THE UNIVERSITY OF SYDNEY

PHD THESIS

DISCIPLINE OF BUSINESS ANALYTICS

OPTIMIZATION AND LEARNING OVER RIEMANNIAN MANIFOLDS

Author: **Andi Han**

Supervisors: Prof. Junbin Gao

A/Prof. Boris Choy

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

The University of Sydney

BUSINESS SCHOOL

2023

To my parents.

Contents

List of Figures

List of Tables

Notation

Vectors, matrices and sets

x, y, z	Generic variables
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Vectors
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	Matrices
$\mathbb{R}, \mathbb{R}^d, \mathbb{R}^{m \times n}$	Real numbers, real vectors of size d and real matrices of size $m \times n$
$\mathbb{R}_+, \mathbb{R}_{++}$	Nonnegative/positive real numbers
$\mathbb{S}^d$	Symmetric matrices of size $d \times d$
$\mathbb{S}^d_+$	Symmetric positive semidefinite matrices of size $d \times d$
$\mathbb{S}^d_{++}$	Symmetric positive definite matrices of size $d \times d$

Riemannian manifold ingredients

$\mathcal{M}, \mathcal{N}$	Smooth manifolds, Riemannian manifolds
$T_x\mathcal{M}$	Tangent space of $\mathcal{M}$ at $x \in \mathcal{M}$
$T\mathcal{M}$	Tangent bundle
$\langle \cdot, \cdot \rangle_2$	Euclidean inner product
$\langle \cdot, \cdot \rangle_x, \langle \cdot, \cdot \rangle$	Riemannian inner product
$\mathrm{Exp}_x(u)$	Exponential map of $u \in T_x\mathcal{M}$ at $x \in \mathcal{M}$
$\mathrm{Exp}_x^{-1}(y), \mathrm{Log}_x(y)$	Inverse exponential map or logarithm map of $y \in \mathcal{M}$ to $T_x\mathcal{M}$
$\mathrm{Retr}_x(u)$	Retraction of $u \in T_x\mathcal{M}$ at $x \in \mathcal{M}$
$\mathrm{Retr}_x^{-1}(y)$	Inverse retraction of $y \in \mathcal{M}$ to $T_x\mathcal{M}$
$\nabla f(x)$	Euclidean gradient of f at $x \in \mathbb{R}^d$

∇	Affine/Riemannian connection
Riem	Riemann curvature tensor
$\mathrm{grad} f(x)$	Riemannian gradient of f at $x \in \mathcal{M}$
$\mathrm{Hess} f(x)$	Riemannian Hessian of f at $x \in \mathcal{M}$
$\Gamma^{c}_{t_0 \to t_1} u, \Gamma^{y}_{x} u$	Parallel transport of $u \in T_x\mathcal{M}$ to $T_y\mathcal{M}$ along a curve c such that $c(t_0) = x, c(t_1) = y$.
$\mathcal{T}_{\xi} u, \mathcal{T}^{y}_{x} u$	Vector transport of $u \in T_x\mathcal{M}$ to $T_y\mathcal{M}$ where $y = \mathrm{Retr}_x(\xi)$

Functions and fields

$\mathfrak{F}(\mathcal{M})$	Real-valued smooth functions on $\mathcal{M}$
$\mathfrak{X}(\mathcal{M})$	Smooth vector fields on $\mathcal{M}$
id	Identity map

Certificate of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes. I declare that any contribution made to the research by others, with whom I have worked at the University of Sydney or elsewhere, is explicitly acknowledged in the thesis.

Andi Han

Abstract

Learning over smooth nonlinear spaces has found wide applications. A principled approach for addressing such problems is to endow the search space with a Riemannian manifold geometry and numerical optimization can be performed intrinsically. Recent years have seen a surge of interest in leveraging Riemannian optimization for nonlinearly-constrained problems. This thesis investigates and improves on the existing algorithms for Riemannian optimization, with a focus on unified analysis frameworks and generic strategies. To this end, the first chapter systematically studies the choice of Riemannian geometries and their impacts on algorithmic convergence, on the manifold of positive definite matrices. The second chapter considers stochastic optimization on manifolds and proposes a unified framework for analyzing and improving the convergence of Riemannian variance reduction methods for nonconvex functions. The third chapter introduces a generic acceleration scheme based on the idea of extrapolation, which achieves optimal convergence rate asymptotically while being empirically efficient.

Acknowledgement

First and foremost, I must express my heartfelt gratitude to my PhD supervisor, Prof. Junbin Gao, who has genuinely provided his unwavering support, invaluable guidance and shared his breadth of knowledge. Meanwhile he has granted me significant degree of research discretion. Without his dedication, the completion of this thesis would not have been possible. My sincere gratitude also extends to my great collaborator/mentor Dr. Bamdev Mishra, who has unreservedly offered his help with domain expertise, and provided invaluable career advice. I also need to thank all my coauthors, including Dr. Pratik Jawanpuria who have offered extensive support, as well as many friends and colleagues who have shared this tough journey with me. I also greatly appreciate the members of the thesis committees for providing constructive feedback that has greatly enhanced the quality of the thesis. At last, I dedicate this thesis to my parents because of their enduring patience in tolerating my stress and complaints and the unconditional support and encouragement during my hardest periods.

Publications

This thesis is based on the following publications and research works:

- **Andi Han**, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao (2021). On Riemannian optimization over positive definite matrices with the Bures-Wasserstein geometry. In *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 8940-8953.

- **Andi Han** and Junbin Gao (2022). Improved variance reduction methods for Riemannian non-convex optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44 (11), 7610-7623.

- **Andi Han**, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao (2023). Riemannian accelerated gradient methods via extrapolation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

The author has further contributed to the following publications, which are not included in the thesis:

- **Andi Han** and Junbin Gao (2021). Riemannian stochastic recursive momentum method for non-convex optimization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2505–2511.

- **Andi Han**, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao (2022). Riemannian block SPD coupling manifold and its application to optimal transport. *Machine Learning*. (Special issue ACML 2022, oral presentation).

- **Andi Han**, Bamdev Mishra, Pratik Jawanpuria, Pawan Kumar, and Junbin Gao (2023). Riemannian Hamiltonian methods for min-max optimization on manifolds. *SIAM Journal on Optimization*, 33(3), 1797-1827.

- **Andi Han**, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao (2023). Learning with symmetric positive definite matrices via generalized Bures-Wasserstein geometry. In *International Conference on Geometric Science of Information (GSI)*, pp. 405-415, 2023. (Oral presentation).

- Saiteja Utpala, **Andi Han**, Pratik Jawanpuria, and Bamdev Mishra (2022). Rieoptax: Riemannian optimization in JAX. In *NeurIPS Workshop on Optimization for Machine Learning*, 2022.

- Saiteja Utpala, **Andi Han**, Pratik Jawanpuria, and Bamdev Mishra (2022). Improved differentially private Riemannian optimization: fast sampling and variance reduction. *Transactions on Machine Learning Research (TMLR)*, 2023.

- Siying Zhang, **Andi Han**, and Junbin Gao (2022). Robust denoising in graph neural networks. In *IEEE Symposium Series On Computational Intelligence (SCCI)*, 2022. (The Best Paper Award).

- Dai Shi, **Andi Han**, Yi Guo and Junbin Gao (2023). Fixed point Laplacian mapping: a geometrically correct manifold learning algorithm. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1-9, 2023.

Authorship Attribution Statement. The three works included in the thesis reflect collaborative efforts. For all the works, I made the main contributions by deriving the theoretical results, conducting majority of the experiments and writing the drafts. The co-authors have helped formulate the ideas in early stage, design the models and experiments, as well as polish the works.

Andi Han, 12/06/2023

Supervisor Statement. As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Junbin Gao, 20/06/2023

Chapter 1

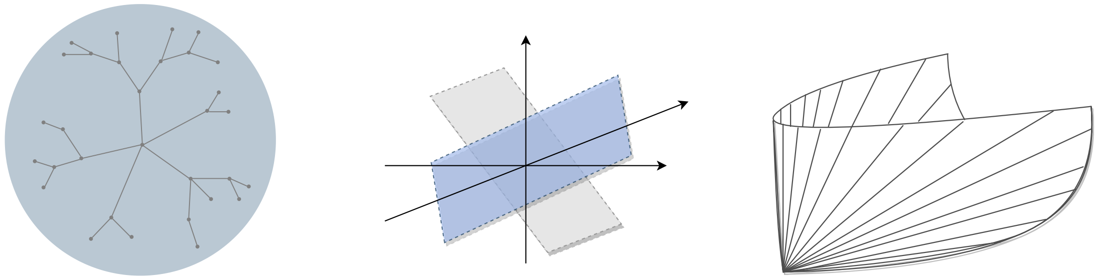
Introduction

Learning and estimation over nonlinear spaces have been rapidly gaining momentum in machine learning, statistics, engineering, and neuroscience, among others. In particular, many problems are concerned with objects that naturally possess nonlinear geometric structure (or constraints), such as orthogonality (Absil et al., 2009; Theis et al., 2009), unit norm (H. Zhang et al., 2016), positive definiteness (Bhatia, 2009), fixed-rank positive semi-definiteness (G. Meyer et al., 2011), hyperbolic (Nickel & Kiela, 2017), unit determinant (Boumal & Absil, 2011a), simplex (Sun et al., 2015), doubly stochasticity (Douik & Hassibi, 2019), to name a few. Being able to identify and incorporate such geometries and constraints in the process of learning and estimation is crucial to ensure efficiency and quality of the results. Common methods for solving the nonlinearly constrained optimization problems include projected gradient methods and relaxation techniques (Nocedal & Wright, 1999; Boyd et al., 2004). Nevertheless, the methods can only accommodate specific problem instances and constraints, and require tailored treatment for each case. Further, both the approaches can be inefficient as they usually lift the problem to a high-dimensional Euclidean ambient search space.

For these reasons, there is a surge of interest in considering optimization on Riemannian manifolds, also known as Riemannian optimization (Absil et al., 2009; Boumal, 2023). In fact, all the aforementioned nonlinear (constraint) spaces form smooth manifolds. When equipped with a smooth inner product structure (i.e., a Riemannian metric), the manifolds become Riemannian manifolds, a class of geometric objects with well-studied tools for both analysis and practical operations (Absil et al., 2009; J. M. Lee, 2018; Boumal, 2023). Riemannian optimization guarantees feasibility with respect to the constraints at each update and provides a flexible framework that directly operates on the intrinsic, low-dimensional search space. This potentially avoids the difficulties of addressing the constraints from the ambient space and thus requires a computational cost scaling with only the intrinsic dimension of the manifolds. Besides the efficiency, Riemannian optimization often results in better-quality solutions by preserving numerical properties of the geometry, such as symmetries and invariants (Absil et al., 2009). More interestingly, many problems that are nonconvex (when considering as constrained optimization in the Euclidean space) turn out to be geodesic (strongly) convex (a generalized notion of convexity) on Riemannian manifolds (Vishnoi, 2018; R. Hosseini & Sra, 2020), thus allowing global optimal solutions to be sought with fast convergence guarantees.

1.1 Learning over nonlinear spaces: from Euclidean space to Riemannian manifolds

Feature representation learning has been central to machine learning and data science, particularly amid the ever-increasing complication of data structure and expanding dimensionality. Among all the methods for representation learning, deep learning methods have revolutionized the fields such as computer vision (LeCun et al., 1998), natural language processing (Bengio et al., 2000), molecular modelling (Jumper et al., 2021), with theoretically guaranteed learning capacity (Hornik et al., 1989). Despite the success of deep learning models, most existing



(a) Hyperbolic geometry (b) Grassmann geometry (c) SPD geometry

Figure 1.1: (1.1a) 2-dimensional Poincaré disk model of hyperbolic geometry, which is well-suited for embedding relational data with hierarchies. (1.1b) Grassmann geometry, the set of subspaces. The figure shows 2-dimensional subspaces in $\mathbb{R}^3$. (1.1c) Symmetric positive definite matrix (SPD) geometry. The figure shows the convex cone defined by a 2×2 SPD matrix.

efforts are concerned with finding good vector representation over the Euclidean space. One downside of the Euclidean representation is that the learning difficulty often escalates with the feature dimensions, a phenomenon known as the curse of dimensionality (Bellman, 1966). One notorious example is the unavoidable large distortion when learning tree-structured data even with unbounded dimensions (Linial et al., 1995). In addition, many tasks often come with latent geometric structure that can be leveraged to enhance learning efficiency and quality (Bronstein et al., 2017, 2021).

Riemannian manifolds have received significant attention over the recent decades for modelling structured data. Specifically, *symmetric positive definite matrices* can be endowed with a Riemannian manifold structure, and have been utilized to represent images via covariance descriptors (Tuzel et al., 2006, 2008), to model water molecule diffusion in human brains (Pennec et al., 2006), to parameterize metrics and kernels (Guillaumin et al., 2009; P. K. Jawanpuria et al., 2015) and for acoustic model compression (Shinohara et al., 2010). *Subspaces*, form the so-called Grassmann manifold, have been exploited to represent image sets, videos (Hamm & Lee, 2008; Turaga et al., 2011), shape spaces (Turaga et al., 2008) and for graph embeddings (Cruceru et al., 2021; B. Zhou et al., 2022). *Spheres* are common in directional statistics (Mardia et al., 2000; Pewsey & García-Portugués, 2021) and have been used for topic modelling (Batmanghelich et al., 2016), text embeddings (Meng et al., 2019). *Hyperbolic* space has been successful in embedding hierarchical relations, such as taxonomies (Nickel & Kiela, 2017), images (Khrulkov et al., 2020), graphs (Chamberlain et al., 2017; Chami et al., 2020). Hyperbolic geometry has been shown to overcome the large distortion issue faced in the Euclidean space due to the exponential growth of distance (Sala et al., 2018). Other applications concern manifolds such as *oblique manifolds* (Absil & Gallivan, 2006), *Lie groups of transformations* (Boumal & Absil, 2011a), *product Riemannian manifolds* with mixed curvatures (Gu et al., 2018), *heterogeneous manifolds* (Di Giovanni et al., 2022), *pseudo-Riemannian manifolds* (Sim et al., 2021; Xiong et al., 2022), and many more.

1.2 Riemannian optimization: the framework for learning on manifolds

Riemannian optimization has become the primary framework for solving learning tasks on manifolds. The key idea is to forgo the extrinsic view of variables restricting to a constraint manifold and instead undertake the intrinsic view that the variables see only the manifold where they are allowed to move freely. Such transition of viewpoint allows unconstrained optimization algorithms to be developed on manifolds by generalizing the many iterative algorithms for solving unconstrained problems in the Euclidean space, which date back to Luenberger (1972); Gabay (1982); Udriste (1994). Although classic optimization algorithms in the Euclidean space rely heavily on the linear space and the Euclidean metric in order to measure the function variations and design the updates between iterates, Riemannian optimization provides an elegant framework with generalized notions from differential geometry, including *exponential map* (or *retraction*) for taking the update, as well as *Riemannian gradient/Hessian* for measuring function variations.

The recent decades have witnessed drastic advancements in Riemannian optimization, from developments of more advanced optimization algorithms, theoretical understanding of the convergence properties to exploration of more general objectives and problem settings. Absil et al. (2009) provides a unified treatment for numerical optimization on matrix manifolds, including line-search and trust-region Newton's methods, and H. Zhang & Sra (2016) offers a framework for analyzing iteration complexities of Riemannian optimization via geodesic convexity. Since then, many works have proposed more advanced algorithms on manifolds, generalizing the ideas in the Euclidean space, such as stochastic gradient based methods (Bonnabel, 2013; Tripuraneni et al., 2018; Kasai et al., 2019), stochastic variance reduction (H. Zhang et al., 2016; Kasai et al., 2018b; Sato et al., 2019; P. Zhou, Yuan, & Feng, 2019), Nesterov acceleration (Y. Liu et al., 2017; H. Zhang & Sra, 2018b; Ahn & Sra, 2020), quasi-Newton methods (Savas & Lim, 2010; W. Huang, Gallivan, & Absil, 2015), cubic-regularized Newton's methods (N. Agarwal et al., 2021), among many other works.

Apart from the algorithmic developments, many studies have also explored more general problem instances on manifolds, including nonsmooth objectives (H. Zhang & Sra, 2016; Chen et al., 2020; W. Huang & Wei, 2022; J. Li, Ma, & Srivastava, 2022), submanifold-constrained objectives (Weber & Sra, 2022a,b), min-max objectives (F. Huang & Gao, 2023; P. Zhang et al., 2022), decentralized settings (Chen et al., 2021; L. Wang & Liu, 2022; J. Li & Ma, 2022), and derivative-free settings (J. Li, Balasubramanian, & Ma, 2022).

1.3 Research questions and contributions

This thesis aims to further advance the developments and expand the potential of Riemannian optimization, with a particular focus on generic and unified strategies. This is achieved through various efforts including providing useful insights and analysis on existing methods, proposing novel algorithms, as well as exploring a variety of problem instances, which lead to applications in a wide range of fields. The thesis is composed of three main chapters, each addressing a research question, which are discussed below.

Research question 1: the choice of Riemannian metric and its impacts on the algorithmic performance of Riemannian optimization.

A metric (an inner product) is crucial for optimization, including defining the local function variations, i.e., the gradient and Hessian. Unlike the Euclidean space where the Euclidean metric is the major choice for optimization algorithms, the nonlinear manifold can be equipped with different Riemannian metrics, which in turn leads to different sets of operations essential for the design of numerical algorithms, such as exponential map. For example, hyperbolic space can be equivalently represented with at least five isometric models, including the most famous Poincaré disk model and Hyperboloid model (Cannon et al., 1997). The space of symmetric positive definite matrices could result in a non-positively curved geometry (Bhatia, 2009), non-negatively curved geometry (Malagò et al., 2018) and a flat geometry (Arsigny et al., 2007), depending on the choice of Riemannian metrics. The study of different Riemannian metrics has been central for statistical analysis and estimation. Nevertheless, comparatively little attention has been given on the choice of Riemannian metrics for the purpose of Riemannian optimization and its implications for algorithm performance.

In Chapter 3, we provide the first systematic study on the implications of Riemannian metric on Riemannian optimization, with a focus on symmetric positive definite (SPD) matrices. Particularly, we show that the recently introduced Bures-Wasserstein metric (Malagò et al., 2018) is a more suitable and robust choice for several Riemannian optimization problems compared to the classic choice of Affine-Invariant metric (Bhatia, 2009), especially over ill-conditioned SPD matrices. This work has been published at NeurIPS 2021 (Han et al., 2021).

Research question 2: Unified framework for accelerating and analyzing variance reduction methods for Riemannian optimization.

Existing studies in the field of Riemannian optimization have been focused on developing more advanced optimization algorithms by generalizing the counterparts in the Euclidean space. Successful generalizations not only should provide efficient numerical procedures for practical problems, but also need to be grounded with theoretical guarantees. Due to the curved geometries of Riemannian manifolds, there exists no ad hoc or 'correct' generalization. For this reason, various designs and analysis frameworks for the algorithms on manifolds are developed.

One example is variance reduction (Johnson & Zhang, 2013; Defazio et al., 2014), which has shown to improve gradient descent and stochastic gradient descent for finite-sum and online optimization in the Euclidean space. The key idea is to construct a variance reduced stochastic estimate of gradient by correcting the difference between full gradient/large-batch gradient and stochastic gradient. Variance reduction literature abounds with the designs of such estimate, including SAGA (Defazio et al., 2014), SVRG (Johnson & Zhang, 2013), SRG (Nguyen et al., 2017a), SPIDER (Fang et al., 2018), just to name the most popular ones. Although many existing works have successfully adapted the varaince reduction techniques for Riemannian optimization, including R-SAGA (Babanezhad et al., 2018) R-SVRG (H. Zhang et al., 2016; Han & Gao, 2021), R-SRG (Kasai et al., 2018b) and R-SPIDER (J. Zhang et al., 2018; P. Zhou, Yuan, Yan, & Feng, 2019), the algorithms are analyzed under different frameworks, with established convergence rates to be both curvature-dependent (H. Zhang et al., 2016) and independent (J. Zhang et al., 2018). This increases the difficulty of comparing the utility across different algorithms on manifolds. In addition, existing analysis of variance reduction on manifolds is suboptimal and incomplete, leaving a performance gap between the Riemannian and Euclidean versions, particularly with the R-SVRG and R-SRG.

In Chapter 4, we propose a unified framework for analyzing variance reduction methods on manifolds, motivated by a recent work (Ji et al., 2020) that proposes batch size adaptation in the Euclidean space. With such a framework, we derive and complete the convergence analysis for both R-SVRG and R-SRG, under both finite-sum and online settings, for both nonconvex and gradient-dominated function classes, with and without the use of general retraction and vector transport. The unified analysis also allows insights to be generated regarding the utility difference between SVRG- and SRG-type of gradient estimates on manifolds. Furthermore, thanks to the batch size adaptation strategy, both R-SVRG and R-SRG potentially require lower gradient complexities, which enhances the computational efficiency of the algorithms. This work has been published on IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI) (Han & Gao, 2021).

Research question 3: Generic acceleration technique for first-order Riemannian optimization.

One of the outstanding questions in the literature of Riemannian optimization is how to design an accelerated (first-order) numerical algorithm such that it achieves the optimal rate of convergence (established in the Euclidean space by Y. Nesterov (2003)). However, classic acceleration strategies depend critically on the flat geometry of the Euclidean space, hence rendering generalization to Riemannian manifolds nontrivial. Nonetheless, a plethora of existing studies have been successful in designing the Nesterov type of acceleration for Riemannian optimization (Y. Liu et al., 2017; H. Zhang & Sra, 2018a; Ahn & Sra, 2020; Kim & Yang, 2022; Jin & Sra, 2022).

Despite the great efforts, Nesterov acceleration on manifolds exhibits several notable deficiencies. First, to achieve the optimal convergence rate, the algorithm requires a careful choice of stepsize and coupling parameters that depend on the smoothness and strong convexity constants, which are often unknown in practical settings. In addition, most of the algorithms for acceleration on manifolds overly rely on the use of (inverse) exponential map, parallel transport, which further amplifies the implementation difficulties. It therefore remains an open question whether there exists a generic yet effective acceleration strategy on manifolds, more favorable both numerically and theoretically.

In Chapter 5, we answer the above question affirmatively by designing an acceleration scheme based on the idea of extrapolation, i.e., a postprocessing averaging step for a sequence of iterates. We show when the iterates are generated from Riemannian gradient descent method, the accelerated scheme achieves the optimal convergence rate asymptotically and is computationally more favorable than the recently proposed Riemannian Nesterov accelerated gradient methods (H. Zhang & Sra, 2018a; Alimisis et al., 2020, 2021; Kim & Yang, 2022).

1.4 Thesis outline

The remainder of the thesis is organized as follows. Chapter 2 provides a preliminary review of concepts and notations from Riemannian geometry and Riemannian optimization. Chapter 3, 4, and 5 are the main chapters addressing the above research questions. Finally Chapter 6 concludes the thesis by summarizing the contributions in a wider context and discusses future research directions.

Chapter 2

Preliminaries

This chapter provides a preliminary overview of the notions from Riemannian geometry and Riemannian optimization, as well as sets the notations for the rest of the thesis.

Although most manifolds we deal with in this thesis can be represented via vectors and matrices, the classic definitions of manifolds come with more abstract notions, such as topological spaces, charts, atlas, and tangent spaces. In fact, the most familiar and intuitive examples of manifolds, such as spheres and hyperboloids, are all embedded submanifolds of a linear space, which constitutes only a small part of the manifold family. Hence, the chapter starts with brief introduction to smooth manifolds, as well as general definitions of tangent space and differential of mappings. Then, we introduce Riemannian metric and Riemannian manifolds, along with the many ingredients and operations of Riemannian manifolds. For Riemannian optimization, this chapter also covers various function classes, including geodesic/retraction (strong) convexity, smoothness and gradient dominance.

Most results are presented without proofs. For more detailed exposition of the topics, readers shall refer to the general texts such as Boothby (1986); J. Lee (2012); J. M. Lee (2018) for Riemannian geometry and Absil et al. (2009); Boumal (2023) for Riemannian optimization.

2.1 Riemannian geometry

2.1.1 Smooth manifolds

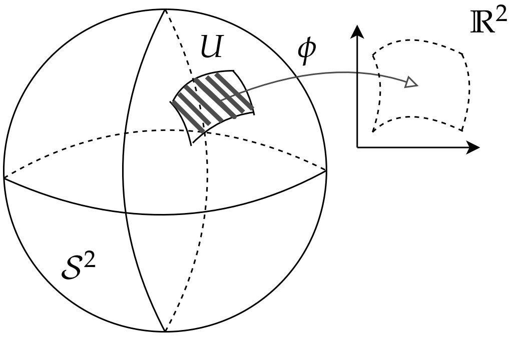


Figure 2.1: Illustration of a chart on a 2-d sphere $\mathcal{S}^2 := \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{x} = 1\}$.

In the simplest terms, a manifold of dimension d is a set that locally resembles $\mathbb{R}^d$. For example, a 2-d sphere, although embedded in $\mathbb{R}^3$ can be locally identified by a 2-d surface (see Figure 2.1). Such resemblance is formally defined via the notion of charts. The union of compatible charts form an atlas, which together with the set forms the manifold. In the rest of the thesis, smoothness is always referred to as C^∞, i.e., infinitely differentiable.

Definition 2.1 (Charts and compatible charts)**.** Suppose M is a set. A chart is the tuple (U, ϕ) where $U \subset M$ is a subset and ϕ is a bijection between U and an open set of $\mathbb{R}^d$. Two charts $(U, \phi), (V, \psi)$ are (smoothly) compatible if either $U \cap V = \emptyset$ or $U \cap V \neq \emptyset$ and satisfy (1) both $\phi(U \cap V), \psi(U \cap V)$ are open sets of $\mathbb{R}^d$; (2) the chart transition map $\phi \circ \psi^{-1}$ is a smooth diffeomorphism (i.e., smooth function and its inverse).

Definition 2.2 (Smooth atlas and manifold)**.** A smooth atlas on M is a collection of compatible charts $A = \{(U_i, \phi_i)\}_{i \in I}$ such that $\cup_{i \in I} U_i = M$. A smooth manifold $\mathcal{M}$ is the tuple (M, A^+) where A^+ is the maximal atlas of A that contains all charts compatible with A.

One trivial example of smooth manifolds is the vector space $\mathbb{R}^d$ with a global identity chart map $\phi = \mathrm{id}$. The definition of manifolds $\mathcal{M}$ via charts and atlas allows manifolds to be locally represented as *linear patches*, which ultimately facilitates tools from calculus to be properly established later. Nevertheless, one can simply treat $\mathcal{M}$ and M equivalently because in most if not all cases, manifolds are described without explicitly constructing an atlas.

2.1.2 Tangent spaces

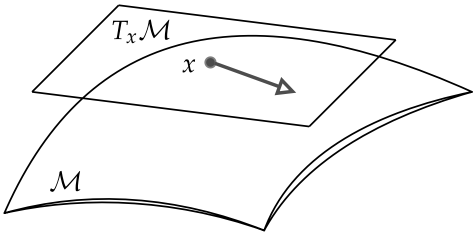


Figure 2.2: Tangent space and tangent vector with respect to an ambient space.

Tangent space is one of the most essential property of a manifold as it represents a linear approximation to the manifold locally around a point. For embedded submanifolds of a vector space, one can imagine the tangent space as a tangent plane to a surface in $\mathbb{R}^d$ as shown in Figure 2.2. This characterization however is made with respect to the ambient space and is usually less desirable. To properly define tangent space intrinsically, consider a smooth curve on the manifold, i.e., $\gamma : I \to \mathcal{M}$ where $I \subseteq \mathbb{R}$ is an interval on the real line. Suppose $\gamma(0) = x$, and denote $\mathfrak{F}(\mathcal{M})$ as the set of smooth functions on manifolds, also called scalar fields. We consider a linear map defined as $v_{\gamma,x} : \mathfrak{F}(\mathcal{M}) \to \mathbb{R}$ such that

$$v_{\gamma,x}(f) := (f \circ \gamma)'(0) = \frac{df(\gamma(t))}{dt}\Big|_{t=0} \tag{2.1}$$

This linear map is known as the directional derivative operator and the tangent space is the collection of all such operators, defined as follows.

Definition 2.3 (Tangent space). For $x \in \mathcal{M}$, the set $T_x\mathcal{M} := \{v_{\gamma,x} \mid \gamma \text{ is smooth}\}$ is called the tangent space to $\mathcal{M}$ at x and the elements $v_{\gamma,x}$ are called tangent vectors.

Although we identify tangent vectors with reference to a curve γ, it is important to notice that there could be infinitely many such curves, which form an equivalence class. With Definition 2.3, it is not difficult to verify that $T_x\mathcal{M}$ is a vector space, i.e., it satisfies $(\alpha v_{\gamma,x} + \beta v_{\delta,x})(f) = \alpha v_{\gamma,x}(f) + \beta v_{\beta,x}(f)$ for any $\alpha, \beta \in \mathbb{R}, f \in \mathfrak{F}(\mathcal{M})$. The dimension of a tangent space is equal to the dimension of the manifold.

Example 2.1. One trivial example of tangent spaces when $\mathcal{M} = \mathbb{R}^d$ is identified as $\mathbb{R}^d$. Another example is when $\mathcal{M}$ is an embedded submanifold of a vector space, e.g. $\mathcal{M} \subset \mathbb{R}^d$. In this case, $\gamma'(t) = \frac{d}{dt}\gamma(t)$ is well-defined. The tangent space thus simplifies to $T_x\mathcal{M} = \{v = \gamma'(0) \mid \gamma \text{ is smooth and } \gamma(0) = x\}$.

It is often useful to study vector fields, which assigns a tangent vector to points on a manifold. Vector field has natural substance in many fields of applications. For example in physics, vector fields are used to model forces and in climate science, vector fields are used to simulate the wind movement (Hutchinson et al., 2021).

Formal definition of vector fields requires the definition of tangent bundle, which is defined as the disjoint union of tangent spaces, i.e., $T\mathcal{M} := \bigcup_{x \in \mathcal{M}} T_x\mathcal{M}$. In addition, let $\pi : T\mathcal{M} \to \mathcal{M}$ be the surjective projection map that $\pi(u) \to x$ for any $u \in T_x\mathcal{M}$. One can construct a (smooth) manifold structure on $T\mathcal{M}$ where vector fields are defined as smooth mapping from $\mathcal{M}$ to $T\mathcal{M}$.

Definition 2.4 (Vector fields)**.** A (smooth) vector field $X : \mathcal{M} \to T\mathcal{M}$ is defined such that $\pi \circ X = \text{id}$. It assigns a tangent vector $X(p) \in T_p\mathcal{M}$ for any $p \in \mathcal{M}$ (also can be written as X_p for notational convenience). The set of smooth vector fields is denoted as $\mathfrak{X}(\mathcal{M})$.

For a vector field $X \in \mathfrak{X}(\mathcal{M})$, its act on a smooth function $f \in \mathfrak{F}(\mathcal{M})$ is a scalar field $Xf \in \mathfrak{F}(\mathcal{M})$ such that for any $p \in \mathcal{M}$, $(Xf)_p = (X_p)(f)$ as defined in (2.1) for $X_p \in T_p\mathcal{M}$. This should be differentiated with fX, which is another vector field, i.e., $fX \in \mathfrak{X}(\mathcal{M})$ and its evaluation at $p \in \mathcal{M}$ is given by $f(p)X_p \in T_p\mathcal{M}$. It can be verified that for any $X, Y \in \mathfrak{X}(\mathcal{M})$, $f, g \in \mathfrak{F}(\mathcal{M})$, $fX + gY \in \mathfrak{X}(\mathcal{M})$.

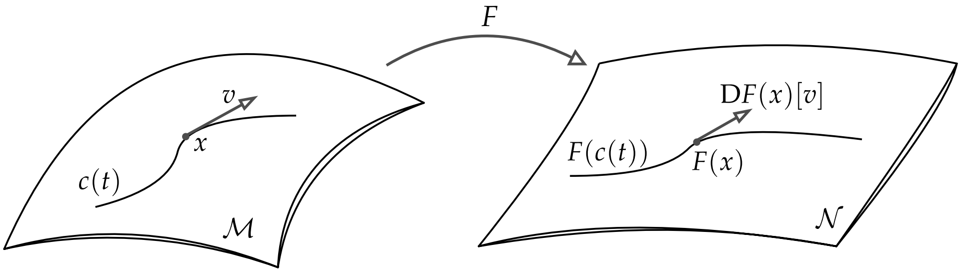


Figure 2.3: Illustration of differential of $F : \mathcal{M} \to \mathcal{N}$.

One important example of vector field on manifolds is Riemannian gradient, which will be defined in the subsequent section where we introduce Riemannian optimization.

2.1.3 Smooth mapping between manifolds

We first define smooth maps between manifolds, where the smoothness on manifolds is characterized as smoothness on the vector space via the chart maps.

Definition 2.5 (Smooth mapping). Let $\mathcal{M}, \mathcal{N}$ be two smooth manifolds. A map $F : \mathcal{M} \to \mathcal{N}$ is smooth, i.e. of C^∞ class if for all $x \in \mathcal{M}$, there exist charts (U, ϕ) of $\mathcal{M}$ that contains x, (V, ψ) of $\mathcal{N}$ that contains $F(x)$ such that $\psi \circ F \circ \phi^{-1} : \phi(U) \to \psi(V)$ is smooth.

Definition 2.6 (Differential of smooth maps). The differential of a smooth map $F : \mathcal{M} \to \mathcal{N}$, denoted as $\mathrm{D}F(x)$ is a linear map from $T_x\mathcal{M}$ to $T_{F(x)}\mathcal{N}$. Its evaluation at $v \in T_x\mathcal{M}$ is denoted as $\mathrm{D}F(x)[v]$, defined as for any $f \in \mathfrak{F}(\mathcal{N})$, $(\mathrm{D}F(x)[v])(f) = v(f \circ F)$ as defined in (2.1).

Proposition 2.1 (Chain rule of differential). *Suppose $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ are smooth manifolds and $F : \mathcal{M}_1 \to \mathcal{M}_2$, $G : \mathcal{M}_2 \to \mathcal{M}_3$ are smooth maps. Then $\mathrm{D}(G \circ F)(x) = \mathrm{D}G(F(x))[\mathrm{D}F(x)]$, which is a linear map from $T_x\mathcal{M}_1$ to $T_{G(F(x))}\mathcal{M}_3$.*

We are particularly interested in the following cases that simplify the abstract definition of differential.

Example 2.2 (Embedded submanifolds of linear spaces)**.** Suppose in Definition 2.6, $\mathcal{M}$ is an embedded submanifold of a linear vector space. Then $\mathrm{D}F(x)[v] = \frac{d}{dt}F(c(t))\big|_{t=0}$ where c is a smooth curve on $\mathcal{M}$ such that $c(0) = x, c'(0) = v$. Such definition naturally holds for the case $\mathcal{M} = \mathbb{R}^d$.

2.1.4 Riemannian manifolds

A Riemannian manifold is a smooth manifold equipped with a smoothly varying inner product structure on each tangent space. Such inner product structure is called a Riemannian metric. This allows to measure the length and angles for tangent vectors. Subsequently, we introduce notions of Levi-Civita connection, distance on manifolds and shortest curves (geodesics), as well as curvature.

Definition 2.7 (Riemannian metric)**.** Let $\mathcal{M}$ be a smooth manifold. For any $x \in \mathcal{M}$, a Riemannian metric $g_x : T_x\mathcal{M} \to T_x\mathcal{M} \to \mathbb{R}$ is a bilinear, symmetric positive definite form that varies smoothly across tangent spaces. Oftentimes, Riemannian metric is written as an inner product $\langle \cdot, \cdot \rangle_x$ where the subscript indicates the base point of the tangent space. The induced norm of a tangent vector $u \in T_x\mathcal{M}$ is given by $\|u\|_x := \sqrt{\langle u, u \rangle_x}$.

To represent a Riemannian metric as mapping from vector fields to scalar fields, we use $g : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{F}(\mathcal{M})$ or $\langle \cdot, \cdot \rangle$ both without the subscript. In some cases, when the tangent space is clear from contexts, we also drop the subscript to represent $\langle \cdot, \cdot \rangle_x$ on $T_x\mathcal{M}$, with a slight abuse of notation.

Definition 2.8 (Riemannian manifold)**.** Let $\mathcal{M}$ be a smooth manifold and g be a Riemannian metric. The pair $(\mathcal{M}, g)$ is a Riemannian manifold.

It should be noted that the choice of different metrics leads to different Riemannian manifolds. When the metric is clear from contexts, we simply use $\mathcal{M}$ to represent the Riemannian manifold. For example, we refer to $\mathbb{R}^d$ as the Euclidean space, which is the vector space $\mathbb{R}^d$ equipped with Euclidean inner product $\langle\cdot,\cdot\rangle_2$. Hence Euclidean space is a special instance of Riemannian manifold.

Riemannian connection. A connection provides a way to differentiate a vector field with respect to another, which is an additional structure for smooth manifolds, parallel to the metric structure. However, we are particularly concerned with the Riemannian connection (also known as Levi-Civita connection), that are affine connections additionally satisfying symmetry and metric compatibility. Riemannian connection, as its name suggests, is specific to smooth manifolds with a metric structure, i.e., Riemannian manifolds. The connections are crucial for defining geodesics and Riemannian Hessian.

Definition 2.9 (Affine connection)**.** For a smooth manifold $\mathcal{M}$, an affine connection is a smooth mapping $\boldsymbol{\nabla} : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$, $(X, Y) \mapsto \boldsymbol{\nabla}_X Y$ that satisfies

(1) $\boldsymbol{\nabla}_{fX+gY} Z = f\boldsymbol{\nabla}_X Z + g\boldsymbol{\nabla}_Y Z$.

(2) $\boldsymbol{\nabla}_X(aY + bZ) = a\boldsymbol{\nabla}_X Y + b\boldsymbol{\nabla}_X Z$.

(3) $\boldsymbol{\nabla}_X(fY) = (Xf)Y + f\boldsymbol{\nabla}_X Y$

for any $X, Y, Z \in \mathfrak{X}(\mathcal{M})$, $f, g \in \mathfrak{F}(\mathcal{M})$, $a, b \in \mathbb{R}$.

For general smooth manifolds, there can be infinitely many affine connections. The next theorem shows that for a Riemannian manifold, there exists a particular affine connection that additionally satisfies two properties.

Theorem 2.1 (Riemannian connection)**.** *For a Riemannian manifold $(\mathcal{M}, g)$, there exists a unique affine connection $\boldsymbol{\nabla}$ that satisfies (1) $\boldsymbol{\nabla}_X Y - \boldsymbol{\nabla}_Y X = [X, Y]$ (symmetry) and (2) $Zg(X, Y) = g(\boldsymbol{\nabla}_Z X, Y) + g(X, \boldsymbol{\nabla}_Z Y)$ (Riemannian metric compatibility). The notation $[X, Y]$ denotes the Lie bracket, which is a vector field, defined as $[X, Y]f = X(Yf) - Y(Xf)$.*

When the manifold is the Euclidean space $\mathbb{R}^d$ with Euclidean inner product, the Riemannian connection reduces to the classic differential of vector fields. From now onward, we only consider Riemannian manifolds with the Riemannian connection.

Riemannian distance and geodesics. For a Riemannian manifold $\mathcal{M}$, and a differentiable curve $\gamma : I \to \mathcal{M}$, where I is an interval on $\mathbb{R}$. Let γ' be the velocity of the curve, which defines a vector field on γ as $\gamma'(t) \in T_{\gamma(t)}\mathcal{M}$. When $\mathcal{M}$ is an embedded submanifold of $\mathbb{R}^d$, then $\gamma'(t) = \lim_{\delta \to 0} \frac{\gamma(t+\delta)-\gamma(t)}{\delta}$.

Riemannian metric provides a measure of the length of such curve as well as the notion of Riemannian distance.

Definition 2.10 (Length of a curve)**.** The length of a differentiable curve $\gamma : I \to \mathcal{M}$ is defined as $\mathrm{length}(\gamma) := \int_I \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)}} dt$.

Definition 2.11 (Riemannian distance)**.** For a Riemannian manifold $\mathcal{M}$, the Riemannian distance is defined as $d : \mathcal{M} \to \mathcal{M} \to \mathbb{R}_+$, such that for any $x, y \in \mathcal{M}$, $d(x,y) := \inf_{\gamma:\gamma(0)=x,\gamma(1)=y} \mathrm{length}(\gamma)$.

In this thesis, we only consider connected Riemannian manifolds, where for each two points, there exists a curve connecting them. This allows the Riemannian distance in Definition 2.11 to satisfy the metric properties, i.e., symmetry, positive definiteness and triangle inequality.

Next we introduce geodesic as generalization of straight lines on manifolds. A geodesic is a curve with zero acceleration, with respect to the Riemannian connection.

Definition 2.12 (Geodesic)**.** For a Riemannian manifold $\mathcal{M}$ with connection $\boldsymbol{\nabla}$, a curve $\gamma : I \to \mathcal{M}$ with open interval I is called a geodesic if it has zero acceleration, i.e., $\boldsymbol{\nabla}_{\gamma'(t)}\gamma'(t) = 0 \in T_{\gamma(t)}\mathcal{M}$ for all $t \in I$.

Definition 2.13 (Geodesic completeness)**.** A Riemannian manifold is geodesic complete if every geodesic can be extended indefinitely, i.e., the interval $I = \mathbb{R}$.

Notably, if the infimum in Definition 2.11 is attained for a curve γ, then γ is a geodesic under constant-speed parameterization. The converse however is not true, i.e., not every geodesic is distance-minimizing. For example, on a sphere, each pair of points can be connected by at least two geodesics. Nonetheless, every geodesic is in fact locally distance-minimizing. Finally, for a geodesic complete Riemannian manifold, every pair of points can always be connected by a geodesic that contains the distance-minimizing geodesic.

Definition 2.14 (Uniquely geodesic)**.** For a subset $\mathcal{X} \subseteq \mathcal{M}$ on a Riemannian manifold, if for any two points $x, y \in \mathcal{X}$, there exists only one geodesic connecting them, the set $\mathcal{X}$ is called uniquely geodesic.

In a uniquely geodesic set $\mathcal{X} \subseteq \mathcal{M}$, Riemannian distance is equal to the length of geodesics. Importantly, there always exists such subset as long as it is sufficiently small. In addition, simply-connected manifolds that are non-positively curved, i.e., all sectional curvatures (defined later) are non-positive, such as hyperbolic space and symmetric positive definite matrices with the affine-invariant metric (Bhatia, 2009), are uniquely geodesic.

Curvature. On a Riemannian manifold, curvature has been critical for convergence of numerical sequences because of its implications for geodesic spreading, angles and Riemannian distance. We particularly introduce Riemann curvature tensor as well as sectional curvature, where the latter notion appears in many convergence analysis of iterative algorithms on Riemannian manifold.

Definition 2.15 (Riemann curvature tensor)**.** The Riemann curvature tensor of a Riemannian manifold $\mathcal{M}$ with connection $\boldsymbol{\nabla}$ is defined as $\mathrm{Riem} : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$ such that for any $X, Y, Z \in \mathfrak{X}(\mathcal{M})$, $\mathrm{Riem}(X, Y)Z := \boldsymbol{\nabla}_X \boldsymbol{\nabla}_Y Z - \boldsymbol{\nabla}_Y \boldsymbol{\nabla}_X Z - \boldsymbol{\nabla}_{[X,Y]} Z$.

It can be shown that Riem is multilinear over $\mathfrak{F}(\mathcal{M})$, i.e., for any $f \in \mathfrak{F}(\mathcal{M})$, it satisfies (1) $\mathrm{Riem}(fX, Y)Z = f\mathrm{Riem}(X, Y)Z$, (2) $\mathrm{Riem}(X, fY)Z = f\mathrm{Riem}(X, Y)Z$,

(3) $\mathrm{Riem}(X,Y)fZ = f\mathrm{Riem}(X,Y)Z$. When evaluated at $p \in \mathcal{M}$, $\mathrm{Riem}_p : T_p\mathcal{M} \times T_p\mathcal{M} \times T_p\mathcal{M} \to T_p\mathcal{M}$ is also multilinear, i.e., $\mathrm{Riem}_p(X_p, Y_p)Z_p$ is linear in each input, $X_p, Y_p, Z_p \in T_p\mathcal{M}$.

Another curvature measure is the sectional curvature. For a manifold with dimension at least two, sectional curvature provides a scalar-valued curvature measure, and is often easier to work with than the Riemann curvature tensor. Sectional curvature measures the curvature of a 2-dimensional linear subspace on each tangent space.

Definition 2.16 (Sectional curvature)**.** For each point x on a Riemannian manifold $\mathcal{M}$, let $\Pi \subseteq T_x\mathcal{M}$ be a linear subspace of dimension 2, spanned by linearly independent tangent vectors $u, v \in T_x\mathcal{M}$, the sectional curvature of Π is

$$K(\Pi) = K(u,v) = \frac{\langle \mathrm{Riem}_x(u,v)v, u\rangle_x}{\|u\|_x^2\|v\|_x^2 - (\langle u,v\rangle_x)^2}.$$

When the tangent vectors are orthonormal with respect to the Riemannian metric, the denominator in Definition 2.16 becomes one. We also remark that sectional curvature uniquely determines the Riemann curvature tensor and hence no information is lost when choosing one over another for analysis.

Remark 2.1 (Relationship with Gaussian curvature)**.** Gaussian curvature provides an intrinsic measure of curvature for a 2-dimensional surface. When $\mathcal{M}$ has dimension 2, the sectional curvature is equivalent to the Gaussian curvature of $\mathcal{M}$. When $\mathcal{M}$ has dimension larger than 2, and for a point $x \in \mathcal{M}$, consider a two-dimensional subspace $\Pi \subset T_x\mathcal{M}$. The union of all geodesics enumerating from x and tangent to Π (i.e., the image of exponential map of Π, defined later), form a two-dimensional submanifold of $\mathcal{M}$. The sectional curvature $K(\Pi)$ turns out to be identical to Gaussian curvature of the submanifold.

Example 2.3. The Euclidean space, sphere with radius 1 and hyperbolic space have constant sectional curvature of 0, 1 and -1 respectively.

2.2 Riemannian optimization

In this section, we introduce ingredients necessary for developing optimization algorithms on Riemannian manifolds. In the Euclidean space, iterative methods for solving unconstrained optimization problems involve updating variables according to some descent directions, which requires the computation of (Euclidean) gradient and/or Hessian. We shall see how such notions can be adapted to Riemannian manifolds for measuring function variations. To update the variables on a manifold, we also need the concepts of exponential map and retraction, which generalize the addition in the Euclidean space. Furthermore, for more advanced algorithms where past update directions are useful for current update, parallel transport and vector transport are needed for relating tangent vectors on different tangent spaces. Next, we also discuss various function classes on Riemannian manifolds, including gradient and function Lipschitzness, smoothness, as well as generalized notions of (strong) convexity and gradient dominance. These are useful for analyzing the convergence and iteration complexities of different numerical algorithms on manifolds.

2.2.1 Riemannian gradient and Hessian

Riemannian gradient and Hessian generalize classic notions of (Euclidean) gradient and Hessian to Riemannian manifolds.

Definition 2.17 (Riemannian gradient)**.** Consider a Riemannian manifold $\mathcal{M}$ and a differentiable function $f : \mathcal{M} \to \mathbb{R}$. The Riemannian gradient of f at $x \in \mathcal{M}$, denoted as $\mathrm{grad} f(x)$, is the unique tangent vector that satisfies $\langle \mathrm{grad} f(x), u \rangle_x = \mathrm{D} f(x)[u]$, for all $u \in T_x\mathcal{M}$.

When $\mathcal{M}$ is an embedded submanifold of the Euclidean space, we have $\mathrm{D} f(x)[u] = \langle \nabla f(x), u \rangle_2$, which is the directional derivative of $f(x)$ along u, where $\nabla f(x)$ denotes the Euclidean gradient. When $\mathcal{M} = \mathbb{R}^d$, one can show

$\mathrm{grad} f(x) = \nabla f(x)$.

Similar to the case of Euclidean gradient, Riemannian gradient can be interpreted as the steepest ascent direction with respect to the Riemannian metric, i.e., $\|\mathrm{grad} f(x)\|_x = \arg\max_{u \in T_x\mathcal{M}: \|u\|_x = 1} \mathrm{D}f(x)[u]$ where the maximum is achieved at $\mathrm{grad} f(x)/\|\mathrm{grad} f(x)\|_x$. Hence, following the direction of negative Riemannian gradient ensures descent in the function value. For this reason, Riemannian steepest descent and Riemannian gradient descent are usually used interchangeably. Nevertheless, the former term is usually referred to Riemannian gradient descent with line-search algorithms and the latter often involves the use of fixed stepsize.

Next we provide the definition of Riemannian Hessian, which is the derivative of gradient vector field $\mathrm{grad} f \in \mathfrak{X}(\mathcal{M})$, via the Riemannian connection $\boldsymbol{\nabla}$.

Definition 2.18 (Riemannian Hessian)**.** Consider a Riemannian manifold $\mathcal{M}$ and a twice-differentiable function $f : \mathcal{M} \to \mathbb{R}$. The Riemannian Hessian of f at x is a symmetric, linear operator, $\mathrm{Hess} f(x) : T_x\mathcal{M} \to T_x\mathcal{M}$, defined as $\mathrm{Hess} f(x)[u] = (\boldsymbol{\nabla}_U \mathrm{grad} f)_x$, for any vector field $U \in \mathfrak{X}(\mathcal{M})$ such that $U_x = u \in T_x\mathcal{M}$.

