $X \in \mathbb{R}^{n\times n}$ and a sparse matrix $A \in \mathbb{R}^{n\times p}$ such that $Y = XA$. Since X is orthogonal, we know that $A = X^\top Y$. This naturally leads to the following matrix version of ODL [Demanet and Hand, 2014, Qu et al., 2016, Spielman et al., 2012, Sun et al., 2017a,b]:

$$\begin{aligned} \min_X \ & \|Y^\top X\|_1 \\ \text{s.t. } & X \in \mathrm{St}(n, n). \end{aligned} \tag{5.38}$$

Here, the ℓ_1 norm is used to promote the sparsity of $A = X^\top Y$, and the constraint set $\mathrm{St}(n, n)$ is known as the orthogonal group, which is a special case of the Stiefel manifold.

Another representative application of (5.38) is robust subspace recovery (RSR) [Lerman and Maunu, 2018b, Lerman et al., 2015, Maunu et al., 2019, 2022]. RSR aims to fit a linear subspace to a dataset corrupted by outliers, which is a fundamental problem in machine learning and data mining. RSR can be described as follows. Given a dataset $Y = [\mathcal{X}, \mathcal{O}]\Gamma \in \mathbb{R}^{n\times(p_1+p_2)}$, where $\mathcal{X} \in \mathbb{R}^{n\times p_1}$ are inlier points spanning a d-dimensional subspace $\mathcal{S}$ of $\mathbb{R}^n$ ($d < p_1$), $\mathcal{O} \in \mathbb{R}^{n\times p_2}$ are outlier points without linear structure, and $\Gamma \in \mathbb{R}^{(p_1+p_2)\times(p_1+p_2)}$ is an unknown permutation, the goal is to recover the inlier space $\mathcal{S}$, or equivalently, to cluster the points into inliers and outliers. For a more comprehensive review of RSR, see the recent survey paper by Lerman and Maunu [2018a]. The dual principal component pursuit (DPCP) is a recently proposed approach to RSR that seeks to learn recursively a basis for the orthogonal complement $\mathcal{S}$ by solving (5.38) when X reduces to a vector, i.e.,

$$\min_{\bar{x}\in\mathbb{R}^n} \ f(\bar{x}) := \|Y^\top \bar{x}\|_1 \quad \text{s.t.} \quad \|\bar{x}\|_2 = 1, \tag{5.39}$$

The idea of DPCP is to first compute a normal vector $\bar{x}$ to a hyperplane $\mathcal{H}$ that contains all inliers $\mathcal{X}$. As outliers are not orthogonal to $\bar{x}$ and the number of outliers is known to be small, the normal vector $\bar{x}$ can be found by solving (5.39). It is shown in Tsakiris and Vidal [2018], Zhu et al. [2018] that under certain conditions, solving (5.39) indeed yields a vector that is orthogonal to $\mathcal{S}$, given that the number of outliers p_2 is at most on the order of $O(p_1^2)$. If d is known, then one can recover $\mathcal{S}$ as the intersection of the $p := n - d$ orthogonal hyperplanes that contain $\mathcal{X}$, which amounts to solving the following matrix optimization problem:

$$\min_{X\in\mathbb{R}^{n\times(n-d)}} \|Y^\top X\|_1 \quad \text{s.t.} \quad X^\top X = I_{n-d}. \tag{5.40}$$

Note that (5.37)-(5.40) are all in the form of (5.1).

5.4.1. Numerical Experiments on Sparse PCA. In this subsection, we conduct experiments to test the performance of our Riemannian ADMM for solving sparse PCA (5.37), and compare it with the performance of ManPG Chen et al. [2020] and Riemannian subgradient method Ferreira and Oliveira [1998], Li et al. [2021]. To apply Riemannian ADMM, we first rewrite (5.37)

as:

$$\begin{aligned} \min_{X,Y} \ & -\frac{1}{2}\mathrm{Tr}(X^\top A^\top AX) + \mu\|Y\|_1 \\ & \text{s.t. } X = Y,\ X \in \mathrm{St}(n,p). \end{aligned} \tag{5.41}$$

Now we see that the nonsmooth function $\|\cdot\|_1$ and the manifold constraint are associated with different variables. Thus, the two difficult terms are separated. Using the Moreau envelope smoothing, the smoothed problem of (5.41) is given by:

$$\begin{aligned} \min_{X,Z} \ & -\frac{1}{2}\mathrm{Tr}(X^\top A^\top AX) + g_\gamma(Z) \\ & \text{s.t. } X = Z,\ X \in \mathrm{St}(n,p), \end{aligned} \tag{5.42}$$

where $g_\gamma(Z) := \min_Y\{\mu\|Y\|_1 + \frac{1}{2\gamma}\|Y - Z\|_F^2\}$. The augmented Lagrangian function of (5.42) is given by

$$\mathcal{L}_{\rho,\gamma}(X, Z; \Lambda) = -\frac{1}{2}\mathrm{Tr}(X^\top A^\top AX) + g_\gamma(Z) + \langle \Lambda, X - Z\rangle + \frac{\rho}{2}\|X - Z\|_F^2.$$

Therefore, one iteration of our Riemannian ADMM 10 for solving (5.41) reduces to:

$$\begin{aligned} X^{k+1} &:= \mathsf{Retr}_{X^k}(-\eta_k \mathsf{proj}_{\mathrm{T}_{X^k}\mathrm{St}(n,p)}(-A^\top AX^k + \Lambda^k + \rho(X^k - Z^k))) \\ Y^{k+1} &:= \mathrm{prox}_{\frac{\mu(1+\rho\gamma)}{\rho}\|\cdot\|_1}\left(X^{k+1} + \frac{1}{\rho}\Lambda^k\right) \\ Z^{k+1} &:= \frac{\gamma}{1+\gamma\rho}\left(\frac{1}{\gamma}Y^{k+1} + \Lambda^k + \rho X^{k+1}\right) \\ \Lambda^{k+1} &:= \Lambda^k + \rho(X^{k+1} - Z^{k+1}). \end{aligned} \tag{5.43}$$

The ManPG [Chen et al., 2020] for solving (5.37) updates the iterates as follows:

$$\begin{aligned} V^k &:= \operatorname*{argmin}_{V \in \mathrm{T}_{X^k}\mathrm{St}(n,p)} \langle -A^\top AX^k, V\rangle + \frac{1}{2t}\|V\|^2 + \mu\|A(X^k + V)\|_1 \\ X^{k+1} &:= \mathsf{Retr}_{X^k}(\alpha V^k), \end{aligned} \tag{5.44}$$

where α and t are stepsizes. The authors of Chen et al. [2020] suggest to solve the V subproblem by using a semi-smooth Newton method. The Riemannian subgradient method (RSG) [Ferreira

and Oliveira, 1998] for solving (5.37) updates the iterates as follows:

$$X^{k+1} := \mathsf{Retr}_{X^k}(-\eta_k \mathsf{proj}_{\mathrm{T}_{X^k}\mathrm{St}(n,p)}(-A^\top A X^k + \mu D^k)), \quad \text{with } D^k \in \partial\|X^k\|_1. \tag{5.45}$$