The symmetry property (self-adjointness) of Riemannian Hessian, claims that $\langle \mathrm{Hess} f(x)[u], v\rangle_x = \langle \mathrm{Hess} f(x)[v], u\rangle_x$ for any $u, v \in T_x\mathcal{M}$ and $x \in \mathcal{M}$.

2.2.2 Moving on manifolds: exponential map and retraction

In the Euclidean space $\mathbb{R}^d$, consider a point x and a direction u (a tangent vector at x). Then the addition $x + u$ can be interpreted as updating x in the direction u. Exponential map generalizes such an idea by following a geodesic enumerating from $x \in \mathcal{M}$ with a velocity $u \in T_x\mathcal{M}$, while staying on the manifold $\mathcal{M}$. However, unlike the Euclidean case, exponential map may not be defined over the entire tangent space, if the manifold is not geodesic complete. Precisely, let $\gamma_{x,u} : I \to \mathcal{M}$ be the unique geodesic such that $\gamma_{x,u}(0) = x, \gamma'_{x,u}(0) = u$, taking

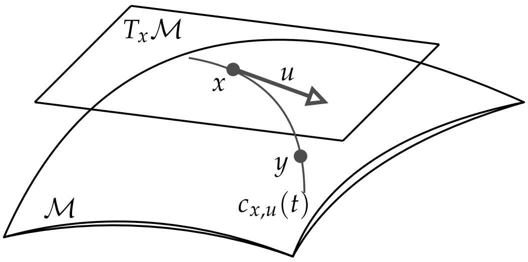


Figure 2.4: Illustration of exponential map and retraction where $c_{x,u}(t)$ is either a geodesic or a retraction curve that satisfies $c_{x,u}(0) = x$, $c_{x,u}(1) = y$ and $c'_{x,u}(0) = u$. Exponential or retraction maps x to y following the direction of u.

to be the maximal for I (J. M. Lee, 2018, Corollary 4.28). Define the subset of tangent spaces $\mathcal{D}_{T\mathcal{M}} := \{(x, u) \in T\mathcal{M} \colon \gamma_{x,u} : I \to \mathcal{M}, [0, 1] \in I\}$. If $\mathcal{M}$ is geodesic complete, then $\mathcal{D}_{T\mathcal{M}} = T\mathcal{M}$ and the exponential map is well-defined over the entire $T_x\mathcal{M}$ for any $x \in \mathcal{M}$.

Definition 2.19 (Exponential map)**.** On a Riemannian manifold $\mathcal{M}$, for any $(x, u) \in \mathcal{D}_{T\mathcal{M}} \subseteq T\mathcal{M}$, the exponential map of u at x is given by $\mathrm{Exp}_x(u) = \gamma_{x,u}(1)$.

Under Definition 2.19, it is not difficult to show the exponential map is smooth on $\mathcal{D}_{T\mathcal{M}}$ and further satisfies the first-order properties $\mathrm{Exp}_x(0) = x$ and $\mathrm{DExp}_x(0)[u] = u$ for any $(x, u) \in \mathcal{D}_{T\mathcal{M}}$, as well as the second-order property of zero acceleration $\gamma''_{x,u}(0) := \nabla_{\gamma'_{x,u}(0)}\gamma'_{x,u}(0) = 0$. Particularly, the first-order properties are crucial to ensure exponential map is a local diffeomorphism (bijective with differentiable inverse). That is, there always exists a (sufficiently small) neighbourhood around the root point such that exponential map has a smooth inverse, which is called inverse exponential map or logarithm map. In a uniquely geodesic domain, exponential map is a diffeomorphism.

Definition 2.20 (Inverse exponential map)**.** Suppose $x, y \in \mathcal{X} \subseteq \mathcal{M}$ where exponential map is a diffeomorphism. The inverse exponential map or logarithm map at a point $x \in \mathcal{M}$, denoted by either Exp_x^{-1} or Log_x is a smooth mapping from $\mathcal{M}$ to $T_x\mathcal{M}$, defined as for any $y \in \mathcal{M}$ such that $\mathrm{Exp}_x(u) = y$, $\mathrm{Log}_x(y) = u$.

Remark 2.2 (Relationship to Riemannian distance). Under the conditions in Definition 2.20, the geodesic $\gamma(t) = \mathrm{Exp}_x(tu)$, $t \in [0,1]$ is the distance minimizing geodesic between x, y, unique up to reparameterization. Hence, the Riemannian distance, $d(x,y) = \|\mathrm{Exp}_x^{-1}(y)\|_x = \|\mathrm{Exp}_y^{-1}(x)\|_y$.

For many practical applications, evaluating exponential map can be extremely expensive particularly for some high-dimensional matrix manifolds. Oftentimes, it is sufficient to consider an approximation to the exponential map, known as retraction.

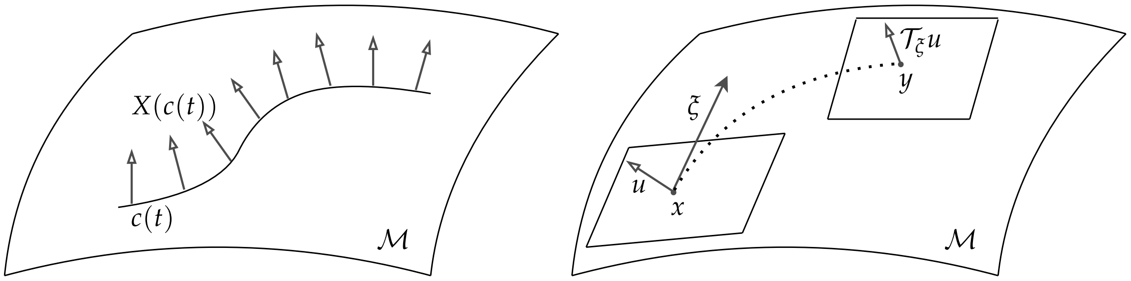
Definition 2.21 (Retraction). A retraction on a manifold $\mathcal{M}$ is a smooth mapping $\mathrm{Retr} : T\mathcal{M} \rightarrow \mathcal{M}$ where it takes any $(x,u) \in T\mathcal{M}$ to $\mathrm{Retr}_x(u)$ that satisfies (1) $\mathrm{Retr}_x(0) = x$ and (2) $\mathrm{DRetr}_x(0)[u] = u$.

Retraction can be seen as first-order approximation of exponential map by satisfying only the first-order properties. This renders exponential map to be a special instance of retraction. Further, retraction can be made into second-order if it satisfies $\boldsymbol{\nabla}_{c'_{x,u}(0)} c'_{x,u}(0) = 0$ where $c_{x,u}(t) := \mathrm{Retr}_x(tu)$ is the retraction curve. In other words, a retraction is second-order if it has zero initial acceleration.

Similar to exponential map, retraction also admits a smooth inverse (locally), called inverse retraction. Such local neighbourhood where retraction is a diffeomorphism is called a totally retractive neighbourhood. Inverse retraction is denoted as $\mathrm{Retr}_x^{-1}(u)$ for $(x,u) \in T\mathcal{M}$ if exists and is similarly defined as in Definition 2.20.

2.2.3 Connecting tangent spaces: parallel transport and vector transport

In the Euclidean space, tangent vectors (or simply vectors) with different root points can be directly compared with one another. On a Riemannian manifold however, we require a specialized tool, called parallel transport, for tangent vec-



(a) Parallel transport

(b) Vector transport

Figure 2.5: (2.5a) A parallel vector field $X(c(t))$ along a given curve $c(t)$, which defines parallel transport along the curve. (2.5b) Vector transport of u along the retraction curve defined by $c(t) = \mathrm{Retr}_x(t\xi)$ with $c(1) = y$.

tors from different tangent spaces to be compared. To define such concept, we require the notion of parallel vector field below.

Definition 2.22 (Parallel vector field)**.** A vector field $X \in \mathfrak{X}(\mathcal{M})$ is called parallel along a curve $c : I \to \mathcal{M}$ if $\boldsymbol{\nabla}_{c'(t)} X(c(t)) = 0$ for all $t \in I$.

It is worth mentioning that here the curve c is not necessarily geodesic and it can be checked that there always exist one and only one such parallel vector field along a given curve c (Boumal, 2023, Theorem 10.34).

Definition 2.23 (Parallel transport)**.** Let $x, y \in \mathcal{M}$ be connected by a curve $c : [t_0, t_1] \to \mathcal{M}$ (not necessarily a geodesic) with $c(t_0) = x, c(t_1) = y$. Parallel transport is a linear map $\Gamma^c_{t_0 \to t_1} : T_x\mathcal{M} \to T_y\mathcal{M}$ (or simply Γ_x^y when the curve is explicit from contexts), defined as $\Gamma^c_{t_0 \to t_1} u = X(c(t_1))$ where X is the parallel vector field along the curve c with $X(c(t_0)) = u \in T_x\mathcal{M}$.

Remark 2.3. Parallel transport admits many useful properties. First, $\Gamma^c_{t_0 \to t_1} \circ \Gamma^c_{t_1 \to t_0} = \mathrm{id}$ and for $t' \in (t_0, t_1)$, we have $\Gamma^c_{t_0 \to t'} \circ \Gamma^c_{t' \to t_1} = \Gamma^c_{t_0 \to t_1}$. More importantly, parallel transport is an isometry with respect to Riemannian metric, i.e., it satisfies $\langle \Gamma^c_{t_0 \to t_1} u, \Gamma^c_{t_0 \to t_1} v \rangle_{c(t_1)} = \langle u, v \rangle_{c(t_0)}$ for any $u, v \in T_{c(t_0)}\mathcal{M}$.

To compute parallel transport, one often needs to solve differential equations on manifolds when no closed-form solutions are available. This can be computationally prohibitive. Thankfully, there exist some first-order approximations to parallel transport, called vector transports, which are often sufficient for practical purposes. Vector transports are defined with respect to a given retraction.

Definition 2.24 (Vector transport)**.** Let $T\mathcal{M} \oplus T\mathcal{M} := \{(u, v) : u, v \in T_x\mathcal{M}, x \in \mathcal{M}\}$ denote the Whitney sum of tangent bundle. Vector transport is a smooth mapping $\mathcal{T} : T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M}$ that satisfies for all $x \in \mathcal{M}$,

(1) there exists a retraction Retr such that $\mathcal{T}_\xi u \in T_{\mathrm{Retr}_x(\xi)}\mathcal{M}$ (*associated retraction*),

(2) $\mathcal{T}_0 v = v$ (*consistency*),

(3) $\mathcal{T}_\xi(au + bv) = a\mathcal{T}_\xi u + b\mathcal{T}_\xi v$ (*linearity*).

Remark 2.4. For most cases, it is desirable to treat vector transport as a mapping between tangent spaces along a curve, similar to parallel transport. For this purpose, we denote $c(t) = \mathrm{Retr}_x(t\xi)$ as the retraction curve where $c(0) = x, c(1) = y$. We denote $\mathcal{T}_x^y : T_x\mathcal{M} \to T_y\mathcal{M}$ as the vector transport, defined as $\mathcal{T}_x^y u := \mathcal{T}_{c'(0)} u = \mathcal{T}_\xi u$. Without mentioning otherwise, we will be using such notation for the rest of the thesis for representing vector transport.

In particular, an isometric vector transport similarly preserves angles and lengths as parallel transport, i.e., $\langle \mathcal{T}_x^y u, \mathcal{T}_x^y v\rangle_y = \langle u, v\rangle_x$.

2.2.4 Function classes on Riemannian manifolds

To properly characterize and analyze the behaviours of optimization algorithms on manifolds, this section discusses various function classes of interest on Riemannian manifolds.

Geodesic convexity. We start with a generalized notion of convexity on manifolds, called geodesic convexity for both functions and sets. We adopt the definition of geodesic convex set in H. Zhang & Sra (2016); Boumal (2023), which is sufficient for the purpose of optimization.

Definition 2.25 (Geodesic convex set)**.** A subset $\mathcal{X} \subseteq \mathcal{M}$ is called geodesic convex if for any pair $x, y \in \mathcal{X}$, there exists a geodesic $\gamma : [0,1] \to \mathcal{M}$ satisfying $\gamma(0) = x, \gamma(1) = y$ and for all $t \in [0,1]$, $\gamma(t) \in \mathcal{X}$.

It is important to note that Definition 2.25 only requires *some*, rather than *all* geodesics connecting x, y to lie entirely in $\mathcal{X}$. The stronger notion of convexity is known as (geodesic) total convexity (Vishnoi, 2018), which we do not pursue in this thesis.

Example 2.4. Any connected, geodesic complete, and non-positively curved Riemannian manifold is geodesic convex. This includes the Euclidean space, hyperbolic space and positive definite matrices with affine-invariant or log-Euclidean metric. For general manifolds, any sufficiently small subset is always geodesic convex.

Definition 2.26 (Geodesic convex function)**.** For a geodesic convex set $\mathcal{X} \subseteq \mathcal{M}$, a real-valued function $f : \mathcal{X} \to \mathbb{R}$ is (strictly) geodesic convex if for all geodesics $\gamma : [0,1] \to \mathcal{M}$ that lies entirely in $\mathcal{X}$, $f \circ \gamma$ is (strictly) convex on $[0,1]$. That is, f is geodesic convex if $f(\gamma(t)) \leq (1-t)f(\gamma(0)) + tf(\gamma(1))$ and f is geodesic strictly convex if the equality only holds when $t = 0, 1$.

Definition 2.27 (Geodesic strongly convex function)**.** For a geodesic convex set $\mathcal{X} \subseteq \mathcal{M}$, a real-valued function $f : \mathcal{X} \to \mathbb{R}$ is called geodesic strongly convex with parameter μ if for all geodesics $\gamma : [0,1] \to \mathcal{M}$ that lies entirely in $\mathcal{X}$, it satisfies $f(\gamma(t)) \leq (1-t)f(\gamma(0)) + tf(\gamma(1)) - \frac{t(1-t)}{2}\mu\|\gamma'(0)\|_{\gamma(0)}$.

Remark 2.5. For geodesic convex functions, any local minimizer is also a global minimizer and for geodesic strictly convex functions, there exists at most one local minimizer, which is the global minimizer.

In this thesis, we focus on differentiable functions on manifolds, with well-defined Riemannian gradient and Hessian. In this case, we have the following equivalent first-order and second-order characterizations of geodesic (strong) convexity in terms of Riemannian gradient and Hessian respectively.

Proposition 2.2 (First-order characterizations of geodesic convexity). *Let $\mathcal{X} \subseteq \mathcal{M}$ be a geodesic convex set and consider $f : \mathcal{M} \to \mathbb{R}$ to be differentiable. Then for all geodesics $\gamma : [0,1] \to \mathcal{M}$ that are contained entirely in $\mathcal{X}$, with $\gamma(0) = x, \gamma'(0) = u$, we have*

(1) $f : \mathcal{X} \to \mathbb{R}$ is geodesic convex if and only if it satisfies $f(\mathrm{Exp}_x(tu)) \geq f(x) + t\langle \mathrm{grad} f(x), u\rangle_x$ for all $t \in [0,1]$;

(2) $f : \mathcal{X} \to \mathbb{R}$ is geodesic strictly convex if and only if $f(\mathrm{Exp}_x(tu)) > f(x) + t\langle \mathrm{grad} f(x), u\rangle_x$ whenever $u \neq 0$, for all $t \in (0,1]$;

(3) $f : \mathcal{X} \to \mathbb{R}$ is geodesic μ-strongly convex for some $\mu > 0$ if and only if $f(\mathrm{Exp}_x(tu)) \geq f(x) + t\langle \mathrm{grad} f(x), u\rangle_x + t^2 \frac{\mu}{2}\|u\|_x^2$ for all $t \in [0,1]$,

where for all the cases, we notice $\gamma(t) = \mathrm{Exp}_x(tu)$. If the geodesic γ happens to be distance minimizing, we can replace $\|u\|_x$ with the Riemannian distance $d(x, \mathrm{Exp}_x(tu))$.

Proposition 2.3 (Second-order characterizations of geodesic convexity). *Under the same settings as in Proposition 2.2, the function $f : \mathcal{X} \to \mathbb{R}$ is (1) geodesic convex if and only if $\mathrm{Hess} f(x) \succeq 0$, (2) geodesic strictly convex if $\mathrm{Hess} f(x) \succ 0$, (3) geodesic strongly convex if and only if $\mathrm{Hess} f(x) \succeq \mu\, \mathrm{id}$, for all $x \in \mathcal{X}$.*

The second-order characterizations in Proposition 2.3 can be also equivalently translated as follows.

Corollary 2.1. *Under settings of Proposition 2.3, the function f is geodesic convex if and only if $\frac{d^2 f(\gamma(t))}{dt^2} \geq 0$ for all valid geodesics contained in $\mathcal{X}$ and geodesic strongly convex if and only if $\frac{d^2 f(\gamma(t))}{dt^2} \geq \mu$ for all valid geodesics with $\|\gamma'(0)\|_{\gamma(0)} = 1$.*

Proof. The proof follows from that $\frac{df(\gamma(t))}{dt} = \langle \mathrm{grad} f(\gamma(t)), \gamma'(t)\rangle_{\gamma(t)}$ and

$$\frac{d^2 f(\gamma(t))}{dt^2} = \langle \mathrm{Hess} f(\gamma(t))[\gamma'(t)], \gamma'(t)\rangle_{\gamma(t)} + \langle \mathrm{grad} f(\gamma(t)), \gamma''(t)\rangle_{\gamma(t)}$$

$$= \langle \mathrm{Hess} f(\gamma(t))[\gamma'(t)], \gamma'(t) \rangle_{\gamma(t)}$$

where we use the fact $\gamma''(t) = 0$ for any geodesic γ. The proof is complete by noticing $\|\gamma'(t)\|_{\gamma(t)} = \|\gamma'(0)\|_{\gamma(0)} = 1$ due to constant velocity of geodesics. □

Remark 2.6. Corollary 2.1 translates the geodesic (strong) convexity of f to the Euclidean (strong) convexity of $f \circ \gamma$.

The same as for the Euclidean case, geodesic convexity leads to the so-called first-order stationarity at optimality.

Proposition 2.4. *If f is geodesic convex on an open geodesic convex set and is differentiable, then x^* is a global minimizer of f is and only if* $\mathrm{grad} f(x^*) = 0$.

Geodesic Lipschitzness and smoothness. Lipschitz continuity and smoothness are standard regularity conditions for functional analysis in the Euclidean space. On general Riemannian manifolds, these notions translate to geodesic Lipschitzness and smoothness.

We first introduce geodesic Lipschitz gradient, which has close connections to bounded Hessian and function smoothness.

Definition 2.28 (Geodesic Lipschitz gradient)**.** A differentiable function $f : \mathcal{M} \to \mathbb{R}$ has geodesic L-Lipschitz gradient if for all $x, y \in \mathcal{M}$ such that $y = \mathrm{Exp}_x(u)$ in the domain of the exponential map, we have

$$\|\Gamma_{\gamma(t)}^{x} \mathrm{grad} f(\gamma(t)) - \mathrm{grad} f(x)\|_x \le L \|tu\|_x,$$

for all $t \in [0, 1]$ and $\gamma(t) = \mathrm{Exp}_x(tu)$.

Proposition 2.5 (Lipschitz gradient and bounded Hessian)**.** *A twice-differentiable function $f : \mathcal{M} \to \mathbb{R}$ has geodesic L-Lipschitz gradient if and only if its Hessian is upper bounded, i.e., $\|\mathrm{Hess} f(x)\|_x := \max_{u \in T_x\mathcal{M} : \|u\|_x = 1} \|\mathrm{Hess} f(x)[u]\|_x \le L$, for all $x \in \mathcal{M}$, where $\|\mathrm{Hess} f(x)\|_x$ denotes the operator norm of Riemannian Hessian.*

Corollary 2.2. *A twice-differentiable function $f : \mathcal{M} \to \mathbb{R}$ has geodesic L-Lipschitz gradient if and only if $\frac{d^2 f(\gamma(t))}{dt^2} \leq L$ for any geodesics satisfying $\|\gamma'(0)\|_{\gamma(0)} = 1$.*

Proof. The proof follows from that of Corollary 2.1. □

Proposition 2.6 (Lipschitz gradient and function smoothness)**.** *If a differentiable function f has geodesic L-Lipschitz gradient, then function f is geodesic L-smooth, which satisfies*

$$|f(y) - f(x) - \langle \mathrm{grad} f(x), u\rangle_x| \leq \frac{L}{2}\|u\|_x^2,$$

for all $x, y \in \mathcal{M}$ such that $y = \mathrm{Exp}_x(u)$.

Further, the notions of Lipschitzness can be defined for higher-order derivatives on Riemannian manifolds. Below we introduce the Hessian Lipschitzness.

Lemma 2.1 (Geodesic Hessian Lipschitzness)**.** *A function f has geodesic ρ-Lipschitz Hessian in $\mathcal{M}$ if for all $x, y \in \mathcal{M}$ such that $y = \mathrm{Exp}_x(u)$ in the domain of the exponential map, we have*

$$\|\Gamma_{\gamma(t)}^{x} \circ \mathrm{Hess} f(\gamma(t)) \circ \Gamma_{x}^{\gamma(t)} - \mathrm{Hess} f(x)\|_x \leq \rho\|tu\|_x^3,$$

for all $t \in [0, 1]$ and $\gamma(t) := \mathrm{Exp}_x(tu)$. If function f has geodesic ρ-Lipschitz Hessian, then function f satisfies

$$|f(y) - f(x) - \langle \mathrm{grad} f(x), u\rangle_x - \frac{1}{2}\langle u, \mathrm{Hess} f(x)[u]\rangle_x| \leq \frac{\rho}{6}\|u\|_x^3$$
$$\|\Gamma_y^x \mathrm{grad} f(y) - \mathrm{grad} f(x) - \mathrm{Hess} f(x)[u]\|_x \leq \frac{\rho}{2}\|u\|^2.$$

Extensions to general retractions and vector transports. All the aforementioned notions, including geodesic convexity and geodesic Lipschitzness and smoothness, also have well-defined analogues for more general retractions and vector transports, which are mainly developed in W. Huang, Gallivan, & Absil (2015).

The following definitions of retraction convexity generalize geodesic convexity by considering general retraction curves other than geodesics.

Definition 2.29 (Retraction convex set)**.** A subset $\mathcal{X} \subseteq \mathcal{M}$ is called a retraction convex set with respect to a retraction Retr if for $x, y \in \mathcal{X}$, there exists a retraction curve $c(t) := \mathrm{Retr}_x(t\xi)$ satisfying $c(1) = y$ and $c(t)$ lies entirely in $\mathcal{X}$.

Definition 2.30 (Retraction convex function)**.** For a retraction convex set $\mathcal{X} \subseteq \mathcal{M}$, a function $f : \mathcal{X} \to \mathbb{R}$ is (strictly) retraction convex if for all $x \in \mathcal{M}$ and retraction curve $c : [0, 1] \to \mathcal{S}$, $f \circ c$ is (strictly) convex in $t \in [0, 1]$.

Definition 2.31 (Retraction strongly convex function)**.** For a retraction convex set $\mathcal{X} \subseteq \mathcal{M}$, a function $f : \mathcal{X} \to \mathbb{R}$ is called retraction μ-strongly convex for some constant $\mu > 0$, if for all $x \in \mathcal{M}$ and retraction curve $c(t) = \mathrm{Retr}_x(t\xi)$ with $\|\xi\|_x = 1$, it satisfies that $\frac{d^2 f(\mathrm{Retr}_x(t\xi))}{dt^2} \geq \mu$ for all $t \geq 0$ such that $c|_{[0,t]}$ lies entirely in $\mathcal{X}$.

There exist similar equivalent characterizations of retraction (strong) convexity, which we refer readers to W. Huang, Gallivan, & Absil (2015); W. Huang, Absil, & Gallivan (2015) for more details.

Next we introduce function smoothness and gradient Lipschitzness with respect to a retraction.

Definition 2.32 (Retraction smoothness)**.** Under settings of Definition 2.31, a function $f : \mathcal{X} \to \mathbb{R}$ is called retraction L-smooth for some constant $L > 0$ with respect to a retraction Retr, if for all $x \in \mathcal{M}$ and retraction curve $c(t) = \mathrm{Retr}_x(t\xi)$ with $\|\xi\|_x = 1$, it satisfies that $\frac{d^2 f(\mathrm{Retr}_x(t\xi))}{dt^2} \leq L$ for all $t \geq 0$ such that $c|_{[0,t]}$ lies entirely in $\mathcal{X}$.

Proposition 2.7. *If $f : \mathcal{X} \to \mathbb{R}$ is retraction L-smooth as per Definition 2.32, then we have for all $x, y \in \mathcal{X}$ such that $y = \mathrm{Retr}_x(u)$ and the constant L, we have*

$$f(y) - f(x) - \langle \mathrm{grad} f(x), u \rangle_x \leq \frac{L}{2} \|u\|_x^2.$$

Proof. The result is proved in (W. Huang, Gallivan, & Absil, 2015, Lemma 3.2). □

It is worth highlighting that unlike the case for exponential map, the use of retraction does not necessarily ensure correspondence between Lipschitz gradient and function smoothness. We particularly define retraction Lipschitzness as follows, where the constant $L_\ell \neq L$ in general.

Definition 2.33 (Retraction Lipschitz gradient)**.** Under settings of Definition 2.32, a function $f : \mathcal{X} \to \mathbb{R}$ has retraction L_ℓ-Lipschitz gradient for some constant $L_\ell > 0$ with respect to a retraction Retr, if for all retraction curves $c : [0, 1] \to \mathcal{X}$ that lies entirely in $\mathcal{X}$ with $c(0) = x, c(1) = y$ such that $y = \mathrm{Retr}_x(u)$, we have

$$\|\Gamma^c_{1\to 0}\mathrm{grad} f(y) - \mathrm{grad} f(x)\|_x \leq L_\ell \|u\|_x,$$

where we recall $\Gamma^c_{1\to 0}$ is the parallel transport along curve c from y to x.

When the retraction is the exponential map, we can see from Proposition 2.6 that $L_\ell = L$ and the retraction curve c becomes a geodesic.

Riemannian PL condition. The Polyak-Łojasiewic (PL) condition (Polyak, 1963) is a sufficient condition for establishing linear convergence of first-order optimization solvers to global optimal solutions. The PL condition is weaker than strong convexity as functions satisfying PL condition can be nonconvex in general. The PL condition can be adapted to Riemannian manifolds in a straightforward manner as follows.

Definition 2.34 (Riemannian PL condition)**.** For a differentiable function $f : \mathcal{M} \to \mathbb{R}$, consider a neighbourhood $\mathcal{X}$ that contains a *global* minimizer x^* of f, i.e., $x^* = \arg\min_{x \in \mathcal{M}} f(x)$. The function satisfies Riemannian PL condition in $\mathcal{X}$ with constant $\tau > 0$ (also called τ-gradient dominance) if for any $x \in \mathcal{X}$, it satisfies

$$f(x) - f(x^*) \leq \tau \|\mathrm{grad} f(x)\|_x^2.$$

However, we shall notice that the global solution x^* needs not to be unique, thus including many nonconvex functions. In particular, as shown in H. Zhang et al. (2016), the objective of computing the leading eigenvector on sphere satisfies the Riemannian PL condition with high probability, while it is both nonconvex in the Euclidean sense and geodesic nonconvex on Riemannian manifolds.

Chapter 3

Optimization on SPD matrices with Bures-Wasserstein geometry

Learning on symmetric positive definite (SPD) matrices is a fundamental problem in various machine learning applications, including metric and kernel learning (Tsuda et al., 2005; Guillaumin et al., 2009; P. K. Jawanpuria et al., 2015; Bhutani et al., 2018; Suárez et al., 2021), medical imaging (Pennec et al., 2006; Pennec, 2020), natural language processing (P. Jawanpuria et al., 2019, 2020a), computer vision (M. T. Harandi et al., 2014; Z. Huang et al., 2017; Z. Huang & Gool, 2017), multi-task learning (P. Jawanpuria & Mishra, 2018; Nimishakavi et al., 2018), domain adaptation (Mahadevan et al., 2019; P. Jawanpuria et al., 2021), modeling time-varying data (Brooks et al., 2019), object detection (Tuzel et al., 2008), and quantum mechanics (Mishra et al., 2020; Luchnikov et al., 2021). The set of SPD matrices of size $n \times n$, defined as $\mathbb{S}_{++}^n := \{\mathbf{X} : \mathbf{X} \in \mathbb{R}^{n \times n}, \mathbf{X}^\top = \mathbf{X}, \text{ and } \mathbf{X} \succ \mathbf{0}\}$, has a smooth manifold structure with a richer geometry than the Euclidean space. When endowed with a metric (inner product structure), the set of SPD matrices becomes a Riemannian manifold (Bhatia, 2009). Hence, numerous existing works (Pennec et al., 2006; Arsigny et al., 2007; Jayasumana et al., 2013; Z. Huang et al., 2015; Z. Huang & Gool, 2017; Z. Lin, 2019; Pennec, 2020) have studied and employed the Riemannian optimization framework for learning over the space of SPD matrices (Absil et al., 2009; Boumal, 2023).

Several Riemannian metrics on $\mathbb{S}^n_{++}$ have been proposed such as the Affine-Invariant (Pennec et al., 2006; Bhatia, 2009), the Log-Euclidean (Arsigny et al., 2007; Quang et al., 2014), and the Log-Cholesky (Z. Lin, 2019), to name a few. One can additionally obtain different families of Riemannian metrics on $\mathbb{S}^n_{++}$ by appropriate parameterization based on the principles of invariance and symmetry (Dryden et al., 2009; Chebbi & Moakher, 2012; Thanwerdas & Pennec, 2019; Pennec, 2020). However, to the best of our knowledge, a systematic study comparing the different metrics for optimizing generic cost functions defined on $\mathbb{S}^n_{++}$ is missing. Practically, the Affine-Invariant (AI) metric seems to be the most widely used metric in Riemannian first-order and second-order algorithms (e.g., steepest descent, conjugate gradients, and trust regions) as it is the only Riemannian SPD metric available in toolboxes specifically for manifold optimization, such as Manopt.jl (Bergmann, 2019), Pymanopt (Townsend et al., 2016), ROPTLIB (W. Huang et al., 2018), and McTorch (Meghwanshi et al., 2018). Moreover, many interesting problems in machine learning are found to be geodesic convex (generalization of Euclidean convexity) under the AI metric, which allows fast convergence of optimization algorithms (H. Zhang et al., 2016; R. Hosseini & Sra, 2020).

Recent works have studied the Bures-Wasserstein (BW) distance on SPD matrices (Malagò et al., 2018; Bhatia et al., 2019; van Oostrum, 2020). It is a well-known result that the Wasserstein distance between two multivariate Gaussian densities is a function of the BW distance between their covariance matrices. Indeed, the BW metric is a Riemannian metric. Under this metric, the necessary tools for Riemannian optimization, including the Riemannian gradient and Hessian expressions, can be efficiently computed (Malagò et al., 2018). Hence, it is a promising candidate for Riemannian optimization on $\mathbb{S}^n_{++}$.

In this chapter, we theoretically and empirically analyze the quality of optimization with the BW geometry and show that it is a viable alternative to the default choice of AI geometry. Our analysis discusses the classes of cost functions (e.g., polynomial) for which the BW metric has better convergence rates than the AI metric. We also discuss cases (e.g., log-det) where the reverse is true. In particular, our contributions for this chapter are as follows.

- We observe that the BW metric has a linear dependence on SPD matrices while the AI metric has a quadratic dependence. We show this impacts the condition number of the Riemannian Hessian and makes the BW metric more suitable to learning ill-conditioned SPD matrices than the AI metric.
- In contrast to the non-positively curved AI geometry, the BW geometry is shown to be non-negatively curved, which leads to a tighter trigonometry distance bound and faster convergence rates for optimization algorithms.
- For both metrics, we analyze the convergence rates of Riemannian steepest descent and trust region methods and highlight the issues arising from the differences in the curvature and condition number of the Riemannian Hessian.
- We verify that common optimization problems that are geodesic convex under the AI metric are also geodesic convex under the BW metric.
- We support our analysis with extensive experiments on applications such as weighted least squares, trace regression, metric learning, and Gaussian mixture model.

3.1 Preliminaries and backgrounds

This section builds upon Chapter 2 where we review the classic Riemannian optimization solvers, Riemannian steepest descent and trust region as representatives for first-order and second-order methods respectively, for the purpose of the subsequent analysis. In addition, the spectrum of Riemannian Hessian is also introduced, which is crucial for quantifying the linear versus quadratic dependence of BW and AI metrics.

Riemannian steepest descent and Riemannian trust region. The *Riemannian steepest descent* method (Udriste, 1994) generalizes the standard gradient descent in the Euclidean space to Riemannian manifolds by ensuring that the updates are along the geodesic and stay on the manifolds. That is, $x_{t+1} = \mathrm{Exp}_{x_t}(-\eta_t \, \mathrm{grad} f(x_t))$ for some step size η_t, often set as fixed or computed via a line-search algorithm.

Second-order methods such as trust region and cubic regularized Newton methods are generalized to Riemannian manifolds (Absil et al., 2007; N. Agarwal et al., 2021). Both the trust region and cubic regularized Newton methods are Hessian-free in the sense that only evaluation of the Hessian acting on a tangent vector, i.e., $\mathrm{Hess} f(x)[u]$ is required. Similar to the Euclidean counterpart, the *Riemannian trust region* method approximates the Newton step by solving a subproblem, i.e.,

$$\min_{u \in T_{x_t}\mathcal{M} : \|u\|_{x_t} \leq \Delta} m_{x_t}(u) = f(x_t) + \langle \mathrm{grad} f(x_t), u \rangle_{x_t} + \frac{1}{2} \langle \mathcal{H}_{x_t}[u], u \rangle_{x_t},$$

where $\mathcal{H}_{x_t} : T_{x_t}\mathcal{M} \to T_{x_t}\mathcal{M}$ is a symmetric and linear operator that approximates the Riemannian Hessian. Δ is the radius of trust region, which may be increased or decreased depending on how model value $m_{x_t}(u)$ changes. The subproblem is solved iteratively using a truncated conjugate gradient algorithm. The next iterate is given by $x_{t+1} = \mathrm{Exp}_{x_t}(u)$ with the optimized u.

Spectrum of Riemannian Hessian. Next, we introduce the eigenvalues and the condition number of the Riemannian Hessian, which we critically rely on for the analysis in the following sections.

Definition 3.1. The minimum and maximum eigenvalues of $\mathrm{Hess} f(x)$ are de-

Table 3.1: Riemannian optimization ingredients for AI and BW geometries.

	Affine-Invariant	Bures-Wasserstein
R.Metric	$g_{\mathrm{ai}}(\mathbf{U}, \mathbf{V}) = \mathrm{tr}(\mathbf{X}^{-1}\mathbf{U}\mathbf{X}^{-1}\mathbf{V})$	$g_{\mathrm{bw}}(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\mathrm{tr}(\mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathbf{V})$
R.Exp	$\mathrm{Exp}_{\mathrm{ai},\mathbf{X}}(\mathbf{U}) = \mathbf{X}\exp(\mathbf{X}^{-1}\mathbf{U})$	$\mathrm{Exp}_{\mathrm{bw},\mathbf{X}}(\mathbf{U}) = \mathbf{X} + \mathbf{U} + \mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathbf{X}\mathcal{L}_{\mathbf{X}}[\mathbf{U}]$
R.Gradient	$\mathrm{grad}_{\mathrm{ai}} f(\mathbf{X}) = \mathbf{X}\nabla f(\mathbf{X})\mathbf{X}$	$\mathrm{grad}_{\mathrm{bw}} f(\mathbf{X}) = 4\{\nabla f(\mathbf{X})\mathbf{X}\}_{\mathrm{S}}$
R.Hessian	$\mathrm{Hess}_{\mathrm{ai}} f(\mathbf{X})[\mathbf{U}] = \mathbf{X}\nabla^2 f(\mathbf{X})[\mathbf{U}]\mathbf{X} + \{\mathbf{U}\nabla f(\mathbf{X})\mathbf{X}\}_{\mathrm{S}}$	$\mathrm{Hess}_{\mathrm{bw}} f(\mathbf{X})[\mathbf{U}] = 4\{\nabla^2 f(\mathbf{X})[\mathbf{U}]\mathbf{X}\}_{\mathrm{S}} + 2\{\nabla f(\mathbf{X})\mathbf{U}\}_{\mathrm{S}} + 4\{\mathbf{X}\{\mathcal{L}_{\mathbf{X}}[\mathbf{U}]\nabla f(\mathbf{X})\}_{\mathrm{S}}\}_{\mathrm{S}} - \{\mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathrm{grad}_{\mathrm{bw}} f(\mathbf{X})\}_{\mathrm{S}}$

fined as $\lambda_{\min} = \min_{\|u\|_x^2=1} \langle \mathrm{Hess} f(x)[u], u\rangle_x$, $\lambda_{\max} = \max_{\|u\|_x^2=1} \langle \mathrm{Hess} f(x)[u], u\rangle_x$. The condition number of $\mathrm{Hess} f(x)$ is defined as $\kappa(\mathrm{Hess} f(x)) := \lambda_{\max}/\lambda_{\min}$.

3.2 Comparing BW with AI for Riemannian optimization

This section starts with an observation of a linear-versus-quadratic dependency between the two metrics. From this observation, we analyze the condition number of the Riemannian Hessian. Then, we further compare the sectional curvature of the two geometries. Together with the differences in the condition number, this allows us to compare the convergence rates of optimization algorithms on the two geometries. We conclude this section by showing geodesic convexity of several generic cost functions under the BW geometry. The proofs for this section are included in Appendix.

AI and BW geometries on SPD matrices. When endowed with a Riemannian metric g, the set of SPD matrices of size n becomes a Riemannian manifold $\mathcal{M} = (\mathbb{S}^n_{++}, g)$. The tangent space at $\mathbf{X}$ is $T_{\mathbf{X}}\mathcal{M} := \{\mathbf{U} : \mathbf{U} \in \mathbb{R}^{n\times n} \text{ and } \mathbf{U}^\top = \mathbf{U}\}$. Under the AI and BW metrics, the Riemannian exponential map, Riemannian gradient, and Hessian are compared in Table 3.1, where we denote $\{\mathbf{A}\}_{\mathrm{S}} :=$

$(\mathbf{A}+\mathbf{A}^\top)/2$ and $\exp(\mathbf{A})$ as the matrix exponential of $\mathbf{A}$. $\mathcal{L}_{\mathbf{X}}[\mathbf{U}]$ is the solution to the matrix linear system $\mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathbf{X}+\mathbf{X}\mathcal{L}_{\mathbf{X}}[\mathbf{U}]=\mathbf{U}$ and is known as the Lyapunov operator. We use $\nabla f(\mathbf{X})$ and $\nabla^2 f(\mathbf{X})$ to represent the first-order and second-order derivatives, i.e., the Euclidean gradient and Hessian, respectively. The derivations in Table 3.1 can be found in Pennec (2020); Bhatia et al. (2019). In the rest of the chapter, we use $\mathcal{M}_{\mathrm{ai}}$ and $\mathcal{M}_{\mathrm{bw}}$ to denote the SPD manifolds under the two metrics.

From Table 3.1, the computational costs for evaluating the AI and BW ingredients are dominated by the matrix exponential/inversion operations and the Lyapunov operator $\mathcal{L}$ computation, respectively. Both at most cost $O(n^3)$, which implies a comparable per-iteration cost of optimization algorithms between the two metric choices. This claim is validated in Section 3.3.

A key observation. From Table 3.1, the Affine-Invariant metric on the SPD manifold can be rewritten as for any $\mathbf{U},\mathbf{V}\in T_{\mathbf{X}}\mathcal{M}$,

$$\langle \mathbf{U},\mathbf{V}\rangle_{\mathrm{ai}} = \mathrm{tr}(\mathbf{X}^{-1}\mathbf{U}\mathbf{X}^{-1}\mathbf{V}) = \mathrm{vec}(\mathbf{U})^\top(\mathbf{X}\otimes\mathbf{X})^{-1}\mathrm{vec}(\mathbf{V}), \tag{3.1}$$

where $\mathrm{vec}(\mathbf{U})$ and $\mathrm{vec}(\mathbf{V})$ are the vectorizations of $\mathbf{U}$ and $\mathbf{V}$, respectively. Note that we omit the subscript $\mathbf{X}$ for inner product $\langle\cdot,\cdot\rangle$ to simplify the notation. The specific tangent space where the inner product is computed should be clear from contexts.

The Bures-Wasserstein metric is rewritten as, for any $\mathbf{U},\mathbf{V}\in T_{\mathbf{X}}\mathcal{M}$,

$$\langle \mathbf{U},\mathbf{V}\rangle_{\mathrm{bw}} = \frac{1}{2}\mathrm{tr}(\mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathbf{V}) = \frac{1}{2}\mathrm{vec}(\mathbf{U})^\top(\mathbf{X}\oplus\mathbf{X})^{-1}\mathrm{vec}(\mathbf{V}), \tag{3.2}$$

where $\mathbf{X}\oplus\mathbf{X}=\mathbf{X}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{X}$ is the Kronecker sum.

Remark 3.1. Comparing Eq. (3.1) and (3.2) reveals that the BW metric has a linear dependence on $\mathbf{X}$ while the AI metric has a quadratic dependence. This suggests that optimization algorithms under the AI metric should be more sensitive to the condition number of $\mathbf{X}$ compared to the BW metric.

The above observation serves as a key motivation for further analysis.

3.2.1 Condition number of Riemannian Hessian at optimality

Throughout the rest of the chapter, we make the following assumptions.

Assumption 3.1. (a). f is at least twice continuously differentiable with a non-degenerate local minimizer $\mathbf{X}^*$. (b). The subset $\mathcal{X} \subseteq \mathcal{M}$ (usually as a neighbourhood of a center point) we consider throughout this chapter is totally normal, i.e., the exponential map is a diffeomorphism.

Assumption 3.1 is easy to satisfy. Particularly, Assumption 3.1(b) is guaranteed for the SPD manifold under the AI metric because its geodesic is unique. Under the BW metric, for a center point $\mathbf{X}$, we can choose the neighbourhood such that $\mathcal{X} = \{\mathrm{Exp}_{\mathbf{X}}(\mathbf{U}) : \mathbf{I} + \mathcal{L}_{\mathbf{X}}[\mathbf{U}] \in \mathbb{S}^n_{++}\}$ as in Malagò et al. (2018). In other words, $\mathcal{X}$ is assumed to be unique-geodesic under both the metrics.

We now formalize the impact of the linear-versus-quadratic dependency, highlighted in Remark 3.1. At a local minimizer $\mathbf{X}^*$ where the Riemannian gradient vanishes, we first simplify the expression for the Riemannian Hessian in Table 3.1.

On $\mathcal{M}_{\mathrm{ai}}$, $\langle \mathrm{Hess}_{\mathrm{ai}} f(\mathbf{X}^*)[\mathbf{U}], \mathbf{U}\rangle_{\mathrm{ai}} = \mathrm{tr}(\nabla^2 f(\mathbf{X}^*)[\mathbf{U}]\mathbf{U}) = \mathrm{vec}(\mathbf{U})^\top \mathbf{H}(\mathbf{X}^*)\mathrm{vec}(\mathbf{U})$, where $\mathbf{H}(\mathbf{X}) \in \mathbb{R}^{n^2 \times n^2}$ is the matrix representation of the Euclidean Hessian $\nabla^2 f(\mathbf{X})$ and $\mathbf{U} \in T_{\mathbf{X}^*}\mathcal{M}_{\mathrm{ai}}$. The maximum eigenvalue of $\mathrm{Hess}_{\mathrm{ai}} f(\mathbf{X}^*)$ is then given by $\lambda^*_{\max} = \max_{\|\mathbf{U}\|^2_{\mathrm{ai}}=1} \mathrm{vec}(\mathbf{U})^\top \mathbf{H}(\mathbf{X}^*)\mathrm{vec}(\mathbf{U})$, where $\|\mathbf{U}\|^2_{\mathrm{ai}} = \mathrm{vec}(\mathbf{U})^\top (\mathbf{X}^* \otimes \mathbf{X}^*)^{-1}\mathrm{vec}(\mathbf{U})$. This is a generalized eigenvalue problem with the solution to be the maximum eigenvalue of $(\mathbf{X}^* \otimes \mathbf{X}^*)\mathbf{H}(\mathbf{X}^*)$. Similarly, $\lambda^*_{\min}$ corresponds to the minimum eigenvalue of $(\mathbf{X}^* \otimes \mathbf{X}^*)\mathbf{H}(\mathbf{X}^*)$.

On $\mathcal{M}_{\mathrm{bw}}$, $\langle \mathrm{Hess}_{\mathrm{bw}} f(\mathbf{X}^*)[\mathbf{U}], \mathbf{U}\rangle_{\mathrm{bw}} = \mathrm{vec}(\mathbf{U})^\top \mathbf{H}(\mathbf{X}^*)\mathrm{vec}(\mathbf{U})$ and the norm is

$\|\mathbf{U}\|^2_{\mathrm{bw}} = \mathrm{vec}(\mathbf{U})^\top(\mathbf{X}^* \oplus \mathbf{X}^*)^{-1}\mathrm{vec}(\mathbf{U})$. The minimum/maximum eigenvalue of $\mathrm{Hess}_{\mathrm{bw}}f(\mathbf{X}^*)$ equals the minimum/maximum eigenvalue of $(\mathbf{X}^* \oplus \mathbf{X}^*)\mathbf{H}(\mathbf{X}^*)$.

Let $\kappa^*_{\mathrm{ai}} := \kappa(\mathrm{Hess}_{\mathrm{ai}}f(\mathbf{X}^*)) = \kappa((\mathbf{X}^* \otimes \mathbf{X}^*)\mathbf{H}(\mathbf{X}^*))$ and $\kappa^*_{\mathrm{bw}} := \kappa(\mathrm{Hess}_{\mathrm{bw}}f(\mathbf{X}^*)) = \kappa((\mathbf{X}^* \oplus \mathbf{X}^*)\mathbf{H}(\mathbf{X}^*))$. The following lemma bounds these two condition numbers.

Lemma 3.1. *For a local minimizer* $\mathbf{X}^*$ *of* $f(\mathbf{X})$*, the condition number of* $\mathrm{Hess}f(\mathbf{X}^*)$ *satisfies*

$$\begin{aligned}\kappa(\mathbf{X}^*)^2/\kappa(\mathbf{H}(\mathbf{X}^*)) \leq &\kappa^*_{\mathrm{ai}} \leq \kappa(\mathbf{X}^*)^2\kappa(\mathbf{H}(\mathbf{X}^*))\\ \kappa(\mathbf{X}^*)/\kappa(\mathbf{H}(\mathbf{X}^*)) \leq &\kappa^*_{\mathrm{bw}} \leq \kappa(\mathbf{X}^*)\kappa(\mathbf{H}(\mathbf{X}^*)).\end{aligned}$$

It is clear that $\kappa^*_{\mathrm{bw}} \leq \kappa^*_{\mathrm{ai}}$ when $\kappa(\mathbf{H}(\mathbf{X}^*)) \leq \sqrt{\kappa(\mathbf{X}^*)}$. This is true for linear, quadratic, higher-order polynomial functions and in general holds for several machine learning optimization problems on the SPD matrices (discussed in Section 3.3).

Case 3.1 (Condition number for linear and quadratic optimization)**.** For a linear function $f(\mathbf{X}) = \mathrm{tr}(\mathbf{XA})$, its Euclidean Hessian matrix is $\mathbf{H}(\mathbf{X}) = \mathbf{0}$. For a quadratic function $f(\mathbf{X}) = \mathrm{tr}(\mathbf{XAXB})$ with $\mathbf{A}, \mathbf{B} \in \mathbb{S}^n_{++}$, $\mathbf{H}(\mathbf{X}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{B} \otimes \mathbf{A}$. Therefore, $\kappa(\mathbf{H}(\mathbf{X}^*))$ is a constant and for ill-conditioned $\mathbf{X}^*$, we have $\kappa(\mathbf{H}(\mathbf{X}^*)) \leq \sqrt{\kappa(\mathbf{X}^*)}$, which leads to $\kappa^*_{\mathrm{ai}} \geq \kappa^*_{\mathrm{bw}}$.

Case 3.2 (Condition number for higher-order polynomial optimization)**.** For an integer $\alpha \geq 3$, consider a function $f(\mathbf{X}) = \mathrm{tr}(\mathbf{X}^\alpha)$ with derived $\mathbf{H}(\mathbf{X}) = \alpha \sum_{l=0}^{\alpha-2}(\mathbf{X}^l \otimes \mathbf{X}^{\alpha-l-2})$. We get $\kappa^*_{\mathrm{ai}} = \alpha \sum_{l=1}^{\alpha-1}((\mathbf{X}^*)^l \otimes (\mathbf{X}^*)^{\alpha-l})$ and $\kappa^*_{\mathrm{bw}} = \alpha(\mathbf{X}^* \oplus \mathbf{X}^*)(\sum_{l=0}^{\alpha-2}(\mathbf{X}^l \otimes \mathbf{X}^{\alpha-l-2}))$. It is apparent that $\kappa^*_{\mathrm{ai}} = \mathcal{O}(\kappa(\mathbf{X}^*)^\alpha)$ while $\kappa^*_{\mathrm{bw}} = \mathcal{O}(\kappa(\mathbf{X}^*)^{\alpha-1})$. Hence, for ill-conditioned $\mathbf{X}^*$, $\kappa^*_{\mathrm{ai}} \geq \kappa^*_{\mathrm{bw}}$.

One counter-example where $\kappa^*_{\mathrm{bw}} \geq \kappa^*_{\mathrm{ai}}$ is the log-det function.

Case 3.3 (Condition number for log-det optimization)**.** For the log-det function $f(\mathbf{X}) = -\log\det(\mathbf{X})$, its Euclidean Hessian is $\nabla^2 f(\mathbf{X})[\mathbf{U}] = \mathbf{X}^{-1}\mathbf{U}\mathbf{X}^{-1}$

and $\mathbf{H}(\mathbf{X}) = \mathbf{X}^{-1} \otimes \mathbf{X}^{-1}$. At a local minimizer $\mathbf{X}^*$, $\mathrm{Hess}_{\mathrm{ai}} f(\mathbf{X}^*)[\mathbf{U}] = \mathbf{U}$ with $\kappa^*_{\mathrm{ai}} = 1$. While on $\mathcal{M}_{\mathrm{bw}}$, we have $\kappa^*_{\mathrm{bw}} = \kappa((\mathbf{X}^* \oplus \mathbf{X}^*)((\mathbf{X}^*)^{-1} \otimes (\mathbf{X}^*)^{-1})) = \kappa((\mathbf{X}^*)^{-1} \oplus (\mathbf{X}^*)^{-1}) = \kappa(\mathbf{X}^*)$. Therefore, $\kappa^*_{\mathrm{ai}} \leq \kappa^*_{\mathrm{bw}}$.

3.2.2 Sectional curvature and trigonometry distance bound

To study the curvature of $\mathcal{M}_{\mathrm{bw}}$, we first show in Lemma 3.2, the existence of a matching geodesic between the Wasserstein geometry of zero-centered non-degenerate Gaussian measures and the BW geometry of SPD matrices. Denote the manifold of such Gaussian measures under the L^2-Wasserstein distance as $(\mathcal{N}_0(\mathbf{\Sigma}), \mathcal{W}_2)$ with $\mathbf{\Sigma} \in \mathbb{S}^n_{++}$.

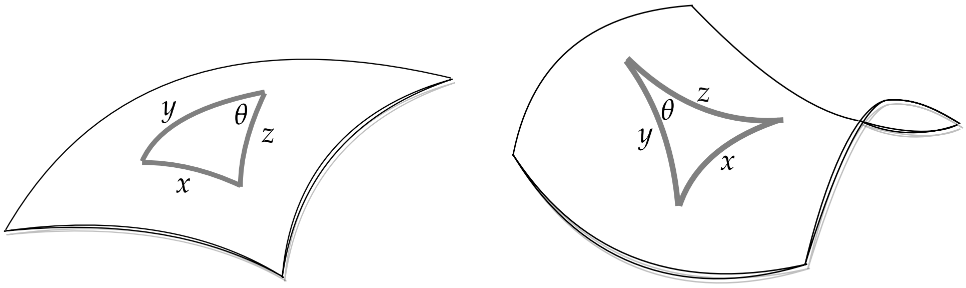
Lemma 3.2. *For any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n_{++}$, a geodesic between $\mathcal{N}_0(\mathbf{X})$ and $\mathcal{N}_0(\mathbf{Y})$ on $(\mathcal{N}_0(\mathbf{\Sigma}), \mathcal{W}_2)$ is given by $\mathcal{N}_0(\gamma(t))$, where $\gamma(t)$ is the geodesic between $\mathbf{X}$ and $\mathbf{Y}$ on $\mathcal{M}_{\mathrm{bw}}$.*

The following lemma builds on a result from the Wasserstein geometry (Ambrosio et al., 2008) and uses Lemma 3.2 to analyze the sectional curvature of $\mathcal{M}_{\mathrm{bw}}$.

Lemma 3.3. *$\mathcal{M}_{\mathrm{bw}}$ is an Alexandrov space with non-negative sectional curvature.*

It is well-known that $\mathcal{M}_{\mathrm{ai}}$ is a non-positively curved space (Cruceru et al., 2021; Pennec, 2020) while, in Lemma 3.3, we show that $\mathcal{M}_{\mathrm{bw}}$ is non-negatively curved. The difference affects the curvature constant in the trigonometry distance bound of Alexandrov space (H. Zhang & Sra, 2016). This bound is crucial in analyzing convergence for optimization algorithms on Riemannian manifolds (H. Zhang & Sra, 2016; H. Zhang et al., 2016). In Section 3.2.3, only local convergence to a minimizer $\mathbf{X}^*$ is analyzed. Therefore, it suffices to consider a neighbourhood Ω around $\mathbf{X}^*$. In such a compact set, the sectional curvature is known to be bounded and we denote the lower bound as K^-.

The following lemma compares the trigonometry distance bounds under the AI and BW geometries. This bound was originally introduced for Alexandrov



(a) Positive curvature (b) Negative curvature

Figure 3.1: Geodesic triangle on spaces with positive and negative curvature.

space with lower bounded sectional curvature (H. Zhang & Sra, 2016). The result for non-negatively curved spaces has been applied in many work (H. Zhang et al., 2016; Sato et al., 2019; Han & Gao, 2021) though without a formal proof. We show the proof in Appendix 3.D, where it follows from the Toponogov comparison theorem (W. Meyer, 1989) on the unit hypersphere and Assumption 3.1.

Lemma 3.4. *Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \Omega$, which forms a geodesic triangle on $\mathcal{M}$. Denote $x = d(\mathbf{Y}, \mathbf{Z}), y = d(\mathbf{X}, \mathbf{Z}), z = d(\mathbf{X}, \mathbf{Y})$ as the geodesic side lengths and let θ be the angle between sides y and z such that $\cos(\theta) = \langle \mathrm{Exp}_{\mathbf{X}}^{-1}(\mathbf{Y}), \mathrm{Exp}_{\mathbf{X}}^{-1}(\mathbf{Z}) \rangle / (yz)$. Then, we have*

$$x^2 \leq \zeta y^2 + z^2 - 2yz\cos(\theta),$$

where ζ is a curvature constant. Under the AI metric, $\zeta = \zeta_{\mathrm{ai}} = \frac{\sqrt{|K_{\mathrm{ai}}^-|}D}{\tanh(\sqrt{|K_{\mathrm{ai}}^-|}D)}$ with D as the diameter bound of Ω, i.e. $\max_{\mathbf{X}_1, \mathbf{X}_2 \in \Omega} d(\mathbf{X}_1, \mathbf{X}_2) \leq D$. Under the BW metric, $\zeta = \zeta_{\mathrm{bw}} = 1$.

It is clear that $\zeta_{\mathrm{ai}} > \zeta_{\mathrm{bw}} = 1$, which leads to a tighter bound under the BW metric.

3.2.3 Convergence analysis

We now analyze the local convergence properties of the Riemannian steepest descent and trust region methods under the two Riemannian geometries. Convergence is established in terms of the Riemannian distance induced from the geodesics. We begin by presenting a lemma that shows in a neighbourhood of $\mathbf{X}^*$, the second-order derivatives of $f \circ \mathrm{Exp}_{\mathbf{X}}$ are both lower and upper bounded.

Lemma 3.5. *In a totally normal neighbourhood Ω around a non-degenerate local minimizer $\mathbf{X}^*$, for any $\mathbf{X} \in \Omega$, it satisfies that $\lambda^*_{\min}/\alpha \leq \frac{d^2}{dt^2} f(\mathrm{Exp}_{\mathbf{X}}(t\mathbf{U})) \leq \alpha\lambda^*_{\max}$, for some $\alpha \geq 1$ and $\|\mathbf{U}\| = 1$. $\lambda^*_{\max} > \lambda^*_{\min} > 0$ are the largest and smallest eigenvalues of* $\mathrm{Hess} f(\mathbf{X}^*)$.

For simplicity of the analysis, we assume such an α is universal under both the Riemannian geometries. We, therefore, can work with a neighbourhood Ω with diameter uniformly bounded by D, where we can choose $D := \min\{D_{\mathrm{ai}}, D_{\mathrm{bw}}\}$ such that α is universal.

One can readily check that under Lemma 3.5 the function f is both μ-geodesic strongly convex and L-geodesic smooth in Ω where $\mu = \lambda^*_{\min}/\alpha$ and $L = \alpha\lambda^*_{\max}$. We now present the local convergence analysis of the two algorithms, which are based on results in H. Zhang & Sra (2016); Absil et al. (2007).

Theorem 3.1 (Local convergence of Riemannian steepest descent)**.** *Under Assumption 3.1 and consider a non-degenerate local minimizer $\mathbf{X}^*$. For a neighbourhood $\Omega \ni \mathbf{X}^*$ with diameter bounded by D on two Riemannian geometries $\mathcal{M}_{\mathrm{ai}}, \mathcal{M}_{\mathrm{bw}}$, running Riemannian steepest descent from $\mathbf{X}_0 \in \Omega$ with a fixed step size $\eta = \frac{1}{\alpha\lambda^*_{\max}}$ yields for $t \geq 2$,*

$$d^2(\mathbf{X}_t, \mathbf{X}^*) \leq \alpha^2 D^2 \kappa^* \left(1 - \min\{\frac{1}{\zeta}, \frac{1}{\alpha^2\kappa^*}\}\right)^{t-2}.$$

Theorem 3.2 (Local convergence of Riemannian trust region)**.** *Under the same settings as in Theorem 3.1, assume further in Ω, it holds that (1) $\|\mathcal{H}_{\mathbf{X}_t} - \mathrm{Hess} f(\mathbf{X}_t)\| \leq \ell \|\mathrm{grad} f(\mathbf{X}_t)\|$ and (2) $\|\nabla^2(f \circ \mathrm{Exp}_{\mathbf{X}_t})(\mathbf{U}) - \nabla^2(f \circ \mathrm{Exp}_{\mathbf{X}_t})(\mathbf{0})\| \leq \rho\|\mathbf{U}\|$ for some ℓ, ρ universal on $\mathcal{M}_{\mathrm{ai}}, \mathcal{M}_{\mathrm{bw}}$. Then running Riemannian trust region from $\mathbf{X}_0 \in \Omega$ yields,*

$$d(\mathbf{X}_t, \mathbf{X}^*) \leq (2\sqrt{\rho} + \ell)(\kappa^*)^2 d^2(\mathbf{X}_{t-1}, \mathbf{X}^*).$$

Theorems 3.1 and 3.2 show that $\mathcal{M}_{\mathrm{bw}}$ has a clear advantage compared to

$\mathcal{M}_{\text{ai}}$ for learning ill-conditioned SPD matrices where $\kappa^*_{\text{bw}} \leq \kappa^*_{\text{ai}}$. For first-order algorithms, $\mathcal{M}_{\text{bw}}$ has an additional benefit due to its non-negative sectional curvature. As $\zeta_{\text{ai}} > \zeta_{\text{bw}} = 1$, the convergence rate degrades on $\mathcal{M}_{\text{ai}}$. Although the convergence is presented in Riemannian distance, it can be readily converted to function value gap by noticing $\frac{\mu}{2} d^2(\mathbf{X}_t, \mathbf{X}^*) \leq f(\mathbf{X}_t) - f(\mathbf{X}^*) \leq \frac{L}{2} d^2(\mathbf{X}_t, \mathbf{X}^*)$. Additionally, we note that these local convergence results hold regardless of whether the function is geodesic convex or not, and similar comparisons also exist for other Riemannian optimization methods.

3.2.4 Geodesic convexity under BW metric for cost functions of interest

Finally we show geodesic convexity of common optimization problems on $\mathcal{M}_{\text{bw}}$. Particularly, we verify that linear, quadratic, log-det optimization, and also certain geometric optimization problems, that are geodesic convex under the AI metric, are also geodesic convex under the BW metric.

Proposition 3.1. *For any* $\mathbf{A} \in \mathbb{S}^n_+$*, where* $\mathbb{S}^n_+ := \{\mathbf{Z} : \mathbf{Z} \in \mathbb{R}^{n \times n}, \mathbf{Z}^\top = \mathbf{Z}, \text{ and } \mathbf{Z} \succeq \mathbf{0}\}$*, the functions* $f_1(\mathbf{X}) = \mathrm{tr}(\mathbf{X}\mathbf{A})$*,* $f_2(\mathbf{X}) = \mathrm{tr}(\mathbf{X}\mathbf{A}\mathbf{X})$*, and* $f_3(\mathbf{X}) = -\log\det(\mathbf{X})$ *are geodesic convex on* $\mathcal{M}_{\text{bw}}$*.*

Based on the result in Proposition 3.1, we also prove geodesic convexity of a reparameterized version of the Gaussian density estimation and mixture model on $\mathcal{M}_{\text{bw}}$ (discussed in Section 3.3). Similar claims on $\mathcal{M}_{\text{ai}}$ can be found in R. Hosseini & Sra (2020).

We further show that monotonic functions on sorted eigenvalues are geodesic convex on $\mathcal{M}_{\text{bw}}$. This is an analogue of (Sra & Hosseini, 2015, Theorem 2.3) on $\mathcal{M}_{\text{ai}}$.

Proposition 3.2. *Let* $\lambda^{\downarrow} : \mathbb{S}^n_{++} \to \mathbb{R}^n_+$ *be the decreasingly sorted eigenvalue map and* $h : \mathbb{R}_+ \to \mathbb{R}$ *be an increasing and convex function. Then* $f(\mathbf{X}) = \sum_{j=1}^k h(\lambda^{\downarrow}_j(\mathbf{X}))$ *for*

$1 \leq k \leq n$ *is geodesic convex on* $\mathcal{M}_{\mathrm{bw}}$*. Examples of such functions include* $f_1(\mathbf{X}) = \mathrm{tr}(\exp(\mathbf{X}))$ *and* $f_2(\mathbf{X}) = \mathrm{tr}(\mathbf{X}^\alpha)$, $\alpha \geq 1$.

3.3 Experiments

In this section, we compare the empirical performance of optimization algorithms under different Riemannian geometries for various problems. In addition to AI and BW, we also include the Log-Euclidean (LE) geometry (Arsigny et al., 2007) in our experiments.

The LE geometry explores the the linear space of symmetric matrices where the matrix exponential acts as a global diffeomorphism from the space to $\mathbb{S}^n_{++}$. The LE metric is defined as

$$\langle \mathbf{U}, \mathbf{V} \rangle_{\mathrm{le}} = \mathrm{tr}(\mathrm{D}_{\mathbf{U}} \log(\mathbf{X}) \mathrm{D}_{\mathbf{V}} \log(\mathbf{X})) \tag{3.3}$$

for any $\mathbf{U}, \mathbf{V} \in T_{\mathbf{X}}\mathcal{M}$, where $\mathrm{D}_{\mathbf{U}} \log(\mathbf{X})$ is the directional derivative of matrix logarithm at $\mathbf{X}$ along $\mathbf{U}$. Following Tsuda et al. (2005); G. Meyer et al. (2011); Quang et al. (2014), for deriving various Riemannian optimization ingredients under the LE metric (3.3), we consider the parameterization $\mathbf{X} = \exp(\mathbf{S})$, where $\mathbf{S} \in \mathbb{S}^n$, i.e., the space of $n \times n$ symmetric matrices. Equivalently, optimization on the SPD manifold with the LE metric is identified with optimization on $\mathbb{S}^n$ and the function of interest becomes $f(\exp(\mathbf{S}))$ for $\mathbf{S} \in \mathbb{S}^n$. While the Riemannian gradient can be computed efficiently by exploiting the directional derivative of the matrix exponential (Al-Mohy & Higham, 2009), deriving the Riemannian Hessian is tricky and we rely on finite-difference Hessian approximations (Boumal, 2015).

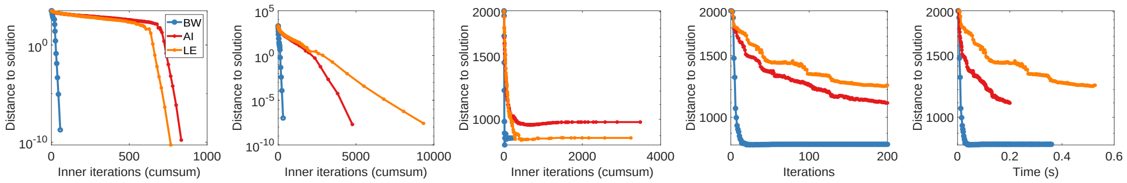
We present convergence mainly in terms of the distance to the solution $\mathbf{X}^*$ whenever applicable. The distance is measured in the Frobenius norm, i.e., $\|\mathbf{X}_t - \mathbf{X}^*\|_{\mathrm{F}}$. When $\mathbf{X}^*$ is not known, convergence is shown in the modified Euclidean gradient norm $\|\mathbf{X}_t \nabla f(\mathbf{X}_t)\|_{\mathrm{F}}$. This is comparable across different metrics as the optimality condition $\mathbf{X}^* \nabla f(\mathbf{X}^*) = \mathbf{0}$ arises from problem structure itself (Journée et al., 2010). We initialize the algorithms with the identity matrix for the AI and BW metrics and zero matrix for the LE metric (i.e., the matrix logarithm of the identity).

We mainly present the results on the Riemannian trust region (RTR) method, which is the method of choice for Riemannian optimization. Note for RTR, the results are shown against the cumulative sum of inner iterations (which are required to solve the trust region subproblem at every iteration). We also include the Riemannian steepest descent (RSD) and Riemannian stochastic gradient (RSGD) (Bonnabel, 2013) methods for some examples. The experiments are conducted in Matlab using the Manopt toolbox (Boumal et al., 2014) on a i5-10500 3.1GHz CPU processor.

In our extended report (Han et al., 2021), we include additional experiments comparing convergence in objective function values for the three geometries. We also present results for the Riemannian conjugate gradient method, and results with different initializations (other than the identity and zero matrices) to further support our claims.

The code can be found on `https://github.com/andyjm3/AI-vs-BW`.

Weighted least squares. We first consider the weighted least squares problem with the symmetric positive definite constraint. The optimization problem is $\min_{\mathbf{X} \in \mathbb{S}^n_{++}} f(\mathbf{X}) = \frac{1}{2}\|\mathbf{A} \odot \mathbf{X} - \mathbf{B}\|^2_{\mathrm{F}}$, which is encountered in for example, SPD matrix completion (R. L. Smith, 2008) where $\mathbf{A}$ is a sparse matrix. The Euclidean gradient and Hessian are $\nabla f(\mathbf{X}) = (\mathbf{A} \odot \mathbf{X} - \mathbf{B}) \odot \mathbf{A}$ and $\nabla^2 f(\mathbf{X})[\mathbf{U}] = \mathbf{A} \odot \mathbf{U} \odot \mathbf{A}$, respectively. Hence, at optimal $\mathbf{X}^*$, the Euclidean Hessian in matrix representation is $\mathbf{H}(\mathbf{X}^*) = \mathrm{diag}(\mathrm{vec}(\mathbf{A} \odot \mathbf{A}))$. We experiment with two choices of $\mathbf{A}$, i.e. $\mathbf{A} = \mathbf{1}_n \mathbf{1}_n^\top$ (`Dense`) and $\mathbf{A}$ as a random sparse matrix (`Sparse`). The former choice for $\mathbf{A}$ leads to well-conditioned $\mathbf{H}(\mathbf{X}^*)$ while the latter choice leads



(a) Dense, `LowCN` (`RTR:iter`) (b) Dense, `HighCN` (`RTR:iter`) (c) Sparse, `HighCN` (`RTR:iter`) (d) Sparse, `HighCN` (`RSD:iter`) (e) Sparse, `HighCN` (`RSD:time`)

Figure 3.2: Weighted least squares problem.

to an ill-conditioned $\mathbf{H}(\mathbf{X}^*)$. Also note that when $\mathbf{A} = \mathbf{1}_n\mathbf{1}_n^\top$, $\kappa^*_{\text{ai}} = \kappa(\mathbf{X}^*)^2$ and $\kappa^*_{\text{bw}} = \kappa(\mathbf{X}^*)$.

We generate $\mathbf{X}^*$ as a SPD matrix with size $n = 50$ and exponentially decaying eigenvalues. We consider two cases with condition numbers $\kappa(\mathbf{X}^*) = 10$ (`LowCN`) and 10^3 (`HighCN`). The matrix $\mathbf{B}$ is generated as $\mathbf{B} = \mathbf{A} \odot \mathbf{X}^*$. Figure 3.2 compares both RSD and RTR for different metrics. When $\mathbf{A}$ is either dense or sparse, convergence is significantly faster on $\mathcal{M}_{\text{bw}}$ than both $\mathcal{M}_{\text{ai}}$ and $\mathcal{M}_{\text{le}}$. The advantage of using $\mathcal{M}_{\text{bw}}$ becomes more prominent in the setting when condition number of $\mathbf{X}^*$ is high. Figure 3.2(e) shows that $\mathcal{M}_{\text{bw}}$ is also superior in terms of runtime.

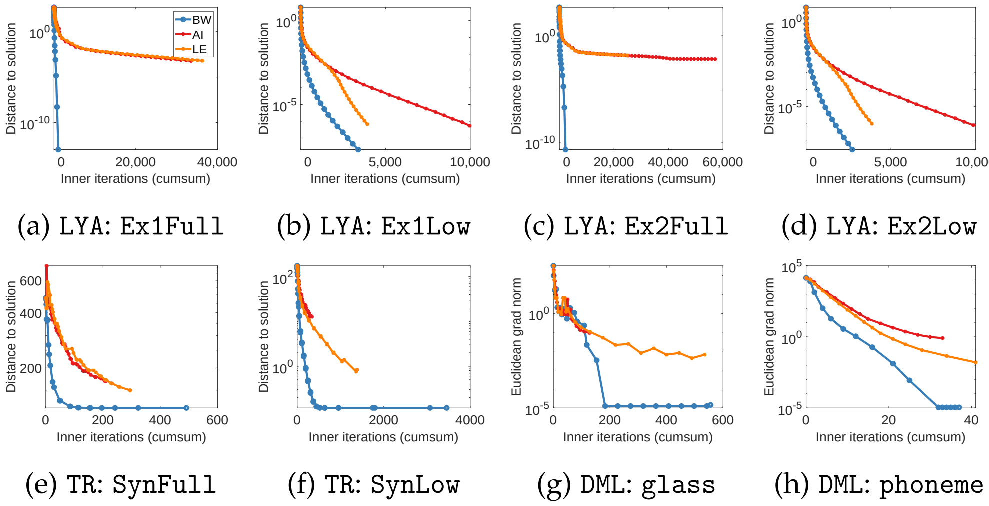
Lyapunov equations. Continuous Lyapunov matrix equation, $\mathbf{AX} + \mathbf{XA} = \mathbf{C}$ with $\mathbf{X} \in \mathbb{S}^n_{++}$, is commonly employed in analyzing optimal control systems and differential equations (Rothschild & Jameson, 1970; Y. Lin & Simoncini, 2015). When $\mathbf{A}$ is stable, i.e., $\lambda_i(\mathbf{A}) > 0$ and $\mathbf{C} \in \mathbb{S}^n_{++}$, the solution $\mathbf{X}^* \succ \mathbf{0}$ and is unique (Lancaster, 1970). When $\mathbf{C} \in \mathbb{S}^n_{+}$ and is low rank, $\mathbf{X}^* \in \mathbb{S}^n_{+}$ is also low rank. We optimize the following problem for solving the Lyapunov equation (Vandereycken & Vandewalle, 2010), i.e., $\min_{\mathbf{X} \in \mathbb{S}^n_{++}} f(\mathbf{X}) = \mathrm{tr}(\mathbf{XAX}) - \mathrm{tr}(\mathbf{XC})$. The Euclidean gradient and Hessian are respectively $\nabla f(\mathbf{X}) = \mathbf{AX} + \mathbf{XA} - \mathbf{C}$ and $\nabla^2 f(\mathbf{X})[\mathbf{U}] = \mathbf{AU} + \mathbf{UA}$ with $\mathbf{H}(\mathbf{X}) = \mathbf{A} \oplus \mathbf{A}$. At optimal $\mathbf{X}^*$, the condition number $\kappa(\mathbf{H}(\mathbf{X}^*)) = \kappa(\mathbf{A})$.

We experiment with two settings for the matrix $\mathbf{A}$, i.e. $\mathbf{A}$ as the Laplace operator on the unit square where we generate 7 interior points so that $n = 49$ (`Ex1`), and $\mathbf{A}$ is a particular Toeplitz matrix with $n = 50$ (`Ex2`). The generated $\mathbf{A}$ matrices are ill-conditioned. The above settings correspond to Examples 7.1 and 7.3 in Y. Lin & Simoncini (2015). Under each setting, $\mathbf{X}^*$ is set to be either full or low rank. The matrix $\mathbf{C}$ is generated as $\mathbf{C} = \mathbf{A}\mathbf{X}^* + \mathbf{X}^*\mathbf{A}$. The full rank $\mathbf{X}^*$ is generated from the full-rank Wishart distribution while the low rank $\mathbf{X}^*$ is a diagonal matrix with $r = 10$ ones and $n - r$ zeros in the diagonal. We label the four cases as `Ex1Full`, `Ex1Low`, `Ex2Full`, and `Ex2Low`. The results are shown in Figures 3.3(a)-(d), where we observe that in all four cases, the BW geometry outperforms both AI and LE geometries.

Trace regression. We consider the regularization-free trace regression model (Slawski et al., 2015) for estimating covariance and kernel matrices (Schölkopf & Smola, 2002; Cai & Zhang, 2015). The optimization problem is $\min_{\mathbf{X} \in \mathbb{S}^d_{++}} f(\mathbf{X}) = \frac{1}{2m} \sum_{i=1}^m (\mathbf{y}_i - \mathrm{tr}(\mathbf{A}_i^\top \mathbf{X}))^2$, where $\mathbf{A}_i = \mathbf{a}_i \mathbf{a}_i^\top$, $i = 1, ..., m$ are some rank-one measurement matrices. Thus, we have $\nabla f(\mathbf{X}) = \sum_{i=1}^m (\mathbf{a}_i^\top \mathbf{X} \mathbf{a}_i - \mathbf{y}_i) \mathbf{A}_i$ and $\nabla^2 f(\mathbf{X})[\mathbf{U}] = \sum_{i=1}^m (\mathbf{a}_i^\top \mathbf{U} \mathbf{a}_i) \mathbf{A}_i$.

We create $\mathbf{X}^*$ as a rank-r Wishart matrix and $\{\mathbf{A}_i\}$ as rank-one Wishart matrices and generate $\mathbf{y}_i = \mathrm{tr}(\mathbf{A}_i \mathbf{X}^*) + \sigma \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, 1)$, $\sigma = 0.1$. We consider two choices, $(m, d, r) = (1000, 50, 50)$ and $(1000, 50, 10)$, which are respectively labelled as `SynFull` and `SynLow`. From Figures 3.3(e)&(f), we also observe that convergence to the optimal solution is faster for the BW geometry.

Metric learning. Distance metric learning (DML) aims to learn a distance function from samples and a popular family of such distances is the Mahalanobis distance, i.e. $d_{\mathbf{M}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{M} (\mathbf{x} - \mathbf{y})}$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. The distance is parameterized by a symmetric positive semi-definite matrix $\mathbf{M}$. We refer readers to this survey (Suárez et al., 2021) for more discussions on this topic. We particularly consider a logistic discriminant learning formulation (Guillaumin



(a) LYA: Ex1Full (b) LYA: Ex1Low (c) LYA: Ex2Full (d) LYA: Ex2Low

(e) TR: SynFull (f) TR: SynLow (g) DML: glass (h) DML: phoneme

Figure 3.3: Lyapunov equation (a, b, c, d), trace regression (e, f), and metric learning (g, h) problems.

et al., 2009). Given a training sample $\{\mathbf{x}_i, y_i\}_{i=1}^N$, denote the link $t_{ij} = 1$ if $y_i = y_j$ and $t_{ij} = 0$ otherwise. The objective is given by $\min_{\mathbf{M} \in \mathbb{S}_{++}^d} f(\mathbf{M}) = -\sum_{i,j} \big(t_{ij} \log p_{ij} + (1 - t_{ij}) \log(1 - p_{ij})\big)$, with $p_{ij} = (1 + \exp(d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)))^{-1}$. We can derive the matrix Hessian as $\mathbf{H}(\mathbf{M}) = \sum_{i,j} p_{ij}(1 - p_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \otimes (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$. Notice $\kappa(\mathbf{H}(\mathbf{M}^*))$ depends on $\mathbf{M}^*$ only through the constants p_{ij}. Thus, the condition number will not be much affected by $\kappa(\mathbf{M}^*)$.

We consider two real datasets, glass and phoneme, from the Keel database (Alcalá-Fdez et al., 2009). The number of classes is denoted as c. The statistics of these two datasets are $(N, d, c) = (241, 9, 7)$ for glass $(5404, 5, 2)$ for phoneme. In Figures 3.3(g)&(h), we similarly see the advantage of using the BW metric compared to the other two metrics that behave similarly.

Log-det maximization. As discussed in Section 3.2.1, log-det optimization is one instance where $\kappa^*_{\mathrm{bw}} \geq \kappa^*_{\mathrm{ai}}$. We first consider minimizing negative log-determinant along with a linear function as studied in C. Wang et al. (2010). For some $\mathbf{C} \in \mathbb{S}_{++}^n$, the objective is $\min_{\mathbf{X} \in \mathbb{S}_{++}^n} f(\mathbf{X}) = \mathrm{tr}(\mathbf{X}\mathbf{C}) - \log\det(\mathbf{X})$. The Euclidean gradient and Hessian are given by $\nabla f(\mathbf{X}) = \mathbf{C} - \mathbf{X}^{-1}$ and $\nabla^2 f(\mathbf{X})[\mathbf{U}] = \mathbf{X}^{-1}\mathbf{U}\mathbf{X}^{-1}$. This problem is geodesic convex under both AI and BW metrics.

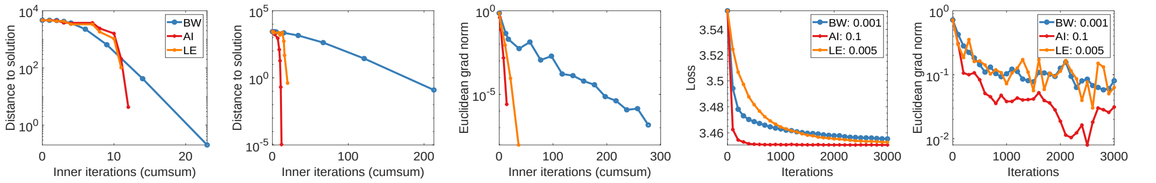
We generate $\mathbf{X}^*$ the same way as in the example of weighted least square with $n = 50$ and set $\mathbf{C} = (\mathbf{X}^*)^{-1}$. We consider two cases with condition number cn = 10 (`LowCN`) and 10^3 (`HighCN`). As expeted, we observe faster convergence of AI and LE metrics over the BW metric in Figures 3.4(a)&(b). This is even more evident when the condition number increases.

Gaussian mixture model. Another notable example of log-det optimization is the Gaussian density estimation and mixture model problem. Following R. Hosseini & Sra (2020), we consider a reformulated problem on augmented samples $\mathbf{y}_i^\top = [\mathbf{x}_i^\top; 1], i = 1, ..., N$ where $\mathbf{x}_i \in \mathbb{R}^d$ are the original samples. The density is parameterized by the augmented covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d+1}$. Notice that the log-likelihood of Gaussian is geodesic convex on $\mathcal{M}_{\rm ai}$, but not on $\mathcal{M}_{\rm bw}$. We, therefore, define $\mathbf{S} = \mathbf{\Sigma}^{-1}$ and the reparameterized log-likelihood is $p_{\mathcal{N}}(\mathbf{Y}; \mathbf{S}) = \sum_{i=1}^N \log \Big((2\pi)^{1-d/2} \exp(1/2) \det(\mathbf{S})^{1/2} \exp(-\frac{1}{2}\mathbf{y}_i^\top \mathbf{S} \mathbf{y}_i) \Big)$, which is now geodesic convex on $\mathcal{M}_{\rm bw}$ due to Proposition 3.1. Hence, we can solve the problem of Gaussian mixture model similar as in R. Hosseini & Sra (2020).

Here, we test on a dataset included in the MixEst package (R. Hosseini & Mash'al, 2015). The dataset has 1580 samples in $\mathbb{R}^2$ with 3 Gaussian components. In Figure 3.4(c), we observe a similar pattern with RTR as in the log-det example. We also include performance of RSGD, which is often preferred for large scale problems. We set the batch size to be 50 and consider a decaying step size, with the best initialized step size shown in Figures 3.4(d)&(e). Following Arthur & Vassilvitskii (2006), the algorithms are initialized with `kmeans++`. We find that the AI geometry still maintains its advantage under the stochastic setting.

3.4 Discussions

In this chapter, we show that the less explored Bures-Wasserstein (BW) geometry for SPD matrices should often be the preferable choice than the Affine-Invariant



(a) LD: `LowCN` (`RTR`)

(b) LD: `HighCN` (`RTR`)

(c) `GMM`: (`RTR`)

(d) `GMM`: (`RSGD`)

(e) `GMM`: (`RSGD`)

Figure 3.4: Log-det maximization (a, b) and Gaussian mixture model (c, d, e) problems.

geometry for optimization, particularly for learning ill-conditioned matrices. We propose three main reasons for this claim:

(1) Riemannian Hessian under BW geometry is often less ill-conditioned compared to the Affine-Invariant (AI) geometry.

(2) BW geometry is non-negatively curved, which provides a tighter trigonometric distance bound and thus leads to better convergence rates.

(3) BW geometry preserves geodesic convexity of some popular cost functions, including linear, quadratic and negative log-det functions.

We also theoretically discuss a 'counter-example' of log-det optimization where the AI geometry enjoys a better second-order conditioning and validate our findings empirically. This issue is addressed in our recent work (Han et al., 2023), where we propose a generalized Bures-Wasserstein (GBW) geometry that is built on a generalization of the Lyapunov operator in the metric:

$$\langle \mathbf{U}, \mathbf{V} \rangle_{\text{gbw}} = \frac{1}{2} \text{vec}(\mathbf{U})^\top (\mathbf{X} \otimes \mathbf{M} + \mathbf{M} \otimes \mathbf{X})^{-1} \text{vec}(\mathbf{V}), \tag{3.4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are symmetric matrices and $\mathbf{M}$ is a given SPD matrix. When $\mathbf{M} = \mathbf{I}$, the metric (3.4) reduces to the special BW metric (3.2). The use of the parameter $\mathbf{M}$ in (3.4) allows great flexibility for optimization algorithms. For one, choosing a particular $\mathbf{M}$ allows preconditioning the Hessian by locally approximating the AI geometry. This leads to improved convergence for log-det optimization.

Our comparisons between AI and BW geometries are based on optimization over generic cost functions. For specific problems, however, there may exist other alternative metrics that potentially work better. This is an interesting research direction to pursue. We also remark that optimization is not the only area where the AI and BW geometries can be compared. It would be useful to qualitatively compare the two metrics for other learning problems on SPD matrices, such as barycenter learning.

Appendices

The appendix sections are organized as follows. Section 3.A reviews the Bures-Wasserstein geometry, particularly the derivation of geodesics. Section 3.B summarizes the Log-Euclidean geometry and operations necessary for Riemannian optimization. Section 3.C, 3.D, 3.E, 3.F include proofs for the main texts.

3.A Bures-Wasserstein geometry of SPD matrices

Here, we include a complete summary of the Bures-Wasserstein geometry. We refer readers to Bhatia et al. (2019); van Oostrum (2020); Malagò et al. (2018) for a more detailed discussion.

The Bures-Wasserstein distance on $\mathbb{S}^n_{++}$ is given by:

$$d_{\rm bw}(\mathbf{X}, \mathbf{Y}) = \Big(\mathrm{tr}(\mathbf{X}) + \mathrm{tr}(\mathbf{Y}) - 2\mathrm{tr}(\mathbf{X}^{1/2}\mathbf{Y}\mathbf{X}^{1/2})^{1/2}\Big)^{1/2}, \tag{3.5}$$

which corresponds to the L^2-Wasserstein distance between zero-centered non-degenerate Gaussian measures. The distance is realized by solving the Procrustes problem, i.e. $d_{\rm bw} = \min_{\mathbf{P} \in \mathbf{O}(n)} \|\mathbf{X}^{1/2} - \mathbf{Y}^{1/2}\mathbf{P}\|_{\rm F}$, where $\mathbf{O}(n)$ denotes the orthogonal group. The minimum is attained when $\mathbf{P}$ is the unitary polar factor of $\mathbf{Y}^{1/2}\mathbf{X}^{1/2}$. The distance defined in (3.5) is indeed a Riemannian distance on $\mathbb{S}^n_{++}$ induced from a Riemannian submersion. That is, the space of SPD matrices can be identified as a quotient space on the general linear group $\mathrm{GL}(n)$ with the action of orthogonal group $\mathbf{O}(n)$. The quotient map $\pi : \mathrm{GL}(n) \to \mathrm{GL}(n)/\mathbf{O}(n)$

thus defines a Riemannian submersion. By endowing a Euclidean metric on $\mathrm{GL}(n)$, we can induce the BW metric on SPD manifold, shown in Table 3.1. Similarly the induced geodesic is given by the following proposition (Bhatia et al., 2019; van Oostrum, 2020).

Proposition 3.3 (Geodesics of $\mathcal{M}_{\rm bw}$ (Bhatia et al., 2019; van Oostrum, 2020))**.** *For any $\mathbf{X}, \mathbf{Y} \in \mathcal{M}_{\rm bw}$, a geodesic γ connecting $\mathbf{X}, \mathbf{Y}$ is given by*

$$\gamma(t) = \Big((1-t)\mathbf{X}^{1/2} + t\mathbf{Y}^{1/2}\mathbf{P}\Big)\Big((1-t)\mathbf{X}^{1/2} + t\mathbf{Y}^{1/2}\mathbf{P}\Big)^\top,$$

where $\mathbf{P} \in O(n)$ is the unitary polar factor of $\mathbf{Y}^{1/2}\mathbf{X}^{1/2}$.

Followed by this proposition, one can derive the Riemannian exponential map as $\mathrm{Exp}_{\mathbf{X}}(\mathbf{U}) = \mathbf{X} + \mathbf{U} + \mathcal{L}_{\mathbf{X}}[\mathbf{U}]\mathbf{X}\mathcal{L}_{\mathbf{X}}[\mathbf{U}]$. The inverse exponential map, also known as the logarithm map only exists in a open set around a center point $\mathbf{X}$. This is because the BW geometry is not unique-geodesic due to the non-negative curvature. Such open neighbourhood around $\mathbf{X}$ is given by $\Omega = \{\mathrm{Exp}_{\mathbf{X}}(\mathbf{U}) : \mathbf{I} + \mathcal{L}_{\mathbf{X}}[\mathbf{U}] \in \mathbb{S}^n_{++}\}$. In this set, the exponential map is a local diffeomorphism from the manifold to the tangent space and the logarithm map is provided by $\mathrm{Log}_{\mathbf{X}}(\mathbf{Y}) = (\mathbf{X}\mathbf{Y})^{1/2} + (\mathbf{Y}\mathbf{X})^{1/2} - 2\mathbf{X}$, for any $\mathbf{X}, \mathbf{Y} \in \Omega$. It is noted that $\mathcal{M}_{\rm bw}$ is geodesic incomplete while $\mathcal{M}_{\rm ai}$ and $\mathcal{M}_{\rm le}$ are geodesic complete. One can follow Takatsu (2008) to complete the space by extending the metric to positive semi-definite matrices.

Relationship between the BW metric and the Procrustes metric. Here we highlight that the BW metric is a special form of the more general Procrustes metric, which is studied in Dryden et al. (2009).

Definition 3.2 (Procrustes metric)**.** For any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n_{++}$, the Procrustes distance is defined as $d_{\rm pc}(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{P} \in \mathbf{O}(n)} \|\mathbf{L}_{\mathbf{X}} - \mathbf{L}_{\mathbf{Y}}\mathbf{P}\|_{\rm F}$, where $\mathbf{X} = \mathbf{L}_{\mathbf{X}}\mathbf{L}_{\mathbf{X}}^\top, \mathbf{Y} = \mathbf{L}_{\mathbf{Y}}\mathbf{L}_{\mathbf{Y}}^\top$ for some decomposition factors $\mathbf{L}_{\mathbf{X}}, \mathbf{L}_{\mathbf{Y}}$.

Thus it is easy to see that under the BW metric, $\mathbf{L}_\mathbf{X} = \mathbf{X}^{1/2}, \mathbf{L}_\mathbf{Y} = \mathbf{Y}^{1/2}$. Another choice of $\mathbf{L}_\mathbf{X}, \mathbf{L}_\mathbf{Y}$ can be the Cholesky factor, which is a lower triangular matrix with positive diagonals. The optimal $\mathbf{P} = \mathbf{U}\mathbf{V}^\top$ is obtained from the singular value decomposition of $\mathbf{L}_\mathbf{Y}^\top \mathbf{L}_\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Under Procrustes metric, one can similarly derive a geodesic as $c(t) = ((1-t)\mathbf{L}_\mathbf{X} + t\mathbf{L}_\mathbf{Y}\mathbf{P})\,((1-t)\mathbf{L}_\mathbf{X} + t\mathbf{L}_\mathbf{Y}\mathbf{P})^\top$, which corresponds to $\gamma(t)$ in Proposition 3.3. This space is also incomplete with non-negative curvature.

3.B Log-Euclidean geometry and its Riemannian gradient computation

This section presents a summary on the Log-Euclidean (LE) geometry (Arsigny et al., 2007; Quang et al., 2014) and derives its Riemannian gradient for Riemannian optimization, which should be of independent interest.

The Log-Euclidean metric is a bi-invariant metric on the Lie group structure of SPD matrices with the group operation $\mathbf{X} \odot \mathbf{Y} := \exp(\log(\mathbf{X}) + \log(\mathbf{Y}))$ for any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n_{++}$. This metric is induced from the Euclidean metric on the space of symmetric matrices, $\mathbb{S}^n$, through the matrix exponential. Hence the LE metric is given by $\langle \mathbf{U}, \mathbf{V} \rangle_{\rm le} = \mathrm{tr}(\mathrm{D}_\mathbf{U} \log(\mathbf{X}) \mathrm{D}_\mathbf{V} \log(\mathbf{X}))$, for $\mathbf{U}, \mathbf{V} \in \mathbb{S}^n$ and the LE distance is $d_{\rm le}(\mathbf{X}, \mathbf{Y}) = \| \log(\mathbf{X}) - \log(\mathbf{Y}) \|_\mathrm{F}$. One can also derive the exponential map associated with the metric as $\mathrm{Exp}_\mathbf{X}(\mathbf{U}) = \exp(\log(\mathbf{X}) + \mathrm{D}_\mathbf{U} \log(\mathbf{X}))$.

Because of the derivative of matrix logarithm in the LE metric, it appears challenging to derive a simple form of Riemannian gradient based on the definition given in the main text. Hence, we follow the works (Tsuda et al., 2005; G. Meyer et al., 2011; Quang et al., 2014) to consider the parameterization of SPD matrices by the symmetric matrices through the matrix exponential. Therefore, the optimization of $f(\mathbf{X}), \mathbf{X} \in \mathbb{S}^n_{++}$ becomes optimization of $g(\mathbf{S}) := f(\exp(\mathbf{S}))$, $\mathbf{S} \in \mathbb{S}^n$, which is a linear space with the Euclidean metric. Then, the Riemannian gradient of $g(\mathbf{S})$ is derived as

$$\mathrm{grad} g(\mathbf{S}) = \{\mathrm{D}_{\nabla f(\exp(\mathbf{S}))} \exp(\mathbf{S})\}_{\mathbf{S}}.$$

To compute the Riemannian gradient, we need to evaluate the directional derivative of matrix exponential along $\nabla f(\exp(\mathbf{S}))$. This can be efficiently computed via the function over a block triangular matrix (Al-Mohy & Higham, 2009). That is, for any $\mathbf{V} \in \mathbb{S}^n$, the directional derivative of $\exp(\mathbf{S})$ along $\mathbf{V}$ is given by the upper block triangular of the following matrix:

$$\exp\left(\begin{bmatrix} \mathbf{S} & \mathbf{V} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}\right) = \begin{bmatrix} \exp(\mathbf{S}) & \mathrm{D}_{\mathbf{V}} \exp(\mathbf{S}) \\ \mathbf{0} & \exp(\mathbf{S}) \end{bmatrix}.$$

This provides an efficient way to compute the Riemannian gradient of $g(\mathbf{S})$ over $\mathbb{S}^n$. However, computing the Riemannian Hessian of $g(\mathbf{S})$, requires further evaluating the directional derivative of $\mathrm{grad}\, g(\mathbf{S})$, which, to the best of our knowledge, is difficult. Thus in experiments, we approach the Hessian with finite difference of the gradient. This is sufficient to ensure global convergence of the Riemannian trust region method (Boumal, 2015).

Remark 3.2 (Practical considerations)**.** For Riemannian optimization algorithms, every iteration requires to evaluate the matrix exponential for a matrix of size $2n \times 2n$, which can be costly. Also, the matrix exponential may result in unstable gradients and updates, particularly when $\nabla g(\mathbf{S})$ involves matrix inversions. This is the case for the log-det optimization problem where $f(\exp(\mathbf{S})) = -\log\det(\exp(\mathbf{S}))$. Hence, $\nabla f(\exp(\mathbf{S})) = (\exp(\mathbf{S}))^{-1}$. Nevertheless, for log-det optimization, we can simplify the function to $f(\exp(\mathbf{S})) = -\mathrm{tr}(\mathbf{S})$, with $\nabla f(\exp(\mathbf{S})) = -\mathbf{I}$.

3.C Proof for Section 3.2.1: Condition number of Riemannian Hessian

Proof of Lemma 3.1. Under AI metric, first note that for any $\mathbf{X} \in \mathbb{S}^n_{++}$,

$$\begin{aligned}\kappa(\mathbf{X} \otimes \mathbf{X}) &= \|\mathbf{X} \otimes \mathbf{X}\|_2 \|(\mathbf{X} \otimes \mathbf{X})^{-1}\|_2 \\ &= \|\mathbf{X} \otimes \mathbf{X}\|_2 \|\mathbf{X}^{-1} \otimes \mathbf{X}^{-1}\|_2 \\ &= \|\mathbf{X}\|_2^2 \|\mathbf{X}^{-1}\|_2^2 = \kappa(\mathbf{X})^2,\end{aligned}$$

where we apply the norm properties for Kronecker product. Next denote the i-th largest eigenvalue as $\lambda_i(\mathbf{A})$ for $1 \leq i \leq d$ where $\mathbf{A} \in \mathbb{R}^{d \times d}$. Then,

$$\begin{aligned}\kappa_{\mathrm{ai}} = \kappa((\mathbf{X} \otimes \mathbf{X})\mathbf{H}(\mathbf{X})) = \frac{\lambda_1((\mathbf{X} \otimes \mathbf{X})\mathbf{H}(\mathbf{X}))}{\lambda_{n^2}((\mathbf{X} \otimes \mathbf{X})\mathbf{H}(\mathbf{X}))} &\geq \frac{\lambda_1((\mathbf{X} \otimes \mathbf{X}))\lambda_{n^2}(\mathbf{H}(\mathbf{X}))}{\lambda_{n^2}((\mathbf{X} \otimes \mathbf{X}))\lambda_1(\mathbf{H}(\mathbf{X}))} \\ &= \kappa((\mathbf{X} \otimes \mathbf{X}))/\kappa(\mathbf{H}(\mathbf{X})) \\ &= \kappa(\mathbf{X})^2/\kappa(\mathbf{H}(\mathbf{X})),\end{aligned}$$

where the first inequality uses the eigenvalue bound for matrix product, i.e. $\lambda_i(\mathbf{A})\lambda_d(\mathbf{B}) \leq \lambda_i(\mathbf{AB}) \leq \lambda_i(\mathbf{A})\lambda_1(\mathbf{B})$ for $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times d}$ (Merikoski & Kumar, 2004). The upper bound on κ^*_{ai} is easily obtained by noting $k(\mathbf{AB}) \leq \kappa(\mathbf{A})\kappa(\mathbf{B})$.

Similarly for the BW metric, we first note that because $\mathbf{X} \in \mathbb{S}^n_{++}$, $\mathbf{X} \oplus \mathbf{X} \in \mathbb{S}^{n^2}_{++}$ by spectrum property of Kronecker sum (Hardy & Steeb, 2019). Then we have

$$\kappa(\mathbf{X} \oplus \mathbf{X}) = \frac{\lambda_1(\mathbf{X} \oplus \mathbf{X})}{\lambda_{n^2}(\mathbf{X} \oplus \mathbf{X})} = \frac{2\lambda_1(\mathbf{X})}{2\lambda_n(\mathbf{X})} = \kappa(\mathbf{X}),$$

where the second equality is again due to the spectrum property. Then the lower and upper bounds of the condition number on $\kappa((\mathbf{X} \oplus \mathbf{X})\mathbf{H}(\mathbf{X}))$ are derived similarly. □

3.D Proofs for Section 3.2.2: Sectional curvature and trigonometry distance bound derivation

Proof of Lemma 3.2. The proof follows by noticing that the push-forward interpolation between two non-degenerate Gaussians is a Gaussian with covariance given by interpolation of the covariances.