We now describe the setup of our numerical experiment. The data matrix $A \in \mathbb{R}^{m\times n}$ is generated randomly whose entries follow the standard Gaussian distribution. We choose μ from $\{0.5, 0.7, 1\}$, n from $\{100, 300, 500\}$, and p from $\{50, 100\}$. In our Riemannian ADMM, we set $\gamma = 10^{-8}$, $\rho = 10^2$ and $\eta_k = \eta = 10^{-2}$. The code of ManPG is downloaded from the authors' website of Chen et al. [2020] and default settings of the parameters are used. In RSG (5.45), we set the stepsize $\eta_k = \eta = 10^{-2}$ as a result of a simple grid search. For all three algorithms, we terminate them when the change of the objective function in two consecutive iterations is smaller than 10^{-8}, which means

$$|F(X^{k+1}) - F(X^k)| < 10^{-8}$$

for ManPG (5.44) and RSG (5.45), and

$$|F(Y^{k+1}) - F(Y^k)| < 10^{-8}$$

for our RADMM (5.43), where $F(X) := -\frac{1}{2}\mathrm{Tr}(X^\top A^\top A X) + \mu\|X\|_1$. Moreover, we also terminate the three algorithms when the maximal iteration number, which is set 1000, is reached. For different combinations of μ, n and p, we report the objective value "obj" ($F(X^k)$ for ManPG and RSG, and $F(Y^k)$ for RADMM), CPU time and the sparsity of the solution "Spa" in Table 5.1. Here the "sparsity" is the percentage of the zero entries of the iterate (X^k for ManPG and RSG, and Y^k for RADMM). Moreover, note that Y^k in RADMM (5.43) is not on the Stiefel manifold, we thus report the constraint violation "infeas", which is defined as $\|(Y^k)^\top Y_k - I_p\|_F$, in Table 5.1 for RADMM. From Table 5.1 we have the following observations: (i) both ManPG and RADMM generated very sparse solutions, while RSG cannot generate sparse solutions; (ii) RSG is very slow. It cannot decrease the objective value to the same level as ManPG and RADMM; (iii) RADMM is always faster than ManPG, sometimes is about 10 to 20 times faster. (iv) In most cases, RADMM yields iterates with better objective value than ManPG, and although Y^k generated by RADMM is not on the Stiefel manifold, the constraint violation is small – usually in the order of $10^{-6} \sim 10^{-8}$.

Settings		RSG			ManPG			RADMM			
μ	(n,p)	obj	CPU	Spa	obj	CPU	Spa	obj	CPU	Spa	infeas
0.5	$(300,50)$	23.9783	0.5725	0	6.1015	1.6808	0.9964	6.0794	0.3550	0.9965	1.14e-6
	$(300,100)$	44.9207	1.4091	0	9.9683	16.9343	0.9966	9.4524	1.0113	0.9964	4.43e-6
	$(500,50)$	34.8607	1.1545	0	4.8868	1.7355	0.9977	4.7141	0.8379	0.9980	7.07e-8
	$(500,100)$	72.1180	2.2447	0	12.0830	15.4234	0.9980	11.7489	1.5738	0.9980	1.00e-7
0.7	$(300,50)$	50.0266	0.5584	0	14.9053	1.7990	0.9965	14.9497	0.2860	0.9967	9.90e-8
	$(300,100)$	99.1306	1.4196	0	29.0171	16.7438	0.9966	28.9101	0.8185	0.9967	1.40e-7
	$(500,50)$	73.4292	1.1515	0	14.3927	1.9293	0.9978	14.2181	0.7760	0.9980	9.90e-8
	$(500,100)$	147.0228	2.2224	0	29.8765	16.9296	0.9980	29.6908	1.2075	0.9980	1.40e-7
1.0	$(300,50)$	99.5018	0.5593	0	29.4374	2.2295	0.9967	29.6217	0.1879	0.9967	1.41e-7
	$(300,100)$	202.9473	1.4154	0	61.5334	16.0349	0.9965	61.0310	0.5699	0.9967	2.00e-7
	$(500,50)$	149.1125	1.1564	0	30.5119	1.8004	0.9980	30.4099	0.4336	0.9980	1.41e-7
	$(500,100)$	295.5895	2.2384	0	59.5210	18.3017	0.9980	59.5309	1.0377	0.9980	2.00e-7

TABLE 5.1. Comparison of RSG (5.45), ManPG (5.44), and RADMM (5.43) for solving (5.37). The results are averaged for 10 repeated experiments with random initializations.

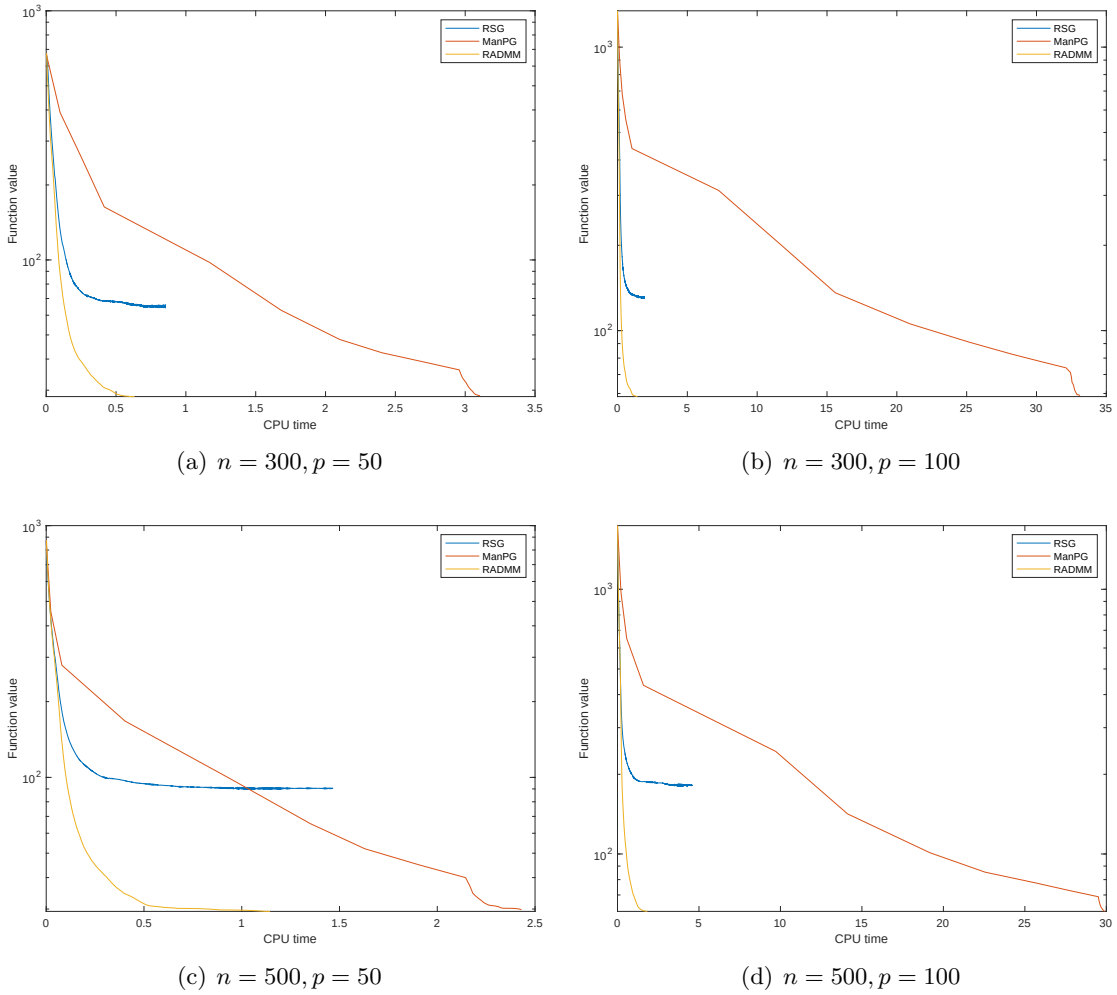
To better illustrate the behavior of the three algorithms, we further draw some figures in Figure 5.1, to show how the objective function value decreases along with the CPU time. From Figure 5.1 we can clearly see that RGS quickly stops decreasing the objective value, while both ManPG and RADMM can decrease the objective value to a much lower level. Moreover, RADMM is much faster than ManPG.