From (Takatsu, 2008, Lemma 2.3), for any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n_{++}$, the geodesic between $\mathcal{N}_0(\mathbf{X})$ and $\mathcal{N}_0(\mathbf{Y})$ under L^2-Wasserstein metric is $\mathcal{N}_0(\omega(t))$, where

$$\omega(t) = ((1-t)\mathbf{I} + t\mathbf{T})\, \mathbf{X}\, ((1-t)\mathbf{I} + t\mathbf{T})\,, \tag{3.6}$$

with $\mathbf{T} = \mathbf{Y}^{1/2}(\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}^{1/2})^{-1/2}\mathbf{Y}^{1/2}$ as the pushforward map from $\mathcal{N}_0(\mathbf{X})$ to $\mathcal{N}_0(\mathbf{Y})$. It is clear that the interpolation of two non-degenerate Gaussian measures is also a non-degenerate Gaussian. To show $\omega(t) = \gamma(t)$, We only need to show $\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2} = \mathbf{T}$, where $\mathbf{P}$ is the unitary polar factor of $\mathbf{Y}^{1/2}\mathbf{X}^{1/2}$. By noting that $\mathbf{P} = \mathbf{Y}^{1/2}(\mathbf{X}\mathbf{Y})^{-1/2}\mathbf{X}^{1/2}$ from eq. (35) in Bhatia et al. (2019), we have $\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2} = \mathbf{Y}(\mathbf{X}\mathbf{Y})^{-1/2}$. On the other hand, $\mathbf{T} = \mathbf{Y}\mathbf{Y}^{-1/2}(\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}^{1/2})^{-1/2}\mathbf{Y}^{1/2} = \mathbf{Y}(\mathbf{X}\mathbf{Y})^{-1/2}$, where the second equality can be seen as follows. Denote $\mathbf{C} := (\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}^{1/2})^{-1/2}$, then

$$\begin{aligned}\mathbf{I} = \mathbf{C}\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}^{1/2}\mathbf{C} &= \mathbf{Y}^{-1/2}\mathbf{C}\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}^{1/2}\mathbf{C}\mathbf{Y}^{1/2} \\ &= \mathbf{Y}^{-1/2}\mathbf{C}\mathbf{Y}^{1/2}\mathbf{X}\mathbf{Y}\mathbf{Y}^{-1/2}\mathbf{C}\mathbf{Y}^{1/2}.\end{aligned}$$

From this result, we have $\mathbf{Y}^{-1/2}\mathbf{C}\mathbf{Y}^{1/2} = (\mathbf{X}\mathbf{Y})^{-1/2}$. This completes the proof. □

Proof of Lemma 3.3. Let $\mu_0, \mu_1, \nu \in \mathcal{N}_0$ with covariance matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{S}^n_{++}$ and denote $\mu_t := ((1-t)\mathrm{id} + tT_{\mu_0 \to \mu_1})_{\#}\mu_0$, which is the interpolated Gaussian measure between μ_0, μ_1. From the matching geodesics in Lemma 3.2, we have $\mu_t \equiv \mathcal{N}_0(\gamma(t))$. Then based on standard Theorem on Wasserstein distance, e.g. (Ambrosio et al., 2008, Theorem 7.3.2), we have $\mathcal{W}_2^2(\mu_t, \nu) \geq (1-t)\mathcal{W}_2^2(\mu_0, \nu) +$

$t\mathcal{W}_2^2(\mu_1, \nu) - t(1-t)\mathcal{W}_2^2(\mu_0, \mu_1)$. Given the accordance between L^2-Wasserstein distance between zero-mean Gaussians and geodesic distance between their corresponding covariance matrices on $\mathcal{M}_{\rm bw}$, we have

$$d_{\rm bw}^2(\gamma(t), \mathbf{Z}) \geq (1-t)d_{\rm bw}^2(\mathbf{X}, \mathbf{Z}) + td_{\rm bw}^2(\mathbf{Y}, \mathbf{Z}) - t(1-t)d_{\rm bw}^2(\mathbf{X}, \mathbf{Y})$$

holds for any $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{S}_{++}^n$. This suggests $\mathcal{M}_{\rm bw}$ is a non-negatively curved Alexandrov space with non-negative sectional curvature. □

Proof of Lemma 3.4. Given $\mathcal{M}_{\rm ai}$ is a non-positively curved space, the proof under AI metric can be found in H. Zhang & Sra (2016), which reduces to proving the claim for hyperbolic space with constant curvature -1. Similarly, for non-negatively curved space, it becomes studying the hypersphere with constant curvature 1. Let $\triangle\tilde{x}\tilde{y}\tilde{z}$ be the comparison triangle on $T_{\mathbf{X}}\mathcal{M}_{\rm bw}$ such that $\tilde{y} = y, \tilde{z} = z$ and θ is the angle between side $\tilde{y}$ and $\tilde{z}$. Because Ω is a uniquely geodesic subset as per Assumption 3.1, we have $d(\mathbf{X}, \mathbf{Y}) = \|{\rm Exp}_{\mathbf{X}}^{-1}(\mathbf{Y})\|_{\rm bw}$ for any $\mathbf{X}, \mathbf{Y} \in \Omega$. Thus, we can immediately see $\tilde{x}^2 = \|{\rm Exp}_{\mathbf{X}}^{-1}(\mathbf{Y}) - {\rm Exp}_{\mathbf{X}}^{-1}(\mathbf{Z})\|_{\rm bw}^2 = y^2 + z^2 - 2yz\cos(\theta)$. Then from the Toponogov Theorem (Theorem 2.2 in W. Meyer (1989)) and the assumption of unique geodesic, we have $x \leq \tilde{x}$, which shows for unit hypersphere:

$$x^2 \leq y^2 + z^2 - 2yz\cos(\theta). \tag{3.7}$$

Next, we see that for the space of constant curvature 0, it satisfies $x^2 = y^2 + z^2 - 2yz\cos(\theta)$. Thus we can focus on where the curvature is positive, i.e. $K > 0$. For such space, we have the following generalized law of cosines (W. Meyer, 1989):

$$\cos(\sqrt{K}x) = \cos(\sqrt{K}y)\cos(\sqrt{K}z) + \sin(\sqrt{K}y)\sin(\sqrt{K}z)\cos(\theta),$$

which can be viewed as a geodesic triangle on unit hypersphere with side lengths $\sqrt{K}x$, $\sqrt{K}y, \sqrt{K}z$. Thus, substituting these side lengths in (3.7) proves the desired result for positively curved space. □

3.E Proofs for Section 3.2.3: Convergence analysis

Proof of Lemma 3.5. The proof follows mainly from the continuity of $\frac{d^2}{dt^2}(f \circ \mathrm{Exp})$ in both $t, \mathbf{X}, \mathbf{U}$. First note at optimality, we have for $\mathbf{U} \in T_{\mathbf{X}^*}\Omega$ with $\|\mathbf{U}\| = 1$, $\lambda^*_{\min} \leq \langle \mathrm{Hess} f(\mathbf{X}^*)[\mathbf{U}], \mathbf{U}\rangle \leq \lambda^*_{\max}$. Because exponential map is a second-order retraction, by standard theory (e.g. Proposition 5.5.5 in Absil et al. (2009)), $\mathrm{Hess} f(\mathbf{X}) = \nabla^2(f \circ \mathrm{Exp}_{\mathbf{X}})(\mathbf{0})$ and $\langle \mathrm{Hess} f(\mathbf{X})[\mathbf{U}], \mathbf{U}\rangle = \frac{d^2}{dt^2} f(\mathrm{Exp}_{\mathbf{X}}(t\mathbf{U}))|_{t=0}$ for any $\mathbf{X} \in \mathcal{M}, \mathbf{U} \in T_{\mathbf{X}}\mathcal{M}$. Thus at optimality, we have

$$\lambda^*_{\min} \leq \frac{d^2}{dt^2} f(\mathrm{Exp}_{\mathbf{X}}(t\mathbf{U}))|_{\mathbf{X}=\mathbf{X}^*, t=0} \leq \lambda^*_{\max}.$$

By the continuity of $\frac{d^2}{dt^2}(f \circ \mathrm{Exp})$, we can always find a constant $\alpha \geq 1$ such that $\lambda^*_{\min}/\alpha \leq \frac{d^2}{dt^2} f(\mathrm{Exp}_{\mathbf{X}}(t\mathbf{U})) \leq \alpha \lambda^*_{\max}$ holds for all $\mathbf{X} \in \Omega$, $\|\mathbf{U}\| = 1$ and t such that $\mathrm{Exp}_{\mathbf{X}}(t\mathbf{U}) \in \Omega$. In general, α scales with the size of Ω. □

Proof of Theorem 3.1. From Theorem 14 in H. Zhang & Sra (2016), we have for either metric,

$$f(\mathbf{X}_t) - f(\mathbf{X}^*) \leq \frac{1}{2}(1 - \min\{\frac{1}{\zeta}, \frac{\mu}{L}\})^{t-2} D^2 L,$$

where L, μ are the constants for geodesic smoothness and strongly convex. As discussed in the main text, $L = \alpha \lambda^*_{\max}$ and $\mu = \lambda^*_{\min}/\alpha$, where $\lambda^*_{\min}$ and $\lambda^*_{\max}$ are eigenvalues under either metric. Based on standard result on μ-geodesic strongly convexity, we have $f(\mathbf{X}_t) - f(\mathbf{X}^*) \geq \frac{\mu}{2} d^2(\mathbf{X}_t, \mathbf{X}^*)$. Combining this result and Lemma 3.4 and 3.1 gives the result. □

Proof of Theorem 3.2. From Theorem 4.13 in Absil et al. (2007), we have for either metric, $d(\mathbf{X}_t, \mathbf{X}^*) \leq c\, d^2(\mathbf{X}_{t-1}, \mathbf{X}^*)$ for some $c \geq (\frac{\rho}{\lambda^*_{\min}} + \lambda^*_{\min} + \ell)(\kappa^*)^2 \geq (2\sqrt{\rho} + \ell)(\kappa^*)^2$. □

3.F Proofs for Section 3.2.4: Geodesic convexity

3.F.1 Proofs

Proof of Proposition 3.1. The main idea is to apply the second-order characterization of geodesic convexity. Let $f_1(\mathbf{X}) = \mathrm{tr}(\mathbf{XA})$ and $f_2(\mathbf{X}) = \mathrm{tr}(\mathbf{XAX})$. For claim of linear function, given any $\mathbf{A} \in \mathbb{S}^n_+$, it can be factorized as $\mathbf{A} = \mathbf{L}^\top \mathbf{L}$ for some $\mathbf{L} \in \mathbb{R}^{m\times n}$. Thus $f_1(\mathbf{X}) = \mathrm{tr}(\mathbf{LXL}^\top)$. Denote $\pi(t) := (1-t)\mathbf{X}^{1/2} + t\mathbf{Y}^{1/2}\mathbf{P}$ and thus the geodesic $\gamma(t) = \pi(t)\pi(t)^\top$. By standard calculus, we can write the first-order and second-order derivatives as

$$\begin{aligned}\frac{df_1(\gamma(t))}{dt} &= 2\mathrm{tr}\left(\mathbf{L}(\mathbf{Y}^{1/2}\mathbf{P} - \mathbf{X}^{1/2})\pi(t)^\top \mathbf{L}^\top\right),\\ \frac{d^2 f_1(\gamma(t))}{dt^2} &= 2\mathrm{tr}\left(\mathbf{L}(\mathbf{Y}^{1/2}\mathbf{P} - \mathbf{X}^{1/2})(\mathbf{Y}^{1/2}\mathbf{P} - \mathbf{X}^{1/2})^\top \mathbf{L}^\top\right) \geq 0.\end{aligned}$$

For claim on quadratic function $f_2(\mathbf{X})$, let $\tilde{\mathbf{X}} := \mathbf{X}^{1/2}, \tilde{\mathbf{Y}} := \mathbf{Y}^{1/2}\mathbf{P}$ and the first-order derivative can be similarly derived as

$$\begin{aligned}\frac{df_2(\gamma(t))}{dt} =& 2\mathrm{tr}(\tilde{\mathbf{Y}}\pi(t)^\top \mathbf{A}\pi(t)\pi(t)^\top) - 2\mathrm{tr}(\tilde{\mathbf{X}}\pi(t)^\top \mathbf{A}\pi(t)\pi(t)^\top)\\ &- 2\mathrm{tr}(\tilde{\mathbf{X}}\pi(t)^\top \pi(t)\pi(t)^\top \mathbf{A}) + 2\mathrm{tr}(\tilde{\mathbf{Y}}\pi(t)^\top \pi(t)\pi(t)^\top \mathbf{A}).\end{aligned}$$

The second-order derivative is derived and simplified as

$$\begin{aligned}\frac{d^2 f_2(\gamma(t))}{dt^2} =& 2\|\tilde{\mathbf{Y}}\pi(t)^\top \mathbf{L}^\top - \tilde{\mathbf{X}}\pi(t)^\top \mathbf{L}^\top\|_{\mathrm{F}}^2 + 2\|\mathbf{L}\tilde{\mathbf{Y}}\pi(t)^\top - \mathbf{L}\tilde{\mathbf{X}}\pi(t)^\top\|_{\mathrm{F}}^2 \qquad (3.8)\\ &+ 4\mathrm{tr}\left((\tilde{\mathbf{X}} - \tilde{\mathbf{Y}})(\tilde{\mathbf{X}} - \tilde{\mathbf{Y}})^\top \{\mathbf{A}\pi(t)\pi(t)^\top\}_{\mathrm{S}}\right) \qquad (3.9)\\ &+ 4\mathrm{tr}\left(\mathbf{L}\left(\tilde{\mathbf{Y}}\pi(t)^\top - \tilde{\mathbf{X}}\pi(t)^\top\right)^2 \mathbf{L}^\top\right) \geq 0, \qquad (3.10)\end{aligned}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. Terms (3.8) and (3.10) are clearly non-negative. Term (3.9) is also non-negative by noting $\{\mathbf{A}\pi(t)\pi(t)^\top\}_{\mathrm{S}} \in \mathbb{S}^n_+$.

To prove the claim on geodesic convexity of $f_3(\mathbf{X}) = -\log\det(\mathbf{X})$, we use the definition of geodesic convexity and applies the fact that $\det(\mathbf{A}\mathbf{A}^\top) = (\det(\mathbf{A}))^2$ and $\det(\mathbf{A}+\mathbf{B}) \geq \det(\mathbf{A}) + \det(\mathbf{B})$ for $\mathbf{A}, \mathbf{B} \succ \mathbf{0}$.

That is, for any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}^n_{++}$ and $t \in [0,1]$, the geodesic $\gamma(t)$ with respect to metric g_{bw} joining $\mathbf{X}, \mathbf{Y}$ is given in Proposition 3.3. Thus,

$$\begin{aligned}
\log\det(\gamma(t)) &= 2\log\det(\pi(t)) && (3.11)\\
&= 2\log\det((1-t)\mathbf{I} + t\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2})\mathbf{X}^{1/2})\\
&= 2\log\det((1-t)\mathbf{I} + t\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2}) + 2\log\det(\mathbf{X}^{1/2})\\
&\geq 2\log((1-t)\det(\mathbf{I}) + t\det(\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2})) + 2\log\det(\mathbf{X}^{1/2})\\
& && (3.12)\\
&\geq 2(1-t)\log\det(\mathbf{I}) + 2t\log\det(\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2}) + 2\log\det(\mathbf{X}^{1/2})\\
& && (3.13)\\
&= 2t\log\det(\mathbf{Y}^{1/2}\mathbf{P}) - 2t\log\det(\mathbf{X}^{1/2}) + 2\log\det(\mathbf{X}^{1/2})\\
&= t\log\det(\mathbf{Y}) + (1-t)\log\det(\mathbf{X}) && (3.14)
\end{aligned}$$

where (3.11) uses the fact that $\det(\mathbf{A}\mathbf{A}^\top) = (\det(\mathbf{A}))^2$ and inequality (3.12) uses the fact that $\det(\mathbf{A}+\mathbf{B}) \geq \det(\mathbf{A}) + \det(\mathbf{B})$ for $\mathbf{A}, \mathbf{B} \in \mathbb{S}^n_{++}$ and from Lemma 1 in van Oostrum (2020), we have $\mathbf{Y}^{1/2}\mathbf{P}\mathbf{X}^{-1/2} \in \mathbb{S}^n_{++}$ with $\mathbf{P}$ as the orthogonal polar factor of $\mathbf{X}^{1/2}\mathbf{Y}^{1/2}$. Inequality (3.13) follows from the concavity of logarithm. Equality (3.14) uses the fact that $\det(\mathbf{P})^2 = 1$ for $\mathbf{P} \in O(n)$. This shows $\log\det$ is geodesically concave. And because logarithm is strictly concave, inequality (3.13) reduces to equality only when $t = 0, 1$. Thus strict geodesic concavity is proved. Now the proof is complete. □

Proposition 3.4. *The log-likelihood of reparameterized Gaussian $f(\mathbf{S}) = p_{\mathcal{N}}(\mathbf{Y}; \mathbf{S})$ is geodesic concave on $\mathcal{M}_{\text{bw}}$.*

Proof of Proposition 3.4. To prove $f(\mathbf{S})$ is geodesic convex, it suffices to show that $f(\mathbf{S}) = \log(\det(\mathbf{S})^{1/2}\exp(-\frac{1}{2}\mathbf{y}_i^\top\mathbf{S}\mathbf{y}_i))$ is geodesic concave. That is, for a geodesic

$\gamma(t)$ connecting $\mathbf{X}, \mathbf{Y}$, we have

$$\begin{aligned} f(\gamma(t)) &= \log\left(\det(\gamma(t))^{1/2}\exp(-\frac{1}{2}\mathbf{y}_i^\top\gamma(t)\mathbf{y}_i)\right) \\ &= \frac{1}{2}\log\det(\gamma(t)) - \frac{1}{2}\mathbf{y}_i^\top\gamma(t)\mathbf{y}_i \\ &\geq \frac{1-t}{2}\log\det(\mathbf{X}) + \frac{t}{2}\log\det(\mathbf{Y}) - \frac{1-t}{2}\mathbf{y}_i^\top\mathbf{X}\mathbf{y}_i - \frac{t}{2}\mathbf{y}_i^\top\mathbf{Y}\mathbf{y}_i \qquad (3.15) \\ &= (1-t)f(\mathbf{X}) + tf(\mathbf{Y}). \end{aligned}$$

where inequality (3.15) follows from Proposition 3.1. We further notice that as log det is strictly geodesically concave, so is $f(\mathbf{S})$. □

Remark 3.3 (Gaussian mixture model)**.** Under the BW metric, consider the reformulated GMM model with K components:

$$\max_{\{\mathbf{S}_j \in \mathbb{S}_{++}^{d+1}\}_{j=1}^K, \{\omega_j\}_{j=1}^{K-1}} L = \sum_{i=1}^n \log\left(\sum_{j=1}^K \frac{\exp(\omega_j)}{\sum_{j=1}^K \exp(\omega_j)} p_{\mathcal{N}}(\mathbf{y}_i; \mathbf{S})\right), \qquad (3.16)$$

where $\omega_K = 0$, $p_{\mathcal{N}}(\mathbf{y}_i; \mathbf{S}) := (2\pi)^{1-d/2}\det(\mathbf{S})^{1/2}\exp(\frac{1}{2} - \frac{1}{2}\mathbf{y}_i^\top\mathbf{S}\mathbf{y}_i)$. It is easy to see that problem (3.16) is geodesically convex for each component. Also, the optimal solution for problem (3.16) is unchanged given the inverse transformation on SPD matrices is one-to-one. That is, if $\mathbf{S}_j^*$ maximizes problem (3.16), $(\mathbf{S}_j^*)^{-1}$ maximizes the problem in R. Hosseini & Sra (2020). Based on Theorem 1 in R. Hosseini & Sra (2020), our local maximizer can be written as parameters of the original GMM problem, i.e. $\{\mu_j, \Sigma_j\}$:

$$(\mathbf{S}_j^*)^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_j^* + \mu_j^*\mu_j^{*T} & \mu_j^* \\ \mu_j^{*\top} & 1 \end{bmatrix}, \quad \text{for } j = 1, 2, ..., K.$$

Proof of Proposition 3.2. The proof is based on (Bhatia et al., 2019, Theorem 6), where we can show the geometric mean under BW geometry also satisfies the convexity with respect to the Loewner ordering. That is, $\gamma(t) \preceq (1-t)\mathbf{X} + t\mathbf{Y}$. Then the proof then follows as in Theorem 2.3 in Sra & Hosseini (2015). □

Chapter 4

Improved variance reduction for Riemannian optimization

This chapter considers the following online and finite-sum optimization problems on a Riemannian manifold $\mathcal{M}$.

$$\min_{x \in \mathcal{M}} f(x) := \begin{cases} \mathbb{E}[f(x;\omega)], & \text{online} \\ \frac{1}{n}\sum_{i=1}^{n} f_i(x), & \text{finite-sum} \end{cases} \tag{4.1}$$

where $f : \mathcal{M} \rightarrow \mathbb{R}$ is a smooth, real-valued, possibly nonconvex function. The finite-sum formulation of minimizing the average of n component functions is a special case of online optimization where ω can be finitely sampled. For some cases, n can be large or possibly infinite where only stochastic gradients are available. This corresponds to the online problem with ω indexed by i. Hence, for notational clarity, we consider the case $f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ and refer to it as finite-sum or online optimization depending on the size of n. Problem (4.1) covers many machine learning applications, including principal component analysis (Sato et al., 2019), low rank matrix completion (Boumal & Absil, 2011b), Riemannian centroid computation (Yuan et al., 2016), independent component analysis (Theis et al., 2009) and many applications in deep learning (Vorontsov et al., 2017; J. Wang et al., 2020).

There is a growing interest to solve problem (4.1) directly over the manifold space via Riemannian gradient based methods. Two basic algorithms are Riemannian steepest descent (R-SD) (Udriste, 1994) and Riemannian stochastic gradient descent (R-SGD) (Bonnabel, 2013). Although R-SD enjoys a faster convergence rate of $O(1/T)$ than $O(1/\sqrt{T})$ of R-SGD for nonconvex optimization (Boumal et al., 2019a; R. Hosseini & Sra, 2020), R-SD requires a full pass over n component functions per iteration. This computation is extremely costly when n is large, thereby prohibiting its applicability for online optimization. On the other hand, despite with higher per-iteration efficiency, R-SGD suffers from high gradient variance, similar to its Euclidean version. Therefore it usually relies on a decaying stepsize to ensure convergence (R. Hosseini & Sra, 2020).

To improve on R-SD and R-SGD and achieve lower gradient complexity, many studies leverage variance reduction techniques from unconstrained optimization in the Euclidean space. This includes Riemannian stochastic variance reduction method (R-SVRG) (H. Zhang et al., 2016; Sato et al., 2019), Riemannian stochastic recursive gradient method (R-SRG) (Kasai et al., 2018b) and Riemannian stochastic path integrated differential estimator (R-SPIDER) (J. Zhang et al., 2018; P. Zhou, Yuan, Yan, & Feng, 2019), which are generalized from the Euclidean counterparts (Reddi, Hefny, et al., 2016; Johnson & Zhang, 2013; Nguyen et al., 2017b; Fang et al., 2018). Among them, R-SPIDER is shown to achieve the optimal complexity for both finite-sum and online optimization (Fang et al., 2018; Arjevani et al., 2023).

Nevertheless, existing convergence results on R-SVRG and R-SRG appear to be incomplete and suboptimal. In particular, convergence of R-SVRG for nonconvex functions under retraction and vector transport (more general than exponential map and parallel transport) is missing. Existing work either analyzes R-SVRG for retraction strongly convex functions (Sato et al., 2019) or for nonconvex functions but restricted to exponential map and parallel transport (H. Zhang et al., 2016). Also, R-SRG in Kasai et al. (2018b) is analyzed in terms of single-loop convergence, which is suboptimal compared to R-SPIDER for finite-sum optimization (Fang et al., 2018). Moreover, analysis under online setting seems to be absent for both methods.

Apart from the gap in the convergence analysis, it is also useful to study if the complexities of existing Riemannian variance reduction methods can be further improved. Indeed, a common feature among these methods is periodic computations of full batch gradient, which potentially limits convergence particularly during early stage of training. This is because at early stage, stochastic gradients are pointing to similar directions and therefore it becomes unnecessary to use exact gradients to correct for deviations (Balles et al., 2017). While approaching optimal point, larger batch gradient becomes increasingly important to reduce variance of stochastic gradients. Furthermore, gradient noise at the outset of training helps to escape sharp minima, leading to higher generalization power (Keskar et al., 2016). Therefore, a reasonable strategy is to gradually increase the batch size throughout optimization path.

In this chapter, we propose a unified and general framework for analyzing and improving the Riemannian variance reduction methods with adaptive batch size. This framework includes the non-adaptive versions as special cases. Under such framework, we show the batch size adaptation improves gradient complexities of Riemannian variance reduction methods. We also close the gap in the convergence analysis of R-SVRG and R-SRG, which matches the state-of-the-art analysis for their Euclidean counterparts. Specifically, our main contributions for this chapter are summarized as follows.

- The proposed framework is more general than H. Zhang et al. (2016); Sato et al. (2019); Kasai et al. (2018b) by considering retraction and vector transport as well as mini-batch stochastic gradients.

- We show that batch size adaptation improves the gradient complexities of

R-SVRG and R-SRG for both general nonconvex functions and gradient dominated functions (see Definition 4.1).

- We first analyze nonconvex R-SVRG under retraction and vector transport following the standard Lyapunov analysis. Then we derive the same complexities (but curvature-free) under the new framework, which requires much simpler analysis without constructing the Lyapunov function and using the trigonometric distance bound. To the best of our knowledge, this is the first curvature-free result for SVRG-type methods on Riemannian manifold.
- Under the new framework, we also prove an improved complexity for R-SRG under double-loop convergence. This result matches the optimal complexity achieved by R-SPIDER, but without requiring a small stepsize.
- In addition, we show the first complexity results of R-SVRG and R-SRG for online optimization.
- Finally, experiments over a number of applications, including principal component analysis, low rank matrix completion and Riemannian Fréchet mean computation, verify the effectiveness of batch size adaptation.

4.1 Related works

Recent progress on accelerating Riemannian gradient methods can be broadly classified into three directions: Nesterov acceleration, adaptive gradient and variance reduction.

Acceleration and adaptive gradient. Several studies consider extending the Nesterov acceleration to Riemannian manifold for retraction (strongly) convex optimization (Ahn & Sra, 2020; H. Zhang & Sra, 2018b; Alimisis et al., 2021).

But for general nonconvex functions, it is unknown whether faster convergence guarantee is maintained. In terms of advancement on R-SGD, recent studies have been devoted to adapting the gradient and stepsize, motivated by the success of adaptive methods on Deep Learning applications. In particular, some successful efforts have been made that generalize AdaGrad, Adam and RMSProp to Riemannian optimization (Kumar Roy et al., 2018; Kasai et al., 2019; Sakai & Iiduka, 2021; Becigneul & Ganea, 2019). These methods can be viewed as preconditioned R-SGD and do not theoretically outperform R-SGD with better complexity.

Variance reduction. The first Riemannian variance reduction method, R-SVRG (H. Zhang et al., 2016; Sato et al., 2019), extends the ideas in Reddi, Hefny, et al. (2016); Johnson & Zhang (2013). By occasionally evaluating full gradient of a reference point, R-SVRG allows for a larger stepsize and hence converges faster particularly around optimal point. But on manifold space, when the reference point is far from current iterates, the use of vector transport can incur unintended distortion. Therefore, inspired by Nguyen et al. (2017b), Kasai et al. (2018b) introduces R-SRG that transports gradients between consecutive iterates. More recently, R-SPIDER (J. Zhang et al., 2018; P. Zhou, Yuan, Yan, & Feng, 2019) hybrids the same recursive gradient estimator with gradient normalization as in Fang et al. (2018). Other related works for reducing variance of R-SGD include Tripuraneni et al. (2018); Babanezhad et al. (2018) where Polyak iterate averaging (Polyak & Juditsky, 1992) and SAGA (Defazio et al., 2014) are also generalized for Riemannian optimization. However, their analysis are limited to retraction (strongly) convex functions. In the Euclidean space, some variance reduction methods that also achieve near-optimal complexities include SNVRG (D. Zhou et al., 2020), Geom-SARAH (Horváth et al., 2022) and PAGE (Z. Li et al., 2021).

Batch size adaptation. Increasing batch size for SGD is also an approach to reduce variance so that stepsize decay is no longer necessary (Balles et al., 2017;

S. L. Smith et al., 2018). This is usually achieved by pre-specifying a strategy for batch size increase, such as exponential or linear (Friedlander & Schmidt, 2012; P. Zhou et al., 2018). Alternatively, adaptively changing the batch size based on gradient variance or model quality often yields improved convergence rates (De et al., 2017; Balles et al., 2017; Sievert & Charles, 2019). For variance reduction methods, SVRG is proved to be robust to inexact gradient at reference point provided that batch size is increasing (Harikandeh et al., 2015; Lei et al., 2017). Still, a fixed increase scheme is considered. To the best of our knowledge, Ji et al. (2020) is the only work that adapts the batch size based on gradient information and proves an improved complexity for generic variance reduction methods.

4.2 Preliminaries and settings

In this section, based on Chapter 2, we describe the settings we consider and introduce some concepts and notations for the rest of the analysis.

Throughout this chapter, we implicitly assume vector transport is isometric, i.e., inner product preserving (see Section 2.2.3 for formal definition). Clearly, parallel transport is an isometric vector transport by definition. Also, there are many ways to construct isometric vector transport, which we discuss in Section 4.4. In addition, we require the notion of Riemannian PL or gradient dominance condition (introduced in Chapter 2). For better reference, we reiterate the definition below.

Definition 4.1 (τ-Gradient Dominance)**.** A differentiable function $f : \mathcal{M} \to \mathbb{R}$ is τ-gradient dominated in $\mathcal{X} \subset \mathcal{M}$ if for any $x \in \mathcal{X}$, there exists a $\tau > 0$ such that

$$f(x) - f(x^*) \leq \tau \|\mathrm{grad} f(x)\|_x^2,$$

where $x^* = \arg\min_{x \in \mathcal{M}} f(x)$ is a global minimizer of f.

With a slight abuse of notation. we in general refer to $x^* \in \mathcal{M}$ as an optimal

Table 4.1: Comparison of IFO gradient complexity on general nonconvex problems. $\Theta := \max\{L, \sqrt{L_l^2 + \theta^2 G^2}\}$, $\Theta_1 := L + \sqrt{L^2 + \varrho_1 (L_l + \theta G)^2 \mu^2 \nu^2}$, $\Theta_2 := L + \sqrt{L^2 + \varrho_2 (L_l + \theta G)^2}$, where $\varrho_1, \varrho_2 > 0$ are parameter-free constants and L, L_l, θ, G, μ, ν are defined in Section 4.4. Under finite-sum setting, $\tilde{B} := \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1 \sigma^2/\beta_s, n\}$ and under online setting, $\tilde{B} := \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1 \sigma^2/\beta_s, \alpha_2 \sigma^2/\epsilon^2\}$. We derive the complexity of R-SVRG and R-SRG under a variety of settings with a unified analysis framework. The new analysis allows R-SRG to match the optimal complexity achieved by R-SPIDER, i.e., $O(n + \sqrt{n}/\epsilon^2)$ under finite-sum setting and $O(1/\epsilon^3)$ under online setting. The adaptive batch size allows complexity of R-SVRG and R-SRG to be further reduced.

General nonconvex		(Retraction and vector transport)		(Exponential map and parallel transport)	
		Finite-sum	Online	Finite-sum	Online
Existing work	R-SVRG (H. Zhang et al., 2016)	—	—	$O\big(n + \frac{L\zeta^{1/2} n^{2/3}}{\epsilon^2}\big)$	—
	R-SRG (Kasai et al., 2018b)	$O\big(n + \frac{\Theta^2}{\epsilon^4}\big)$	—	$O\big(n + \frac{L^2}{\epsilon^4}\big)$	—
	R-SPIDER (P. Zhou, Yuan, Yan, & Feng, 2019) (J. Zhang et al., 2018)	$O\big(n + \frac{\Theta\sqrt{n}}{\epsilon^2}\big)^*$	$O\big(\frac{\Theta\sigma}{\epsilon^3}\big)$	$O\big(n + \frac{L\sqrt{n}}{\epsilon^2}\big)^*$	$O\big(\frac{L\sigma}{\epsilon^3}\big)$
This work	R-SVRG	$O\big(n + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big)$	$O\big(\frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big)$	$O\big(n + \frac{L n^{2/3}}{\epsilon^2}\big)$	$O\big(\frac{L\sigma^{4/3}}{\epsilon^{10/3}}\big)$
	R-SRG	$O\big(n + \frac{\Theta_2 \sqrt{n}}{\epsilon^2}\big)$	$O\big(\frac{\Theta_2 \sigma}{\epsilon^3}\big)$	$O\big(n + \frac{L\sqrt{n}}{\epsilon^2}\big)$	$O\big(\frac{L\sigma}{\epsilon^3}\big)$
	R-AbaSVRG	$O\big(\tilde{B} + \frac{\Theta_1 \tilde{B}}{n^{1/3}\epsilon^2} + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big)$	$O\big(\frac{\Theta_1 \tilde{B}}{\sigma^{2/3}\epsilon^{4/3}} + \frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big)$	$O\big(\tilde{B} + \frac{L\tilde{B}}{n^{1/3}\epsilon^2} + \frac{L n^{2/3}}{\epsilon^2}\big)$	$O\big(\frac{L\tilde{B}}{\sigma^{2/3}\epsilon^{4/3}} + \frac{L\sigma^{4/3}}{\epsilon^{10/3}}\big)$
	R-AbaSRG	$O\big(\tilde{B} + \frac{\Theta_2 \tilde{B}}{\sqrt{n}\epsilon^2} + \frac{\Theta_2 \sqrt{n}}{\epsilon^2}\big)$	$O\big(\frac{\Theta_2 \tilde{B}}{\sigma\epsilon} + \frac{\Theta_2 \sigma}{\epsilon^3}\big)$	$O\big(\tilde{B} + \frac{L\tilde{B}}{\sqrt{n}\epsilon^2} + \frac{L\sqrt{n}}{\epsilon^2}\big)$	$O\big(\frac{L\tilde{B}}{\sigma\epsilon} + \frac{L\sigma}{\epsilon^3}\big)$

[*] P. Zhou, Yuan, Yan, & Feng (2019) presents finite-sum complexities of R-SPIDER as minimum of finite-sum and online complexities, which simply applies online choices of parameters to finite-sum setting.

point within its neighbourhood $\mathcal{X}$, can be either local or global. Only Section 4.7 considers the existence of global minimizer as per Definition 4.1.

Further, this chapter measures algorithm quality by the total IFO complexity to achieve ϵ-accurate solution, defined as follows.

Definition 4.2 (ϵ-accurate solution and IFO complexity)**.** ϵ-accurate solution from a stochastic algorithm is an output x with expected gradient norm no larger than ϵ. That is, $\mathbb{E}\|\mathrm{grad} f(x)\|_x \leq \epsilon$. Incremental First-Order (IFO) oracle (A. Agarwal & Bottou, 2015) takes a component index i and a point $x \in \mathcal{X}$ and outputs an unbiased stochastic gradient $\mathrm{grad} f_i(x) \in T_x\mathcal{M}$. IFO complexity counts the total number of IFO oracle calls.

Notations. For notational clarity, we omit the subscripts for norm and inner product. Specific indication to the tangent space should be clear from contexts. Also, we denote $[n] := \{1, ..., n\}$ and $\mathbb{1}_{\{\cdot\}}$ as the indicator function. $\mathrm{grad} f_{\mathcal{I}}(x) := \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \mathrm{grad} f_i(x)$ is a mini-batch Riemannian stochastic gradient on $T_x\mathcal{M}$, where $\mathcal{I} \subset [n]$ is an index set with cardinality $|\mathcal{I}|$. When $\mathcal{I} \equiv [n]$, we obtain the full gradient as $\mathrm{grad} f(x) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{grad} f_i(x)$.

4.3 Algorithms

4.3.1 Riemannian SGD and variance reduction

A default solution for optimizing problem (4.1) is to use R-SGD that iteratively updates

$$x_{t+1} = \mathrm{Retr}_{x_t}\big(-\eta_t \, \mathrm{grad} f_{\mathcal{I}_t}(x_t)\big), \tag{4.2}$$

where $\eta_t > 0$ is the stepsize. The updates move along the retraction curve from current iterate with the direction determined by negative gradient. When $\mathcal{M} \equiv \mathbb{R}^d$, (4.2) reduces to $x_{t+1} = x_t - \eta_t \nabla f_{\mathcal{I}_t}(x_t)$, which is the standard SGD update in the Euclidean space. Variance reduction techniques leverage previous gradient information to construct a modified stochastic gradient with variance that decreases as training progresses.

R-SVRG adopts a double loop structure where, at the start of each epoch (i.e. outer loop), a snapshot point $\tilde{x}$ is fixed and full gradient is evaluated. Within each inner iteration, mini-batch stochastic gradients are computed for the current iterate x_t as well as for the snapshot point. A modified gradient v_t at x_t is then constructed as in (4.3) by adjusting deviations according to the difference between the stochastic gradient and full gradient at $\tilde{x}$. Since Riemannian gradients of x_t and $\tilde{x}$ are defined on disjoint tangent spaces, vector transport is used to combine gradient information.

$$v_t = \mathrm{grad} f_{\mathcal{I}_t}(x_t) - \mathcal{T}_{\tilde{x}}^{x_t}\big(\mathrm{grad} f_{\mathcal{I}_t}(\tilde{x}) - \mathrm{grad} f(\tilde{x})\big). \tag{4.3}$$

Instead of using gradient information from a distant reference point, R-SRG recursively modifies stochastic gradients based on the previous iterate. That is, after computing batch gradient $v_0 = \mathrm{grad} f(x_0)$ on an initial point, a modified gradient is constructed within each inner loop as

$$v_t = \mathrm{grad} f_{\mathcal{I}_t}(x_t) - \mathcal{T}_{x_{t-1}}^{x_t}\big(\mathrm{grad} f_{\mathcal{I}_t}(x_{t-1}) - v_{t-1}\big). \tag{4.4}$$

This is followed by a standard retraction update $x_{t+1} = \mathrm{Retr}_{x_t}(-\eta_t v_t)$. Note for both R-SVRG (Sato et al., 2019; H. Zhang et al., 2016) and R-SRG (Kasai et al., 2018b), stochastic gradient $\mathrm{grad} f_{i_t}(x_t)$ rather than mini-batch gradient $\mathrm{grad} f_{\mathcal{I}_t}(x_t)$ is considered in (4.3) and (4.4). We show in the analysis sections that the mini-batch formulation allows more flexible choices of the stepsize and inner loop size and potentially speeds up the algorithms in distributed settings.

R-SPIDER employs the same recursive gradient estimator as in (4.4). A fundamental difference is the use of normalized gradient for its update, which is given by $x_{t+1} = \mathrm{Retr}_{x_t}\big(-\eta_t \frac{v_t}{\|v_t\|}\big)$. Therefore, it requires a changing stepsize η_t

proportional to the desired accuracy ϵ and depend on $\|v_t\|$. Also, R-SPIDER does not adopt the inner-outer loop framework. This results in distinct convergence analysis that shows progress every iteration by bounding the distance between consecutive iterates.

4.3.2 Proposed Riemannian variance reduction with batch size adaptation

In this work, we propose a unified framework motivated by the idea in Ji et al. (2020), for adapting batch size based on the norm of modified gradients in the previous epoch. It is believed that the gradient norm decreases as optimization proceeds and hence is indicative of optimization stages.

Our primary analysis is based on the inner-outer loop formulation of R-SVRG and R-SRG. For R-SPIDER, we defer its analysis to Appendix because we notice the use of variable stepsize imposes some difficulties in generalizing this adaptive strategy. By assuming a bounded gradient norm, we can similarly prove its convergence. Nevertheless, the total complexity can be worse than its original complexity.

The proposed Riemannian adaptive batch-size SVRG (R-AbaSVRG) and SRG (R-AbaSRG) are shown in Algorithms 1 and 2. Let s and t respectively represent the outer loop and inner loop index. For each epoch, B^s is set as $\frac{\alpha_1 \sigma^2 m}{\sum_t \|v_t^{s-1}\|^2}$ where α_1 is a sufficiently large constant and m, σ^2 are the size of inner loop and the variance of stochastic gradient respectively. As training progresses, B^s would gradually increase to n under finite-sum setting and to $\alpha_2 \sigma^2 / \epsilon^2$ under online setting. Without-replacement sampling is employed to construct batch gradients. This is to ensure that full batch gradient can be computed under finite-sum setting, thus recovering vanilla R-SVRG and R-SRG. Under online setting, it makes no theoretical difference between with- and without-replacement sampling as n approaches infinity. Here we consider setting the initial point (or the reference point for R-SVRG) as the last iterate from previous epoch. This is in contrast to some update rules such as uniform selection in R-SRG (Kasai et al., 2018b) or Riemannian centroid in R-SVRG (Sato et al., 2019). Especially for R-SRG, this simple modification allows us to derive double loop convergence, which is stronger than single loop convergence in Kasai et al. (2018b) under finite-sum setting (details in Section 4.6).

4.4 Assumptions

We first present three sets of assumptions as follows that are necessary for convergence analysis. Assumption 4.1 is standard for all Riemannian variance reduction methods and is sufficient for SRG-type methods. Assumption 4.2 is further required for analysing SVRG-type methods and Assumption 4.3 is needed to establish convergence of R-SVRG under traditional Lyapunov analysis. All assumptions are common in the analysis of optimization algorithms using retraction and vector transport, see for example W. Huang, Gallivan, & Absil (2015); Kasai et al. (2018b); Sato et al. (2019); P. Zhou, Yuan, Yan, & Feng (2019).

Assumption 4.1.

(4.1.1) Function f and its components $f_i, i = 1, ..., n$ are at least twice continuously differentiable.

(4.1.2) Iterate sequences produced by algorithms stay continuously in a neighbourhood $\mathcal{X} \subset \mathcal{M}$ around an optimal solution x^*. Additionally, $\mathcal{X}$ is a totally retractive neighbourhood of x^* where retraction Retr is a diffeomorphism.

(4.1.3) Norms of Riemannian gradient and Riemannian Hessian are bounded. That is, for all $x \in \mathcal{X}$ and any component function f_i, there exists constants $G, H > 0$ where $\|\text{grad} f_i(x)\| \leq G$ and $\|\text{Hess} f_i(x)\| \leq H$ hold.

(4.1.4) Variance of Riemannian gradient is bounded. That is, for all $x \in \mathcal{X}$, $\mathbb{E}\|\mathrm{grad} f_i(x) - \mathrm{grad} f(x)\|^2 \leq \sigma^2$.

(4.1.5) Function f is retraction L-smooth with respect to retraction R. That is, for all $x, y = \mathrm{Retr}_x(\xi) \in \mathcal{X}$, there exists a constant $L > 0$ such that

$$f(y) \leq f(x) + \langle \mathrm{grad} f(x), \xi \rangle + \frac{L}{2}\|\xi\|^2.$$

(4.1.6) Function f is average retraction L_l-Lipschitz. That is, for all $x, y \in \mathcal{X}$, there exists a constant $L_l > 0$ such that

$$\mathbb{E}\|\mathrm{grad} f_i(x) - \Gamma_y^x \mathrm{grad} f_i(y)\| \leq L_l \|\xi\|,$$

where Γ_y^x is the parallel transport from y to x along the retraction curve $c(t) := \mathrm{Retr}_x(t\xi)$ with $c(0) = x, c(1) = y$.

(4.1.7) (Lemma 3.5 in W. Huang, Gallivan, & Absil (2015)) Difference between vector transport $\mathcal{T}$ and parallel transport Γ associated with the same retraction Retr is bounded. That is, for all $x, y = \mathrm{Retr}_x(\xi) \in \mathcal{X}$ and $u \in T_x\mathcal{M}$, there exists a constant $\theta \geq 0$, such that

$$\|\mathcal{T}_x^y u - \Gamma_x^y u\| \leq \theta \|\xi\| \|u\|.$$

Assumptions (4.1.2) to (4.1.4) clearly hold for compact manifolds, including sphere, (compact) Stiefel and Grassmann manifolds. For non-compact manifolds, like symmetric positive definite (SPD) matrices, the assumptions hold by choosing a sufficiently small neighbourhood $\mathcal{X}$. Note that Assumption (4.1.4) is introduced to bound the deviation resulting from inexact batch gradient. For vanilla R-SRG and R-SVRG, this assumption is not required. Assumption (4.1.5) generalizes the notion of smoothness in the Euclidean space to Riemannian manifold and is satisfied if $\frac{d^2 f(\mathrm{Retr}_x(t\xi))}{dt^2} \leq L$ for all $x \in \mathcal{X}, \xi \in T_x\mathcal{M}$ with

$\|\xi\| = 1$ (Kasai et al., 2018b, Lemma 3.5). In a compact set $\mathcal{X}$, we can simply choose $L = \sup_{x\in\mathcal{X},t,\xi} \frac{d^2 f(\mathrm{Retr}_x(t\xi))}{dt^2}$. Finally, Assumption (4.1.6) and (4.1.7) are required given we use vector transport to approximate parallel transport. Based on W. Huang, Absil, & Gallivan (2015); W. Huang, Gallivan, & Absil (2015), these two assumptions can be derived by requiring the vector transport $\mathcal{T}$ to be isometric and satisfy $\|\mathcal{T}_x^y u - \mathrm{DRetr}_x(\xi)[u]\| \leq c_0\|\xi\|\|u\|$, where $\mathrm{DRetr}_x(\xi)[u]$ is the differentiated retraction. The latter condition is ensured by Taylor approximation in a compact set $\mathcal{X}$ (Kasai et al., 2018b) and the following remark discusses the condition of isometric vector transport.

Remark 4.1. First we remark that parallel transport trivially satisfies Assumption (4.1.6) and (4.1.7) with $L_l = L$ and $\theta = 0$. In addition, one can follow W. Huang, Gallivan, & Absil (2015) to construct other isometric vector transports that meet these two assumptions. For the manifold of SPD matrices of size $d \times d$, denoted as $\mathcal{S}_{++}^d$, an isometric vector transport can be derived by parallelization. That is, for any $\mathbf{X}, \mathbf{Y} \in \mathcal{S}_{++}^d$, $\mathcal{T}_{\mathbf{X}}^{\mathbf{Y}}\xi = \mathbf{B}_{\mathbf{Y}}\mathbf{B}_{\mathbf{X}}^{\flat}$, where $\mathbf{B}_{\mathbf{X}} \in \mathbb{R}^{d\times d}$ is the orthonormal bases on $T_{\mathbf{X}}\mathcal{S}_{++}^d$ and $\mathbf{B}_{\mathbf{X}}^{\flat} : T_{\mathbf{X}}\mathcal{S}_{++}^d \to \mathbb{R}$ such that $\mathbf{B}_{\mathbf{X}}^{\flat}\mathbf{U} = \langle \mathbf{B}_{\mathbf{X}}, \mathbf{U}\rangle_{\mathbf{X}}$. Similar construction also exists for Stiefel and Grassmann manifolds (W. Huang, 2013).

Assumption 4.2.

(4.2.1) The neighbourhood $\mathcal{X}$ is also a totally normal neighbourhood of x^* where exponential map is a diffeomorphism.

(4.2.2) (Lemma 3 in W. Huang, Absil, & Gallivan (2015)) There exists $\mu, \nu, \delta_{\mu,\nu} > 0$ where for all $x, y = \mathrm{Retr}_x(\xi) \in \mathcal{X}$ with $\|\xi\| \leq \delta_{\mu,\nu}$, we have $\|\xi\| \leq \mu\, d(x,y)$ and $d(x,y) \leq \nu\|\xi\|$.

These two assumptions are also standard as in Sato et al. (2019). Assumption (4.2.1) ensures that the Riemannian distance can be expressed in terms of the inverse exponential map. Assumption (4.2.2) hence relates the exponential map with the retraction. Indeed, we have $\|\mathrm{Retr}_x^{-1}(y)\| \leq \mu\|\mathrm{Exp}_x^{-1}(y)\|$ and

$\|\mathrm{Exp}_x^{-1}(y)\| \leq \nu \|\mathrm{Retr}_x^{-1}(y)\|$. This assumption is also satisfied with $\mathcal{X}$ sufficiently small (Ring & Wirth, 2012, Lemma 6).

Assumption 4.3.

(4.3.1) The neighbourhood $\mathcal{X}$ is compact with its diameter upper bounded by D, i.e., $\max_{x,y \in \mathcal{X}} d(x,y) \leq D$. In addition, $\mathcal{X}$ has sectional curvature lower bounded by κ.

(4.3.2) For all $x, y \in \mathcal{X}$, there exists a constant $c_R > 0$ such that $\|\mathrm{Retr}_x^{-1}(y) - \mathrm{Exp}_x^{-1}(y)\| \leq c_R \|\mathrm{Retr}_x^{-1}(y)\|^2$.

Assumption (4.3.1) is required to establish the trigonometric distance bound (Lemma 4.5), which is an important result for proving convergence for first-order algorithms (H. Zhang & Sra, 2016). This assumption is natural as for any compact set on Riemannian manifold, its diameter is upper bounded and the curvature is both lower and upper bounded. Although Assumption (4.3.2) can be implied from Assumption (4.2.2) by triangle inequality, we state it separately because it is a common assumption as in Sato et al. (2019).

Remark 4.2. Assumptions 4.2 and 4.3 can be satisfied by further bounding the neighbourhood $\mathcal{X}$. These two sets of Assumptions introduce additional constraints on exponential map that bound its difference with retraction. This is because SVRG-type algorithms require tracing the distances between a remote snapshot point and the iterate sequence, which can only be characterized by the exponential map. This is in contrast with the recursive gradient estimator that only depends on successive iterates.

Algorithm 1: R-AbaSVRG

```
Input: stepsize $\eta$, epoch size $S$, inner loop size $m$, mini-batch size $b$, adaptive batch size parameters $\alpha_1, \alpha_2, \beta_1$, initialization $\tilde{x}^0$, desired accuracy $\epsilon$.
for $s = 1, ..., S$ do
    $x_0^s = \tilde{x}^{s-1}$.
    $B^s = \begin{cases} \min\{\alpha_1 \sigma^2 / \beta_s, n\}, & \text{(finite-sum)} \\ \min\{\alpha_1 \sigma^2 / \beta_s, \alpha_2 \sigma^2 / \epsilon^2\}, & \text{(online)} \end{cases}$
    Draw a sample $\mathcal{B}^s$ from $[n]$ of size $B^s$ without replacement.
    $v_0^s = \mathrm{grad} f_{\mathcal{B}^s}(x_0^s)$.
    $\beta_{s+1} = 0$.
    for $t = 0, ..., m-1$ do
        Draw a sample $\mathcal{I}_t^s$ from $[n]$ of size $b$ with replacement.
        $v_t^s = \mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s}\big(\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - v_0^s\big)$.
        $x_{t+1}^s = \mathrm{Retr}_{x_t^s}(-\eta v_t^s)$.
        $\beta_{s+1} = \beta_{s+1} + \|v_t^s\|^2 / m$.
    end for
    $\tilde{x}^s = x_m^s$.
end for
Output: $\tilde{x}$ uniformly selected at random from $\{\{x_t^s\}_{t=0}^{m-1}\}_{s=1}^{S}$.
```

4.5 Convergence guarantees for Riemannian SVRG and AbaSVRG

Riemannian adaptive batch size SVRG is presented in Algorithm 1 where the batch size B^s is adjusted based on the accumulated gradient information from last epoch. By simply fixing $B^s = n, s = 1, ..., S$, Algorithm 1 reduces to the vanilla R-SVRG under finite-sum setting.

4.5.1 Finite-sum R-SVRG under standard analysis

We first prove nonconvex convergence for finite-sum R-SVRG under standard Lyapunov analysis, which is currently missing in the literature. Theorem 4.1 generalizes the analysis for exponential map and parallel transport (H. Zhang et al., 2016, Theorem 2) to the more general retraction and vector transport.

Theorem 4.1. *Suppose Assumptions 4.1, 4.2 and 4.3 hold and consider Algorithm 1* *with full batch gradients $B^s = n, s = 1, ..., S$ under finite-sum setting. There exist some global constants $a_1, a_2, \mu_0 \in (0,1)$ and $\psi > 0$ such that by choosing $\eta = \frac{\mu_0 b}{(L_l + \theta G)\mu n^{a_1}(\zeta\nu^2 + 2c_R D)^{a_2}}$, $m = \lfloor n^{3/2a_1}/2b\mu_0(\zeta\nu^2 + 2c_R D)^{1-2a_2} \rfloor$, $b \leq n^{a_1}$, the output $\tilde{x}$ after running $T = Sm$ iterations satisfies*

$$\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \frac{(L_l + \theta G) n^{a_1} (\zeta\nu^2 + 2c_R D)^{a_2} \Delta}{bT\psi},$$

where $\Delta := f(\tilde{x}^0) - f(x^*)$, $\zeta \geq 1$ is a curvature constant defined in Lemma 4.5 (Appendix) along with other parameters defined in the Assumptions. Setting $a_1 = 2/3, a_2 = 1/2$, the IFO complexity to achieve ϵ-accurate solution is

$$O\Big(n + \frac{(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{1/2} n^{2/3}}{\epsilon^2}\Big).$$

Proof sketch. We first derive bounds on the norm of modified gradients $\|v_t^s\|^2$ and also on the distance between current iterates and the reference point within an epoch, $d^2(x_t^s, x_0^s)$. Then we construct a Lyapunov function $f(x_t^s) + c_t d^2(x_t^s, x_0^s)$. We therefore can show the norm of gradient at current iterate is upper bounded by the difference in Lyapunov functions at consecutive iterates. In the process, the trigonometric distance bound is applied to relate $d^2(x_t^s, x_0^s)$ to $d^2(x_{t+1}^s, x_0^s)$. By carefully choosing parameters and managing the coefficients c_t, we obtain the desired result. □

Theorem 4.1 is an extension of (H. Zhang et al., 2016, Theorem 2), which is analyzed with exponential map and parallel transport. Because we use retraction and vector transport as approximation, more parameters are involved. But if we choose the special exponential map and parallel transport, the parameters simplify as $L_l = L, \theta = 0, \nu = 1, c_R = 0$ and the complexity reduces to $O(n + \frac{Ln^{2/3}\zeta^{1/2}}{\epsilon^2})$ as in H. Zhang et al. (2016).

4.5.2 R-AbaSVRG and R-SVRG under new analysis

In this section, we first show convergence and gradient complexity of R-AbaSVRG. As a corollary, we derive convergence results of vanilla R-SVRG with much simpler analysis. The new formulation also allows analysis of R-SVRG under online setting, which is to the best of our knowledge, novel on Riemannian manifold. Define the sigma algebras $\mathcal{F}_t^s := \{\mathcal{B}^1, ..., \mathcal{I}_{m-1}^1, \mathcal{B}^2, ..., \mathcal{I}_{m-1}^2, ..., \mathcal{B}^s, ..., \mathcal{I}_{t-1}^s\}$. From Algorithm 1, v_{t-1}^s and x_t^s are measurable in $\mathcal{F}_t^s$. Thus, conditional on $\mathcal{F}_t^s$, randomness at current iteration t only comes from sampling $\mathcal{I}_t^s$ or $\mathcal{B}^s$. We first present a lemma that bounds the estimation error of the SVRG-type modified gradient v_t^s to the full gradient $\text{grad}f(x_t^s)$.

Lemma 4.1. *Suppose Assumptions 4.1 and 4.2 hold and consider Algorithm 1. Then we have the estimation bound as*

$$\mathbb{E}[\|v_t^s - \text{grad}f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{t}{b}(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 \sum_{i=0}^{t-1} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{\sigma^2}{B^s}.$$

Proof sketch. The idea is to first show that $\mathbb{E}[\|v_t^s - \text{grad}f(x_t^s)\|^2 | \mathcal{F}_t^s]$ can be bounded by $d^2(x_t^s, x_0^s)$ and $\|v_0^s - \text{grad}f(x_0^s)\|^2$. Then simply applying triangle inequality recursively along with Assumption (4.2.2), we bound $d^2(x_t^s, x_0^s)$ by $O(t \sum_{i=0}^{t-1} \|v_i^s\|^2)$. And $\mathbb{E}[\|v_0^s - \text{grad}f(x_0^s)\|^2 | \mathcal{F}_0^s]$ are bounded based on Assumption (4.1.4). Then the law of iterated expectation is applied to complete the proof. □

This suggests the deviation of modified gradient v_t^s to the full gradient can be controlled with proper choice of parameters. When choosing $B^s = n$ as in the vanillas R-SVRG, the second term vanishes and we hence obtain a tighter bound on the estimation error. Next, based on this lemma, we analyze convergence for R-AbaSVRG as follows.

Theorem 4.2 (Convergence of R-AbaSVRG). *Suppose Assumptions 4.1 and 4.2 hold and consider Algorithm 1. Choose a fixed stepsize* $\eta \leq \frac{2-\frac{2}{\alpha}}{L+\sqrt{L^2+4(1-\frac{1}{\alpha})\frac{(L_l+\theta G)^2\mu^2\nu^2m^2}{b}}}$ *for*

$\alpha \geq 4$. Then under both finite-sum and online settings, output $\tilde{x}$ after running $T = Sm$ iterations satisfies

$$\mathbb{E}\|\mathrm{grad}f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \frac{\epsilon^2}{2},$$

where $\Delta := f(\tilde{x}^0) - f(x^*)$ and ϵ is the desired accuracy.

Proof sketch. The analysis is motivated by Ji et al. (2020). We start with the retraction L-smoothness to obtain $f(x_{t+1}^s) - f(x_t^s) \leq -\frac{\eta}{2}\|\mathrm{grad}f(x_t^s)\|^2 + \frac{\eta}{2}\|v_t^s - \mathrm{grad}f(x_t^s)\|^2 - (\frac{\eta}{2} - \frac{L\eta^2}{2})\|v_t^s\|^2$. This is different to the bound in Ji et al. (2020), i.e., $f(x_{t+1}^s) - f(x_t^s) \leq \frac{\eta}{2}\|v_t^s - \mathrm{grad}f(x_t^s)\|^2 - (\frac{\eta}{2} - \frac{L\eta^2}{2})\|v_t^s\|^2$. Our bound is apparently stronger and we can also directly bound $\|\mathrm{grad}f(x_t^s)\|^2$ based on this inequality. This allows for a simpler proof and a choice of larger stepsize than Ji et al. (2020). Taking the expectation and applying Lemma 4.1, we can bound $\mathbb{E}\|\mathrm{grad}f(x_t^s)\|^2$ by cumulative sum of $\mathbb{E}\|v_t^s\|^2$. Then carefully selecting the parameters yields the result. □

Now we show the gradient complexities of R-AbaSVRG and R-SVRG respectively in Corollary 4.1 and 4.2.

Corollary 4.1 (IFO complexity of R-AbaSVRG)**.** *With same Assumptions in Theorem 4.2, choose $b = m^2$, $\eta = \frac{3}{2L+2\sqrt{L^2+3(L_l+\theta G)^2\mu^2\nu^2}} (\alpha = 4)$. Set $m = \lfloor n^{1/3} \rfloor$ under finite-sum setting and $m = (\frac{\sigma}{\epsilon})^{2/3}$ under online setting. The IFO complexity of Algorithm 1 to achieve ϵ-accurate solution is given by*

$$\begin{cases} O\big(\tilde{B} + \frac{\Theta_1 \tilde{B}}{n^{1/3}\epsilon^2} + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_1 \tilde{B}}{\sigma^{2/3}\epsilon^{4/3}} + \frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big), & \textit{(online)} \end{cases}$$

where $\Theta_1 := L + \sqrt{L^2 + \varrho_1(L_l + \theta G)^2\mu^2\nu^2}$ with $\varrho_1 > 0$ is a constant that does not depend on any parameter. $\tilde{B}$ is the average batch size, i.e., $\tilde{B} := \frac{1}{S}\sum_{s=1}^S \min\{\alpha_1\sigma^2/\beta_s, n\}$ under finite-sum setting and $\tilde{B} := \frac{1}{S}\sum_{s=1}^S \min\{\alpha_1\sigma^2/\beta_s, \alpha_2\sigma^2/\epsilon^2\}$ under online setting.

Corollary 4.2 (Convergence and IFO complexity of R-SVRG under new analysis)**.** *With the same assumptions as in Theorem 4.2 and consider Algorithm 1 with fixed batch size $B^s = B$ for $s = 1, ..., S$. Choose a fixed stepsize $\eta \leq \frac{2}{L+\sqrt{L^2+4\frac{(L_l+\theta G)^2\mu^2\nu^2m^2}{b}}}$. Output $\tilde{x}$ after running $T = Sm$ iterations satisfies $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B}$. If we further choose $b = m^2, \eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2\mu^2\nu^2}}$ and the following parameters*

$$\begin{aligned} B &= n, \quad m = \lfloor n^{1/3} \rfloor \quad \textit{(finite-sum)} \\ B &= \frac{2\sigma^2}{\epsilon^2}, \quad m = (\frac{\sigma}{\epsilon})^{2/3} \quad \textit{(online)} \end{aligned}$$

IFO complexity to obtain ϵ-accurate solution is

$$\begin{cases} O\big(n + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big), & \textit{(online)} \end{cases}$$

We first compare the complexities of vanilla R-SVRG under two analysis frameworks. From Theorem 4.1, the complexity is $O(n + \frac{(L_l+\theta G)(\zeta\nu^2+2c_R D)^{1/2}n^{2/3}}{\epsilon^2})$, which is the same as $O(n + \frac{\Theta_1 n^{2/3}}{\epsilon^2})$ in Corollary 4.2 up to a constant.

Remark 4.3. Under standard analysis of Lyapunov function, the complexity is further controlled by the curvature constant ζ. We note that such constant does not occur in the complexity with our new analysis. The is because we replace the bounded curvature assumption, i.e., Assumption (4.3.1) with Assumption (4.2.2) that relates Riemannian distance to inverse retraction. Thus the curvature ζ is hidden in the constants μ, ν.

Nevertheless, this leads to much simpler analysis using the triangle inequality, rather than the complex trigonometric distance bound and the Lyapunov function. Moreover, such analysis allows insights to be drawn between SVRG and SRG-type methods under a unified framework (see Section 4.6).

Comparing with R-SD that requires a complexity of $O(n + \frac{n}{\epsilon^2})$, R-SVRG is superior with complexity lower by a factor of $O(n^{1/3})$. The new analysis also provides a complexity of $O\big(\frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big)$ under online setting, which is the first online complexity established on SVRG-type methods over Riemannian manifold. This corresponds to the best known rate $O\big(\frac{1}{\epsilon^{10/3}}\big)$ for SVRG-based algorithms on Euclidean space, such as SCSG (Lei et al., 2017) and ProxSVRG+ (Z. Li & Li, 2018). Compared with the $O(\frac{1}{\epsilon^4})$ complexity of R-SGD, R-SVRG under online setting outperforms R-SGD by a factor of $O\big(\frac{1}{\epsilon^{2/3}}\big)$.

From Theorem 4.2, R-AbaSVRG enjoys the same convergence rate as vanilla R-SVRG. This shows that the iteration complexities to achieve ϵ-accurate solution are identical. Therefore with the same choices of parameters, R-AbaSVRG requires $O\big(\tilde{B} + \frac{\Theta_1 \tilde{B}}{n^{1/3}\epsilon^2} + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big)$ under finite-sum setting and $O\big(\frac{\Theta_1 \tilde{B}}{\sigma^{2/3}\epsilon^{4/3}} + \frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big)$ under online setting. These complexities can be theoretically much lower than R-SVRG from the definition of $\tilde{B}$. That is, because $\tilde{B} \leq n$ under finite-sum setting, the complexity of R-AbaSVRG is at most $O\big(n + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big)$, which matches the complexity of R-SVRG. Similar argument holds for online setting.

Lastly, we comment on the choice of parameters. Theorem 4.1 suggests a choice of $m = O(n/b)$ with $b \leq n^{2/3}$ while both Corollary 4.1 and 4.2 simply select $b = m^2 = n^{2/3}$. Similar to Ji et al. (2020), our new analysis does not easily allow more flexible choices of b and m as we do not construct any nontrivial auxiliary variable to achieve this purpose.

4.6 Convergence guarantees for Riemannian SRG and AbaSRG

The key steps of R-AbaSRG in Algorithm 2 are nearly identical to R-AbaSVRG except that the modified gradient v_t^s is constructed recursively from v_{t-1}^s. We first similarly present a bound on the gradient estimation error for SRG-type modified gradient.

Algorithm 2: R-AbaSRG

```
Input: stepsize $\eta$, epoch length $S$, inner loop size $m$, mini-batch size $b$,
  adaptive batch size parameters $\alpha_1, \alpha_2, \beta_1$, initialization $\tilde{x}^0$, desired accuracy
  $\epsilon$.
for $s = 1, ..., S$ do
  $x_0^s = \tilde{x}^{s-1}$.
  $B^s = \begin{cases} \min\{\alpha_1 \sigma^2 / \beta_s, n\}, & \text{(finite-sum)} \\ \min\{\alpha_1 \sigma^2 / \beta_s, \alpha_2 \sigma^2 / \epsilon^2\}, & \text{(online)} \end{cases}$
  Draw a sample $\mathcal{B}^s$ from $[n]$ of size $B^s$ without replacement.
  $v_0^s = \text{grad} f_{\mathcal{B}^s}(x_0^s)$.
  $x_1^s = \text{Retr}_{x_0^s}(-\eta \, v_0^s)$.
  $\beta_{s+1} = \|v_0^s\|^2 / m$.
  for $t = 1, ..., m-1$ do
    Draw a sample $\mathcal{I}_t^s$ from $[n]$ of size $b$ with replacement.
    $v_t^s = \text{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s} \big( \text{grad} f_{\mathcal{I}_t^s}(x_{t-1}^s) - v_{t-1}^s \big)$.
    $x_{t+1}^s = \text{Retr}_{x_t^s}(-\eta \, v_t^s)$.
    $\beta_{s+1} = \beta_{s+1} + \|v_t^s\|^2 / m$.
  end for
  $\tilde{x}^s = x_m^s$.
end for
Output: $\tilde{x}$ uniformly selected at random from $\{\{x_t^s\}_{t=0}^{m-1}\}_{s=1}^{S}$.
```

Lemma 4.2. *Suppose Assumption 4.1 hold and consider Algorithm 2. Then we obtain the estimation bound as*

$$\mathbb{E}[\|v_t^s - \text{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{(L_l + \theta G)^2 \eta^2}{b} \sum_{i=0}^{t} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{\sigma^2}{B^s}.$$

Proof sketch. We first bound $\mathbb{E}[\|v_t^s - \text{grad} f(x_t^s)\|^2 | \mathcal{F}_t^s]$ by $\|v_{t-1}^s\|^2$ and $\|v_{t-1}^s - \text{grad} f(x_{t-1}^s)\|^2$. Applying the bound recursively with proper conditioning yields the result. □

Compared with Lemma 4.1, the bound for SRG-type gradient is tighter than SVRG-type gradient as the first term on the right hand side is smaller by a factor of $O(t)$. This is because the use of recursive gradients does not accumulate errors, in contrast to using gradients of a distant reference point. Next, we prove convergence and complexity results for R-SRG and R-AbaSRG.

Theorem 4.3 (Convergence of R-AbaSRG). *Suppose Assumption 4.1 holds and consider Algorithm 2. Choose a fixed stepsize* $\eta \leq \frac{2-\frac{2}{\alpha}}{L+\sqrt{L^2+4(1-\frac{1}{\alpha})\frac{(L_l+\theta G)^2 m}{b}}}$ *for* $\alpha \geq 4$*. Then under both finite-sum and online setting, output* $\tilde{x}$ *after running* $T = Sm$ *iterations satisfies* $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \frac{\epsilon^2}{2}$.

Corollary 4.3 (IFO complexity of R-AbaSRG). *With the same Assumptions and settings in Theorem 4.3, choose* $b = m$, $\eta = \frac{3}{2L+2\sqrt{L^2+3(L_l+\theta G)^2}} (\alpha = 4)$. *Set* $m = \lfloor n^{1/2} \rfloor$ *under finite-sum setting and* $m = \frac{\sigma}{\epsilon}$ *under online setting. The IFO complexity of Algorithm 1 to obtain* ϵ*-accurate solution is*

$$\begin{cases} O\big(\tilde{B} + \frac{\Theta_2 \tilde{B}}{\sqrt{n}\epsilon^2} + \frac{\Theta_2\sqrt{n}}{\epsilon^2}\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_2 \tilde{B}}{\sigma\epsilon} + \frac{\Theta_2 \sigma}{\epsilon^3}\big), & \textit{(online)} \end{cases}$$

where $\Theta_2 := L + \sqrt{L^2 + \varrho_2(L_l+\theta G)^2}$ *with* $\varrho_2 > 0$ *independent of any parameter.* $\tilde{B}$ *is the same average batch size defined in Corollary 4.1.*

Corollary 4.4 (Double-loop convergence and IFO complexity of R-SRG). *With the same assumptions in Theorem 4.3 and consider Algorithm 2 with fixed batch size* $B^s = B$*, for* $s = 1, ..., S$*. Consider a stepsize* $\eta \leq \frac{2}{L+\sqrt{L^2+4\frac{(L_l+\theta G)^2 m}{b}}}$*. After running* $T = Sm$ *iterations, output* $\tilde{x}$ *satisfies* $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B}$. *If we further choose* $b = m$, $\eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2}}$ *and following parameters*

$$B = n, \quad m = \lfloor n^{1/2} \rfloor, \quad \textit{(finite-sum)}$$
$$B = \frac{2\sigma^2}{\epsilon^2}, \quad m = \frac{\sigma}{\epsilon}, \quad \textit{(online)}$$

IFO complexity to obtain ϵ*-accurate solution is*

$$\begin{cases} O\big(n + \frac{\Theta_2\sqrt{n}}{\epsilon^2}\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_2 \sigma}{\epsilon^3}\big), & \textit{(online)} \end{cases}$$

We first compare Corollary 4.4 with the existing complexity result on R-SRG. In Kasai et al. (2018b), they only prove single-loop convergence for R-SRG under finite-sum setting where $\tilde{x}^s$ for the next epoch is uniformly chosen from iterates within the current epoch. This leads to a complexity of $O(n + \frac{\Theta^2}{\epsilon^4})$, with $\Theta := \max\{L, \sqrt{L_l^2 + \theta^2 G^2}\}$. However, this is suboptimal when $n \leq O(\frac{1}{\epsilon^4})$. Indeed, under such condition, the optimal complexity has been proved to be $O(n + \frac{L\sqrt{n}}{\epsilon^2})$ in the Euclidean space (Fang et al., 2018). By simply choosing $\tilde{x}^s$ as the last iterate of current epoch, we prove double-loop convergence for R-SRG with a complexity of $O(n + \frac{\Theta_2\sqrt{n}}{\epsilon^2})$, matching the lower bound up to some constants.

Furthermore, we prove the first online complexity for R-SRG, which is $O(\frac{\Theta_2 \sigma}{\epsilon^3})$. The complexity of $O(\epsilon^{-3})$ has been recently proved to be optimal for online optimization in the Euclidean space (Arjevani et al., 2023). Thus, R-SRG achieves the optimal complexity under both finite-sum and online settings.

Remark 4.4 (Comparison to R-SPIDER)**.** R-SPIDER (J. Zhang et al., 2018; P. Zhou, Yuan, Yan, & Feng, 2019) also achieves the same optimal complexities as in Corollary 4.4. Nevertheless, R-SPIDER bears high relevance to R-SRG. In fact, the only key difference of R-SPIDER is to normalize gradient v_t^s before taking a retraction step. Therefore, by selecting a small stepsize $\eta = O(\frac{\epsilon}{L})$, they can bound distances between successive iterates $d(x_t, x_{t+1})$ by a small quantity $O(\epsilon)$. Also, we highlight that our result is stronger than P. Zhou, Yuan, Yan, & Feng (2019) because the output $\tilde{x}$ satisfies $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \epsilon^2$, which implies $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\| \leq \epsilon$ (Definition 4.2 of ϵ-accurate solution) by Jensen's inequality. The latter is considered and analyzed in P. Zhou, Yuan, Yan, & Feng (2019).

Corollary 4.4 suggests the gradient normalization in R-SPIDER is non-essential for acceleration. Similar claims are also made in Z. Wang, Ji, et al. (2019). Thus we note that R-SPIDER is equivalent to R-SRG with a variable stepsize $\eta / \|v_t^s\|$. But in practical settings, R-SRG has an advantage of using a large and fixed stepsize. More discussions are in Section 4.8.

Comparing with R-SVRG, R-SRG strictly improves on the complexity by a factor of $O(n^{1/6})$ under finite-sum setting and $O\big((\frac{\sigma}{\epsilon})^{1/3}\big)$ under online setting. Similar to R-AbaSVRG, R-AbaSRG maintains the same iteration complexity as R-SRG and thus with the same choices of inner loop size m and mini batch size b, ϵ-accurate solution can be returned with potentially much lower total IFO complexity.

Lastly, the strict parameter choices, $b = m = \sqrt{n}$ are unnecessary to achieve the optimal rate. That is, consider R-SRG under finite-sum setting with the choice $mb = n$. From the proof of Corollary 4.4, the number of epochs required to achieve ϵ-accurate solution is $S = \frac{2\Delta}{m\eta\epsilon^2} = \frac{L+\sqrt{L^2+4(L_l+\theta G)^2\frac{m}{b}}}{m\epsilon^2} \leq \frac{2L\sqrt{1+4\frac{(L_l+\theta G)^2 m}{L^2 b}}}{m\epsilon^2}$. Then total IFO complexity is given by $S(n+2mb) \leq n + \frac{6L\sqrt{b^2+4\frac{(L_l+\theta G)^2 n}{L^2}}}{\epsilon^2}$. Hence as long as $b \leq \sqrt{n}$, total complexity is at most $O(n + \frac{\sqrt{n}}{\epsilon^2})$ ignoring constants. This suggests that we can freely choose $b \in [1, \sqrt{n}]$ and $m \in [\sqrt{n}, n]$ as long as $mb = n$. stepsize can also be selected larger when choosing a larger mini-batch size. We remark that the total complexity does not improve for larger mini-batch size. But it potentially provides linear speedups in distributed systems where b stochastic gradients are computed in parallel (Goyal et al., 2017).

4.7 Convergence under gradient dominance

As an important class of nonconvex functions, gradient dominated functions (Definition 4.1) assume existence of a global solution x^* where function value difference of any point to x^* is upper bounded by its gradient. This condition allows linear convergence to be established for nonconvex functions. It is worth noticing that retraction ς-strongly convex function is $\frac{1}{2\varsigma}$-gradient dominated.[1]

Common strategy of adapting variance reduction methods to gradient dominance condition is by restarting (H. Zhang et al., 2016; J. Zhang et al., 2018). We

[1] Proof of this claim can be seen in (H. Zhang et al., 2016, Corollary 5). Retraction ς-strongly convex f satisfies $f(y) \geq f(x) + \langle \mathrm{grad} f(x), \xi \rangle + \frac{\varsigma}{2}\|\xi\|^2$, for all $x, y = R_x(\xi) \in \mathcal{M}$.

Algorithm 3: R-GD-VR

```
**Input:** Initial accuracy $\epsilon_0$ and desired accuracy $\epsilon$, initialization $x_0$.
**for** $k = 1, ..., K$ **do**
    $\epsilon_k = \frac{\epsilon_{k-1}}{2}$ and set other parameters (args) according to the solver choice.
    (R-SVRG):    $x_k = \text{R-AbaSVRG}(x_{k-1}, \epsilon_k, B_k, \text{args})$
    (R-AbaSVRG):    $x_k = \text{R-AbaSVRG}(x_{k-1}, \epsilon_k, \text{args})$
    (R-SRG):    $x_k = \text{R-AbaSRG}(x_{k-1}, \epsilon_k, B_k, \text{args})$
    (R-AbaSRG):    $x_k = \text{R-AbaSRG}(x_{k-1}, \epsilon_k, \text{args})$
**end for**
**Output:** $x_K$.
```

hence provide a unified framework in Algorithm 3. For vanilla R-SVRG and R-SRG, we consider Algorithm 1 and 2 respectively with batch size $B_k = n$ under finite-sum setting and $B_k = \frac{2\sigma^2}{\epsilon_k^2}$.

The idea is to gradually shrink the desired accuracy at each mega epoch, thus requiring increasing number of iterations S_k. By running sufficient number of mega epochs, output x_K is guaranteed to be ϵ-accurate. We first show the linear convergence for any variance reduction method under gradient dominance condition.

Theorem 4.4. *Suppose Assumptions 4.1 and 4.2 hold and suppose function f is τ-gradient dominated. Consider Algorithm 3 with any solver and choose appropriate parameters to return ϵ_k-accurate solution. Then at mega epoch k, iterate x_k satisfies $\mathbb{E}\|\text{grad} f(x_k)\| \leq \frac{\epsilon_0}{2^k}$ and $\mathbb{E}[f(x_k) - f(x^*)] \leq \frac{\tau\epsilon_0^2}{4^k}$.*

Based on this result, we show the IFO complexities of R-AbaSVRG, R-SVRG in Corollary 4.5 and R-AbaSRG and R-SRG in Corollary 4.6.

Corollary 4.5 (Complexities of R-AbaSVRG and R-SVRG)**.** *Consider R-AbaSVRG solver with the following parameters at each mega epoch. $\eta = \frac{3}{2L+2\sqrt{L^2+3(L_l+\theta G)^2\mu^2\nu^2}}$, $b_k = m_k^2$, where $m_k = \lfloor n^{1/3} \rfloor$ under finite-sum setting and $m_k = (\frac{\sigma}{\epsilon_k})^{2/3}$ under online* *setting. To achieve ϵ-accurate solution, it requires IFO complexity of*

$$\begin{cases} O\big(\sum_{k=1}^{K} \tilde{B}_k(1+\frac{\Theta_1\tau}{n^{1/3}}) + (\Theta_1 n^{2/3}\tau)\log(\frac{1}{\epsilon})\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_1\tau\sum_{k=1}^{K}\tilde{B}_k\epsilon_k^{2/3}}{\sigma^{2/3}} + \frac{\Theta_1\tau\sigma^{4/3}}{\epsilon^{4/3}}\big), & \textit{(online)} \end{cases}$$

where the average batch size at mega epoch k is $\tilde{B}_k := \frac{1}{S_k}\sum_{s=1}^{S_k}\min\{\alpha_1\sigma^2/\beta_s, n\}$ under finite-sum setting and $\tilde{B}_k := \frac{1}{S_k}\sum_{s=1}^{S_k}\min\{\alpha_1\sigma^2/\beta_s, \alpha_2\sigma^2/\epsilon_k^2\}$ under online setting.

Consider R-SVRG solver with the same parameters except for $\eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2\mu^2\nu^2)}}$ and $B_k = n$ under finite-sum setting and $B_k = \frac{2\sigma^2}{\epsilon_k^2}$ under online setting. To achieve ϵ-accurate solution, it requires IFO complexity of

$$\begin{cases} O\big((n+\Theta_1\tau n^{2/3})\log(\frac{1}{\epsilon})\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_1\tau\sigma^{4/3}}{\epsilon^{4/3}}\big), & \textit{(online)} \end{cases}$$

Corollary 4.6 (Complexities of R-AbaSRG and R-SRG)**.** *Consider R-AbaSRG solver with $\eta = \frac{3}{2L+2\sqrt{L^2+3(L_l+\theta G)^2}}$, $b_k = m_k$ where $m_k = \lfloor n^{1/2}\rfloor$ under finite-sum setting and $m_k = \frac{\sigma}{\epsilon_k}$ under online setting. To achieve ϵ-accurate solution, it requires IFO complexity of*

$$\begin{cases} O\big(\sum_{k=1}^{K} \tilde{B}_k(1+\frac{\Theta_2\tau}{n^{1/2}}) + (\Theta_2 n^{1/2}\tau)\log(\frac{1}{\epsilon})\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_2\tau\sum_{k=1}^{K}\tilde{B}_k\epsilon_k}{\sigma} + \frac{\Theta_2\tau\sigma}{\epsilon}\big), & \textit{(online)} \end{cases}$$

Consider R-SRG solver with the same parameters except for $\eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2}}$ and $B_k = n$ under finite-sum setting and $B_k = \frac{2\sigma^2}{\epsilon_k^2}$ under online setting. To achieve ϵ-accurate solution, it requires IFO complexity of

$$\begin{cases} O\big((n+\Theta_2\tau n^{1/2})\log(\frac{1}{\epsilon})\big), & \textit{(finite-sum)} \\ O\big(\frac{\Theta_2\tau\sigma}{\epsilon}\big), & \textit{(online)} \end{cases}$$

Apart from the results, we also prove in Appendix that under gradient dominance condition, R-SD requires a complexity of $O\big((n + L\tau n)\log(\frac{1}{\epsilon})\big)$ and R-SGD requires $O\big(\frac{LG^2}{\epsilon^2}\big)$. These results are consistent with the results established in the Euclidean space (Polyak, 1963; Karimi et al., 2016).

Compared with R-SD and R-SGD, R-SVRG requires fewer gradient queries, with a factor of $O(n^{1/3})$ lower than R-SD and a factor of $O\big(\frac{1}{\epsilon^{2/3}}\big)$ lower than R-SGD. R-SRG further improves on the complexities by $O(n^{1/6})$ and $O\big(\frac{1}{\epsilon^{1/3}}\big)$ under finite-sum and online settings respectively.

Similar to the general nonconvex case, these results can be further improved by batch size adaptation. For example, consider R-AbaSVRG under finite-sum setting with complexity given by $O\big(\sum_{k=1}^{K}\tilde{B}_k(1+\frac{\Theta_1\tau}{n^{1/3}}) + (\Theta_1 n^{2/3}\tau)\log(\frac{1}{\epsilon})\big)$. By definition, $\sum_{k=1}^{K}\tilde{B}_k(1+\frac{\Theta_1\tau}{n^{1/3}}) \leq \sum_{k=1}^{K} n(1+\frac{\Theta_1\tau}{n^{1/3}}) = (n+\Theta_1 n^{2/3}\tau)\log(\frac{1}{\epsilon})$. Hence the complexity is at worst the same as the vanilla R-SVRG, which is $O\big((n + \Theta_1 n^{2/3}\tau)\log(\frac{1}{\epsilon})\big)$. These arguments also hold for R-AbaSRG and online setting.

Existing result (Kasai et al., 2018b) shows that R-SRG under gradient dominance condition requires a complexity of $O\big((n+\tau^2\Theta^2)\log(\frac{1}{\epsilon^2})\big)$. This is because the inner-loop convergence does not require restarting the algorithm and simply running $O\big(\log(\frac{1}{\epsilon^2})\big)$ outer iterations is sufficient to achieve linear convergence. Comparing with the rate of $O\big((n+\Theta_2\tau n^{1/2})\log(\frac{1}{\epsilon})\big)$ under the current analysis, we again highlight a trade-off between the sample size and the desired accuracy. When n is small relative to ϵ, our rate is superior. The complexity we prove for R-SRG matches that of R-SPIDER (P. Zhou, Yuan, Yan, & Feng, 2019) up to some constants.

Finally, we comment on the convergence under strongly convex functions in Remark 4.5, and also discuss the convergence under the more restricted exponential map and parallel transport in Remark 4.6.

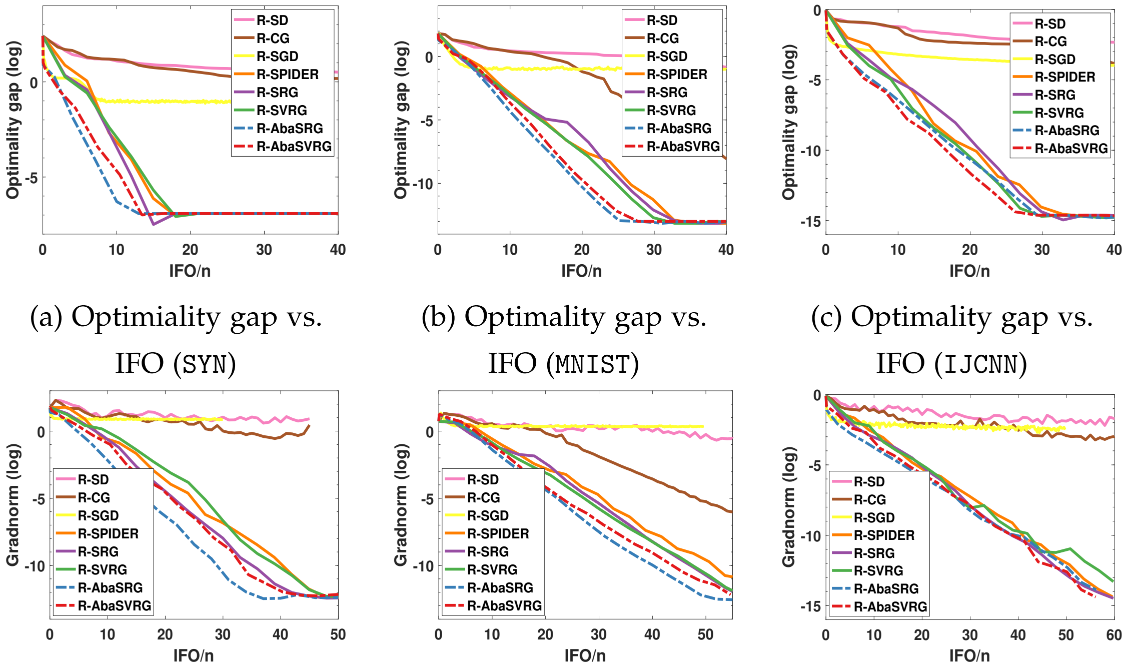
Remark 4.5 (Convergence under strongly convex functions)**.** The results in this section can be readily extended for the retraction strongly convex functions, a special instance of gradient dominated functions. For example, under finite-sum setting, suppose f is retraction ς-strongly convex, R-SVRG requires a complexity of $O\big((n + \Theta_1 \varsigma^{-1} n^{2/3}) \log(\frac{1}{\epsilon})\big)$ and R-SRG requires a complexity of $O\big((n + \Theta_2 \varsigma^{-1} n^{1/2}) \log(\frac{1}{\epsilon})\big)$.

Remark 4.6 (Convergence under exponential map and parallel transport)**.** Our analysis of retraction and vector transport easily adapts to the exponential map and parallel transport for both general nonconvex and gradient dominated functions. That is, we can simply replace the assumptions of retraction L-smooth and L_l-Lipschitz by geodesic L-smoothness and L-Lipschitzness (H. Zhang et al., 2016). Therefore, Θ_1, Θ_2 reduce to L as $\theta = 0$, $\mu = \nu = 1$. See table 4.1 for comparisons. In general, $\Theta_1, \Theta_2 > L$ and hence the complexity bounds become tighter under the exponential map and parallel transport. The improvement on the complexity however, does not bring practical advantages due to the expensive computation of the operations (P. Zhou, Yuan, Yan, & Feng, 2019).

Note that the curvature constant ζ that appears in the standard complexity results of R-SVRG does not occur under the new analysis. In addition, the constants μ, ν (regulated by the curvature) also vanish when considering the exponential map and parallel transport. This leads to improved complexity bounds that are curvature-free, which are the first such results to the best of our knowledge. However, whether the curvature affects the convergence rate of Riemannian optimization remains an open question. Similar discussions in this regard can be found in Criscitiello & Boumal (2019).

4.8 Experiments

This section empirically evaluates batch size adaptation on variance reduction algorithms over a number of tasks. To make a comparison with some first-order baseline methods, we also include results from R-SD, R-SGD as well as the Riemannian conjugate gradient (R-CG) (Absil et al., 2009). Except for R-



(a) Optimality gap vs. IFO (SYN)

(b) Optimality gap vs. IFO (MNIST)

(c) Optimality gap vs. IFO (IJCNN)

(d) Gradient norm vs. IFO (SYN)

(e) Gradient norm vs. IFO (MNIST)

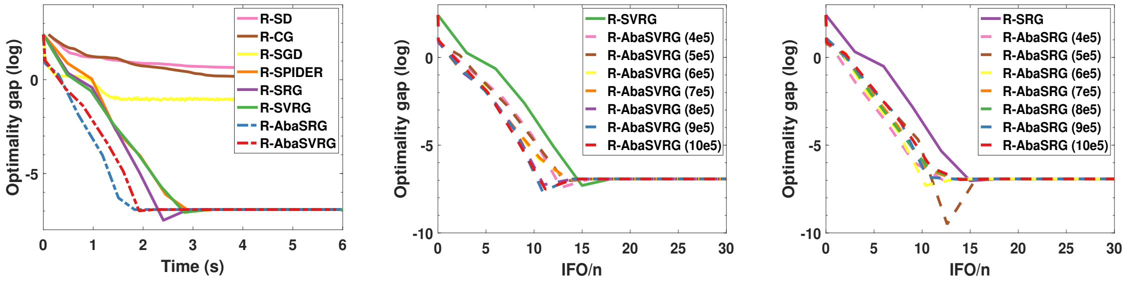
(f) Gradient norm vs. IFO (IJCNN)

Figure 4.1: PCA problem on Grassmann manifold

SD and R-CG that have inbuilt line search algorithm, all other methods require fine-tuning stepsize. For simplicity, we consider a fixed stepsize η for SVRG and SRG based methods. Following Kasai et al. (2018b); P. Zhou, Yuan, Yan, & Feng (2019), we set a decaying stepsize for R-SGD, i.e., $\eta_k = \eta(1 + \eta\lambda_\eta k)$ and an adaptive stepsize for R-SPIDER, i.e., $\eta_k = \alpha_\eta^{\lfloor k/p \rfloor} \cdot \beta_\eta$ where k is the iteration index and p is the batch gradient frequency for R-SPIDER.

Remark 4.7 (Stepsize of R-SPIDER)**.** In theory, R-SPIDER requires a small stepsize proportional to desired accuracy (J. Zhang et al., 2018; P. Zhou, Yuan, Yan, & Feng, 2019). Such choice of stepsize empirically slows down convergence particularly for initial epochs where gradient is large. The adaptive stepsize generally performs better, as seen from P. Zhou, Yuan, Yan, & Feng (2019).

Some global parameter settings are as follows. For variance reduction methods and their adaptive batch size versions, we set inner loop size m, mini-batch size b and batch gradient frequency p to be $m = b = p = \sqrt{n}$, in accordance with



(a) Optimality gap vs. time (SYN)

(b) Sensitivity of R-AbaSVRG to c_β

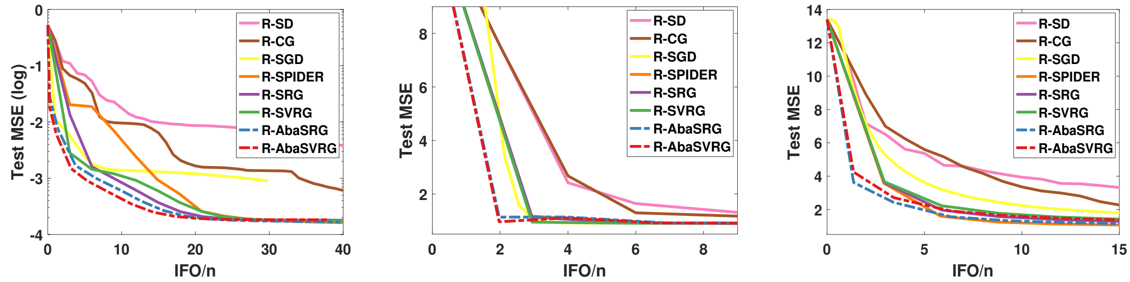
(c) Sensitivity of R-AbaSRG to c_β

Figure 4.2: Additional PCA results on Syn dataset

theory. We set $\lambda_\eta = 0.01$ for R-SGD and select α_η from $\{0.1, 0.2, ..., 0.8, 0.85, 0.9, 0.95, 0.99\}$ and β_η from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ for R-SPIDER. This search grid is more extensive than the one adopted in P. Zhou, Yuan, Yan, & Feng (2019) as we found for some applications, a smaller search grid is unable to ensure convergence. We set adaptive batch size $B^s = \min\{n, c_\beta/\beta_s\}, s > 1$. Initial batch size B^1 is set to be 50 and therefore we only need to tune c_β. To achieve fairness in comparisons, we first tune stepsize η on vanilla variance reduction methods. Then the best tuned η is fixed for their adaptive versions, where c_β is tuned accordingly. We select η from $\{1, 2, ..., 9\} \times 10^q$ and c_β from $\{1, 3, ..., 15\} \times 10^l$, where q, l are to be determined for each problem.

Practical implementation of batch size adaptation. On Riemannian manifolds, due to the error caused by vector transport, inexact batch gradients at initial epochs can further deviate when inner loop size m is large. Hence practically, we set $m_s = \min\{B^s, \sqrt{n}\}$. Also, mini-batch size is set to be $b_s = \min\{B^s, \sqrt{n}\}$ because it is unreasonable for batch gradient to be less exact than mini-batch gradients.

All experiments are coded in Matlab based on the ManOpt package (Boumal et al., 2014) on a i5-8600 3.1GHz CPU processor. The codes are available on `https://github.com/andyjm3/R-AbaVR`.

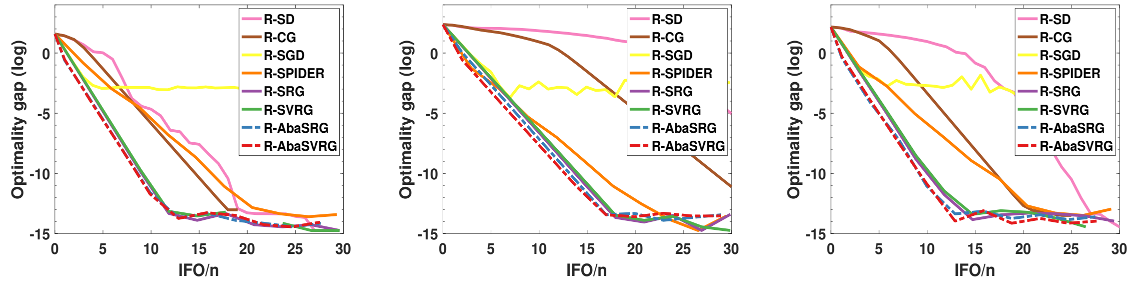


(a) Test MSE vs. IFO (SYN)

(b) Test MSE vs. IFO (Netflix)

(c) Test MSE vs. IFO (Movielens)

Figure 4.3: LRMC problem on Grassmann manifold



(a) Optimality gap vs. IFO (SYN)

(b) Optimality gap vs. IFO (YaleB)

(c) Optimality gap vs. IFO (Kylberg)

Figure 4.4: RKM problem on SPD manifold

4.8.1 PCA and LRMC on Grassmann manifold

We first consider principal component analysis (PCA) and low rank matrix completion (LRMC) over the set of subspaces, which is often identified as the Grassmann manifold.

Preliminaries on Grassmann manifold. Grassmann manifold $\mathcal{G}(r, d)$, is the set of r-dimensional subspaces in $\mathbb{R}^d$ ($r \leq d$). Points on Grassmann manifold are equivalence classes of column orthonormal matrices under the orthogonal group $\mathcal{O}(r)$. That is, a point on Grassmann manifold can be represented by a column orthonormal matrix $\mathbf{U} \in \mathbb{R}^{d \times r}$ such that $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_r$ and any point is deemed equivalent to $\mathbf{U}$ if they can be represented as $\mathbf{U}\mathbf{R}$ for any $\mathbf{R} \in \mathcal{O}(r)$. Recall that Stiefel manifold $\mathrm{St}(r, d)$ is the set of column orthonormal matrices in $\mathbb{R}^{d \times r}$. Grassmann manifold can be identified as a quotient manifold of Stiefel manifold, written as $\mathrm{St}(r, d)/\mathcal{O}(r)$.

To satisfy the assumptions, we consider the polar-based retraction, which is commonly used for Riemannian optimization (P. Zhou, Yuan, Yan, & Feng, 2019; Boumal et al., 2014; Absil et al., 2009). That is, $R_{\mathbf{X}}(\mathbf{V}) = \mathrm{pf}(\mathbf{X} + \mathbf{V})$, where pf extracts the polar factor from polar decomposition. The inverse retraction is $R_{\mathbf{X}}^{-1}(\mathbf{Y}) = \mathbf{Y}(\mathbf{X}^\top \mathbf{Y})^{-1} - \mathbf{X}$. The associated vector transport is the orthogonal projection to the horizontal space, i.e. $\mathcal{T}_{\mathbf{X}}^{\mathbf{Y}}(\mathbf{V}) = (\mathbf{I} - \mathbf{Y}\mathbf{Y}^\top)\mathbf{V}$. However, such vector transport is not isometric. Following Remark 4.1, one can construct isometric vector transport that involves singular value decomposition (W. Huang, 2013), which can be expensive. Nevertheless, from the experiments, the use of non-isometric vector transport appears to work well.

The PCA problem

The PCA problem considers minimizing reconstruction error between projected and original samples over the set of orthonormal projection matrices $\mathbf{U} \in \mathrm{St}(r, d)$, which is $\min_{\mathbf{U} \in \mathrm{St}(r,d)} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_i\|^2$, where $\mathbf{x}_i \in \mathbb{R}^d, i = 1, ..., n$ represent data samples. Note the objective function is invariant under the action of orthogonal group. That is, $f(\mathbf{U}) = f(\mathbf{U}\mathbf{R})$ for $\mathbf{R} \in O(r)$. Thus, the optimization search space is Grassmann manifold and the problem is equivalent to $\min_{\mathbf{U} \in \mathcal{G}(r,d)} -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}_i$.