We also compare our RADMM (5.43) with SOC [Lai and Osher, 2014] and MADMM [Kovnatsky et al., 2016]. Before we present the numerical comparisons, we remind the reader that there is no convergence guarantee for SOC and MADMM. The SOC (5.4) algorithm for solving problem (5.37) actually solves the following equivalent problem:

$$\begin{aligned} &\min_{X,Y} \ -\frac{1}{2}\mathrm{Tr}(X^\top A^\top A X) + \mu\|X\|_1 \\ &\text{s.t. } X = Y,\ Y \in \mathrm{St}(n,p). \end{aligned} \tag{5.46}$$

The SOC iterates as follows.

$$\begin{aligned} X^{k+1} &:= \ \operatorname*{argmin}_X -\frac{1}{2}\mathrm{Tr}(X^\top A^\top A X) + \mu\|X\|_1 + \langle \Lambda^k, X - Y^k\rangle + \frac{\rho}{2}\|X - Y^k\|_F^2 \\ Y^{k+1} &:= \ \operatorname*{argmin}_{Y\in\mathrm{St}(n,p)} \langle \Lambda^k, X^{k+1} - Y\rangle + \frac{\rho}{2}\|X^{k+1} - Y\|_F^2 \\ \Lambda^{k+1} &:= \ \Lambda^k + \rho(X^{k+1} - Y^{k+1}). \end{aligned} \tag{5.47}$$



(a) $n = 300, p = 50$ (b) $n = 300, p = 100$

(c) $n = 500, p = 50$ (d) $n = 500, p = 100$

FIGURE 5.1. Comparison of the CPU time (in seconds) consumed among the ManPG, RADMM and Riemannian gradient methods for solving (5.37) with $\mu = 1$. Each figure is averaged for 10 repeated experiments with random initializations.

In our numerical experiment, we chose to solve the X-subproblem using the proximal gradient method. The MADMM (5.5) solves (5.41), and iterates as follows:

$$
\begin{aligned}
X^{k+1} &:= \operatorname*{argmin}_{X \in \mathrm{St}(n,p)} -\frac{1}{2}\mathrm{Tr}(X^\top A^\top A X) + \langle \Lambda^k, X - Y^k \rangle + \frac{\rho}{2}\|X - Y^k\|_F^2 \\
Y^{k+1} &:= \operatorname*{argmin}_{Y} \mu\|Y\|_1 + \langle \Lambda^k, X^{k+1} - Y \rangle + \frac{\rho}{2}\|X^{k+1} - Y\|_F^2 \qquad (5.48) \\
\Lambda^{k+1} &:= \Lambda^k + \rho(X^{k+1} - Y^{k+1}).
\end{aligned}
$$

In our numerical experiment, we chose to solve the X-subproblem using a Riemannian gradient method.

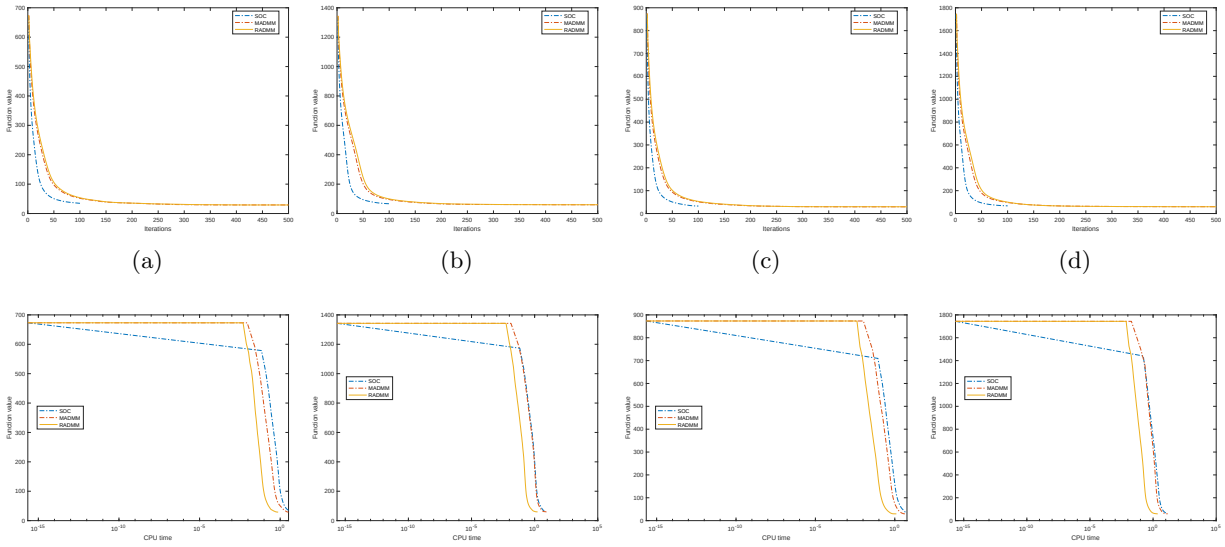
We test our RADMM with SOC and MADMM with the following parameters: for SOC we set $\rho = 50$ and $\eta = 10^{-2}$, where η is the stepsize for the proximal gradient method for solving the X-subproblem; for MADMM we set $\rho = 100$ and $\eta = 10^{-2}$, where η is the stepsize for the Riemannian gradient method for solving the X-subproblem; for RADMM we set $\rho = 100$, $\eta = 10^{-2}$ and $\gamma = 10^{-8}$. The parameters are obtained via simple grid searches, also we randomly initialize three algorithms at the same starting point. For all the three algorithms we record the function value and sparsity for the sequence on the manifold, i.e. X^k for MADMM and RADMM, and Y^k for SOC. For each algorithm, we terminate after 100 iterations. We present the function value change curve in Figure 5.2. We also report the objective function values of the outputs (denoted as "obj"), the sparsity (the percentage of zero entries, denoted as "Spa") and the constraint violation ($\|X^k - Y^k\|_F$ for all three algorithms, denoted as "infeas") in Table 5.2. From the top row of Figure 5.2 we can see that SOC is more efficient in terms of the iteration number, but from the bottom row of Figure 5.2 we see that RADMM is more efficient in terms of the CPU time. This is exactly because all steps in our RADMM are very easy to compute, and so the per-iteration complexity is very cheap.