We first consider a synthetic dataset with $(n, d, r) = (10^5, 200, 5)$, which is generated by a random normal matrix in $\mathbb{R}^{n \times d}$ with r significant columns. Then we also experiment on two practical datasets, MNIST dataset (LeCun et al., 1998) with $(n, d, r) = (60000, 784, 5)$ and ijcnn1 dataset from LibSVM (Chang & Lin, 2011) with $(n, d, r) = (49990, 22, 5)$. We set $q = -3, l = 5$ for synthetic and MNIST datasets and $q = -1, l = 2$ for ijcnn. Fig. 4.1 presents convergence results for the PCA problem in terms of both optimality gap and gradient norm. Optimality gap is computed as the function value difference between iterates to the optimal point, returned by the PCA function in Matlab. From the figures, it is clear that variance reduction with batch size adaptation outperforms their full batch size versions, especially on large datasets like synthetic and MNIST. Due to small batch size in the initial epochs, R-AbaSVRG and R-AbaSRG behave similarly to R-SGD with rapid function value decrease, while still maintaining fast convergence around optimal point due to variance reduction. similar observations can be made in terms of gradient norm decrease. Fig. 4.2 shows additional results on synthetic dataset. Specifically, Fig. 4.2a illustrates how optimality gap decreases with algorithm runtime, which aligns closely with Fig. 4.1a. This suggests the additional cost of tracing gradient norm within each epoch is negligible. Also from Fig. 4.2b, 4.2c, we find that R-AbaSVRG and R-AbaSRG are insensitive to the parameter c_β as long as it is sufficiently large.

The LRMC problem

Given a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ with missing entries, the LRMC problem aims to recover the full matrix by assuming a low rank structure. Denote Ω as an index set corresponding to observed entries and $\mathcal{P}_\Omega$ as an operator that projects the known entries. Formally, $\Omega := \{(i,j) \,|\, A_{ij} \text{ is observed}\}$. $\mathcal{P}_\Omega(A_{ij}) = A_{ij}$ if $(i,j) \in \Omega$ and $\mathcal{P}_\Omega(A_{ij}) = 0$ otherwise. Then the problem is to $\min_{\mathbf{U},\mathbf{V}} \|\mathcal{P}_\Omega(\mathbf{A}) - \mathcal{P}_\Omega(\mathbf{UV})\|^2$, with $\mathbf{U} \in \mathbb{R}^{d \times r}, \mathbf{V} \in \mathbb{R}^{r \times n}$. Since the factorization into $\mathbf{U}, \mathbf{V}$ is not unique and depends only on the column space of $\mathbf{U}$, the problem is defined on Grassmann manifold $\mathcal{G}(r,d)$. Denote $\mathbf{a}_1, ..., \mathbf{a}_n$ as column vectors of $\mathbf{A}$ and $\mathcal{P}_{\Omega_i}, i = 1, ..., n$ as the corresponding projection for the i-th column. We can reformulate LRMC into $\min_{\mathbf{U} \in \mathcal{G}(r,d), \mathbf{v}_i \in \mathbb{R}^r} \frac{1}{n} \sum_{i=1}^n \|\mathcal{P}_{\Omega_i}(\mathbf{a}_i) - \mathcal{P}_{\Omega_i}(\mathbf{U}\mathbf{v}_i)\|^2$. Note given $\mathbf{U}$, $\mathbf{v}_i$ has a closed form solution given by the least square.

A baseline dataset with $n = 20000, d = 100, r = 5$ is generated similarly as in Kasai et al. (2018a). We set condition number of the generated matrix as $\text{cn} = 50$, Oversampling ratio is set as $\text{os} = 8$, which determines the number of known entries given by os $\times (n + d - r)r$. The known entries are subsequently perturbed by injecting Gaussian noise with a noise level $\epsilon = 10^{-10}$. In general, the larger the condition number, the smaller the oversampling ratio, the higher the noise level, the more difficult the LRMC problem is. In addition, we consider two movie recommendation datasets as follows. Netflix prize (Bennett & Lanning, 2007) contains over 100 million movie ratings, which are integers from 1 to 5. We first choose a random subset of 10 million instances and subsequently include movies and users with more than 100 observed entries. This leaves 1372 movies (n) rated by 13088 users (d). Movielens-1M (Harper & Konstan, 2015) is a dataset with 6040 users (d) and 3706 movies (n). For these two datasets, we randomly extract 20 ratings per user as test sets, which results in 15% and 12% of total observed entries for testing. We set $q = -2, -5, -5$, $l = 2, 8, 8$ for synthetic, Netflix and Movielens datasets respectively.

Fig. 4.3 presents test mean square error (MSE) on three datasets. We include training MSE results in the Appendix, which display similar patterns. From Fig. 4.3, we see that batch size adaptation accelerates variance reduction methods particularly for the first few epochs and thus perform no worse than their vanilla versions.

4.8.2 RKM on SPD manifold

We also consider the task of computing Riemannian Karcher mean (RKM) on $d \times d$ symmetric positive definite (SPD) manifold $\mathcal{S}^d_{++}$. Given n sample points $\mathbf{X}_1, ..., \mathbf{X}_n \in \mathcal{S}^d_{++}$, Riemannian Karcher mean with respect to the affine-invariant Riemannian metric (AIRM) (Pennec et al., 2006), involves solving the problem $\min_{\mathbf{C} \in \mathcal{S}^d_{++}} \frac{1}{n} \sum_{i=1}^n \| \log(\mathbf{C}^{-1/2} \mathbf{X}_i \mathbf{C}^{-1/2}) \|^2_F$, where $\log(\cdot)$ represents the principal matrix logarithm.

We follow Kasai et al. (2018b); Jeuris et al. (2012) to use an efficient retraction, $R_{\mathbf{X}}(\mathbf{V}) = \mathbf{X} + \mathbf{V} + \frac{1}{2}\mathbf{V}\mathbf{X}^{-1}\mathbf{V}$ along with the isometric vector transport in Remark

4.1 that satisfies the assumptions.

We first test on a synthetic dataset with $(n, d, \text{cn}) = (5000, 10, 20)$ generated as in (Bini & Iannazzo, 2013). In addition, we compare algorithms on Extended Yale B dataset (Wright et al., 2008) that collects 2414 (n) frontal face images of 38 individuals under various lighting conditions and Kylberg dataset (Kylberg, 2011) that contains 4480 (n) images of 28 different texture classes. Original images are resized to 32×32 pixels and region covariance descriptors are constructed for each image. Particularly, we generate 8-dimensional feature vectors consisting of pixel locations, intensity, first- and second-order pixel gradients and edge orientation at each pixel location (Pang et al., 2008). As a result, we obtain n 8×8 SPD matrices for which we calculate Riemannian Karcher mean. For all datasets, we set $q = -2$, $l = 5$. The optimal solution is obtained by the relaxed Richardson iteration (Bini & Iannazzo, 2013). From Fig. 4.4, we observe that R-AbaSVRG and R-AbaSRG still perform better compared to R-SVRG and R-SRG. The improvement is not as significant as in the PCA and LRMC problem because all methods converge rapidly and therefore batch size adaptation only takes place in the first epoch.

4.8.3 Additional experiment results

To further evaluate sensitivity of batch size adaptation, we also include results on synthetic datasets with different characteristics in Appendix 4.F for all three applications, such as large-scale, high-dimension, high-rank, ill-conditioning. We find in general, R-AbaSVRG and R-AbaSRG are insensitive when characteristics of dataset vary and perform comparatively better across all methods considered.

At last, we empirically compare R-SRG and R-SPIDER with matching complexities. We notice a similar performance for PCA and LRMC problem while R-SPIDER fails on RKM problem. One reason is that the search grid might not be extensive enough to reflect the best performance of R-SPIDER. For more difficult

LRMC problems, we find that R-SPIDER can converge faster near optimal point (Appendix 4.F). This is reasonable as gradient normalization allows magnitude of each step to be dictated precisely by the adaptive stepsize, which gives more flexibility than fixed stepsize. However, it also requires more effort in tuning two stepsize parameters α_η, β_η, which poses difficulty particularly for large datasets in high dimensions.

4.9 Discussions

In this chapter, we show that the batch size adaption can improve the complexity bounds and empirically accelerate the convergence of vanilla Riemannian variance reduction methods. Additionally, we show that the unified framework allows convergence analysis to be simplified and improved for vanilla R-SVRG and R-SRG.

In the Euclidean space, variance reduction has shown its popularity, not only for general smooth problems, but also for nonsmooth optimization (Z. Li & Li, 2018; Reddi, Sra, Poczos, & Smola, 2016; Pham et al., 2020), convex-constrained optimization (Reddi, Sra, Póczos, & Smola, 2016; Yurtsever et al., 2019), derivative-free optimization (S. Liu et al., 2018; Ji et al., 2019), compositional optimization (Lian et al., 2017; Yu & Huang, 2017), to name a few. In addition, variance reduction also improves the convergence of stochastic quasi-Newton methods (Moritz et al., 2016; X. Wang et al., 2017) and even second-order optimization algorithms (Shen et al., 2019; Z. Wang, Zhou, et al., 2019).

On the manifold space however, only a few studies consider extending Riemannian variance reduction to the broader settings. For nonsmooth optimization, Riemannian stochastic proximal gradient method with recursive variance reduction is proposed on Stiefel manifold (B. Wang et al., 2022). In Weber & Sra (2019), Riemannian stochastic Frank-Wolfe is introduced for constrained Riemannian optimization where the constrained set is geodesically convex. Other works also incorporate variance reduction for Quasi-Newton (Kasai et al., 2018a) and cubic-regularized Newton method (D. Zhang & Tajbakhsh, 2020) on Riemannian manifold.

For all the aforementioned Riemannian variance reduction methods, we believe the proposed framework is helpful for further accelerating the convergence. To validate such claim, we show in Appendix that batch size adaptation similarly improves on the complexity of Riemannian proximal stochastic recursive gradient (B. Wang et al., 2022) for nonsmooth composite optimization. The batch size is however adapted based on the norm of the generalized gradient defined by the Riemannian proximal mapping.

Furthermore, with the new analysis in this chapter, we suspect that variance reduction methods (in different settings) can be more easily generalized to Riemannian manifold with simplified analysis. Particularly, In D. Zhang & Tajbakhsh (2020) where SVRG-based variance reduction is considered, the analysis is curvature-dependent and involves the trigonometric distance bound. This can be potentially simplified under the proposed framework and derive curvature-free bounds.

Appendices

The appendix sections are structured as follows. Section 4.A presents several lemmas used widely for the subsequent proofs. Section 4.B derives the convergence and complexity of R-SVRG under classic Lyapunov analysis. Section 4.C proves convergence of R-AbaSVRG and R-SVRG under the unified framework. Section 4.D proves convergence of R-AbaSRG and R-SRG. Section 4.E analyzes existing algorithms under gradient dominance condition. Section 4.F include addition experiment results. Finally Section 4.G proposes and analyzes batch size adaptation for nonsmooth composite optimization on Riemannian manifolds.

4.A Useful lemmas

Lemma 4.3 (Variance bound for sampling without replacement)**.** *Consider a set of population vectors* $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ *in* $\mathbb{R}^D$ *with* $\sum_{i=1}^N \mathbf{x}_i = 0$*, and a subset* $\mathcal{I}$ *of cardinality* b*, which is uniformly drawn at random from* $[N]$ *without replacement. Then*

$$\mathbb{E}_{\mathcal{I}} \| \frac{1}{b} \sum_{i \in \mathcal{I}} x_i \|^2 \leq \frac{1}{Nb} \frac{N-b}{N-1} \sum_{i=1}^{N} \|x_i\|^2.$$

Proof. See Lemma A.1 in (Lei et al., 2017). □

Lemma 4.4 (Retraction Lipschitzness with vector transport)**.** *Suppose* f *is average retraction* L_l*-Lipschitz as in Assumption (4.1.6) and norm of gradient is bounded by* G*. Also suppose difference between parallel transport* Γ_y^x *and vector transport* $\mathcal{T}_y^x$ *under* *same retraction is bounded as in Assumption (4.1.7). Then for all* $x, y = \mathrm{Retr}_x(\xi) \in \mathcal{X}$*,*

$$\mathbb{E}\|\mathrm{grad} f_i(x) - \mathcal{T}_y^x \mathrm{grad} f_i(y)\| \le (L_l + \theta G)\|\xi\|,$$

where expectation is taken with respect to index i *and* θ *is parameter defined in Assumption (4.1.7).*

Proof.

$$\begin{aligned}
&\mathbb{E}\|\mathrm{grad} f_i(x) - \mathcal{T}_y^x \mathrm{grad} f_i(y)\| \\
&= \mathbb{E}\|\mathrm{grad} f_i(x) - \Gamma_y^x \mathrm{grad} f_i(y) + \Gamma_y^x \mathrm{grad} f_i(y) - \mathcal{T}_y^x \mathrm{grad} f_i(y)\| \\
&\le \mathbb{E}\|\mathrm{grad} f_i(x) - \Gamma_y^x \mathrm{grad} f_i(y)\| + \mathbb{E}\|\Gamma_y^x \mathrm{grad} f_i(y) - \mathcal{T}_y^x \mathrm{grad} f_i(y)\| \\
&\le L_l\|\xi\| + \theta\|\xi\|\mathbb{E}\|\mathrm{grad} f_i(y)\| \\
&\le (L_l + \theta G)\|\xi\|,
\end{aligned}$$

where the first inequality is by triangle inequality and the last two inequalities follow from Assumptions (1.6) and (1.7) and the bounded gradient. □

4.B Proof of Theorem 4.1

The proofs in this section are inspired by Reddi, Hefny, et al. (2016); H. Zhang et al. (2016). We first present the trigonometric distance bound (H. Zhang & Sra, 2016) that extends law of cosines on Euclidean space to Riemannian manifold with bounded sectional curvature. Next we show that the norm of gradient is bounded by difference in a properly constructed Lyapunov function. Then telescoping this result completes the proof.

Lemma 4.5 (Trigonometric distance bound)**.** *If* a, b, c *are side lengths of a geodesic triangle in a length space with curvature lower bounded by* κ*, and* θ *is the angle between* *sides b and c,*

$$a^2 \leq \frac{\sqrt{|\kappa|}c}{\tanh(\sqrt{|\kappa|}c)} b^2 + c^2 - 2bc\cos(\theta).$$

Assume Assumption 4.3 holds and define the following curvature constant

$$\zeta := \begin{cases} \frac{\sqrt{|\kappa|}D}{\tanh(\sqrt{|\kappa|}D)}, & \text{if } \kappa < 0 \\ 1, & \text{if } \kappa \geq 0 \end{cases}$$

where D is the diameter of compact set $\mathcal{X}$. Then for a, b, c as side lengths of a geodesic triangle in $\mathcal{X}$,

$$a^2 \leq \zeta b^2 + c^2 - 2bc\cos(\theta).$$

Proof. See Lemma 5 in H. Zhang & Sra (2016). □

Lemma 4.6. *Suppose Assumptions 4.1, 4.2 and 4.3 hold. Let*

$$c_t = c_{t+1} + c_{t+1}\eta\lambda + c_{t+1}\frac{(\zeta\nu^2 + 2c_R D)(L_l + \theta G)^2\mu^2\eta^2}{b} + \frac{L(L_l + \theta G)^2\mu^2\eta^2}{2b},$$
$$\delta_t = \eta - \frac{c_{t+1}\eta}{\lambda} - \frac{L\eta^2}{2} - c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2.$$

Suppose we choose $\{c_t\}, \eta$ and $\lambda > 0$ such that $\delta_t > 0$. Then iterate sequence $\{x_t^s\}$ produced by Algorithm 1 with full batch gradient $B^s = n$ satisfies

$$\|\text{grad}f(x_t^s)\|^2 \leq \frac{\mathbb{E}[R_t^s - R_{t+1}^s | \mathcal{F}_t^s]}{\delta_t},$$

with $R_t^s := f(x_t^s) + c_t d^2(x_t^s, x_0^s)$, for $s = 1, ..., S, t = 0, ..., m-1$.

Proof. By retraction L-smoothness and taking expectation with respect to $\mathcal{F}_t^s$, we have

$$\begin{aligned}\mathbb{E}[f(x_{t+1}^s)|\mathcal{F}_t^s] &\leq f(x_t^s) - \eta\langle \text{grad}f(x_t^s), \mathbb{E}[v_t^s|\mathcal{F}_t^s]\rangle + \frac{L\eta^2}{2}\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] \\ &= f(x_t^s) - \eta\|\text{grad}f(x_t^s)\|^2 + \frac{L\eta^2}{2}\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s]\end{aligned}$$

We first establish a bound on norm of modified gradient v_t^s.

$$\begin{aligned}
&\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s}\big(\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - v_0^s\big)\|^2|\mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s}\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s}\mathrm{grad} f(x_0^s) + \mathrm{grad} f(x_t^s)\|^2|\mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s}\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s}\mathrm{grad} f(x_0^s)\|^2|\mathcal{F}_t^s] + \|\mathrm{grad} f(x_t^s)\|^2 \\
&\leq \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s}\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s)\|^2|\mathcal{F}_t^s] + \|\mathrm{grad} f(x_t^s)\|^2 \\
&\leq \frac{(L_l + \theta G)^2 \mu^2}{b} d^2(x_t^s, x_0^s) + \|\mathrm{grad} f(x_t^s)\|^2,
\end{aligned}$$

where third equality is due to unbiasedness of stochastic gradient. The first inequality holds due to $\mathbb{E}\|x - \mathbb{E}[x]\|^2 \leq \mathbb{E}\|x\|^2$ and the last inequality is by Lemma 4.4 ans Assumption (4.2.2). Then we use Lemma 4.5 to bound distance $d^2(x_{t+1}^s, x_0^s)$. For a geodesic triangle $\triangle x_{t+1}^s x_t^s x_0^s$, we have

$$\begin{aligned}
\mathbb{E}[d^2(x_{t+1}^s, x_0^s)|\mathcal{F}_t^s] &\leq \mathbb{E}[\zeta d^2(x_{t+1}^s, x_t^s) + d^2(x_t^s, x_0^s) - 2\langle \mathrm{Exp}_{x_t^s}^{-1}(x_{t+1}^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle|\mathcal{F}_t^s] \\
&\leq \mathbb{E}[\zeta \nu^2 \eta^2 \|v_t^s\|^2 + d^2(x_t^s, x_0^s) - 2\langle \mathrm{Exp}_{x_t^s}^{-1}(x_{t+1}^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle|\mathcal{F}_t^s],
\end{aligned} \tag{4.5}$$

where the second inequality is by Assumption (4.2.2). Also note that

$$\begin{aligned}
&-2\langle \mathrm{Exp}_{x_t^s}^{-1}(x_{t+1}^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle \\
&= 2\langle \mathrm{Retr}_{x_t^s}^{-1}(x_{t+1}^s) - \mathrm{Exp}_{x_t^s}^{-1}(x_{t+1}^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle - 2\langle \mathrm{Retr}_{x_t^s}^{-1}(x_{t+1}^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle \\
&\leq 2\|\mathrm{Retr}_{x_t^s}^{-1}(x_{t+1}^s) - \mathrm{Exp}_{x_t^s}^{-1}(x_{t+1}^s)\|\|\mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\| + 2\eta\langle v_t^s, \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle \\
&\leq 2c_R D\eta^2\|v_t^s\|^2 + 2\eta\langle v_t^s, \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle,
\end{aligned}$$

where the last inequality uses Assumption (4.3.1) and (4.3.2) with $\|\mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\| =$

$d(x_t^s, x_0^s) \leq D$. Substitute this result back to (4.5) gives

$$\begin{aligned}
&\mathbb{E}[d^2(x_{t+1}^s, x_0^s)|\mathcal{F}_t^s] \\
&\leq \mathbb{E}[(\zeta\nu^2 + 2c_R D)\eta^2\|v_t^s\|^2 + d^2(x_t^s, x_0^s) + 2\eta\langle v_t^s, \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle|\mathcal{F}_t^s] \\
&= (\zeta\nu^2 + 2c_R D)\eta^2\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] + d^2(x_t^s, x_0^s) + 2\eta\langle \mathrm{grad}f(x_t^s), \mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\rangle \\
&\leq (\zeta\nu^2 + 2c_R D)\eta^2\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] + d^2(x_t^s, x_0^s) + 2\eta(\frac{1}{2\lambda}\|\mathrm{grad}f(x_t^s)\|^2 + \frac{\lambda}{2}\|\mathrm{Exp}_{x_t^s}^{-1}(x_0^s)\|^2) \\
&= (\zeta\nu^2 + 2c_R D)\eta^2\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] + (1+\eta\lambda)d^2(x_t^s, x_0^s) + \frac{\eta}{\lambda}\|\mathrm{grad}f(x_t^s)\|^2.
\end{aligned}$$

The second inequality is due to Young's inequality $\langle a, b\rangle \leq \frac{1}{2\lambda}\|b\|^2 + \frac{\lambda}{2}\|a\|^2$ with parameter $\lambda > 0$. Now construct a Lyapunov function $R_t^s := f(x_t^s) + c_t d^2(x_t^s, x_0^s)$. Then,

$$\begin{aligned}
&\mathbb{E}[R_{t+1}^s|\mathcal{F}_t^s] \qquad (4.6)\\
&= \mathbb{E}[f(x_{t+1}^s) + c_{t+1}d^2(x_{t+1}^s, x_0^s)|\mathcal{F}_t^s] \\
&\leq f(x_t^s) - \eta\|\mathrm{grad}f(x_t^s)\|^2 + \frac{L\eta^2}{2}\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] \\
&+ c_{t+1}\big((\zeta\nu^2 + 2c_R D)\eta^2\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] + (1+\eta\lambda)d^2(x_t^s, x_0^s) + \frac{\eta}{\lambda}\|\mathrm{grad}f(x_t^s)\|^2\big) \\
&= f(x_t^s) - (\eta - \frac{c_{t+1}\eta}{\lambda})\|\mathrm{grad}f(x_t^s)\|^2 + (c_{t+1} + c_{t+1}\eta\lambda)d^2(x_t^s, x_0^s) \\
&+ \big(\frac{L\eta^2}{2} + c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2\big)\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_t^s] \\
&\leq f(x_t^s) - (\eta - \frac{c_{t+1}\eta}{\lambda})\|\mathrm{grad}f(x_t^s)\|^2 + (c_{t+1} + c_{t+1}\eta\lambda)d^2(x_t^s, x_0^s) \\
&+ \big(\frac{L\eta^2}{2} + c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2\big)\big(\frac{(L_l + \theta G)^2\mu^2}{b}d^2(x_t^s, x_0^s) + \|\mathrm{grad}f(x_t^s)\|^2\big) \\
&= f(x_t^s) - \big(\eta - \frac{c_{t+1}\eta}{\lambda} - \frac{L\eta^2}{2} - c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2\big)\|\mathrm{grad}f(x_t^s)\|^2 \\
&+ \big(c_{t+1} + c_{t+1}\eta\lambda + c_{t+1}\frac{(\zeta\nu^2 + 2c_R D)(L_l + \theta G)^2\mu^2\eta^2}{b} + \frac{L(L_l + \theta G)^2\mu^2\eta^2}{2b}\big)d^2(x_t^s, x_0^s) \\
&= R_t^s - \delta_t\|\mathrm{grad}f(x_t^s)\|^2,
\end{aligned}$$

with $c_t = c_{t+1} + c_{t+1}\eta\lambda + c_{t+1}\frac{(\zeta\nu^2+2c_R D)(L_l+\theta G)^2\mu^2\eta^2}{b} + \frac{L(L_l+\theta G)^2\mu^2\eta^2}{2b}$ and $\delta_t := \eta - \frac{c_{t+1}\eta}{\lambda} - \frac{L\eta^2}{2} - c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2$. Suppose we choose parameters such that $\delta_t >$

0. Then, we have the desired result. □

Lemma 4.7. *With the same assumptions and settings in Lemma 4.6, choose $c_m = 0$ and define $\tilde{\delta} := \min_{0 \leq t \leq m-1} \delta_t$. Denote $T = Sm$ as the total number of iterations and $\Delta = f(\tilde{x}^0) - f(x^*)$. Then output $\tilde{x}$ from Algorithm 1 with full batch gradient $B^s = n$ satisfies*

$$\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \frac{\Delta}{T\tilde{\delta}}.$$

Proof. Summing over result over $t = 0, ..., m-1$ from Lemma 4.6 and taking expectation with respect to $\mathcal{F}_0^s$ yields

$$\sum_{t=0}^{m-1} \mathbb{E}[\|\text{grad}f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{\mathbb{E}[R_0^s - R_m^s | \mathcal{F}_0^s]}{\tilde{\delta}} = \frac{\mathbb{E}[f(x_0^s) - f(x_0^{s+1}) | \mathcal{F}_0^s]}{\tilde{\delta}},$$

where we note that $R_0^s = f(x_0^s)$ and $R_m^s = f(x_m^s) = f(x_0^{s+1})$ for $c_m = 0$. Telescoping this inequality from $s = 1, ..., S$ and taking full expectation, we have

$$\frac{1}{T} \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\text{grad}f(x_t^s)\|^2 \leq \frac{f(\tilde{x}^0) - \mathbb{E}[f(x_m^S)]}{T\tilde{\delta}} \leq \frac{\Delta}{T\tilde{\delta}}.$$

Finally, by noting that output $\tilde{x}$ is uniformly drawn at random from all iterates and thus $\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 = \frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|\text{grad}f(x_t^s)\|^2$, the proof is complete. □

Now we are ready to prove Theorem 4.1.

Theorem 4.1. Suppose Assumptions 4.1, 4.2 and 4.3 hold and consider Algorithm 1 with full batch gradient $B^s = n$. Choose stepsize $\eta = \frac{\mu_0 b}{(L_l + \theta G)\mu n^{a_1} (\zeta \nu^2 + 2c_R D)^{a_2}}$, $m = \lfloor n^{3/2a_1} / 2b\mu_0 (\zeta\nu^2 + 2c_R D)^{1-2a_2} \rfloor$, $b \leq n^{a_1}$, where $a_1, a_2, \mu_0 \in (0, 1)$. Then for a constant $\psi > 0$ such that

$$\psi \leq \frac{\mu_0}{\mu}\Big(1 - \frac{L\mu_0(e-1)}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{2-a_2}\mu} - \frac{L\mu_0 b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{a_1}} \\ - \frac{L\mu_0^2 (e-1) b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{3/2a_1}}\Big),$$

the output $\tilde{x}$ after running $T = Sm$ iterations satisfies

$$\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \frac{(L_l + \theta G)n^{a_1}(\zeta\nu^2 + 2c_R D)^{a_2}\Delta}{bT\psi},$$

where $\Delta := f(\tilde{x}^0) - f(x^*)$. By choosing $a_1 = 2/3, a_2 = 1/2$, the total IFO complexity to achieve ϵ-accurate solution is $O\Big(n + \frac{(L_l+\theta G)n^{2/3}(\zeta\nu^2+2c_R D)^{1/2}}{\epsilon^2}\Big)$.

Proof. First note that $c_t = c_{t+1}(1 + \eta\lambda + \frac{(\zeta\nu^2+2c_R D)(L_l+\theta G)^2\mu^2\eta^2}{b}) + \frac{L(L_l+\theta G)^2\mu^2\eta^2}{2b} = c_{t+1}(1+\phi) + \frac{L(L_l+\theta G)^2\mu^2\eta^2}{2b}$, where $\phi := \eta\lambda + \frac{(\zeta\nu^2+2c_R D)(L_l+\theta G)^2\mu^2\eta^2}{b}$. Choose $\eta = \frac{\mu_0 b}{(L_l+\theta G)\mu n^{a_1}(\zeta\nu^2+2c_R D)^{a_2}}, \mu_0 \in (0,1)$ and $\lambda = \frac{(L_l+\theta G)\mu(\zeta\nu^2+2c_R D)^{1-a_2}}{n^{a_1/2}}$ gives

$$c_t = (1+\phi)c_{t+1} + \frac{L\mu_0^2 b}{2n^{2a_1}(\zeta\nu^2 + 2c_R D)^{2a_2}}, \tag{4.7}$$

Applying (4.7) recursively to c_0 and noting $c_m = 0$, we have

$$c_0 = \frac{L\mu_0^2 b}{2n^{2a_1}(\zeta\nu^2 + 2c_R D)^{2a_2}} \frac{(1+\phi)^m - 1}{\phi}. \tag{4.8}$$

It is noted that the sequence $\{c_t\}_{t=0}^{m-1}$ is a decreasing sequence and achieves its maximum at c_0. Therefore we derive a bound on c_0. Note that

$$\begin{aligned}\phi &= \frac{\mu_0 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}}{n^{3/2a_1}} + \frac{\mu_0^2 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}}{n^{2a_1}} \\ &\in \Big(\frac{\mu_0 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}}{n^{3/2a_1}}, \frac{2\mu_0 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}}{n^{3/2a_1}}\Big).\end{aligned} \tag{4.9}$$

Choosing $m = \lfloor n^{3/2a_1}/2b\mu_0(\zeta\nu^2 + 2c_R D)^{1-2a_2}\rfloor$ suggests

$$\phi \leq \frac{2\mu_0 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}}{n^{3/2a_1}} \leq \frac{1}{m}, \text{ and } (1+\phi)^m \leq e, \tag{4.10}$$

where e is the Euler's constant. Note for the second inequality, we loosely use $\leq$

instead of $<$ for consistency. Then applying (4.9) and (4.10) into (4.8), we have

$$c_0 \leq \frac{L\mu_0^2 b}{2n^{2a_1}(\zeta\nu^2 + 2c_R D)^{2a_2}} \times \frac{n^{3/2a_1}(e-1)}{\mu_0 b(\zeta\nu^2 + 2c_R D)^{1-2a_2}} = \frac{L\mu_0(e-1)}{2n^{1/2a_1}(\zeta\nu^2 + 2c_R D)}.$$

Next we consider a lower bound on $\tilde{\delta}$.

$$\begin{aligned}
\tilde{\delta} &= \min_t \Big(\eta - \frac{c_{t+1}\eta}{\lambda} - \frac{L\eta^2}{2} - c_{t+1}(\zeta\nu^2 + 2c_R D)\eta^2\Big) \\
&\geq \Big(\eta - \frac{c_0\eta}{\lambda} - \frac{L\eta^2}{2} - c_0(\zeta\nu^2 + 2c_R D)\eta^2\Big) \\
&\geq \eta\Big(1 - \frac{L\mu_0(e-1)}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{2-a_2}\mu} - \frac{L\mu_0 b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{a_1}} \\
&\quad - \frac{L\mu_0^2(e-1)b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{3/2a_1}}\Big) \\
&\geq \frac{b\psi}{(L_l + \theta G)n^{a_1}(\zeta\nu^2 + 2c_R D)^{a_2}},
\end{aligned}$$

where $\psi > 0$ is a constant such that the last inequality holds. That is, we choose ψ satisfying

$$\begin{aligned}
0 < \psi \leq \frac{\mu_0}{\mu}\Big(1 &- \frac{L\mu_0(e-1)}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{2-a_2}\mu} - \frac{L\mu_0 b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{a_1}} \\
&- \frac{L\mu_0^2(e-1)b}{2(L_l + \theta G)(\zeta\nu^2 + 2c_R D)^{a_2}\mu n^{3/2a_1}}\Big).
\end{aligned}$$

This condition holds by setting a sufficiently small $\mu_0 \in (0,1)$ and also $b \leq n^{a_1}$. The requirement on b is to ensure the third term and fourth term do not increase with n. Therefore, combining this result with Lemma 4.7 yields

$$\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \frac{(L_l + \theta G)n^{a_1}(\zeta\nu^2 + 2c_R D)^{a_2}\Delta}{bT\psi}.$$

To achieve ϵ-accurate solution, it is sufficient to require $\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \epsilon^2$. That is, $\mathbb{E}\|\text{grad}f(\tilde{x})\| \leq \sqrt{\mathbb{E}\|\text{grad}f(\tilde{x})\|^2} \leq \epsilon$ by Jensen's inequality. Therefore, we require at least

$$\begin{aligned} S = \frac{(L_l + \theta G) n^{\alpha_1} (\zeta \nu^2 + 2 c_R D)^{\alpha_2}}{b m \psi \epsilon^2} &= \lceil \frac{2 \mu_0 (L_l + \theta G)(\zeta \nu^2 + 2 c_R D)^{1-\alpha_2} n^{-\alpha_1/2}}{\psi \epsilon^2} \rceil \\ &= O\Big(1 + \frac{(\zeta \nu^2 + 2 c_R D)^{1-\alpha_2} n^{-\alpha_1/2}}{\epsilon^2} \Big) \end{aligned}$$

number of epochs. Each epoch requires $n + 2mb$ IFO calls, which is $n + \lfloor n^{3/2\alpha_1} / \mu_0 (\zeta \nu^2 + 2 c_R D)^{1 - 2\alpha_2} \rfloor = O\big(n + n^{3/2\alpha_1} (\zeta \nu^2 + 2 c_R D)^{2\alpha_2 - 1}\big)$. Hence the total complexity is given by

$$\begin{aligned} & O\Big(\big(1 + \frac{(\zeta \nu^2 + 2 c_R D)^{1-\alpha_2} n^{-\alpha_1/2}}{\epsilon^2} \big) \big(n + n^{3/2\alpha_1} (\zeta \nu^2 + 2 c_R D)^{2\alpha_2 - 1} \big) \Big) \\ &= O\Big(n + \frac{n^{\alpha_1} (\zeta \nu^2 + 2 c_R D)^{\alpha_2}}{\epsilon^2} + \frac{(\zeta \nu^2 + 2 c_R D)^{1-\alpha_2} n^{1-\alpha_1/2}}{\epsilon^2} + n^{3/2\alpha_1} (\zeta \nu^2 + 2 c_R D)^{2\alpha_2 - 1} \big) \Big). \end{aligned}$$

With the standard choice of $\alpha_1 = \frac{2}{3}$ and $\alpha_2 = \frac{1}{2}$, we obtain the desired result. □

4.C Convergence Analysis for R-AbaSVRG

Proof of Lemma 4.1. First note that $\mathcal{F}_0^s \subseteq \mathcal{F}_t^s$, for $0 \leq t \leq m-1$ and therefore it holds that $\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] = \mathbb{E}[\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_t^s] | \mathcal{F}_0^s]$. Hence we first consider bounding $\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_t^s]$ as

$$\begin{aligned} & \mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_t^s] \\ &= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \big(\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - v_0^s \big) - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_t^s] \\ &= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f(x_0^s) \\ &\quad + \mathcal{T}_{x_0^s}^{x_t^s} v_0^s - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_t^s] \\ &= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_t^s] \\ &\quad + \mathbb{E}[\langle \mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f(x_0^s), v_0^s - \mathrm{grad} f(x_0^s) \rangle | \mathcal{F}_t^s] \\ &\quad + \|v_0^s - \mathrm{grad} f(x_0^s)\|^2 \\ &= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) - \mathrm{grad} f(x_t^s) + \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_t^s] \end{aligned}$$

$$
\begin{aligned}
&+ \|v_0^s - \mathrm{grad} f(x_0^s)\|^2 \\
&\leq \mathbb{E}[\|\mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s)\|^2 | \mathcal{F}_t^s] + \|v_0^s - \mathrm{grad} f(x_0^s)\|^2 \\
&\leq \frac{1}{b}\mathbb{E}[\|\mathrm{grad} f_i(x_t^s) - \mathcal{T}_{x_0^s}^{x_t^s} \mathrm{grad} f_i(x_0^s)\|^2 | \mathcal{F}_t^s] + \|v_0^s - \mathrm{grad} f(x_0^s)\|^2 \\
&\leq \frac{1}{b}(L_l + \theta G)^2 \|\mathrm{Retr}_{x_0^s}^{-1}(x_t^s)\|^2 + \|v_0^s - \mathrm{grad} f(x_0^s)\|^2 \\
&\leq \frac{1}{b}(L_l + \theta G)^2 \mu^2 d^2(x_t^s, x_0^s) + \|v_0^s - \mathrm{grad} f(x_0^s)\|^2. \qquad (4.11)
\end{aligned}
$$

The fourth equality is based on the facts that $\mathrm{grad} f_{\mathcal{I}_t^s}(x)$ is unbiased estimator of $\mathrm{grad} f(x)$ and also the isometric property of vector transport $\mathcal{T}_{x_0^s}^{x_t^s}$. Note that $\mathcal{T}_{x_0^s}^{x_t^s}$ depends on both x_0^s and x_t^s, which are measurable in $\mathcal{F}_t^s$. Therefore $\mathbb{E}[\mathcal{T}_{x_0^s}^{x_{t-1}^s} \mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) | \mathcal{F}_t^s] = \mathcal{T}_{x_0^s}^{x_{t-1}^s} \mathbb{E}[\mathrm{grad} f_{\mathcal{I}_t^s}(x_0^s) | \mathcal{F}_t^s] = \mathcal{T}_{x_0^s}^{x_{t-1}^s} \mathrm{grad} f(x_0^s)$. The first inequality is due to $\mathbb{E}\|x - \mathbb{E}[x]\|^2 \leq \mathbb{E}\|x\|^2$ and the second inequality is due to independence of with replacement sampling. The last two inequalities are from Assumption (4.2.2) and Lemma 4.4. Taking expectation with respect to $\mathcal{F}_0^s$ gives $\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{1}{b}(L_l + \theta G)^2 \mu^2 \mathbb{E}[d^2(x_t^s, x_0^s) | \mathcal{F}_0^s] + \mathbb{E}[\|v_0^s - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s]$. Next, we further simplify (4.11) by telescoping iterates within epoch s. Note that by triangle inequality and Assumption (4.2.2),

$$
\begin{aligned}
d^2(x_t^s, x_0^s) &\leq \big(d(x_t^s, x_{t-1}^s) + d(x_{t-1}^s, x_{t-2}^s) + \cdots + d(x_1^s, x_0^s)\big)^2 \\
&\leq \nu^2 \eta^2 \big(\|v_{t-1}^s\| + \cdots + \|v_0^s\|\big)^2 \leq \nu^2 \eta^2 t \sum_{i=0}^{t-1} \|v_i^s\|^2, \qquad (4.12)
\end{aligned}
$$

where the last inequality follows from $\| \sum_{i=1}^d w_i \|^2 \leq d \sum_{i=1}^d \|w_i\|^2$. On the other hand, by Lemma 4.3 and variance bound assumption (4.1.4), we have

$$
\begin{aligned}
\mathbb{E}[\|v_0^s - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s] &= \mathbb{E}[\|\mathrm{grad} f_{\mathcal{B}^s}(x_0^s) - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s] \\
&= \mathbb{E}[\| \frac{1}{B^s} \sum_{i \in \mathcal{B}^s} \mathrm{grad} f_i(x_0^s) - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{n - B^s}{n - 1} \frac{1}{n B^s} \sum_{i=1}^n \|\mathrm{grad} f_i(x_0^s) - \mathrm{grad} f(x_0^s)\|^2
\end{aligned}
$$

$$\leq \frac{n - B^s}{n-1} \frac{\sigma^2}{B^s} \leq \mathbb{1}_{\{B^s < n\}} \frac{\sigma^2}{B^s}. \tag{4.13}$$

Note if $\mathcal{B}^s$ is chosen from $[n]$ with replacement or under online setting where n approaches infinity, we simply have $\mathbb{E}[\|v_0^s - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s] = \frac{1}{B^s} \mathbb{E}[\|\mathrm{grad} f_i(x_0^s) - \mathrm{grad} f(x_0^s)\|^2 | \mathcal{F}_0^s] \leq \frac{\sigma^2}{B^s}$, which does not vanish when $B^s = n$. Substituting (4.12) and (4.13) back to (4.11) gives the desired result. □

Proof of Theorem 4.2. By retraction L-smoothness in Assumption (4.1.5),

$$\begin{aligned}
f(x_{t+1}^s) - f(x_t^s) &\leq -\eta \langle \mathrm{grad} f(x_t^s), v_t^s \rangle + \frac{L\eta^2}{2} \|v_t^s\|^2 \\
&= -\frac{\eta}{2} \|\mathrm{grad} f(x_t^s)\|^2 - \frac{\eta}{2} \|v_t^s\|^2 + \frac{\eta}{2} \|v_t^s - \mathrm{grad} f(x_t^s)\|^2 + \frac{L\eta^2}{2} \|v_t^s\|^2 \\
&= -\frac{\eta}{2} \|\mathrm{grad} f(x_t^s)\|^2 + \frac{\eta}{2} \|v_t^s - \mathrm{grad} f(x_t^s)\|^2 - (\frac{\eta}{2} - \frac{L\eta^2}{2}) \|v_t^s\|^2.
\end{aligned}$$

Rearranging the term and taking expectation with respect to $\mathcal{F}_0^s$ yields

$$\begin{aligned}
&\mathbb{E}[\|\mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{2}{\eta} \mathbb{E}[f(x_t^s) - f(x_{t+1}^s) | \mathcal{F}_0^s] + \mathbb{E}[|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] - (1 - L\eta) \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{2}{\eta} \mathbb{E}[f(x_t^s) - f(x_{t+1}^s) | \mathcal{F}_0^s] + \frac{t}{b} (L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 \sum_{i=0}^{t-1} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{\sigma^2}{B^s} \\
&- (1 - L\eta) \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_t^s].
\end{aligned}$$

Summing this result over $t = 0, ..., m-1$ gives

$$\begin{aligned}
&\sum_{t=0}^{m-1} \mathbb{E}[\|\mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{2}{\eta} \mathbb{E}[f(x_0^s) - f(x_m^s) | \mathcal{F}_0^s] + \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2}{b} \sum_{t=0}^{m-1} t \sum_{i=0}^{t} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s} \\
&- (1 - L\eta) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{2}{\eta} \mathbb{E}[f(x_0^s) - f(x_m^s) | \mathcal{F}_0^s] + \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 m^2}{b} \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s}
\end{aligned}$$

$$
\begin{aligned}
&- (1 - L\eta) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\
&= \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] - (1 - L\eta - \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 m^2}{b}) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\
&\quad + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s}. \qquad (4.14)
\end{aligned}
$$

The second inequality uses the fact $t \leq m-1$. Telescoping (4.14) from $s = 1, ..., S$ and taking expectation over all randomness gives

$$
\begin{aligned}
\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\text{grad} f(x_t^s)\|^2 &\leq -(1 - L\eta - \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 m^2}{b}) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 \\
&\quad + \frac{2}{\eta}\mathbb{E}[f(\tilde{x}^0) - f(x_m^S)] + \sum_{s=1}^{S} \mathbb{E}[\mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s}] \\
&\leq \frac{2\Delta}{\eta} - (1 - L\eta - \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 m^2}{b}) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 \\
&\quad + \sum_{s=1}^{S} \mathbb{E}[\mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s}], \qquad (4.15)
\end{aligned}
$$

where $\Delta := f(\tilde{x}^0) - f(x^*)$ and we use the fact that $\mathbb{E}[f(x_m^S)] \geq f(x^*)$. Since $\mathcal{B}^s$ depends on whether finite-sum or online setting is considered, we consider these two cases separately.

(1) Under the finite-sum setting,

$$
\mathbb{1}_{\{B^s < n\}} \frac{1}{B^s} = \frac{1}{\min\{\alpha_1 \sigma^2 / \beta_s, n\}} \leq \frac{\beta_s}{\alpha_1 \sigma^2} \leq \frac{\beta_s}{\alpha \sigma^2},
$$

where we choose $\alpha_1 \geq \alpha$. Note also from the definition of β_s and the choice of $\beta_1 \leq \epsilon^2 S$, we have

$$
\sum_{s=1}^{S} \mathbb{E}[\beta_s] = \beta_1 + \frac{1}{m} \sum_{s=1}^{S-1} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 \leq \epsilon^2 S + \frac{1}{m} \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2, \qquad (4.16)
$$

Combining these two results and substituting into (4.15) gives

$$\begin{aligned}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\mathrm{grad}f(x_t^s)\|^2 &\leq -(1-L\eta-\frac{(L_l+\theta G)^2\mu^2\nu^2\eta^2m^2}{b})\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|v_t^s\|^2+\frac{2\Delta}{\eta}\\ &\quad+\frac{m}{\alpha}[\epsilon^2S+\frac{1}{m}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|v_t^s\|^2]\\ &= -(1-L\eta-\frac{(L_l+\theta G)^2\mu^2\nu^2\eta^2m^2}{b}-\frac{1}{\alpha})\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|v_t^s\|^2\\ &\quad+\frac{2\Delta}{\eta}+\frac{\epsilon^2mS}{\alpha}.\end{aligned} \tag{4.17}$$

Let $\eta \leq \frac{2-\frac{2}{\alpha}}{L+\sqrt{L^2+4(1-\frac{1}{\alpha})\frac{(L_l+\theta G)^2\mu^2\nu^2m^2}{b}}}$, which is the larger root of the equation $1-L\eta-\frac{(L_l+\theta G)^2\mu^2\nu^2\eta^2m^2}{b}-\frac{1}{\alpha}=0$. The other root is smaller than zero. Therefore, this choice of η can ensure coefficients before $\mathbb{E}\|v_t^s\|^2$ is smaller than zero. Then dividing (4.17) by $T=Sm$ yields,

$$\mathbb{E}\|\mathrm{grad}f(\tilde{x})\|^2=\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\mathrm{grad}f(x_t^s)\|^2\leq\frac{2\Delta}{T\eta}+\frac{\epsilon^2}{\alpha},$$

where we note that output $\tilde{x}$ is uniformly drawn at random from $\{\{x_t^s\}_{t=0}^{m-1}\}_{s=1}^{S}$.