Settings	SOC			MADMM			RADMM		
(n, p)	obj	Spa	infeas	obj	Spa	infeas	obj	Spa	infeas
$(300, 50)$	34.8851	0.7609	0.0060	29.2059	0.9967	0.0000	29.1197	0.9967	0.0000
$(300, 100)$	66.6870	0.6018	0.0072	59.6483	0.9967	0.0000	59.8210	0.9967	0.0000
$(500, 50)$	32.7199	0.8819	0.0040	29.4007	0.9980	0.0000	29.5003	0.9742	0.0000
$(500, 100)$	67.2337	0.7558	0.0082	59.7878	0.9977	0.0000	59.4491	0.9980	0.0000

TABLE 5.2. Comparison of SOC, MADMM and RADMM for solving (5.37) with $\mu = 1$. The results are averaged for 10 repeated experiments with random initializations.

5.4.2. Numerical Experiments on ODL and DPCP. In this section, we test Algorithm 10 on the DPCP problem (5.40), which can be equivalently written as:

$$\begin{aligned} &\min_{X,W} \ \|W\|_1 \\ &\text{s.t., } W = Y^\top X, \ X \in \mathrm{St}(n, p). \end{aligned} \tag{5.49}$$



(a) (b) (c) (d)

(a) $n = 300, p = 50$ (b) $n = 300, p = 100$ (c) $n = 500, p = 50$ (d) $n = 500, p = 100$

FIGURE 5.2. Comparison of SOC, MADMM and RADMM for solving (5.37) with $\mu = 1$. The first row is the comparison of function value decrease w.r.t. number of iterations, and the second row is w.r.t. CPU time consumed. Each figure is averaged for 10 repeated experiments with random initializations.

Simple calculation shows that Algorithm 10 for the DPCP problem (5.40) iterates as follows.

$$
\begin{aligned}
X^{k+1} &:= \mathsf{Retr}_{X^k}(-\eta_k \mathsf{proj}_{T_{X^k}\,\mathrm{St}(n,p)}(Y\Lambda^k + \rho Y(Y^\top X^k - Z^k))) \\
W^{k+1} &:= \mathrm{prox}_{\frac{1+\rho\gamma}{\rho}\|\cdot\|_1}(Y^\top X^{k+1} + \frac{1}{\rho}\Lambda^k) \\
Z^{k+1} &:= \frac{1}{1/\gamma + \rho}\left(\frac{1}{\gamma}W^{k+1} + \Lambda^k + \rho Y^\top X^{k+1}\right) \\
\Lambda^{k+1} &:= \Lambda^k + \rho(Y^\top X^{k+1} - Z^{k+1}).
\end{aligned}
\tag{5.50}
$$

We compare the RADMM with iteratively reweighted least squares (IRLS) Lerman and Maunu [2018b], Tsakiris and Vidal [2018], projected subgradient method (PSGM) Zhu et al. [2018][1] and manifold proximal point algorithm (ManPPA) Chen et al. [2021a]. Note that the objective of the

[1]We remark that Maunu et al. [2019] proposed a similar Riemannian gradient descent algorithm for RSR by operating on the subspace rather than its orthogonal complement.

problem:

$$\begin{aligned} &\min_{X} \ F(X) := \|Y^\top X\|_1 \\ &\text{s.t.} \ \ X \in \mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} | X^\top X = I_p\}. \end{aligned} \tag{5.51}$$

is separable column-wisely:

$$\begin{aligned} &\min_{x_1,\ldots,x_p} \ \sum_{i=1}^{p} \|Y^\top x_i\|_1 \\ &\text{s.t.} \ \ \{x_1, ..., x_p\} \text{ is orthonormal set.} \end{aligned} \tag{5.52}$$

PSGM and ManPPA conduct the minimization column-wisely. Therefore, in our experiment, we can only record the function value at the outputs of PSGM and ManPPA. Meanwhile the IRLS algorithm that we implemented here is a variant of the original column-wise algorithm for solving (5.40) Lerman and Maunu [2018b], Tsakiris and Vidal [2018,?]. IRLS iterates as follows: first we find the initialization by $X^0 := \mathrm{argmin}_{X\in\mathrm{St}(n,p)} \|Y^\top X\|_F^2$ and then the iterate is updated by

$$X^{k+1} \leftarrow \underset{X\in\mathrm{St}(n,p)}{\mathrm{argmin}} \sum_i \|X^\top Y_i\|_2^2 / \max\{\delta, \|(X^k)^\top Y_i\|_2\}. \tag{5.53}$$

We follow the same experiment setting as Chen et al. [2021a]. More specifically, we construct the data to be $Y = [SR, O]$, $S \in \mathbb{R}^{n\times d}$ with orthogonal column vectors, $R \in \mathbb{R}^{d\times p_1}$, $O \in \mathbb{R}^{N\times p_2}$ both with random Gaussian entries. Here p_1 and p_2 are the numbers of inliers and outliers respectively as described in Chen et al. [2020]. In our experiment we set $p = 5$, $p_1 = 500$ and $p_2 = 1167$, with different choice of n. For our RADMM algorithm we set $\rho = 40$, $\gamma = 4 \cdot 10^{-9}$, $\eta = 2 \cdot 10^{-4}$. For other algorithms, we use their default parameter settings from Chen et al. [2021a], Tsakiris and Vidal [2018], Zhu et al. [2018]. For all the algorithm, we terminate them if the difference between two consecutive function values is smaller than 10^{-6}, i.e.

$$|F(X^{k+1}) - F(X^k)| < 10^{-6}.$$

We initialize IRLS and RADMM with the same initial point as in Zhu et al. [2018]. Note that PSGM and ManPPA sequentially solves the column-wise problems, and therefore they do not need the initial point to be on the Stiefel manifold. In Figure 5.3, we show how the objective function

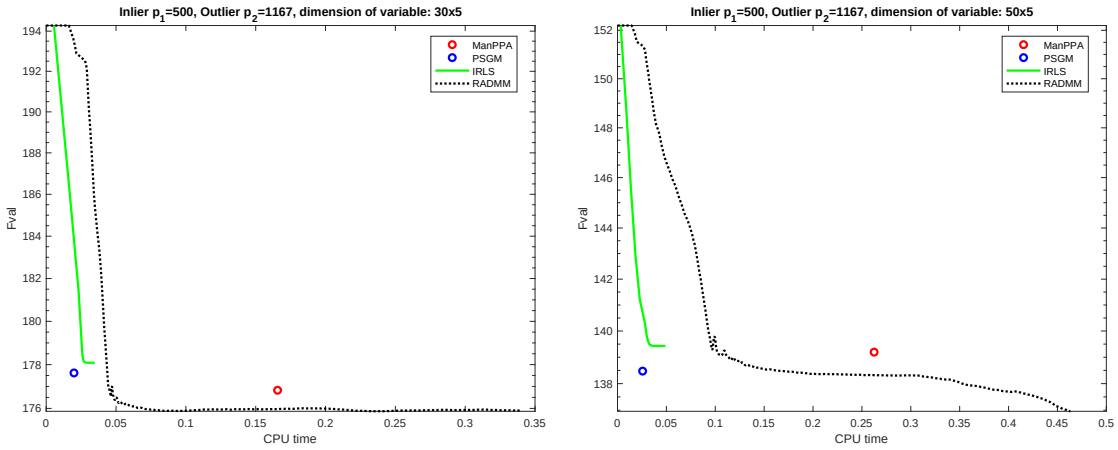


FIGURE 5.3. Function value $\|Y^\top X^k\|_1$ versus CPU time. In this experiment we set $n \in \{30, 50\}$, $p = 5$, $p_1 = 500$ and $p_2 = 1167$.

value changes along with the CPU time. We also record the CPU time and final objective function value in Table 5.3. For RADMM, we also include the constraint violation (i.e. $\|W^k - Y^\top X^k\|_F$, denoted as "infeas" in the table) in Table 5.3. It can be seen from Figure 5.3 and Table 5.3 that RADMM outputs the other three algorithms in terms of the objective function value.