(2) Similarly, under the online setting,

$$\mathbb{1}_{\{B^s<n\}}\frac{1}{B^s}=\frac{1}{\min\{\alpha_1\sigma^2/\beta_s,\alpha_2\sigma^2/\epsilon^2\}}=\max\{\frac{\beta_s}{\alpha_1\sigma^2},\frac{\epsilon^2}{\alpha_2\sigma^2}\}\leq\frac{\beta_s+\epsilon^2}{\alpha\sigma^2}, \tag{4.18}$$

where the last inequality uses the fact that $\max\{a,b\}\leq a+b$ and $\alpha_1,\alpha_2\geq\alpha$. Following the same procedure and choice of η, we have

$$\mathbb{E}\|\mathrm{grad}f(\tilde{x})\|^2=\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\mathrm{grad}f(x_t^s)\|^2\leq\frac{2\Delta}{T\eta}+\frac{2\epsilon^2}{\alpha}.$$

Hence, by choosing $\alpha\geq 2$ for finite-sum setting and $\alpha\geq 4$ for online setting, we have

$$\mathbb{E}\|\mathrm{grad}f(\tilde{x})\|^2\leq\frac{2\Delta}{T\eta}+\frac{\epsilon^2}{2}.$$

For simplicity, we consider $\alpha \geq 4$ for both cases. □

Proof of Corollary 4.1. Consider the following parameter setting, $b = m^2$, $\alpha = 4$ and $\eta = \frac{3}{2L + 2\sqrt{L^2 + 3(L_l + \theta G)^2 \mu^2 \nu^2}}$. To obtain ϵ-accurate solution, we require at least

$$S = \frac{4\Delta}{\epsilon^2 m \eta} = \frac{8\Delta}{3\epsilon^2 m}\big(L + \sqrt{L^2 + 3(L_l + \theta G)^2 \mu^2 \nu^2}\big) = O\big(\frac{\Theta_1}{m\epsilon^2}\big)$$

where $\Theta_1 := L + \sqrt{L^2 + \varrho_1 (L_l + \theta G)^2 \mu^2 \nu^2}$, where $\varrho_1 > 0$ is a constant that does not depend on any parameter. Define average batch size $\tilde{B}$ as

$$\tilde{B} := \frac{1}{S}\sum_{s=1}^{S} B^s = \begin{cases} \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1 \sigma^2/\beta_s, n\}, & \text{(finite-sum)} \\ \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1 \sigma^2/\beta_s, \alpha_2 \sigma^2/\epsilon^2\}, & \text{(online)} \end{cases} \tag{4.19}$$

Then one epoch requires $\tilde{B} + 2mb = O(\tilde{B} + m^3)$ IFO calls. Choosing $m = \lfloor n^{1/3} \rfloor$ under finite-sum setting and $m = (\frac{\sigma}{\epsilon})^{2/3}$ under online setting, the total IFO complexity is given by

$$\begin{aligned} O\big(S(\tilde{B} + m^3)\big) = O\big(S\tilde{B} + Sm^3\big) &= O\big(\frac{\Theta_1 \tilde{B}}{m\epsilon^2} + \frac{\Theta_1 m^2}{\epsilon^2}\big) \\ &= \begin{cases} O\big(\tilde{B} + \frac{\Theta_1 \tilde{B}}{n^{1/3}\epsilon^2} + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big), & \text{(finite-sum)} \\ O\big(\frac{\Theta_1 \tilde{B}}{\sigma^{2/3}\epsilon^{4/3}} + \frac{\Theta_1 \sigma^{4/3}}{\epsilon^{10/3}}\big), & \text{(online)} \end{cases} \end{aligned}$$

which completes the proof. □

Proof of Corollary 4.2. From (4.14),

$$\begin{aligned} \sum_{t=0}^{m-1} \mathbb{E}[\|\mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] &\leq \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s) | \mathcal{F}_0^s] + \mathbb{1}_{\{B<n\}}\frac{m\sigma^2}{B} \\ &\quad - (1 - L\eta - \frac{(L_l + \theta G)^2 \mu^2 \nu^2 \eta^2 m^2}{b}) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s]. \end{aligned} \tag{4.20}$$

Choosing a choice of fixed stepsize $\eta \leq \frac{2}{L+\sqrt{L^2+4\frac{(L_l+\theta G)^2\mu^2\nu^2m^2}{b}}}$, which ensures $1 - L\eta - \frac{(L_l+\theta G)^2\mu^2\nu^2\eta^2m^2}{b} \geq 0$. Telescoping this (4.20) from $s = 1, ..., S$ and dividing by $T = Sm$ gives

$$\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\text{grad}f(x_t^s)\|^2 \leq \frac{2\Delta}{T\eta} + \mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B}.$$

Note that output $\tilde{x}$ satisfies $\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 = \frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\text{grad}f(x_t^s)\|^2$. Under the finite-sum setting where $B = n$, $\mathbb{1}_{\{B<n\}}\frac{m\sigma^2}{B} = 0$, we have $\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta}$. Under the online setting where $B = \frac{2\sigma^2}{\epsilon^2}$, $\mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B} = \frac{\epsilon^2}{2}$, $\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \frac{\epsilon^2}{2}$. Given $b = m^2$ and $\eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2\mu^2\nu^2}}$, under both finite-sum and online settings, to obtain ϵ-accurate solution, we require at least

$$S = O\Big(\frac{\Delta}{m\eta\epsilon^2}\Big) = O\Big(\frac{\Delta}{m\epsilon^2}\big(L + \sqrt{L^2 + 4(L_l+\theta G)^2\mu^2\nu^2}\big)\Big) = O\big(\frac{\Theta_1}{m\epsilon^2}\big).$$

Hence, we obtain the same iteration complexity as adaptive batch size version. Note for one epoch, we require $B + 2mb = O(B + m^3)$ IFO calls. With the same choice of $m = \lfloor n^{1/3}\rfloor$ under finite-sum setting and $m = (\frac{\sigma}{\epsilon})^{2/3}$ under online setting, total IFO complexity is given by

$$O\big(S(B+m^3)\big) = \begin{cases} O\big(n + \frac{\Theta_1 n^{2/3}}{\epsilon^2}\big), & \text{(finite-sum)} \\ O\big(\frac{\Theta_1\sigma^{4/3}}{\epsilon^{10/3}}\big), & \text{(online)} \end{cases}$$

□

4.D Convergence Analysis for R-AbaSRG

Proof of Lemma 4.2. Note that similarly, we can write $\mathbb{E}[\|v_t^s - \text{grad}f(x_t^s)\|^2|\mathcal{F}_0^s] = \mathbb{E}[\mathbb{E}[\|v_t^s - \text{grad}f(x_t^s)\|^2|\mathcal{F}_t^s]|\mathcal{F}_0^s]$ and we first bound $\mathbb{E}[\|v_t^s - \text{grad}f(x_t^s)\|^2|\mathcal{F}_t^s]$. To this end,

$$\begin{aligned}
&\mathbb{E}[\|v_t^s - \mathrm{grad}f(x_t^s)\|^2 | \mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad}f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f_{\mathcal{I}_t^s}(x_{t-1}^s) + \mathcal{T}_{x_{t-1}^s}^{x_t^s} v_{t-1}^s - \mathrm{grad}f(x_t^s)\|^2 | \mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad}f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f_{\mathcal{I}_t^s}(x_{t-1}^s) - \mathrm{grad}f(x_t^s) + \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f(x_{t-1}^s) \\
&\quad + \mathcal{T}_{x_{t-1}^s}^{x_t^s} v_{t-1}^s - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] \\
&= \mathbb{E}[\|\mathrm{grad}f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f_{\mathcal{I}_t^s}(x_{t-1}^s) - \mathrm{grad}f(x_t^s) + \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] \\
&\quad + \mathbb{E}[\|v_{t-1}^s - \mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] \\
&\leq \mathbb{E}[\|\mathrm{grad}f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f_{\mathcal{I}_t^s}(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] + \mathbb{E}[\|v_{t-1}^s - \mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] \\
&= \frac{1}{b}\mathbb{E}[\|\mathrm{grad}f_i(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\mathrm{grad}f_i(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] + \mathbb{E}[\|v_{t-1}^s - \mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_t^s] \\
&\leq \frac{1}{b}(L_l + \theta G)^2 \eta^2 \|v_{t-1}^s\|^2 + \|v_{t-1}^s - \mathrm{grad}f(x_{t-1}^s)\|^2.
\end{aligned}$$

Note the expectation is taken with respect to randomness of sample $\mathcal{I}_t^s$ where both x_{t-1}^s and x_t^s are measurable. The vector transport $\mathcal{T}_{x_{t-1}^s}^{x_t^s}$ is therefore fixed conditional on $\mathcal{F}_t^s$. Hence, the third equality holds due to unbiasedness. The first inequality is due to $\mathbb{E}\|x - \mathbb{E}[x]\|^2 \leq \mathbb{E}\|x\|^2$ and the last inequality is from Lemma 4.4. Therefore we have $\mathbb{E}[\|v_t^s - \mathrm{grad}f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{1}{b}(L_l + \theta G)^2\eta^2 \mathbb{E}[\|v_{t-1}^s\|^2 | \mathcal{F}_0^s] + \mathbb{E}[\|v_{t-1}^s - \mathrm{grad}f(x_{t-1}^s)\|^2 | \mathcal{F}_0^s]$. Recursively applying this inequality gives

$$\begin{aligned}
&\mathbb{E}[\|v_t^s - \mathrm{grad}f(x_t^s)\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{(L_l + \theta G)^2 \eta^2}{b} \sum_{i=0}^{t-1} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{E}[\|v_0^s - \mathrm{grad}f(x_0^s)\|^2 | \mathcal{F}_0^s] \\
&\leq \frac{(L_l + \theta G)^2 \eta^2}{b} \sum_{i=0}^{t} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] + \mathbb{E}[\|v_0^s - \mathrm{grad}f(x_0^s)\|^2 | \mathcal{F}_0^s], \qquad (4.21)
\end{aligned}$$

where we note that $\mathbb{E}[\|v_0^s - \mathrm{grad}f(x_0^s)\|^2 | \mathcal{F}_0^s] \leq \mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B}$ by similar argument in (4.15). Combining this inequality with (4.21) completes the proof. □

Proof of Theorem 4.3. Here we adopt a similar procedure as the proof of R-AbaSVRG.

By retraction L-smoothness, we have

$$\begin{aligned}f(x_{t+1}^s) - f(x_t^s) &\leq -\eta \langle \mathrm{grad} f(x_t^s), v_t^s \rangle + \frac{L\eta^2}{2} \|v_t^s\|^2 \\ &= -\frac{\eta}{2}\|\mathrm{grad} f(x_t^s)\|^2 - \frac{\eta}{2}\|v_t^s\|^2 + \frac{\eta}{2}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 + \frac{L\eta^2}{2}\|v_t^s\|^2 \\ &= -\frac{\eta}{2}\|\mathrm{grad} f(x_t^s)\|^2 + \frac{\eta}{2}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 - (\frac{\eta}{2} - \frac{L\eta^2}{2})\|v_t^s\|^2.\end{aligned}$$

Taking expectation of this inequality with respect to $\mathcal{F}_0^s$ and summing over $t = 0, ..., m-1$ gives

$$\begin{aligned}&\sum_{t=0}^{m-1} \mathbb{E}[\|\mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \\ &\leq \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] + \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \\ &\quad - (1 - L\eta) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\ &\leq \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] - (1 - L\eta) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s} \\ &\quad + \frac{(L_l + \theta G)^2 \eta^2}{b} \sum_{t=0}^{m-1} \sum_{i=0}^{t} \mathbb{E}[\|v_i^s\|^2 | \mathcal{F}_0^s] \\ &\leq \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] - (1 - L\eta) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s} \\ &\quad + \frac{(L_l + \theta G)^2 \eta^2 m}{b} \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\ &= \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] - \Big(1 - L\eta - \frac{(L_l + \theta G)^2 \eta^2 m}{b}\Big) \sum_{t=0}^{m-1} \mathbb{E}[\|v_t^s\|^2 | \mathcal{F}_0^s] \\ &\quad + \mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s},\end{aligned} \tag{4.22}$$

where the second first inequality is by Lemma 4.2 and the third inequality is due to the fact that $t \leq m - 1$. Summing this inequality over $s = 1, ..., S$ and taking full expectation, we have

$$\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\mathrm{grad} f(x_t^s)\|^2$$

$$\leq \frac{2\Delta}{\eta} - \Big(1 - L\eta - \frac{(L_l + \theta G)^2 \eta^2 m}{b}\Big) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 + \sum_{s=1}^{S} \mathbb{E}[\mathbb{1}_{\{B^s < n\}} \frac{m\sigma^2}{B^s}],$$

where $\Delta := f(\tilde{x}^0) - f(x^*)$. Same as in (4.16), we can show $\sum_{s=1}^S \mathbb{E}[\beta_s] \leq \epsilon^2 S + \frac{1}{m}\sum_{s=1}^S \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2$, with the choice $\beta_1 \leq \epsilon^2 S$. (1) Under finite-sum setting, $\mathbb{1}_{\{B^s<n\}} \frac{1}{B^s} \leq \frac{\beta_s}{c_\beta \sigma^2} \leq \frac{\beta_s}{\alpha\sigma^2}$ where we choose $\alpha_1 > \alpha$. This gives

$$\begin{aligned}
&\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\mathrm{grad} f(x_t^s)\|^2 \\
&\leq \frac{2\Delta}{\eta} - \Big(1 - L\eta - \frac{(L_l + \theta G)^2 \eta^2 m}{b}\Big) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 + \frac{m}{\alpha}\sum_{s=1}^{S} \mathbb{E}[\beta_s] \\
&\leq \frac{2\Delta}{\eta} - \Big(1 - \frac{1}{\alpha} - L\eta - \frac{(L_l + \theta G)^2 \eta^2 m}{b}\Big) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 + \frac{Sm\epsilon^2}{\alpha}.
\end{aligned}$$

Let $\eta \leq \frac{2 - \frac{2}{\alpha}}{L + \sqrt{L^2 + 4(1-\frac{1}{\alpha})\frac{(L_l+\theta G)^2 m}{b}}}$, which is the larger root of the equation $1 - \frac{1}{\alpha} - L\eta - \frac{(L_l+\theta G)^2\eta^2 m}{b} = 0$. Dividing both sides by $T = Sm$ gives

$$\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 = \frac{1}{T}\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\mathrm{grad} f(x_t^s)\|^2 \leq \frac{2\Delta}{T\eta} + \frac{\epsilon^2}{\alpha},$$

where $\tilde{x}$ is uniformly selected at random from $\{\{x_t^s\}_{t=0}^{m-1}\}_{s=1}^S$. (2) Under online setting, from (4.18), we have $\mathbb{1}_{\{B^s<n\}}\frac{1}{B^s} \leq \frac{\beta_s + \epsilon^2}{\alpha\sigma^2}$, where we choose $\alpha_1, \alpha_2 \geq \alpha$. This results in

$$\begin{aligned}
\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\mathrm{grad} f(x_t^s)\|^2 \leq \frac{2\Delta}{\eta} - \Big(1 - \frac{1}{\alpha} - L\eta - \frac{(L_l + \theta G)^2 \eta^2 m}{b}\Big) \sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|v_t^s\|^2 \\
+ \frac{2Sm\epsilon^2}{\alpha}.
\end{aligned}$$

Choose the same $\eta \leq \frac{2 - \frac{2}{\alpha}}{L + \sqrt{L^2 + 4(1-\frac{1}{\alpha})\frac{(L_l+\theta G)^2 m}{b}}}$, we have

$$\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 = \frac{1}{T}\sum_{s=1}^{S} \sum_{t=0}^{m-1} \mathbb{E}\|\mathrm{grad} f(x_t^s)\|^2 \leq \frac{2\Delta}{T\eta} + \frac{2\epsilon^2}{\alpha}.$$

By simply setting $\alpha \geq 4$ for both finite-sum and online setting, we can ensure

$$\mathbb{E}\|\mathrm{grad} f(\tilde{x})\|^2 \leq \frac{2\Delta}{T\eta} + \frac{\epsilon^2}{2}.$$

□

Proof of Corollary 4.3. By choosing $\alpha = 4, b = m$, $\eta = \frac{3}{2L+2\sqrt{L^2+3(L_l+\theta G)^2}}$, to ensure $\mathbb{E}\|\mathrm{grad} f(\tilde{x})\| \leq \epsilon$, we require at least

$$S = \frac{4\Delta}{m\eta\epsilon^2} = \frac{8\Delta}{3m\epsilon^2}(L + \sqrt{L^2 + 3(L_l + \theta G)^2}) = O(\frac{\Theta_2}{m\epsilon^2}),$$

with $\Theta_2 := L + \sqrt{L^2 + \varrho_2(L_l + \theta G)^2}$ where $\varrho_2 > 0$ is a constant that does not depend on any parameters. Let $\tilde{B}$ be the average batch size defined in (4.19). That is, $\tilde{B} = \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1\sigma^2/\beta_s, n\}$ under finite-sum setting and $\tilde{B} = \frac{1}{S}\sum_{s=1}^{S} \min\{\alpha_1\sigma^2/\beta_s, \alpha_2\sigma^2/\epsilon^2\}$ under online setting. Then one epoch requires $\tilde{B} + 2mb = O(\tilde{B} + m^2)$ IFO calls. Consider the choice of $m = \lfloor n^{1/2} \rfloor$ and $m = \frac{\sigma}{\epsilon}$ under finite-sum and online setting respectively. The total IFO complexity is given by

$$O\big(S\tilde{B} + Sm^2\big) = O\big(\frac{\Theta_2\tilde{B}}{m\epsilon^2} + \frac{\Theta_2 m}{\epsilon^2}\big) = \begin{cases} O\big(\tilde{B} + \frac{\Theta_2\tilde{B}}{\sqrt{n}\epsilon^2} + \frac{\Theta_2\sqrt{n}}{\epsilon^2}\big), & \text{(finite-sum)} \\ O\big(\frac{\Theta_2\tilde{B}}{\sigma\epsilon} + \frac{\Theta_2\sigma}{\epsilon^3}\big), & \text{(online)} \end{cases}$$

□

Proof of Corollary 4.4. From (4.22), we have

$$\begin{aligned} \sum_{t=0}^{m-1} \mathbb{E}[\|\mathrm{grad} f(x_t^s)\|^2|\mathcal{F}_0^s] \leq \frac{2}{\eta}\mathbb{E}[f(x_0^s) - f(x_m^s)|\mathcal{F}_0^s] + \mathbb{1}_{\{B<n\}}\frac{m\sigma^2}{B} \\ - \Big(1 - L\eta - \frac{(L_l + \theta G)^2\eta^2 m}{b}\Big)\sum_{t=0}^{m-1}\mathbb{E}[\|v_t^s\|^2|\mathcal{F}_0^s]. \end{aligned}$$

Consider stepsize choice $\eta \leq \frac{2}{L+\sqrt{L^2+4\frac{(L_l+\theta G)^2 m}{b}}}$, which ensures $1 - L\eta - \frac{(L_l+\theta G)^2\eta^2 m}{b} \geq$

0. Summing this result over $s = 1, ..., S$ and dividing by $T = Sm$ yields

$$\mathbb{E}\|\text{grad}f(\tilde{x})\|^2 = \frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|\text{grad}f(x_t^s)\|^2 \leq \frac{2\Delta}{T\eta} + \mathbb{1}_{\{B<n\}}\frac{\sigma^2}{B}.$$

Considering the choice of $b = m$ and $\eta = \frac{2}{L+\sqrt{L^2+4(L_l+\theta G)^2}}$ and following exactly the same procedures as in proof of Corollary 2.2, we require at least

$$S = O\Big(\frac{\Delta}{m\eta\epsilon^2}\Big) = O\Big(\frac{\Delta}{m\epsilon^2}\big(L + \sqrt{L^2 + 4(L_l + \theta G)^2}\big)\Big) = O\Big(\frac{\Theta_2}{m\epsilon^2}\Big).$$

One epoch requires $B + 2mb = O(B + m^2)$ IFO complexity. With the same choice of $m = \lfloor n^{1/2} \rfloor$ under finite-sum setting and $m = \frac{\sigma}{\epsilon}$ under online setting, total IFO complexity is given by

$$O\big(S(B + m^2)\big) = \begin{cases} O\big(n + \frac{\Theta_2\sqrt{n}}{\epsilon^2}\big), & \text{(finite-sum)} \\ O\big(\frac{\Theta_2\sigma}{\epsilon^3}\big), & \text{(online)} \end{cases}$$

□

4.E Convergence under gradient dominance condition

Proof of Theorem 4.4. At mega epoch k, we have $\mathbb{E}\|\text{grad}f(x_k)\| \leq \epsilon_k = \frac{\epsilon_0}{2^k}$ and $\mathbb{E}[f(x_k) - f(x^*)] \leq \tau\mathbb{E}\|\text{grad}f(x_k)\|^2 \leq \frac{\tau\epsilon_0^2}{4^k}$. □

Proof of Corollary 4.5. Define $\Delta_k := \mathbb{E}[f(x_k) - f(x^*)]$. At mega epoch k, to obtain ϵ_k-accurate solution, we require number of epochs

$$S_k = \frac{4\Delta_{k-1}}{m_k\eta\epsilon_k^2} = \frac{8\Theta_1\Delta_{k-1}}{3m_k\epsilon_k^2} \leq \frac{8\Theta_1}{3m_k\epsilon_k^2}\tau\mathbb{E}\|\text{grad}f(x_{k-1})\|^2 \leq \frac{8\Theta_1\tau}{3m_k}\frac{\epsilon_{k-1}^2}{\epsilon_k^2} = \frac{32\Theta_1\tau}{3m_k},$$

where the first inequality uses the definition of gradient dominance and the second inequality is by the fact that x_{t-1} is output from the preceding mega epoch and hence has gradient bounded by desired accuracy ϵ_{k-1}. The last equality is from the choice of ϵ_k. Define the average batch size $\tilde{B}_k = \frac{1}{S_k}\sum_{s=1}^{S_k}\min\{\alpha_1\sigma^2/\beta_s, n\}$ under finite-sum settings and $\tilde{B}_k = \frac{1}{S_k}\sum_{s=1}^{S_k}\min\{\alpha_1\sigma^2/\beta_s, \alpha_2\sigma^2/\epsilon_k^2\}$ under online settings. Then IFO complexity at mega epoch k is

$$S_k(\tilde{B}_k + 2m_k b_k) = O\big(S_k\tilde{B} + S_k m_k^3\big) = \begin{cases} O\big(\tilde{B}_k + \frac{\Theta_1 \tilde{B}_k \tau}{n^{1/3}} + \Theta_1 n^{2/3}\tau\big), & \text{(finite-sum)} \\ O\big(\frac{\Theta_1 \tilde{B}_k \tau \epsilon_k^{2/3}}{\sigma^{2/3}} + \frac{\Theta_1 \sigma^{4/3}\tau}{\epsilon_k^{4/3}}\big), & \text{(online)} \end{cases} \tag{4.23}$$

To ensure $\mathbb{E}\|\text{grad}f(x_K)\|^2 \leq \epsilon^2$, it is equivalent to requiring $\epsilon_K^2 = \frac{\epsilon_0^2}{2^{2K}} \leq \epsilon^2$. Therefore, we require at least $K = \log(\frac{\epsilon_0}{\epsilon})$ mega epochs. Accordingly, under finite-sum setting, the complexity in (4.23) depends on mega epoch k only through $\tilde{B}_k$. So total IFO complexity after running K mega epochs is simply $O\big(\sum_{k=1}^K \tilde{B}_k(1 + \frac{\Theta_1\tau}{n^{1/3}}) + (\Theta_1 n^{2/3}\tau)\log(\frac{1}{\epsilon})\big)$. Under online setting, both $\tilde{B}_k$ and ϵ_k of its complexity depend on mega epoch k. Hence we need to sum this result from $k = 1, ..., K = \log(\frac{\epsilon_0}{\epsilon})$. Note that $\sum_{k=1}^K \frac{1}{\epsilon_k^{4/3}} = \frac{2^{4/3}}{\epsilon_0^{4/3}}\frac{(2^k)^{4/3}-1}{2^{4/3}-1} \leq 2^{4/3}(\frac{2^k}{\epsilon_0})^{4/3} = O\big((\frac{1}{\epsilon})^{4/3}\big)$. Hence, total IFO complexity can be written as $O\big(\frac{\Theta_1\tau\sum_{k=1}^K \tilde{B}_k\epsilon_k^{2/3}}{\sigma^{2/3}} + \frac{\Theta_1\tau\sigma^{4/3}}{\epsilon^{4/3}}\big)$. Similarly, R-SVRG with fixed batch size $B_k = n$ under finite-sum cases and $B_k = \frac{2\sigma^2}{\epsilon_k^2}$ under online cases requires complexities $O\big((n + \Theta_1\tau n^{2/3})\log(\frac{1}{\epsilon})\big)$ and $O\big(\frac{\Theta_1\tau\sigma^{4/3}}{\epsilon^{4/3}}\big)$ respectively. The proof is exactly the same except that we replace $\tilde{B}_k$ with B_k. □

Proof of Corollary 4.6. The proof is exactly the same as that for R-AbaSVRG and R-SVRG and hence skipped. □

Next, we provide complexity results for R-SD and R-SGD under gradient dominance condition. We simply restart the algorithms similar to variance reduction methods.

Algorithm 4: R-GD-SD/SGD

```
Input: Initial accuracy $\epsilon_0$ and desired accuracy $\epsilon$, initialization $x_0$.
for $k = 1, ..., K$ do
    $\epsilon_k = \frac{\epsilon_{k-1}}{2}$.
    Set $T_k$ sufficient to achieve $\epsilon_k$-accurate solution and choose stepsize $\eta$
    accordingly.
    $x_0^k = x^{k-1}$.
    for $t = 1, ..., T_k$ do
        (R-SD): $x_t^k = \mathrm{Retr}_{x_{t-1}^k}(-\eta\, \mathrm{grad} f(x_{t-1}^k))$.
        (R-SGD): $x_t^k = \mathrm{Retr}_{x_{t-1}^k}(-\eta\, \mathrm{grad} f_{i_t^k}(x_{t-1}^k))$, with $i_t^k \in [n]$ random.
    end for
    $x^k$ is chosen uniformly at random from $\{x_t^k\}_{t=0}^{T_k - 1}$.
end for
Output: $x^K$.
```

Theorem 4.5 (IFO complexity of R-SD and R-SGD under gradient dominance condition)**.** *Suppose f is retraction L-smooth and also τ-gradient dominated. Consider Algorithm 4 with R-SD solver. Then total IFO complexity to achieve ϵ-accurate solution is given by $O\big((n + L\tau n)\log(\frac{1}{\epsilon})\big)$. Suppose additionally that f has G-bounded gradient. That is, $\|\mathrm{grad} f_i(x)\| \leq G$, with i being a random index from $[n]$. Consider Algorithm 4 with R-SGD solver. Total IFO complexity to achieve ϵ-accurate solution is $O\big(\frac{LG^2}{\epsilon^2}\big)$.*

Proof. The proof idea is similar to that of Theorem 4.4. We first consider a single epoch k. By retraction L-smoothness,

$$
\begin{aligned}
f(x_{t+1}^k) &\leq f(x_t^k) + \langle \mathrm{grad} f(x_t^k), -\eta \mathrm{grad} f(x_t^k)\rangle + \frac{L}{2}\| - \eta \mathrm{grad} f(x_t^k)\|^2 \\
&= f(x_t^k) - (\eta - \frac{L\eta^2}{2})\|\mathrm{grad} f(x_t^k)\|^2.
\end{aligned}
$$

Choose $\eta = \frac{1}{L}$ and summing this inequality from $t = 0, ..., T_k - 1$ gives

$$
\frac{1}{T_k}\sum_{t=0}^{T_k - 1} \mathbb{E}\|\mathrm{grad} f(x_t^k)\|^2 \leq \frac{2L\mathbb{E}[f(x_0^k) - f(x_{T_k}^k)]}{T_k} \leq \frac{2L\Delta_{k-1}}{T_k},
$$

where $\Delta_{k-1} := \mathbb{E}[f(x^{k-1}) - f(x^*)]$. Note the update rule of x^k gives $\mathbb{E}\|\mathrm{grad} f(x^k)\|^2 = \frac{1}{T_k}\sum_{t=0}^{T_k-1} \mathbb{E}\|\mathrm{grad} f(x_t^k)\|^2$. Therefore, to ensure $\mathbb{E}\|\mathrm{grad} f(x^k)\|^2 \leq \epsilon_k^2$, we require at least

$$T_k = \frac{2L\Delta_{k-1}}{\epsilon_k^2} \leq \frac{2L\tau\mathbb{E}\|\mathrm{grad}f(x^{k-1})\|^2}{\epsilon_k^2} \leq \frac{2L\tau\epsilon_{k-1}^2}{\epsilon_k^2} = 8L\tau.$$

IFO complexity of a single epoch is given by $8L\tau n = O(n + L\tau n)$. By similar argument, to ensure $\mathbb{E}\|\mathrm{grad}f(x^K)\|^2 \leq \epsilon^2$, we require $\log(\frac{1}{\epsilon})$ epochs. Hence the total IFO complexity of R-SD is given as $O\big((n + L\tau n)\log(\frac{1}{\epsilon})\big)$. This result matches the complexity of Euclidean gradient descent under gradient dominance condition (see Reddi, Hefny, et al. (2016); Polyak (1963)). Similarly, for R-SGD, we have

$$\begin{aligned}\mathbb{E}[f(x_{t+1}^k)] &\leq \mathbb{E}[f(x_t^k) + \langle \mathrm{grad}f(x_t^k), -\eta\mathrm{grad}f_{i_t^k}(x_t^k)\rangle + \frac{L}{2}\| - \eta\mathrm{grad}f_{i_t^k}(x_t^k)\|^2] \\ &= \mathbb{E}[f(x_t^k)] - \eta\mathbb{E}\|\mathrm{grad}f(x_t^k)\|^2 + \frac{L\eta^2G^2}{2}.\end{aligned}$$

Choosing $\eta = \frac{z}{\sqrt{T_k}}$ where $z > 0$ is a constant and summing over $t = 0, ..., T_k - 1$, we have

$$\frac{1}{T_k}\sum_{t=0}^{T_k-1}\mathbb{E}\|\mathrm{grad}f(x_t^k)\|^2 \leq \frac{\Delta_{k-1}}{z\sqrt{T_k}} + \frac{LG^2z}{2\sqrt{T_k}}.$$

Choose $z = \sqrt{\frac{2\Delta_{k-1}}{LG^2}}$ to minimize right hand side as $\frac{\sqrt{2LG^2\Delta_{k-1}}}{\sqrt{T_k}}$. Hence to ensure $\mathbb{E}\|\mathrm{grad}f(x^k)\|^2 \leq \epsilon_k^2$, we require at least

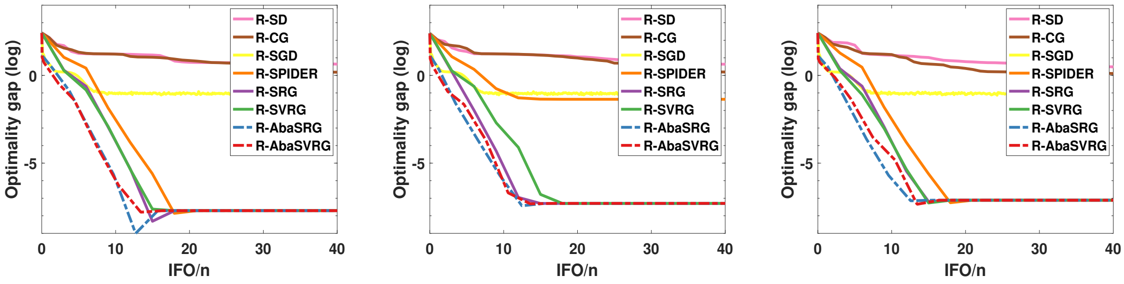
$$T_k = \frac{2LG^2\Delta_{k-1}}{\epsilon_k^4} \leq \frac{2LG^2\epsilon_{k-1}^2}{\epsilon_k^4} = \frac{8LG^2}{\epsilon_k^2}.$$

IFO complexity of a single epoch is therefore $O(\frac{LG^2}{\epsilon_k^2})$. To achieve ϵ-accurate solution, we require $\log(\frac{1}{\epsilon})$ epochs and hence, the total IFO complexity of R-SGD is $O(\frac{LG^2}{\epsilon^2})$. □

4.F Additional experiment results

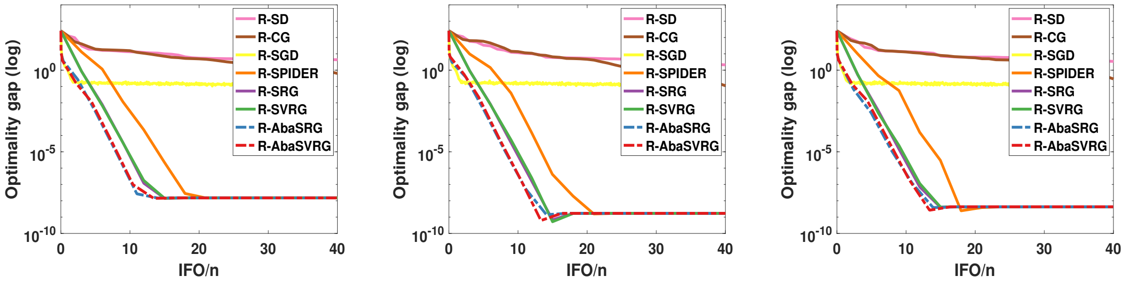
4.F.1 PCA problem on Grassmann manifold

We here present results on synthetic datasets by varying n and d and also examine result sensitivity on all datasets by conducting three independent runs.



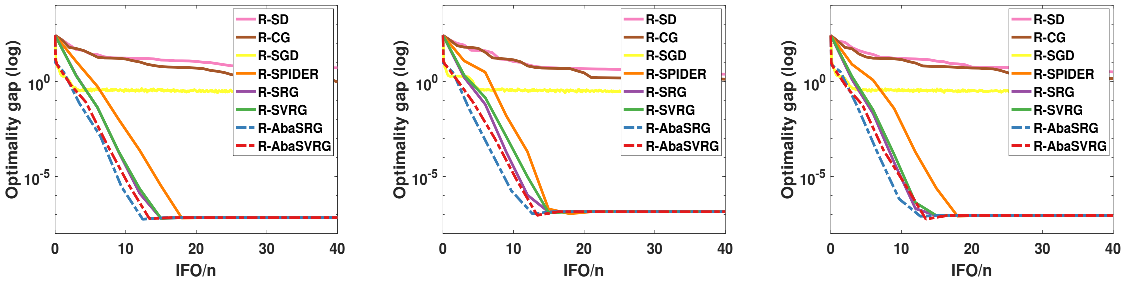
(a) Run 1 (b) Run 2 (c) Run 3

Figure 4.5: Synthetic dataset with $n = 100000, d = 200, r = 5$.



(a) Run 1 (b) Run 2 (c) Run 3

Figure 4.6: Synthetic dataset with $n = 200000, d = 200, r = 5$.

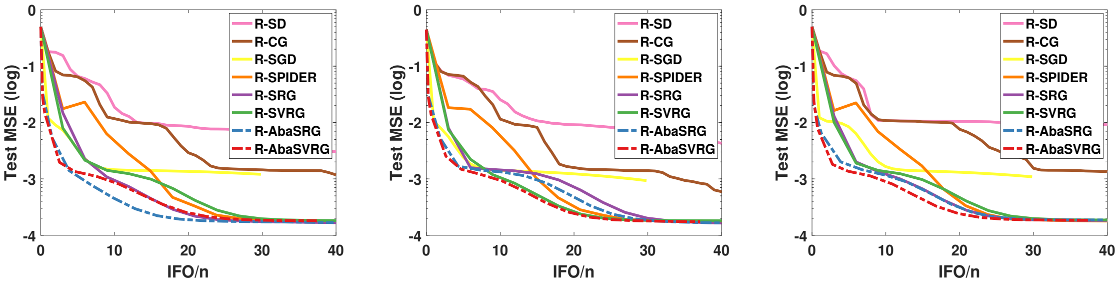


(a) Run 1 (b) Run 2 (c) Run 3

Figure 4.7: Synthetic dataset with $n = 100000, d = 300, r = 5$.

4.F.2 LRMC on Grassmann manifold

Additional results on synthetic datasets. We first present three independent runs in Fig. 4.8 to test the sensitivity of batch size adaptation on baseline synthetic dataset with $n = 20000, d = 100, r = 5, \mathrm{cn} = 50, \mathrm{os} = 8, \varepsilon = 10^{-10}$. We also compare algorithms on datasets with different characteristics. Specifically, we consider a large-scale dataset with $n = 40000$, a high dimensional dataset with $d = 200$, a high-rank dataset with $r = 10$, an ill-conditioned dataset with $\mathrm{cn} = 100$, a low-sampling dataset with $\mathrm{os} = 4$ and a noisy dataset with $\varepsilon = 10^{-8}$. Test MSE results are presented in Fig. 4.9.

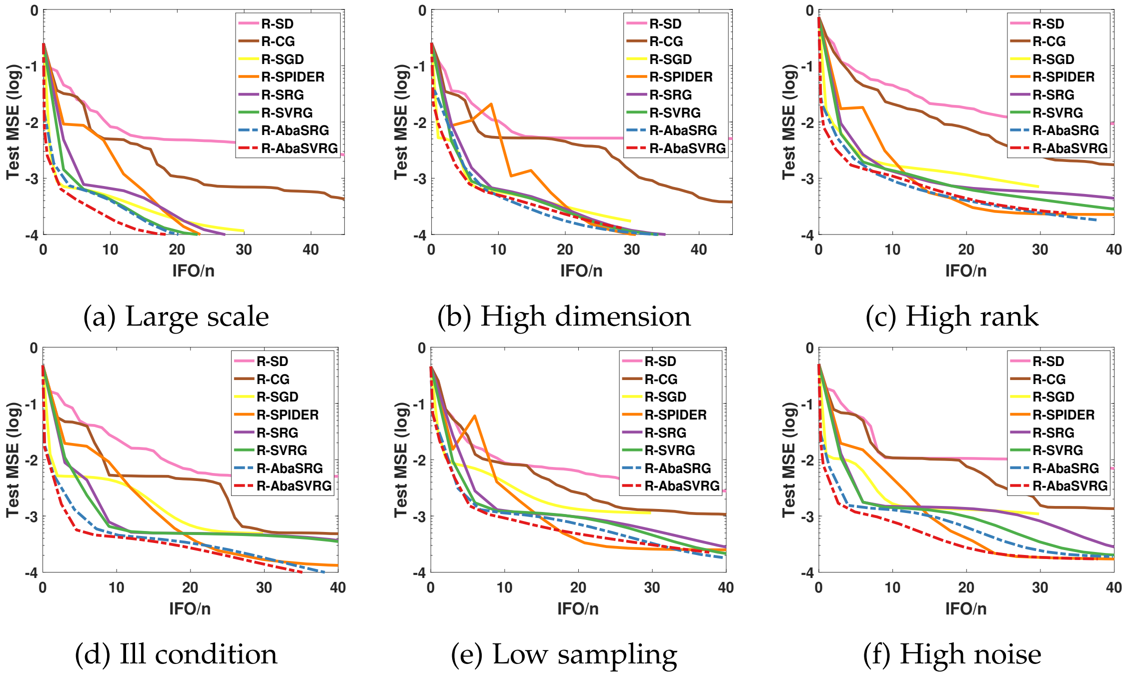


(a) Run 1

(b) Run 2

(c) Run 3

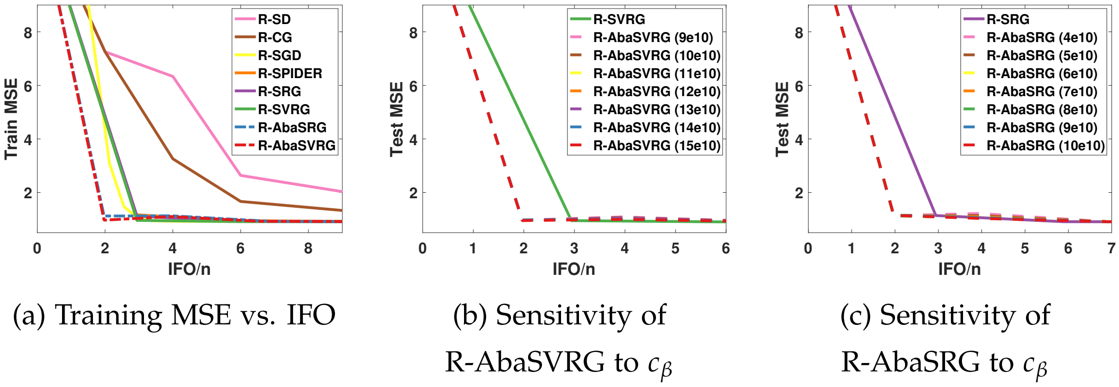
Figure 4.8: LRMC Result sensitivity on baseline synthetic dataset



(a) Large scale
(b) High dimension
(c) High rank
(d) Ill condition
(e) Low sampling
(f) High noise

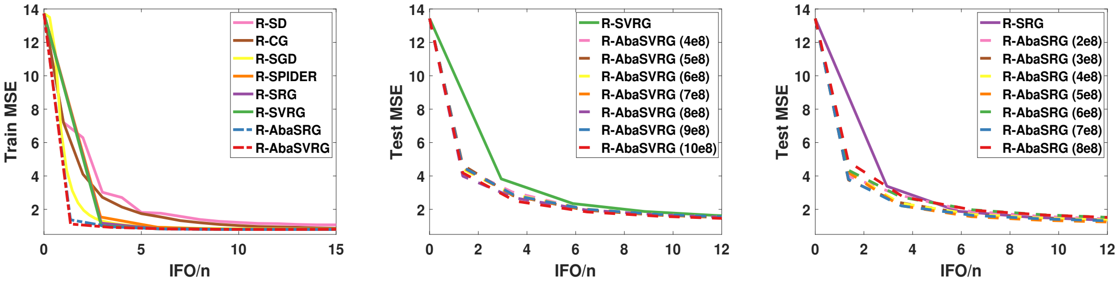
Figure 4.9: LRMC results on datasets with different characteristics.

Additional results for Netflix and Movielens dataset. We present training MSE results on Netflix and Movielens datasets accompanying test MSE results in the main text. Also, we examine sensitivity of R-AbaSVRG and R-AbaSRG to parameter c_β.



(a) Training MSE vs. IFO
(b) Sensitivity of R-AbaSVRG to c_β
(c) Sensitivity of R-AbaSRG to c_β

Figure 4.10: Additional LRMC results on Netflix dataset.



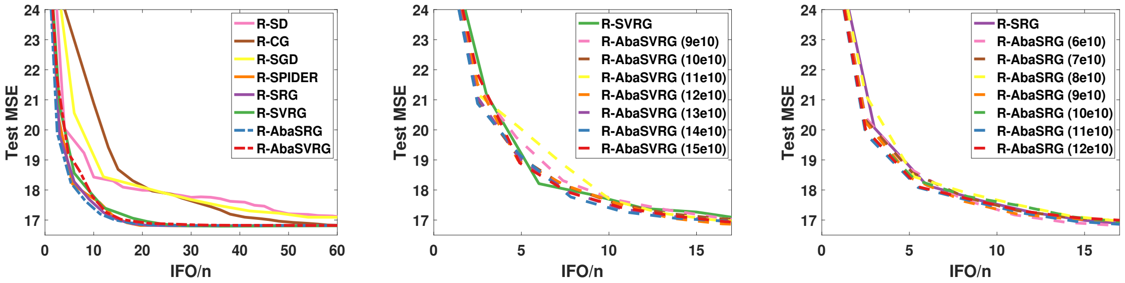
(a) Training MSE vs. IFO

(b) Sensitivity of R-AbaSVRG to c_β

(c) Sensitivity of R-AbaSRG to c_β

Figure 4.11: Additional LRMC results on Movielens dataset.

Additonal results on Jester dataset. We also consider Jester dataset Goldberg et al. (2001) that contains continuous ratings in $[-10, 10]$ from 24983 (d) users on 100 jokes (n). We extract 10 ratings per user as test set. We choose $q = -6, l = 10$.



(a) Test MSE vs. IFO

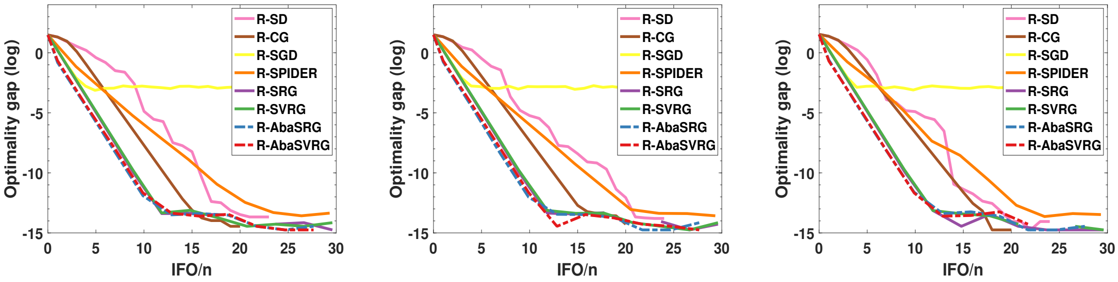
(b) Sensitivity of R-AbaSVRG to c_β

(c) Sensitivity of R-AbaSRG to c_β

Figure 4.12: LRMC results on Jester dataset.

4.F.3 RKM on SPD manifold

Additional results on synthetic datasets. Similar to PCA and LRMC, result sensitivity on baseline synthetic dataset with $(n, d, \text{cn}) = (5000, 10, 20)$ is evaluated by presenting three independent results in Fig. 4.13. We also evaluate algorithms on datasets with large samples $n = 10000$, with high dimension $d = 30$ and with high condition number $\text{cn} = 50$. Optimality gap results are presented in Fig. 4.14.

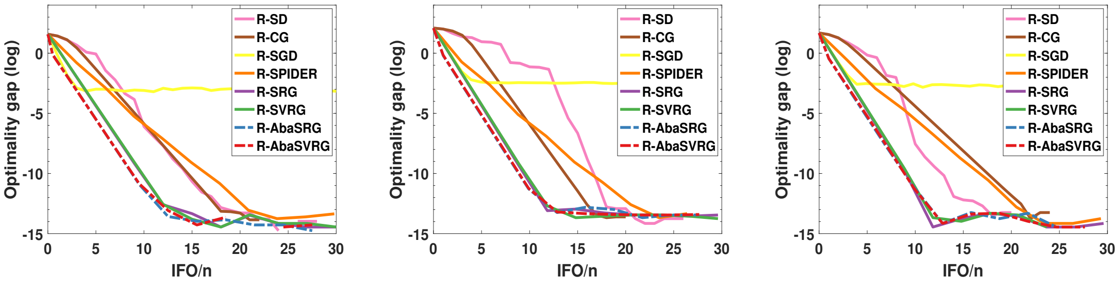


(a) Run 1

(b) Run 2

(c) Run 3

Figure 4.13: RKM Result sensitivity on baseline synthetic dataset.



(a) Large scale

(b) High dimension

(c) Ill condition

Figure 4.14: RKM Result on datasets with different characteristics.

4.G Batch size adaptation for Riemannian proximal gradient methods

To validate the claim that the proposed batch size adaptation is helpful for broader settings, we consider the example of nonsmooth optimization with composite functions. The problem is

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x), \tag{4.24}$$

where f_i, $i = 1, ..., n$ are smooth, possibly nonconvex functions on Riemannian manifold while h is retraction-convex and nonsmooth (in the manifold sense). The convexity of h is to ensure the proximal mapping is well-defined. Similarly, we refer to problem (4.24) as finite-sum and online settings depending on the

Algorithm 5: R-AbaPSRG

```
Input: stepsize $\eta$, epoch length $S$, inner loop size $m$, mini-batch size $b$,
  adaptive batch size parameters $\alpha_1, \alpha_2, \beta_1$, initialization $\tilde{x}^0$, desired accuracy
  $\epsilon$.
for $s = 1, ..., S$ do
  $x_0^s = \tilde{x}^{s-1}$.
  Draw a sample $\mathcal{B}^s$ from $[n]$ of size $B^s$ without replacement, where
  $B^s = \begin{cases} \min\{\alpha_1 \sigma^2/\beta_s, n\}, & \text{(finite-sum)} \\ \min\{\alpha_1 \sigma^2/\beta_s, \alpha_2 \sigma^2/\epsilon^2\}, & \text{(online)} \end{cases}$
  $v_0^s = \mathrm{grad} f_{\mathcal{B}^s}(x_0^s)$.
  Solve $\xi_0^s = \arg\min_{\xi \in T_{x_0^s}\mathcal{M}} \langle v_0^s, \xi \rangle + \frac{1}{2\eta}\|\xi\|^2 + h(\mathrm{Retr}_{x_0^s}(\xi))$.
  $x_1^s = \mathrm{Retr}_{x_0^s}(\xi_0^s)$.
  $\beta_{s+1} = \|\xi_0^s\|^2/(m\eta^2)$.
  for $t = 1, ..., m-1$ do
    Draw a sample $\mathcal{I}_t^s$ from $[n]$ of size $b$ with replacement.
    $v_t^s = \mathrm{grad} f_{\mathcal{I}_t^s}(x_t^s) - \mathcal{T}_{x_{t-1}^s}^{x_t^s}\big(\mathrm{grad} f_{\mathcal{I}_t^s}(x_{t-1}^s) - v_{t-1}^s\big)$.
    Solve $\xi_t^s = \arg\min_{\xi \in T_{x_t^s}\mathcal{M}} \langle v_t^s, \xi \rangle + \frac{1}{2\eta}\|\xi\|^2 + h(\mathrm{Retr}_{x_t^s}(\xi))$.
    $x_{t+1}^s = \mathrm{Retr}_{x_t^s}(\xi_t^s)$.
    $\beta_{s+1} = \beta_{s+1} + \|\xi_t^s\|^2/(m\eta^2)$
  end for
  $\tilde{x}^s = x_m^s$.
end for
Output: $\tilde{x}$ uniformly selected at random from $\{\{x_t^s\}_{t=0}^{m-1}\}_{s=1}^S$.
```

size of n. If the manifold is the Stiefel manifold (or any embedded submanifold of the Euclidean space), one can solve the problem efficiently using the Riemannian proximal stochastic recursive gradient (B. Wang et al., 2022). For general manifolds, we refer readers to W. Huang & Wei (2022) while no analysis has been made for nonconvex functions under stochastic settings.

Here we present the Riemannian proximal stochastic recursive gradient with batch size adaptation (R-AbaPSRG) in Algorithm 5, with a particular focus on the embedded submanifold of the Euclidean space. One can follow similar ideas in the main text to design SVRG-based method. Hence, with a slight abuse of notation, for the subsequent section, $\langle \cdot, \cdot \rangle, \| \cdot \|$ denote the Euclidean inner product and norm respectively.

Compared to R-AbaSRG for smooth optimization, Algorithm 5 simply replaces the retraction step with the proximal mapping after constructing the modified gradient. However, now the batch size is adapted based on the norm of ξ_t^s rather than v_t^s, which corresponds to the generalized modified gradient. Also we highlight in Algorithm 5, the proximal subproblem is defined with $h(\mathrm{Retr}_x(\xi))$. This can pose difficulties for both the analysis and practical implementation. Thus we follow B. Wang et al. (2022) to approximate $h(\mathrm{Retr}_x(\xi))$ with $h(x+\xi)$ for embedded submanifolds where the addition is performed in the ambient Euclidean space. Such manifolds include Sphere, Stiefel and fixed-rank manifolds.

Hence, the subproblem becomes $\xi^* = \arg\min_{\xi \in T_x\mathcal{M}} \langle v, \xi \rangle + \frac{1}{2\eta}\|\xi\|^2 + h(x + \xi)$. Accordingly we define the generalized gradient $G(x, v, \eta) = -\xi^*/\eta$. Hence the update becomes $x_{t+1}^s = \mathrm{Retr}_{x_t^s}(-\eta\, G(x_t^s, v_t^s, \eta))$ so that we can analyze similarly as for the smooth optimization.

Note that the simplified subproblem is a function defined on the tangent space of embedded submanifold usually with standard inner product. This is equivalent to optimizing in the Euclidean space without nonlinear retraction, which can be solved efficiently.

In Section 4.G.1 below, we first briefly review some preliminary knowledge for Riemannian nonsmooth optimization along with necessary assumptions and important lemmas for convergence analysis. See B. Wang et al. (2022); W. Huang & Wei (2022) for more detailed treatments on the topic.

4.G.1 Preliminaries, assumptions and lemmas

We first define the stationary point of problem (4.24), which is based on (B. Wang et al., 2022, Definition 2), as well as the ϵ-approximate solution and IFO complexity.

Definition 4.3 (Stationary point of composite function)**.** A point $x^* \in \mathcal{M}$ is a stationary point of problem (4.24) if $0 \in \hat{\partial} F(x^*) := \mathrm{grad} f(x^*) + \mathcal{P}_{x^*}(\partial h(x^*))$, where $\hat{\partial} F$ is the generalized Clarke subdifferential defined in (S. Hosseini et al.,

2018) and $\mathcal{P}_{x^*}$ is the projection to tangent space $T_{x^*}\mathcal{M}$. Moreover, if $\xi^* = 0$, i.e. $G(x^*, v, \eta) = 0$, x^* is a stationary point.