Settings	PSGM		IRLS		ManPPA		RADMM		
(n, p)	obj	CPU	obj	CPU	obj	CPU	obj	CPU	infeas
$(30, 5)$	180.59	0.0131	177.66	0.0230	177.90	0.1164	173.28	0.3177	0.0003
$(50, 5)$	141.66	0.0215	142.61	0.0404	138.78	0.1820	136.62	0.3971	0.0007
$(70, 5)$	125.94	0.0429	118.97	0.0881	119.50	0.3532	116.39	0.4526	0.0074

TABLE 5.3. Summary of function value, CPU time (seconds) of proposed RADMM Algorithm (5.50), comparing with PSGM Zhu et al. [2018], IRLS Lerman and Maunu [2018b], Tsakiris and Vidal [2018] and ManPPA Chen et al. [2021a] algorithm. The results are averaged for 10 repeated experiments with random generated data. In this experiment we set $p_1 = 500$ and $p_2 = 1167$.

We also compare our RADMM (5.50) with SOC Lai and Osher [2014] and MADMM Kovnatsky et al. [2016]. The SOC (5.4) algorithm for problem (5.40) actually solves the following equivalent problem:

$$\min_{X,W} \|Y^\top X\|_1$$
$$\text{s.t., } X = W,\ W \in \mathrm{St}(n, p),$$

and it iterates as:

$$
\begin{aligned}
X^{k+1} &:= \operatorname*{argmin}_{X} \|Y^\top X\|_1 + \langle \Lambda^k, X - W^k\rangle + \frac{\rho}{2}\|X - W^k\|_F^2 \\
W^{k+1} &:= \operatorname*{argmin}_{W\in \mathrm{St}(n,p)} \langle \Lambda^k, X^{k+1} - W\rangle + \frac{\rho}{2}\|X^{k+1} - W\|_F^2 \\
\Lambda^{k+1} &:= \Lambda^k + \rho(X^{k+1} - W^{k+1}).
\end{aligned} \tag{5.54}
$$

In our experiment, we chose to solve the X-subproblem by a subgradient method Beck [2017]. MADMM (5.5) solves (5.49), and updates the iterates as follows:

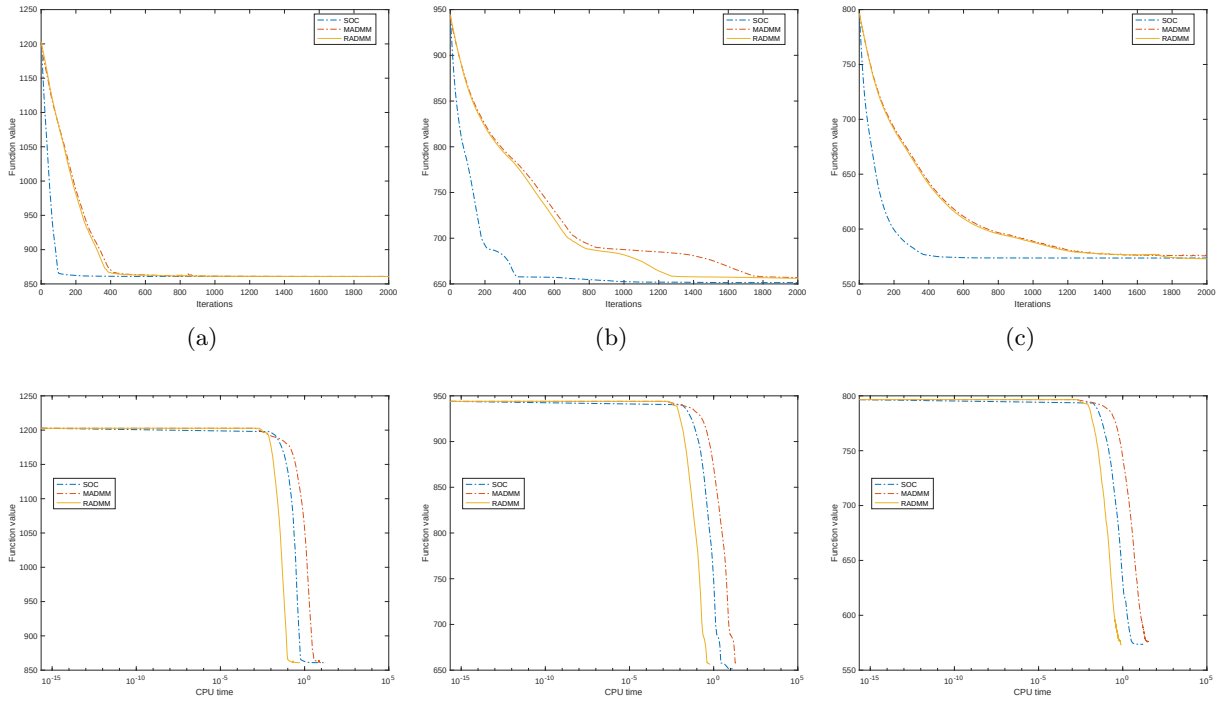
$$
\begin{aligned}
X^{k+1} &:= \operatorname*{argmin}_{X\in \mathrm{St}(n,p)} \langle \Lambda^k, Y^\top X - W^k\rangle + \frac{\rho}{2}\|Y^\top X - W^k\|_F^2 \\
W^{k+1} &:= \operatorname*{argmin}_{W} \|W\|_1 + \langle \Lambda^k, Y^\top X^{k+1} - W\rangle + \frac{\rho}{2}\|Y^\top X^{k+1} - W\|_F^2 \\
\Lambda^{k+1} &:= \Lambda^k + \rho(Y^\top X^{k+1} - W^{k+1}).
\end{aligned} \tag{5.55}
$$

In our experiment, we chose to solve the X-subproblem by a Riemannian gradient descent method.

The parameters are set as follows. For SOC we set $\rho = 50$ and $\eta = 5 \cdot 10^{-6}$, where η is the stepsize for the subgradient step; for MADMM we set $\rho = 50$ and $\eta = 10^{-6}$, where η is the stepsize for the X update; for RADMM we set $\rho = 50$, $\eta = 10^{-4}$ and $\gamma = 10^{-9}$. Again, the parameters are obtained via simple grid searches, also we randomly initialize three algorithms at the same starting point. For all the three algorithms we record the function value for the sequence on the manifold, i.e. X^k for MADMM and RADMM, and W^k for SOC. We terminate the algorithms after 2000 iterations. We record the objective function values in Figure 5.4. We also report the objective function values of the final output (denoted as "obj") and the constraint violation ($\|X^k - W^k\|_F$ for SOC and $\|Y^\top X^k - W^k\|_F$ for MADMM and RADMM, denoted as "infeas") in Table 5.4. It can be seen from Figure 5.4 and Table 5.4 that RADMM is more efficient in terms of CPU time, despite small constraint violation.

5.5. Conclusions

In this chapter, we proposed a Riemannian ADMM for solving a class of Riemannian optimization problem with nonsmooth objective function.



(a) (b) (c)

(a) $(n, p) = (30, 5)$ (b) $(n, p) = (50, 5)$ (c) $(n, p) = (70, 5)$

FIGURE 5.4. Comparison of SOC, MADMM and RADMM for solving (5.40). The first row is the comparison of function value decrease w.r.t. number of iterations, and the second row is w.r.t. CPU time consumed. Each figure is averaged for 10 repeated experiments with random initializations.

Settings	SOC		MADMM		RADMM	
(n, p)	obj	infeas	obj	infeas	obj	infeas
$(30, 5)$	860.9367	0.0000	860.8601	0.0019	860.8394	0.0021
$(50, 5)$	651.4294	0.0000	656.9796	0.0062	656.1095	0.0066
$(70, 5)$	551.2766	0.0000	564.4312	0.0137	563.8032	0.0097

TABLE 5.4. Comparison of SOC, MADMM and RADMM for solving (5.40). The results are averaged for 10 repeated experiments with random initializations.

All steps of our Riemannian ADMM are easy to compute and implement, which gives the potential to be applied to solving large-scale problems. Our method is based on a Moreau envelop smoothing technique. How to design ADMM for solving (5.1) without smoothing remains an open question for future work.

Bibliography

P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

N. Agarwal, N. Boumal, B. Bullins, and C. Cartis. Adaptive regularization with cubics on manifolds. *Mathematical Programming*, 188(1):85–134, 2021.

F. Alimisis, P. Davies, B. Vandereycken, and D. Alistarh. Distributed principal component analysis with limited communication. *Advances in Neural Information Processing Systems*, 34, 2021.

C. Audet and W. Hare. *Derivative-free and blackbox optimization*, volume 2. Springer, 2017.

K. Balasubramanian and S. Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Foundations of Computational Mathematics*, pages 1–42, 2021.

K. Balasubramanian, S. Ghadimi, and A. Nguyen. Stochastic multilevel composition optimization algorithms with level-independent convergence rates. *SIAM Journal on Optimization*, 32(2): 519–544, 2022.

A. Beck. *First-order methods in optimization*. SIAM, 2017.

T. Bendory, Y. C. Eldar, and N. Boumal. Non-convex phase retrieval from STFT measurements. *IEEE Transactions on Information Theory*, 64(1):467–484, 2017.

G. C. Bento, O. P. Ferreira, and J. G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *Journal of Optimization Theory and Applications*, 173(2):548–562, 2017.

D. A. Bini and B. Iannazzo. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700–1710, 2013.

S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

P. B. Borckmans, S. E. Selvan, N. Boumal, and P.-A. Absil. A Riemannian subgradient algorithm for economic dispatch with valve-point effect. *Journal of Computational and Applied Mathematics*, 255:848–866, 2014.

N. Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

N. Boumal and P. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In *Advances in neural information processing systems*, pages 406–414, 2011.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42):1455–1459, 2014. URL `https://www.manopt.org`.

N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, 2018.

S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

D. Burago, Y. Burago, and S. Ivanov. *A course in metric geometry*, volume 33. American Mathematical Society, 2022.

C. J. Burges. *Dimension reduction: A guided tour*. Now Publishers Inc, 2010.

H. Cai, D. Mckenzie, W. Yin, and Z. Zhang. Zeroth-order regularized optimization (zoro): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2):687–714, 2022.

Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 10705–10715, 2018.

T. Carson, D. G. Mixon, and S. Villar. Manifold optimization for k-means clustering. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 73–77. IEEE, 2017.

Z. Charles and J. Konečnỳ. Convergence and accuracy trade-offs in federated learning and meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2575–2583. PMLR, 2021.

A. Chattopadhyay, S. E. Selvan, and U. Amato. A derivative-free Riemannian Powell's method, minimizing hartley-entropy-based ICA contrast. *IEEE transactions on neural networks and learning systems*, 27(9):1983–1990, 2015.

F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4, 2021.

P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

S. Chen, S. Ma, A. M.-C. So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM Journal on Optimization*, 30(1):210–239, 2020.

S. Chen, Z. Deng, S. Ma, and A. M.-C. So. Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning. *IEEE Transactions on Signal Processing*, 69:4759–4773, 2021a.

S. Chen, A. Garcia, M. Hong, and S. Shahrampour. Decentralized Riemannian gradient descent on the Stiefel manifold. In *International Conference on Machine Learning*, pages 1594–1605. PMLR, 2021b.

S. Chen, A. Garcia, M. Hong, and S. Shahrampour. On the local linear rate of consensus on the Stiefel manifold. *arXiv preprint arXiv:2101.09346*, 2021c.

M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE transactions on neural networks and learning systems*, 28(12):2859–2871, 2016.

A. Conn, K. Scheinberg, and L. Vicente. *Introduction to Derivative-Free Optimization*, volume 8. SIAM, 2009.

A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

L. Demanet and P. Hand. Scaling law for recovering the sparsest element in a subspace. *Information and Inference*, 3(4):295–309, 2014.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

P. Diaconis, S. Holmes, and M. Shahshahani. Sampling from a manifold. In *Advances in modern statistical theory and applications: a Festschrift in honor of Morris L. Eaton*, pages 102–125. Institute of Mathematical Statistics, 2013.

M. P. Do Carmo. *Riemannian geometry*, volume 6. Springer, 1992.

D. Drieß, P. Englert, and M. Toussaint. Constrained bayesian optimization of combined interaction force/task space controllers for manipulations. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 902–907. IEEE, 2017.

J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

O. Ferreira and P. Oliveira. Subgradient algorithm on Riemannian manifolds. *Journal of Optimization Theory and Applications*, 97(1):93–104, 1998.

R. S. Fong and P. Tino. Stochastic derivative-free optimization on riemannian manifolds. In *Population-Based Optimization on Riemannian Manifolds*, pages 105–137. Springer, 2022.

M. Forina, R. Leardi, A. C, and S. Lanteri. *PARVUS: An Extendable Package of Programs for Data Exploration*. Elsevier, Amsterdam, 01 1998. ISBN 0-444-43012-1.

P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

K. A. Gallivan and P. Absil. Note on the convex hull of the Stiefel manifold. *Technical note*, 2010.

E. S. Gawlik and M. Leok. High-order retractions on matrix manifolds using projected polynomials. *SIAM Journal on Matrix Analysis and Applications*, 39(2):801–828, 2018.

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

S. Ghadimi, A. Ruszczynski, and M. Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

D. Golovin, J. Karro, G. Kochanski, C. Lee, and X. Song. Gradientless descent: High-dimensional zeroth-order optimization. *arXiv preprint arXiv:1911.06317*, 2019.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

A. Grammenos, R. Mendoza Smith, J. Crowcroft, and C. Mascolo. Federated principal component analysis. *Advances in Neural Information Processing Systems*, 33:6453–6464, 2020.

P. Grohs and S. Hosseini. Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. *IMA Journal of Numerical Analysis*, 36(3):1167–1192, 2016a.

P. Grohs and S. Hosseini. ε-subgradient algorithms for locally Lipschitz functions on Riemannian manifolds. *Advances in Computational Mathematics*, 42(2):333–360, 2016b.

M. Harandi, M. Salzmann, and R. Hartley. Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods. *IEEE transactions on pattern analysis and machine intelligence*, 40(1):48–62, 2017.

S. Hosseini. Convergence of nonsmooth descent methods via Kurdyka–Lojasiewicz inequality on Riemannian manifolds. *Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn (2015,(INS Preprint No. 1523))*, 2015.

S. Hosseini and A. Uschmajew. A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM Journal on Optimization*, 27(1):173–189, 2017.

S. Hosseini, W. Huang, and R. Yousefpour. Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM Journal on Optimization*, 28(1):596–619, 2018.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

E. P. Hsu. *Stochastic Analysis on Manifolds*, volume 38. American Mathematical Soc., 2002.

W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding to Riemannian optimization for sparse principal component analysis. *arXiv preprint arXiv:1909.05485*, 2019.

W. Huang and K. Wei. Riemannian proximal gradient methods. *Mathematical Programming*, 194: 371–413, 2022.

W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

K. G. Jamieson, R. Nowak, and B. Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2012.

N. Jaquier and L. Rozo. High-dimensional Bayesian optimization via nested Riemannian manifolds. *Advances in Neural Information Processing Systems*, 33, 2020.

N. Jaquier, L. Rozo, S. Calinon, and M. Bürger. Bayesian optimization meets Riemannian manifolds in robot learning. In *Conference on Robot Learning*, pages 233–246. PMLR, 2020.

B. Jiang, T. Lin, S. Ma, and S. Zhang. Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis. *Computational Optimization and Applications*, 72(1): 115–157, 2019.

I. Jolliffe, N.Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003.

M. Journee, Y. Nesterov, P. Richtarik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, 2010.

O. Kachan. Persistent homology-based projection pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 856–857, 2020.

T. Kaneko, S. Fiori, and T. Tanaka. Empirical arithmetic averaging over the compact Stiefel manifold. *IEEE Transactions on Signal Processing*, 61(4):883–894, 2012.

S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

H. Kasai, H. Sato, and B. Mishra. Riemannian stochastic recursive gradient algorithm. In *International Conference on Machine Learning*, pages 2516–2524, 2018.

J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *ECCV*, pages 680–696, 2016.

R. Lai and S. Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.

G. Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Springer Nature, 2020.

J. Larson, M. Menickelly, and S. M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Science & Business Media, 2007.

J. M. Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

G. Lerman and T. Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106 (8):1380–1410, 2018a.

G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of IMA*, 7:277–336, 2018b.

G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15:363–410, 2015.

J. Leygonie, S. Oudot, and U. Tillmann. A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics*, pages 1–63, 2021.

C.-L. Li, K. Kandasamy, B. Póczos, and J. Schneider. High dimensional Bayesian optimization via restricted projection pursuit models. In *Artificial Intelligence and Statistics*, pages 884–892, 2016.

J. Li and S. Ma. Federated learning on Riemannian manifolds. *Appl. Set-Valued Anal. Optim. 5 (2023), 213-232*, 2023.

J. Li, S. Ma, and T. Srivastava. A Riemannian ADMM. *arXiv preprint arXiv:2211.02163*, 2022.

J. Li, K. Balasubramanian, and S. Ma. Stochastic zeroth-order riemannian derivative estimation and optimization. *Mathematics of Operations Research*, 48(2):1183–1211, 2023.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

X. Li, S. Chen, Z. Deng, Q. Qu, Z. Zhu, and A. M. C. So. Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM J. Optimization*, 31(3):1605–1634, 2021.

L. Lin, B. St. Thomas, H. Zhu, and D. B. Dunson. Extrinsic local regression on manifold-valued data. *Journal of the American Statistical Association*, 112(519):1261–1273, 2017.

L. Lin, D. Lazar, B. Sarpabayeva, and D. B. Dunson. Robust optimization and inference on manifolds. *arXiv preprint arXiv:2006.06843*, 2020.

C. Lu, J. Feng, Z. Lin, and S. Yan. Nonconvex sparse spectral clustering by alternating direction method of multipliers and its convergence analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

C. Lyu, K. Huang, and H.-N. Liang. A unified gradient regularization family for adversarial examples. In *2015 IEEE International Conference on Data Mining*, pages 301–309. IEEE, 2015.

S. Ma. Alternating direction method of multipliers for sparse principal component analysis. *Journal of the Operations Research Society of China*, 1(2):253–274, 2013.

A. I. Maass, C. Manzie, D. Nesic, J. H. Manton, and I. Shames. Tracking and regret bounds for online zeroth-order Euclidean and Riemannian optimization. *SIAM Journal on Optimization*, 32 (2):445–469, 2022.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

G. Malinovskiy, D. Kovalev, E. Gasanov, L. Condat, and P. Richtarik. From local sgd to local fixed-point methods for federated learning. In *International Conference on Machine Learning*, pages 6692–6701. PMLR, 2020.

A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe. Automatic LQR tuning based on Gaussian process global optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 270–277. IEEE, 2016.

A. Marco, P. Hennig, S. Schaal, and S. Trimpe. On the design of LQR kernels for efficient controller learning. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5193–5200. IEEE, 2017.

J. Matyas. Random optimization. *Automation and Remote control*, 26(2):246–253, 1965.

T. Maunu, T. Zhang, and G. Lerman. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20:1–59, 2019.

T. Maunu, C. Yu, and G. Lerman. Stochastic and private nonconvex outlier-robust PCA. *Journal of Machine Learning Research*, 190:173–188, 2022.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

J. W. Milnor and J. D. Stasheff. *Characteristic classes*. Number 76. Princeton university press, 1974.

B. Mishra, H. Kasai, P. Jawanpuria, and A. Saroop. A Riemannian gossip approach to subspace learning on Grassmann manifold. *Machine Learning*, pages 1–21, 2019.

A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34:14606–14619, 2021.

J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

J. Mockus. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media, 2012.

M. Mutny and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems*, pages 9005–9016, 2018.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7 (4):308–313, 1965.

A. S. Nemirovski and D. B. Yudin. Problem complexity and method efficiency in optimization. *Wiley & Sons*, 1983.

Y. Nesterov. Random gradient-free minimization of convex functions. *Technical Report. Center for Operations Research and Econometrics (CORE), Catholic University of Louvain*, 2011.

Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 1999.

C. Oh, E. Gavves, and M. Welling. BOCK: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3868–3877, 2018.

S. Park and H. Zhao. Spectral clustering based on learning similarity matrix. *Bioinformatics*, 34 (12):2069–2076, 2018.

R. Pathak and M. J. Wainwright. Fedsplit: An algorithmic framework for fast federated optimization. *Advances in Neural Information Processing Systems*, 33:7057–7066, 2020.

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in subspace: linear sparsity using alternating directions. *IEEE Trans. Information Theory*, 62(10):5855–5880, 2016.

R. Rabadán and A. J. Blumberg. *Topological Data Analysis for Genomics and Evolution: Topology in Biology*. Cambridge University Press, 2019.

E. Rio. Moment inequalities for sums of dependent random variables under projective conditions. *Journal of Theoretical Probability*, 22(1):146–163, 2009.

P. Rolland, J. Scarlett, I. Bogunovic, and V. Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pages 298–307, 2018.

A. Ruszczyński. A linearization method for nonsmooth stochastic programming problems. *Mathematics of Operations Research*, 12(1):32–49, 1987.

A. Ruszczynski. A stochastic subgradient method for nonsmooth nonconvex multilevel composition optimization. *SIAM Journal on Control and Optimization*, 59(3):2301–2320, 2021.

A. Ruszczynski and W. Syski. Stochastic approximation method with gradient averaging for unconstrained problems. *IEEE Transactions on Automatic Control*, 28(12):1097–1105, 1983.

H. Sato. Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses. *SIAM Journal on Optimization*, 32(4):2690–2717, 2022.

K. Scheinberg. Finite difference gradient approximation: To randomize or not? *INFORMS Journal on Computing*, 34(5):2384–2388, 2022.

S. M. Shah. Distributed optimization on Riemannian manifolds for multi-agent networks. *arXiv preprint arXiv:1711.11196*, 2017.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, volume 65. John Wiley & Sons, 2005.

D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *COLT*, 2012.

C. Stein. A bound for the error in the Normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. The Regents of the University of California, 1972.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Trans. Information Theory*, 63(2):853–884, 2017a.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere ii: Recovery by Riemannian trust-region method. *IEEE Trans. Information Theory*, 63(2):885–914, 2017b.

J. Sun, Q. Qu, and J. Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

A. Themelis and P. Patrinos. Douglas–Rachford splitting and admm for nonconvex optimization: Tight convergence results. *SIAM Journal on Optimization*, 30(1):149–181, 2020.

J. Townsend, N. Koep, and S. Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016. URL `http://jmlr.org/papers/v17/16-177.html`.

N. Tripuraneni, N. Flammarion, F. Bach, and M. I. Jordan. Averaging stochastic gradient descent on Riemannian manifolds. In *Conference On Learning Theory*, pages 650–687, 2018.

R. Tron, B. Afsari, and R. Vidal. Riemannian consensus for manifolds with bounded curvature. *IEEE Transactions on Automatic Control*, 58(4):921–934, 2012.

M. C. Tsakiris and R. Vidal. Dual principal component pursuit. *Journal of Machine Learning Research*, 2018.

K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, 6(Jun):995–1018, 2005.

C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng. AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

L. W. Tu. *An Introduction to Manifolds*. Springer Science & Universitext, 2011.

B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

B. Wang, S. Ma, and L. Xue. Riemannian stochastic proximal gradient methods for nonsmooth optimization over the Stiefel manifold. *The Journal of Machine Learning Research*, 23(1):4599–4631, 2022a.

J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33: 7611–7623, 2020a.

L. Wang, R. Fonseca, and Y. Tian. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33, 2020b.

T. Wang. On sharp stochastic zeroth-order Hessian estimators over Riemannian manifolds. *Information and Inference: A Journal of the IMA*, 12(2):787–813, 2023.

T. Wang, Y. Huang, and D. Li. From the Greene–Wu convolution to gradient estimation over Riemannian manifolds. *arXiv:2108.07406*, 2021.

X. Wang, Z. Tu, Y. Hong, Y. Wu, and G. Shi. Online optimization over Riemannian manifolds. *Journal of Machine Learning Research*, 24(84):1–67, 2023.

Y. Wang, S. Du, S. Balakrishnan, and A. Singh. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 1356–1365, 2018.

Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Z. Wang, B. Liu, S. Chen, S. Ma, L. Xue, and H. Zhao. A manifold proximal linear method for sparse spectral clustering with application to single-cell RNA sequencing data analysis. *INFORMS Journal on Optimization*, 4(2):200–214, 2022b.

M. Weber and S. Sra. Nonconvex stochastic optimization on manifolds via Riemannian Frank-Wolfe methods. *arXiv preprint arXiv:1910.04194*, 2019.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidenite programming. In *CVPR*, 2004.

X. Xiao, Y. Li, Z. Wen, and L. Zhang. A regularized semi-smooth Newton method with projection steps for composite convex programs. *Journal of Scientific Computing*, 76(1):364–389, 2018.

L. Yang, T. K. Pong, and X. Chen. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. *SIAM Journal on Imaging Sciences*, 10(1):74–110, 2017.

W. H. Yang, L.-H. Zhang, and R. Song. Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific J. Optimization*, 10(2):415–434, 2014.

K. Yuan, I. Chatzinikolaidis, and Z. Li. Bayesian optimization for whole-body control of high-degree-of-freedom robots through reduction of dimensionality. *IEEE Robotics and Automation Letters*, 4(3):2268–2275, 2019.

J. Zeng, W. Yin, and D.-X. Zhou. Moreau envelope augmented Lagrangian method for nonconvex optimization with linear constraints. *Journal of Scientific Computing*, 91(2):1–36, 2022.

H. Zhang and S. Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR, 2016.

H. Zhang, S. J Reddi, and S. Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 29:4592–4600, 2016a.

H. Zhang, S. J. Reddi, and S. Sra. Fast stochastic optimization on Riemannian manifolds. *ArXiv e-prints*, pages 1–17, 2016b.

J. Zhang and S. Zhang. A cubic regularized Newton's method over Riemannian manifolds. *arXiv preprint arXiv:1805.05565*, 2018.

J. Zhang, S. Ma, and S. Zhang. Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis. *Mathematical Programming Series A*, 184:445–490, 2020.

J. Zhang, W. Pu, and Z.-Q. Luo. On the iteration complexity of smoothed proximal alm for nonconvex optimization problem with convex constraints. *arXiv preprint arXiv:2207.06304*, 2022.

P. Zhou, X. Yuan, S. Yan, and J. Feng. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Y. Zhou, C. Bao, C. Ding, and J. Zhu. A semismooth Newton based augmented Lagrangian method for nonsmooth optimization on matrix manifolds. *Mathematical Programming*, pages 1–61, 2022.

P. Zhu and A. V. Knyazev. Angles between subspaces and their tangents. *Journal of Numerical Mathematics*, 21(4):325–340, 2013.

Z. Zhu, Y. Wang, D. Robinson, D. Naiman, R. Vidal, and M. Tsakiris. Dual principal component pursuit: Improved analysis and efficient algorithms. *Advances in Neural Information Processing Systems*, 31, 2018.

R. Zimmermann and K. Hüper. Computing the Riemannian logarithm on the Stiefel manifold: metrics, methods and performance. *arXiv preprint arXiv:2103.12046*, 2021.

H. Zou and L. Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.

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