Definition 4.4 (ϵ-accurate solution and IFO complexity)**.** Output x from a stochastic algorithm is an ϵ-accurate solution if $\mathbb{E}\|G(x, \mathrm{grad} f(x), \eta)\| \leq \epsilon$. Similarly, an IFO oracle call takes in an index i and outputs $\mathrm{grad} f_i(x)$ as in the smooth setting.

We need to make the same assumptions as in Assumption 4.1 in the main text, but only for function f and its components f_i. In addition, we require assumptions for the nonsmooth term $h(x)$. For the purpose, we again consider the neighbourhood $\mathcal{X} \subseteq \mathcal{M}$ around a stationary point x^*.

Assumption 4.4.

(4.4.1) Function h is convex (in the Euclidean sense). That is, for any $x, y \in \mathcal{X}$, it satisfies $h((1-t)x + t(y)) \leq (1-t)f(x) + tf(y)$ for $t \in [0, 1]$.

(4.4.2) Function h is L_h-Lipschitz continuous (in the Euclidean sense). That is, for any $x, y \in \mathcal{X}$, $|h(x) - h(y)| \leq L_h\|x - y\|$.

Note that the convexity and Lipschitzness are notions in the Euclidean sense because we use $h(x + \xi)$ instead of $h(\mathrm{Retr}_x(\xi))$. The l_1 norm is one common example that satisfies the assumption.

Lastly, given that we replace retraction with addition for the subproblem, we need to assume the difference is bounded as in B. Wang et al. (2022). This assumption naturally follows from the Taylor approximation and can be ensured in a compact set $\mathcal{X}$.

Assumption 4.5. For any $x \in \mathcal{X}, \xi \in T_x\mathcal{M}$, there exists a constant $M > 0$ such that $\|\mathrm{Retr}_x(\xi) - (x + \xi)\| \leq M\|\xi\|^2$.

Denote $g(\xi; v) := \langle v, \xi \rangle + \frac{1}{2\eta}\|\xi\|^2 + h(x + \xi)$ where one can verify that $g(\xi; v)$ is $(1/\eta)$-strongly convex due to the convexity of h. The following lemma is modified from (B. Wang et al., 2022, Lemma 8).

Lemma 4.8. *Suppose Assumption 4.4 holds. Then the optimized ξ_t^s satisfies $g(\xi_t^s; v_t^s) - g(0; v_t^s) \leq -\frac{\eta}{2}\|G(x_t^s, v_t^s, \eta)\|^2$.*

Proof. By strongly convexity of g, we have for $\xi_t^s, 0 \in T_{x_t^s}\mathcal{M}$,

$$g(0; v_t^s) \geq g(\xi_t^s; v_t^s) + \langle \hat{\partial} g(\xi_t^s; v_t^s), -\xi_t^s \rangle + \frac{1}{2\eta}\|\xi_t^s\|^2 = g(\xi_t^s; v_t^s) + \frac{1}{2\eta}\|\xi_t^s\|^2. \quad (4.25)$$

where the equality is due to the optimality condition of $g(\xi; v_t^s)$ and ξ_t^s is the optimal solution. That is, $\langle \hat{\partial} g(\xi_t^s; v_t^s), u \rangle = 0$ for any $u \in T_{x_t^s}\mathcal{M}$. Hence this is equivalent to $h(x_t^s + \xi_t^s) - h(x_t^s) \leq -\langle v_t^s, \xi_t^s \rangle - \frac{1}{\eta}\|\xi_t^s\|^2$. Applying the definitions of $g(\xi; v)$ and generalized gradient completes the proof. □

4.G.2 Convergence analysis

Lemma 4.9 (Gradient estimation bound of R-AbaPSRG)**.** *Suppose Assumption 4.1 (in the main text) holds for function f and consider Algorithm 5. Then we have the following bound on the estimation error.*

$$\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 | \mathcal{F}_0^s] \leq \frac{(L_l + \theta G)^2 \eta^2}{b} \sum_{i=0}^{t} \mathbb{E}[\|G(x_i^s, v_i^s, \eta)\|^2 | \mathcal{F}_0^s] + \mathbb{1}_{\{B^s < n\}} \frac{\sigma^2}{B^s}.$$

Proof. The proof follows from the proof of Lemma 4.2 where we replace v_t^s with the generalized gradient. □

The following lemma bounds the iteration function value gap.

Lemma 4.10. *Suppose Assumption 4.1 and 4.4 hold. Then for the optimized ξ_t^s, we have*

$$\begin{aligned}&F(x_{t+1}^s) - F(x_t^s) \\ &\leq \eta \langle v_t^s - \mathrm{grad} f(x_t^s), G(x_t^s, v_t^s, \eta) \rangle + \big(\frac{L\eta^2}{2} - \eta\big)\|G(x_t^s, v_t^s, \eta)\|^2 + h(x_{t+1}^s) - h(x_t^s + \xi_t^s).\end{aligned}$$

Proof. Firstly $F(x_{t+1}^s) - F(x_t^s) \leq -\eta \langle \mathrm{grad} f(x_t^s), G(x_t^s, v_t^s, \eta) \rangle + \frac{L\eta^2}{2}\|G(x_t^s, v_t^s, \eta)\|^2 +$

$h(x_{t+1}^s) - h(x_t^s)$ by retraction L-smoothness of f. Also from the definition of g, we have

$$\begin{aligned} g(\xi_t^s; v_t^s) - g(0; v_t^s) &= \langle v_t^s, \xi_t^s \rangle + \frac{1}{2\eta} \|\xi_t^s\|^2 + h(x_t^s + \xi_t^s) - h(x_t^s) \\ &= -\eta \langle v_t^s, G(x_t^s, v_t^s, \eta) \rangle + \frac{\eta}{2} \|G(x_t^s, v_t^s, \eta)\|^2 + h(x_t^s + \xi_t^s) - h(x_t^s). \end{aligned}$$

Substitute this result in the above inequality and apply Lemma 4.8 yields the result. □

Now we present a lemma that bounds the norm of generalized gradient.

Lemma 4.11. *We can show* $\|G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|^2 \leq 2\|G(x_t^s, v_t^s, \eta)\|^2 + 2\|v_t^s - \mathrm{grad} f(x_t^s)\|^2$.

Proof. Let $\tilde{\xi}_t^s := \arg\min_{\xi \in T_{x_t^s}\mathcal{M}} g(\xi; \mathrm{grad} f(x_t^s)) = -\eta G(x_t^s, \mathrm{grad} f(x_t^s), \eta)$.

$$\begin{aligned} \|G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|^2 &\leq 2\|G(x_t^s, v_t^s, \eta)\|^2 + 2\|G(x_t^s, v_t^s, \eta) - G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|^2 \\ &= 2\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{2}{\eta^2}\|\xi_t^s - \tilde{\xi}_t^s\|^2 \\ &\leq 2\|G(x_t^s, v_t^s, \eta)\|^2 + 2\|v_t^s - \mathrm{grad} f(x_t^s)\|^2, \qquad (4.26) \end{aligned}$$

where the last inequality is due to (B. Wang et al., 2022, Lemma 12) derived from the optimality condition of the subproblem. □

Theorem 4.6 (Convergence of R-AbaPSRG). *Suppose Assumptions 1, 4.4 and 4.5 hold. Consider Algorithm 5 and choose a stepsize* $\eta \leq \frac{\frac{1}{2} - \frac{2}{\alpha}}{\tilde{L} + \sqrt{\tilde{L}^2 + (1 - \frac{4}{\alpha}) \frac{(L_l + \theta G)^2 m}{2b}}}$, *then under both finite-sum and online settings, we have*

$$\mathbb{E}\|G(\tilde{x}, \mathrm{grad} f(\tilde{x}), \eta)\|^2 \leq \frac{11\Delta}{T\eta} + \frac{13}{8}\epsilon^2,$$

where $\Delta := F(\tilde{x}^0) - F(x^*)$.

Proof. By Lemma 4.10, we have

$$
\begin{aligned}
&F(x_{t+1}^s) - F(x_t^s) \\
&\leq \eta \langle v_t^s - \mathrm{grad} f(x_t^s), G(x_t^s, v_t^s, \eta)\rangle + \big(\frac{L\eta^2}{2} - \eta\big)\|G(x_t^s, v_t^s, \eta)\|^2 + h(x_{t+1}^s) - h(x_t^s + \xi_t^s) \\
&\leq \frac{\eta}{2}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 + \big(\frac{L\eta^2}{2} - \frac{\eta}{2}\big)\|G(x_t^s, v_t^s, \eta)\|^2 + L_h\|\mathrm{Retr}_{x_t^s}(\xi_t^s) - (x_t^s + \xi_t^s)\| \\
&\leq \frac{\eta}{2}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 + \big(\frac{L\eta^2}{2} - \frac{\eta}{2}\big)\|G(x_t^s, v_t^s, \eta)\|^2 + L_h M\|\xi_t^s\|^2 \\
&= \frac{\eta}{2}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 + (\tilde{L}\eta^2 - \frac{\eta}{2})\|G(x_t^s, v_t^s, \eta)\|^2,
\end{aligned}
$$

where we denote $\tilde{L} := \frac{L}{2} + L_h M$. The second inequality applies Lipschitz continuity of h and $\langle a, b\rangle \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$. The third inequality is due to Assumption 4.5. Taking the expectation with respect to $\mathcal{F}_0^s$ and applying Lemma 4.9 yield

$$
\begin{aligned}
&\mathbb{E}[F(x_{t+1}^s) - F(x_t^s)|\mathcal{F}_0^s] \\
&\leq \frac{\eta}{2}\mathbb{E}[\|v_t^s - \mathrm{grad} f(x_t^s)\|^2|\mathcal{F}_0^s] + (\tilde{L}\eta^2 - \frac{\eta}{2})\mathbb{E}[\|G(x_t^s, v_t^s, \eta)\|^2|\mathcal{F}_0^s] \\
&\leq \frac{(L_l + \theta G)^3\eta^2}{2b}\sum_{i=0}^{t}\mathbb{E}[\|G(x_i^s, v_i^s, \eta)\|^2|\mathcal{F}_0^s] + \mathbb{1}_{\{B^s<n\}}\frac{\eta\sigma^2}{2B^s} \\
&\quad + (\tilde{L}\eta^2 - \frac{\eta}{2})\mathbb{E}[\|G(x_t^s, v_t^s, \eta)\|^2|\mathcal{F}_0^s].
\end{aligned}
$$

Telescoping the inequality from $t = 0, ..., m-1$ gives

$$
\begin{aligned}
&\mathbb{E}[F(x_m^s) - F(x_0^s)|\mathcal{F}_0^s] \\
&\leq \frac{(L_l + \theta G)^2\eta^3}{2b}\sum_{t=0}^{m-1}\sum_{i=0}^{t}\mathbb{E}[\|G(x_i^s, v_i^s, \eta)\|^2|\mathcal{F}_0^s] + \mathbb{1}_{\{B^s<n\}}\frac{m\eta\sigma^2}{2B^s} \\
&\quad + (\tilde{L}\eta^2 - \frac{\eta}{2})\sum_{t=0}^{m-1}\mathbb{E}[\|G(x_t^s, v_t^s, \eta)\|^2|\mathcal{F}_0^s] \\
&\leq -\eta\big(\frac{1}{2} - \tilde{L}\eta - \frac{(L_l + \theta G)^2\eta^2 m}{2b}\big)\sum_{t=0}^{m-1}\mathbb{E}[\|G(x_t^s, v_t^s, \eta)\|^2|\mathcal{F}_0^s] + \mathbb{1}_{\{B^s<n\}}\frac{m\eta\sigma^2}{2B^s}.
\end{aligned}
$$

Then summing over $s = 1, ..., S$ and taking full expectation gives

$$-\Delta \leq -\eta\big(\frac{1}{2} - \tilde{L}\eta - \frac{(L_l + \theta G)^2\eta^2 m}{2b}\big) \sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \mathbb{1}_{\{B^s < n\}} \sum_{s=1}^{S} \mathbb{E}[\frac{m\eta\sigma^2}{2B^s}],$$

where $\Delta := F(\tilde{x}^0) - F(x^*)$. The following analysis follows closely as that of R-AbaSRG.

(1) Under the finite-sum setting,

$$\begin{aligned} -\Delta &\leq -\eta\big(\frac{1}{2} - \frac{1}{\alpha} - \tilde{L}\eta - \frac{(L_l + \theta G)^2\eta^2 m}{2b}\big) \sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{Sm\epsilon^2\eta}{2\alpha} \\ &\leq -\frac{\eta}{4} \sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{Sm\epsilon^2\eta}{2\alpha}, \end{aligned}$$

where we choose $\eta \leq \frac{\frac{1}{2} - \frac{2}{\alpha}}{\tilde{L} + \sqrt{\tilde{L}^2 + (1 - \frac{4}{\alpha})\frac{(L_l + \theta G)^2 m}{2b}}}$. Thus we have

$$\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 \leq \frac{4\Delta}{T\eta} + \frac{2\epsilon^2}{\alpha}.$$

(2) Under the online setting, similarly we have $\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 \leq \frac{4\Delta}{T\eta} + \frac{4\epsilon^2}{\alpha}$. Choose $\alpha = 8$ for both cases, we have under both finite-sum and online settings,

$$\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 \leq \frac{4\Delta}{T\eta} + \frac{\epsilon^2}{2}.$$

Note the ϵ-accurate solution is with respect to $\mathbb{E}\|G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|$. Thus we further need the following bound. From Lemma 4.9, we have

$$\begin{aligned} &\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 \\ &\leq \frac{1}{T}\left(\frac{(L_l + \theta G)^2\eta^2 m}{b} + \frac{1}{8}\right)\sum_{s=1}^{S}\sum_{t=0}^{m-1} \mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{\epsilon^2}{8}. \end{aligned}$$

Finally, from Lemma 4.11 and the above inequality, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|^2 \\
&\leq \frac{2}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{2}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|v_t^s - \mathrm{grad} f(x_t^s)\|^2 \\
&\leq \left(2 + \frac{2(L_l+\theta G)^2\eta^2 m}{b} + \frac{1}{4}\right)\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|G(x_t^s, v_t^s, \eta)\|^2 + \frac{\epsilon^2}{4} \\
&\leq \left(\frac{2(L_l+\theta G)^2\eta^2 m}{b} + \frac{9}{4}\right)(\frac{4\Delta}{T\eta} + \frac{\epsilon^2}{2}) + \frac{\epsilon^2}{4} \\
&\leq \frac{11}{4}(\frac{4\Delta}{T\eta} + \frac{\epsilon^2}{2}) + \frac{\epsilon^2}{4} = \frac{11\Delta}{T\eta} + \frac{13}{8}\epsilon^2,
\end{aligned}
$$

where the last inequality is due to the choice of η. Finally, we notice that $\tilde{x}$ is uniformly chosen from all the iterates, which gives

$$
\mathbb{E}\|G(x_t^s, \mathrm{grad} f(\tilde{x}), \eta)\|^2 = \frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{m-1}\mathbb{E}\|G(x_t^s, \mathrm{grad} f(x_t^s), \eta)\|^2,
$$

thus completing the proof. □

Corollary 4.7 (IFO complexity of R-AbaPSRG). *With the same assumptions as in Theorem 4.6, choose $b = m$, $\eta = \frac{1}{4\tilde{L}^2+2\sqrt{4\tilde{L}+(L_l+\theta G)^2}}$ ($\alpha = 8$). Set $m = \lfloor n^{1/2} \rfloor$ under finite-sum setting and $m = \frac{\sigma}{\epsilon}$ under online setting. The IFO complexity of Algorithm 5 to obtain $\sqrt{2}\epsilon$-accurate solution is*

$$
\begin{cases}
O\big(\tilde{B} + \frac{\Theta_3\tilde{B}}{\sqrt{n}\epsilon^2} + \frac{\Theta_3\sqrt{n}}{\epsilon^2}\big), & \textit{(finite-sum)} \\
O\big(\frac{\Theta_3\tilde{B}}{\sigma\epsilon} + \frac{\Theta_3\sigma}{\epsilon^3}\big), & \textit{(online)}
\end{cases}
$$

where $\Theta_3 := \tilde{L} + \sqrt{\tilde{L}^2 + \varrho_3(L_l+\theta G)^2}$ and $\rho_3 > 0$ is parameter-free. The average batch size is defined the same way as in the smooth setting.

Proof. To return an $\sqrt{2}\epsilon$-accurate solution, we require $\frac{11\Delta}{T\eta} \leq \frac{3}{8}\epsilon^2$, which gives

$$
S \geq \frac{88\Delta}{3m\eta\epsilon^2} = \frac{88\Delta}{3m\epsilon^2}\big(4\tilde{L} + 2\sqrt{4\tilde{L}^2 + (L_l+\theta G)^2}\big) = O(\frac{\Theta_3}{m\epsilon^2}).
$$

Hence the IFO complexity is similarly derived as in the smooth setting. □

Chapter 5

Generic acceleration for Riemannian optimization via extrapolation

In contrast to Chapters 3 and 4 that focus on particular manifold (i.e., SPD manifold in Chapter 3) or problem types (i.e., finite-sum/online problems in Chapter 4), this chapter considers the general optimization problem on Riemannian manifolds

$$\min_{x \in \mathcal{M}} f(x), \tag{5.1}$$

where $\mathcal{M}$ is a Riemannian manifold. Optimization problem of the form (5.1) naturally appears in various fields of applications and is more general than the one considered in Chapter 4. Examples include principal component analysis (Edelman et al., 1998; H. Zhang et al., 2016), matrix completion and factorization (Keshavan & Oh, 2009; Vandereycken, 2013; Boumal & Absil, 2015; P. Jawanpuria & Mishra, 2018), dictionary learning (Cherian & Sra, 2016; M. Harandi et al., 2013), cross-lingual translation (P. Jawanpuria et al., 2020a,b, 2021), and optimal transport (Shi et al., 2021; P. Jawanpuria et al., 2021; Mishra et al., 2021; Han et al., 2022), etc.

The Riemannian gradient descent method[1] (Udriste, 1994; H. Zhang & Sra,

[1] Here we use Riemannian gradient descent rather than steepest descent to emphasize the fixed stepsize as opposed to variable stepsize computed by line-search in the previous chapters.

2016; Absil et al., 2009; Boumal, 2023) has been considered as the default solver for problem (5.1). Existing works have also explored generalizing Nesterov acceleration (Y. E. Nesterov, 1983) to Riemannian manifolds, including (Y. Liu et al., 2017; Ahn & Sra, 2020; H. Zhang & Sra, 2018a; Alimisis et al., 2020; Jin & Sra, 2022; Kim & Yang, 2022; Criscitiello & Boumal, 2022a). However, they primarily involve exponential map, inverse exponential map, and parallel transport, which are computationally expensive operations. In addition, the Nesterov acceleration based methods require the knowledge of smoothness and strong convexity constants, which are often unknown. Furthermore, recent studies (Hamilton & Moitra, 2021; Criscitiello & Boumal, 2022b) show that global acceleration cannot be achieved on manifolds in general. In particular, they claim for Hadamard manifolds with strictly negative curvature, such as hyperbolic and positive definite manifolds, acceleration is impossible for smooth, geodesic strongly convex functions, when domain expands.

In this chapter, we focus on an extrapolation based strategy to produce an accelerated sequence. The core idea is to compute extrapolation as a linear combination of the iterates where the weights depend nonlinearly on the iterates. Existing works (Aitken, 1927; Shanks, 1955; Brezinski et al., 2018; Wynn, 1956; Sidi et al., 1986; Walker & Ni, 2011; Scieur et al., 2020) have explored such strategy in the Euclidean setting. Recently, it has been shown in Scieur et al. (2020) that such nonlinear acceleration (Euclidean) scheme achieves optimal convergence rates asymptotically without knowing the function-specific constants.

A natural question is *can such extrapolation idea be generalized to Riemannian manifolds so that we achieve acceleration?* The nonlinear structure of manifolds imposes key technical challenges such as averaging on manifolds, distortion due to varying metric, computationally expensive operations, like exponential map and parallel transport, to name a few. Nevertheless, we answer the above question affirmatively and our contributions of this chapter are as follows.

- We propose an acceleration strategy for Riemannian optimization based on the idea of extrapolation, which we call the Riemannian nonlinear acceleration (RiemNA) strategy. We analyze several averaging schemes that generalize weighted averaging in the Euclidean space from various perspectives.

- When the iterates are generated by the Riemannian gradient descent method, we show RiemNA achieves the optimal asymptotic first-order convergence rate. We show the convergence is robust to the choice of different averaging schemes on manifolds.
- A salient feature is that convergence of RiemNA holds under general retraction and vector transport. This is in contrast to existing analyses for Riemannian accelerated gradient methods which employ exponential map and parallel transport (Y. Liu et al., 2017; H. Zhang & Sra, 2018a; Ahn & Sra, 2020; Kim & Yang, 2022).
- We empirically demonstrate the superiority of RiemNA over state-of-the-art methods both in terms of convergence speed and computational efficiency.

5.1 Preliminaries on metric distortion and related works

Metric distortion. Building on the preliminary chapter, i.e., Chapter 2, we first introduce several results bounding the metric distortion, which are essential for the subsequent analysis. Due to the curved geometry of Riemannian manifolds, many of the metric properties in the linear space are lost. The following geometric lemmas on manifolds provide standard bounds on the metric distortion.

Lemma 5.1 (Ahn & Sra (2020); Sun et al. (2019)). *Consider a compact subset $\mathcal{X} \subseteq \mathcal{M}$ with unique geodesic. Let $x, y = \mathrm{Exp}_x(u) \in \mathcal{X}$ for some $u \in T_x\mathcal{M}$. Then for any $v \in T_x\mathcal{M}$, we have $d(\mathrm{Exp}_x(u+v), \mathrm{Exp}_y(\Gamma_x^y v)) \leq \min\{\|u\|, \|v\|\} C_\kappa(\|u\| + \|v\|)$, where $\mathcal{X}$ has curvature upper bounded by κ in magnitude and $C_\kappa(r) := \cosh(\sqrt{\kappa} r) -$*

$\sinh(\sqrt{\kappa}r)/(\sqrt{\kappa}r)$.

Lemma 5.2 (Ahn & Sra (2020); Karcher (1977); Mangoubi & Smith (2018); Sun et al. (2019))**.** *For a compact subset $\mathcal{X} \subseteq \mathcal{M}$ with unique geodesic, there exists constants $C_0 > 0$, $C_1, C_2 \geq 1$ that depend on the curvature and diameter of $\mathcal{X}$ such that for all $x, y, z \in \mathcal{X}$, $u \in T_x\mathcal{M}$ we have*

(1). $\|\Gamma_y^z \Gamma_x^y u - \Gamma_x^z u\|_z \leq C_0 d(x,y) d(y,z) \|u\|_x$.

(2). $C_1^{-1} d(x,y) \leq \|\mathrm{Exp}_z^{-1}(x) - \mathrm{Exp}_z^{-1}(y)\|_z \leq C_2 d(x,y)$.

(3). $d\big(\mathrm{Exp}_x(u), \mathrm{Exp}_y(\Gamma_x^y u)\big) \leq C_3 d(x,y)$.

Related works on Riemannian acceleration. Generalizing Nesterov acceleration strategy (Y. E. Nesterov, 1983) from the Euclidean space to Riemannian manifolds for geodesic (strongly) convex functions has been explored in Y. Liu et al. (2017); H. Zhang & Sra (2018a); Ahn & Sra (2020); Kim & Yang (2022); Jin & Sra (2022). Works such as Alimisis et al. (2020); Duruisseaux & Leok (2022) have approached acceleration on manifolds inspired by the continuous dynamics formulation of the Nesterov acceleration in the Euclidean space (Su et al., 2014; Wibisono et al., 2016). Lastly, acceleration has also been studied for specific manifolds, including sphere and hyperbolic manifolds (Martínez-Rubio, 2022) and the Stiefel manifold (Siegel, 2019).

In the next section, we explore Riemannian nonlinear acceleration based on an extrapolation strategy for iterates generated from a Riemannian solver. Since our convergence analysis is local, the contributions can benefit both geodesic (strongly) convex functions and many nonconvex functions. Further, our convergence rates hold beyond the use of exponential map and parallel transport, which are the primary focus of the aforementioned works.

5.2 Riemannian nonlinear acceleration

We generalize the nonlinear acceleration strategy for Riemannian optimization via a weighted Riemannian averaging on the manifold. For a set of weights $\{c_i\}_{i=0}^k$ and points $\{x_i\}_{i=0}^k$ on the manifold, we define the weighted Riemannian average $\bar{x}_{c,x}$ as

$$\bar{x}_{c,x} = \tilde{x}_k, \qquad \tilde{x}_i = \mathrm{Exp}_{\tilde{x}_{i-1}}\Big(\frac{c_i}{\sum_{j=0}^i c_j}\mathrm{Exp}_{\tilde{x}_{i-1}}^{-1}(x_i)\Big), \tag{Avg.1}$$

for $i = 0, ..., k$ and $\tilde{x}_{-1} = x_0$. When $\mathcal{M}$ is the Euclidean space, (Avg.1) recovers the weighted mean as $\bar{x}_{c,x} = \sum_{i=0}^k c_i x_i$ (see Lemma 5.11 in Appendix 5.C).

The coefficients $\{c_i\}_{i=0}^k$ are determined by minimizing a weighted combination of the residuals $\mathrm{Exp}_{x_i}^{-1}(x_{i+1}) \in T_{x_i}\mathcal{M}, i = 0, ..., k$. Specifically, we consider the following optimization problem:

$$\min_{c \in \mathbb{R}^{k+1}: c^\top 1 = 1} \| \sum_{i=0}^k c_i r_i \|_{x_k}^2 + \lambda \|c\|_2^2, \tag{5.2}$$

which is a linear system of dimension $k+1$ and has a simple closed-form solution (see Proposition 5.2 in Appendix 5.D). Here, $r_i = \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) \in T_{x_k}\mathcal{M}$ and $\Gamma_{x_i}^{x_k}$ is the parallel transport from x_i to x_k.

Our Riemannian nonlinear acceleration (RiemNA) strategy is presented in Algorithm 6, which takes a sequence of non-diverging iterates from any solver as input and constructs an extrapolation using coefficients $\{c_i\}_{i=0}^k$ that solve (5.2). The extrapolation is performed in parallel to the update of the iterate sequence. Note that when the manifold $\mathcal{M}$ is the Euclidean space, Algorithm 6 exactly recovers the nonlinear acceleration algorithm in Scieur et al. (2020).

Algorithm 6: Riemannian nonlinear acceleration (RiemNA)

```
**Input:** Iterate sequence $x_0, ..., x_{k+1}$. Regularization parameter $\lambda$.
Compute $r_i = \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1}), i = 0, ..., k$.
Solve $c^* = \arg\min_{c \in \mathbb{R}^{k+1}: c^\top 1 = 1} \| \sum_{i=0}^{k} c_i r_i \|_{x_k}^2 + \lambda \|c\|_2^2$.
**Output:** $\tilde{x}_{c^*,x} = \tilde{x}_k$ computed from $\tilde{x}_i = \mathrm{Exp}_{\tilde{x}_{i-1}}\Big(\frac{c_i^*}{\sum_{j=0}^{i} c_j^*} \mathrm{Exp}_{\tilde{x}_{i-1}}^{-1}(x_i)\Big)$, with $\tilde{x}_{-1} = x_0$.
```

5.3 Convergence acceleration for Riemannian gradient descent

This section analyzes the convergence acceleration of RiemNA (Algorithm 6) when the iterates are generated by the Riemannian gradient descent (RGD) method (Absil et al., 2009). In particular, we show that the extrapolated point (output of Algorithm 6) is a good estimate of the optimal solution and bound its distance to optimality. We start by making the following assumption throughout the chapter.

Assumption 5.1. Let $x^* \in \mathcal{M}$ be a (strictly) local minimizer of f. The iterates generated, i.e., $x_0, x_1, \ldots$ stay within a neighbourhood $\mathcal{X}$ around x^* with unique geodesic. Furthermore, the sequence of iterates is non-divergent, i.e., $d(x_k, x^*) = O(d(x_0, x^*))$ for all $k \geq 0$.

The former condition in Assumption 5.1 ensures the exponential map is invertible and is standard for analyzing accelerated gradient methods on manifolds (Ahn & Sra, 2020; Jin & Sra, 2022; Kim & Yang, 2022). This condition is satisfied for any non-positively curved manifolds, such as symmetric positive definite (SPD) manifold with the affine-invariant metric (Bhatia, 2009). In addition, this also holds true for any sufficiently small subset $\mathcal{X}$ of any manifold.

Linear iterates and error decomposition in the Euclidean space. First, we recall that the convergence analysis for *Euclidean* nonlinear acceleration (Scieur et al., 2020) relies critically on a sequence of linear fixed-point iterates that satisfy

$\hat{x}_i - x^* = G(\hat{x}_{i-1} - x^*)$ for some positive semi-definite and contractive matrix G (with $\|G\|_2 < 1$). The main idea is to show that the algorithm converges optimally on $\hat{x}_i$ and then bound the deviation arising from the nonlinearity. In particular, let $\{x_i\}_{i=0}^k$ be the given iterates and $\{\hat{x}_i\}_{i=0}^k$ be the linear iterates defined above. Consider $c^*, \hat{c}^*$ as the coefficients solving (5.2) in the Euclidean setup using $\{x_i\}_{i=0}^k$, $\{\hat{x}_i\}_{i=0}^k$ respectively. The convergence analysis in Scieur et al. (2020) aims to bound each term from the error decomposition:

$$\sum_{i=0}^k c_i^* x_i - x^* = \underbrace{\sum_{i=0}^k \hat{c}_i^* \hat{x}_i - x^*}_{\text{Linear term}} + \underbrace{\sum_{i=0}^k (c_i^* - \hat{c}_i^*)\hat{x}_i}_{\text{Stability}} + \underbrace{\sum_{i=0}^k c_i^* (x_i - \hat{x}_i)}_{\text{Nonlinearity}} . \tag{5.3}$$

From linearized iterates to iterates on manifolds. On general Riemannian manifolds, due to the curved geometry, it becomes nontrivial to generalize the error decomposition (5.3) to manifolds. Nevertheless, we start with identification of linearized iterates on manifolds in the tangent space of x^*. For notational convenience, we denote $\Delta_x := \mathrm{Exp}_{x^*}^{-1}(x)$ for any $x \in \mathcal{X}$. We now consider the linearized iterates $\hat{x}_i$ produced by the following progression as

$$\Delta_{\hat{x}_i} = G[\Delta_{\hat{x}_{i-1}}], \tag{5.4}$$

for some $G : T_{x^*}\mathcal{M} \to T_{x^*}\mathcal{M}$ as a self-adjoint, positive semi-definite operator with $\|G\|_{x^*} \leq \sigma < 1$, where we denote $\|A\|_{x^*}$ as the operator norm for any linear operator A on the tangent space $T_{x^*}\mathcal{M}$. In fact, we show in Lemma 5.3 that the progression of iterates from the Riemannian gradient descent method is locally linear on the tangent space of the local minimizer x^*, thus satisfying (5.4) up to some error term. This requires the following regularity assumption on the objective function f.

Assumption 5.2. The function f has geodesic Lipschitz gradient and Lipschitz Hessian.

Remark 5.1. Assumption 5.2 is used to ensure sufficient smoothness of the function such that the Riemannian gradient and Hessian are bounded at optimality.

Lemma 5.3. *Under Assumptions 5.1 and 5.2, suppose the iterates generated by the Riemannian gradient descent method are* $x_{i+1} = \mathrm{Exp}_{x_i}(-\eta\, \mathrm{grad} f(x_i))$*. Then, we have*

$$\Delta_{x_i} = \big(\mathrm{id} - \eta\, \mathrm{Hess} f(x^*)\big)[\Delta_{x_{i-1}}] + \varepsilon_i$$

where id *denotes the identity operator and* $\|\varepsilon_i\|_{x^*} = O(d^2(x_i, x^*))$ *and* $\varepsilon_0 = 0$*.*

Lemma 5.3 suggests that it is reasonable to consider the linearized iterates $\{\hat{x}_k\}$ defined in (5.4) where $G = \mathrm{id} - \eta\, \mathrm{Hess} f(x^*)$. It is clear that for a strictly local minimizer x^*, there exist $\mu, L > 0$ such that $\mu\, \mathrm{id} \preceq \mathrm{Hess} f(x^*) \preceq L\, \mathrm{id}$. This is irrespective of whether the function f is geodesic strongly convex or has geodesic Lipschitz gradient. Thus, for proper choices of η, we can always ensure G is positive semi-definite and contractive.

In this chapter, the convergence analysis focuses on the case when $G = \mathrm{id} - \eta\, \mathrm{Hess} f(x^*)$ and $\{x_i\}$ are given by Riemannian gradient descent to simplify the bounds. However, we highlight that most of the analysis holds for more general and symmetric G.

Hence, the error in the manifold weighted average $\bar{x}_{c^*,x}$ computed from (Avg.1) leads to the decomposition (due to triangle inequality of Riemannian distance):

$$d(\bar{x}_{c^*,x}, x^*) \leq \underbrace{d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*)}_{\text{Linear term}} + \underbrace{d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}})}_{\text{Stability}} + \underbrace{d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x})}_{\text{Nonlinearity}},$$

where we denote $\hat{c}^*$ as the coefficients solving (5.2) with the residuals $\hat{r}_i = \Delta_{\hat{x}_{i+1}} - \Delta_{\hat{x}_i}$ from the linearized iterates $\{\hat{x}_i\}$ in (5.4) and $\bar{x}_{\hat{c}^*,\hat{x}}$, $\bar{x}_{c^*,\hat{x}}$ as weighted average computed using pairs $\{(\hat{c}_i^*, \hat{x}_i)\}_{i=0}^k$ and $\{(c_i^*, \hat{x}_i)\}_{i=0}^k$ respectively. Before we bound each of the error term, we first present a lemma relating the averaging on manifolds to averaging on the tangent space.

Lemma 5.4. *Under Assumption 5.1, for some coefficients $\{c_i\}_{i=0}^k$ with $\sum_{i=0}^k c_i = 1$ and any iterate sequence $\{x_i\}_{i=0}^k$, consider $\bar{x}_{c,x}$ computed from* (Avg.1) *via the given coefficients and the iterates. Then, we have $\Delta_{\bar{x}_{c,x}} = \sum_{i=0}^k c_i \Delta_{x_i} + e$, where $\|e\|_{x^*} = O(d^3(x_0, x^*))$.*

Remark 5.2. Lemma 5.4 shows that the error between the averaging on the manifold and averaging on the tangent space is on the order of $O(d^3(x_0, x^*))$. This relies heavily on the metric distortion bound given in Lemmas 5.1 and 5.2, which only holds for the case of exponential map and parallel transport. Nevertheless, we highlight that when the general retraction and vector transport are used, we can follow the idea of (Tripuraneni et al., 2018, Lemma 12) to show the error is on the order of $O(d^2(x_0, x^*))$. Please see Proposition 5.3 in Appendix 5.E and Section 5.5 for more details where we discuss convergence under a more general setup.

Error bound from the linear term. We show that extrapolation using the linearized iterates converges in a near-optimal rate, via the regularized Chebyshev polynomial. This generalizes the development of Scieur et al. (2020) (in the Euclidean setting) to manifolds.

Definition 5.1 (Regularized Chebyshev polynomial (Scieur et al., 2020))**.** The regularized Chebyshev polynomial of degree k, in the range of $[0, \sigma]$ with a regularization parameter α, denoted as $C_{k,\alpha}^{[0,\sigma]}(x)$ is defined as

$$C_{k,\alpha}^{[0,\sigma]}(x) = \arg\min_{p \in \mathcal{P}_k^1} \max_{x \in [0,\sigma]} p^2(x) + \alpha \|p\|_2^2$$

where we denote $\mathcal{P}_k^1 := \{p \in \mathbb{R}[x] : \deg(p) = k, p(1) = 1\}$ as the set of polynomials of degree k with coefficients summing to 1 and $\|p\|_2$ is the Euclidean norm of the coefficients of the polynomial p. We write the maximum valued as

$S_{k,\alpha}^{[0,\sigma]} := \sqrt{\max_{x\in[0,\sigma]}(C_{k,\alpha}^{[0,\sigma]}(x))^2 + \alpha\|C_{k,\alpha}^{[0,\sigma]}(x)\|_2^2}$.

In Lemma 5.5, we present the error bound coming from the linear term, which follows from the definition of regularized Chebyshev polynomial and Lemma 5.4. Due to the curvature of the manifold, we observe an additional error term $O(d^3(x_0, x^*))$ compared to the Euclidean counterpart, which becomes insignificant as approaching optimality.

Lemma 5.5 (Error from the linear term). *Under Assumption 5.1, let $\bar{x}_{\hat{c}^*,\hat{x}}$ be computed from* (Avg.1) *using $\{(\hat{c}_i^*, \hat{x}_i)\}_{i=0}^k$. Then we can bound*

$$d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*) \leq \frac{d(x_0, x^*)}{1-\sigma}\sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{d^2(x_0, x^*)}\|\hat{c}^*\|_2^2} + \epsilon_1,$$

where $\bar{\lambda} = \lambda/d^2(x_0, x^*)$ and $\epsilon_1 = O(d^3(x_0, x^*))$.

Error bound from coefficient stability. We now bound the deviation between the optimal coefficients computed via the Riemannian gradient descent iterates $\{x_i\}$ and the linearized iterates $\{\hat{x}_i\}$. To this end, we require the following result on the coefficients.

Lemma 5.6 (Bound on norm of coefficients). *Under Assumptions 5.1 and 5.2, let the coefficients $c^*, \hat{c}^*$ be solved from* (5.2) *using $\{x_i\}, \{\hat{x}_i\}$ respectively, where $\{x_i\}$ are given by the Riemannian gradient descent and $\{\hat{x}_i\}$ satisfy* (5.4)*. Then, we have $\|c^*\|_2 \leq \sqrt{\frac{\sum_{i=0}^k d^2(x_i,x_{i+1})+\lambda}{(k+1)\lambda}}$ and $\|c^* - \hat{c}^*\|_2 \leq \frac{1}{\lambda}\left(\frac{2d(x_0,x^*)}{1-\sigma}\psi + (\psi)^2\right)\|\hat{c}^*\|_2$ for some $\psi = O(d^2(x_0, x^*))$.*

It should be noted that in the Euclidean space, $\psi = \sum_{i=0}^k \|\Delta_{x_i} - \Delta_{\hat{x}_i}\|_2 = \|x_i - \hat{x}_i\|_2$ and also can be shown to have an order of $O(d^2(x_0, x^*))$ under certain Lipschitz conditions on the function (Scieur et al., 2020, Proposition 3.8). On manifolds, the term ψ suffers from additional distortion coming from the metric, which is also on the order $O(d^2(x_0, x^*))$.

Based on Lemma 5.6, the error from coefficient stability can now be bounded as follows. The proof follows from linearizing the weighted average on the tangent space $T_{x^*}\mathcal{M}$ where we bound the deviation arising from the coefficients. Hence, an extra error ϵ_2 appears in the bound.

Lemma 5.7 (Error from coefficient estimation). *Under the same settings as in Lemma 5.6, let $\bar{x}_{\hat{c}^*,\hat{x}}$, $\bar{x}_{c^*,\hat{x}}$ be computed from* (Avg.1) *using $\{(\hat{c}_i^*, \hat{x}_i)\}_{i=0}^k$ and $\{(c_i^*, \hat{x}_i)\}_{i=0}^k$ respectively. Then,*

$$d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}}) \leq \frac{C_1}{\lambda(1-\sigma)}\Big(\frac{2d^2(x_0,x^*)}{1-\sigma}\psi + d(x_0,x^*)(\psi)^2\Big)\|\hat{c}^*\|_2 + \epsilon_2,$$

for some $\psi = O(d^2(x_0,x^*)), \epsilon_2 = O(d^3(x_0,x^*))$.

Error bound from nonlinearity. Next, we show that the nonlinearity term can be bounded in Lemma 5.8, which follows a similar idea of linearization on a fixed tangent space. Additional error ϵ_3 is again due to the curvature of the manifold, which vanishes when $\mathcal{M}$ is the Euclidean space.

Lemma 5.8 (Error from the nonlinearity). *Under the same settings as in Lemma 5.6, we have*

$$d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x}) \leq C_1\sqrt{\frac{\sum_{i=0}^k d^2(x_i,x_{i+1}) + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^k\sum_{j=0}^i \|\varepsilon_j\|_{x^*}\Big) + \epsilon_3,$$

where $\|\varepsilon_j\|_{x^*} = O(d^2(x_j,x^*))$ is defined in Lemma 5.3 and $\epsilon_3 = O(d^3(x_0,x^*))$.

Finally, we combine Lemmas 5.5, 5.7, 5.8 to obtain the following convergence result for Algorithm 6 when the iterates are generated from the Riemannian gradient descent (RGD).

Theorem 5.1 (Convergence of RiemNA with RGD). *Under Assumptions 5.1, 5.2, let $\{x_i\}_{i=0}^k$ be the iterates given by the Riemannian gradient descent method, i.e., $x_{i+1} = \mathrm{Exp}_{x_i}(-\eta\, \mathrm{grad} f(x_i))$ and $\{\hat{x}_i\}_{i=0}^k$ be the linearized iterates satisfying $\Delta_{\hat{x}_i} = G[\Delta_{\hat{x}_{i-1}}]$*

with $G = \mathrm{id} - \eta\, \mathrm{Hess} f(x^*)$ contractive, i.e., $\|G\|_{x^*} \leq \sigma < 1$. Then, Algorithm 6 with regularization parameter λ produces $\bar{x}_{c^*,x^*}$ that satisfies

$$d(\bar{x}_{c^*,x}, x^*) \leq d(x_0, x^*) \frac{S^{[0,\sigma]}_{k,\bar{\lambda}}}{1-\sigma} \sqrt{1 + \frac{C_1^2 d^2(x_0,x^*) \big(\frac{2d(x_0,x^*)}{1-\sigma}\psi + (\psi)^2\big)^2}{\lambda^3}}$$
$$+ C_1 \sqrt{\frac{\sum_{i=0}^k d^2(x_i, x_{i+1}) + \lambda}{(k+1)\lambda}} \Big(\sum_{i=0}^{k} \sum_{j=0}^{i} \|\varepsilon_j\|_{x^*} \Big) + \epsilon_1 + \epsilon_2 + \epsilon_3,$$

where $\psi = O(d^2(x_0, x^*))$, $\epsilon_1, \epsilon_2, \epsilon_3 = O(d^3(x_0, x^*))$ and $\varepsilon_i = O(d^2(x_i, x^*))$ is defined in Lemma 5.3.

We prove that even with additional distortion from the curved geometry of the manifold, the asymptotic optimal convergence is still guaranteed. This is mainly due to the fact that all errors incurred by the metric distortion, i.e., $\epsilon_1, \epsilon_2, \epsilon_3$ are on the order of at least $O(d^2(x_0, x^*))$, which is primarily attributed to Lemma 5.4.

Proposition 5.1 (Asymptotic optimal convergence rate of RiemNA with RGD iterates)**.** *Under the same settings as in Theorem 5.1, set $\lambda = O(d^s(x_0, x^*))$ for $s \in (2, \frac{8}{3})$. Then, $\lim_{d(x_0,x^*) \to 0} \frac{d(\bar{x}_{c^*,x}, x^*)}{d(x_0,x^*)} \leq \frac{1}{1-\sigma} \frac{2}{\beta^{-k} + \beta^k}$, where $\beta = \frac{1-\sqrt{1-\sigma}}{1+\sqrt{1-\sigma}}$.*

Remark 5.3. The asymptotic optimal convergence rate holds as long as $\epsilon_1, \epsilon_2, \epsilon_3$ are on the order of at least $O(d^2(x_0, x^*))$ such that $\lim_{d(x_0,x^*) \to 0} \frac{1}{d(x_0,x^*)}(\epsilon_1 + \epsilon_2 + \epsilon_3) = 0$.

Remark 5.4. Suppose at a (strictly) local minimizer, $0 \prec \mu\, \mathrm{id} \preceq \mathrm{Hess} f(x^*) \preceq L\, \mathrm{id}$. Then, by choosing $\eta = \frac{1}{L}$, we have $\sigma = 1 - \frac{\mu}{L}$. This corresponds to the optimal convergence rate obtained by Nesterov acceleration (Y. Nesterov, 2003) and its Riemannian extensions such as Y. Liu et al. (2017); Ahn & Sra (2020); Kim & Yang (2022) for geodesic strongly convex functions.

Implementation and complexity. Algorithm 7 presents an implementation for the proposed RiemNA strategy when the iterates are given by Riemannian gra-

Algorithm 7: RGD+RiemNA

```
**Input:** Initialization $x_0$, stepsize $\eta$, regularization parameter $\lambda$, and memory depth $m$.
Set $t = 0$.
**while** $t \leq T$ **do**
    **for** $i = 1, ..., m$ **do**
        $x_i = \mathrm{Exp}_{x_{i-1}}(-\eta\, \mathrm{grad} f(x_{i-1}))$.
        $t = t + 1$.
    **end for**
    $r_i = -\eta\, \Gamma_{x_i}^{x_{m-1}} \mathrm{grad} f(x_i),\ i = 0, ..., m-1$.
    Solve $c^* = \arg\min_{c \in \mathbb{R}^m : c^\top 1 = 1} \| \sum_{i=0}^k c_i r_i \|_{x_{m-1}}^2 + \lambda \|c\|_2^2$.
    Set $\bar{x}_{c^*,x} = \tilde{x}_{m-1}$ computed from $\tilde{x}_i = \mathrm{Exp}_{\tilde{x}_{i-1}}(\frac{c_i^*}{\sum_{j=0}^i c_j^*} \mathrm{Exp}_{\tilde{x}_{i-1}}^{-1}(x_i))$, with $\tilde{x}_{-1} = x_0$.
    Restart with $x_0 = \bar{x}_{c^*,x}$.
**end while**
```

dient descent (RGD) method with fixed stepsize. Specifically, we run RGD to produce the iterate sequence $x_0, \ldots, x_{m-1}$, where m is the memory depth. Then, we compute $\bar{x}_{c^*,x}$ with these iterates by Algorithm 6. We then restart Riemannian gradient descent with $x_0 = \bar{x}_{c^*,x}$ for the next epoch. It should be noted that in this case, we do not require the inverse exponential map for computing the residuals.

RGD+RiemNA requires T RGD updates and $\lceil T/m \rceil$ calls to RiemNA. Overall, Algorithm 7 needs $T + \lceil T/m \rceil m$ calls to the exponential map and $\lceil T/m \rceil m$ calls each to the parallel transport and the inverse exponential map operations. This is as efficient as the most practical implementation of the Riemannian Nesterov accelerated gradient methods (H. Zhang & Sra, 2018a; Kim & Yang, 2022) (discussed in Appendix 5.A.2) that require $2T$ calls each to the exponential and inverse exponential map operations.

5.4 Alternative averaging schemes

In this section, we propose alternative averaging schemes on manifolds used for extrapolation. For the iterates obtained from the Riemannian gradient descent method, we show the schemes ensure the same asymptotically optimal convergence rate obtained in Proposition 5.1.

The first scheme we consider is based on the following equality in the Euclidean space for the weighted mean, i.e., $\sum_{i=0}^{k} c_i x_i = x_k - (\sum_{i=0}^{k-1} c_i)(x_k - x_{k-1}) - (\sum_{i=0}^{k-2} c_i)(x_{k-1} - x_{k-2}) - \cdots - c_0(x_1 - x_0)$. Accordingly, let $\theta_i = \sum_{j=0}^{i} c_j, i = 0, ..., k - 1$. We define an alternative weighted averaging as

$$\bar{x}_{c,x} = \mathrm{Exp}_{x_k}\Big(- \sum_{i=0}^{k-1} \theta_i \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1})\Big). \tag{Avg.2}$$

Based on the earlier analysis, to show the convergence of $\bar{x}_{c,x}$ defined in (Avg.2), we only require to show that Lemma 5.4 holds for the new scheme, with an error of order at least $O(d^2(x_0, x^*))$. We formalize this claim in the next lemma and show the error is in fact on the order of $O(d^3(x_0, x^*))$.

Lemma 5.9. *Under Assumption 5.1, for some coefficients $\{c_i\}_{i=0}^k$ with $\sum_{i=0}^k c_i = 1$ and iterates $\{x_i\}_{i=0}^k$, consider $\bar{x}_{c,x} = \mathrm{Exp}_{x_k}\big(- \sum_{i=0}^{k-1} \theta_i \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1})\big)$, $\theta_i = \sum_{j=0}^{i} c_j$. Then, we have $\|\Delta_{\bar{x}_{c,x}} - \sum_{i=0}^{k} c_i \Delta_{x_i}\|_{x^*} = O(d^3(x_0, x^*))$.*

Lemma 5.9 allows convergence under the averaging scheme (Avg.2) to be established exactly following the same steps as before. This is sufficient to show that the same convergence bounds hold, i.e., Theorem 5.1 and Proposition 5.1.

Weighted Fréchet mean. In addition, we discuss the weighted Fréchet mean in Appendix 5.B, which can also be used in place of the two aforementioned averaging schemes. We have provided similar error bounds as in Lemma 5.9 that lead to similar convergence guarantees.

5.5 Convergence under general retraction and vector transport

In this section, we generalize our convergence results for RiemNA with general retraction and vector transport operations. To the best of our knowledge, Riemannian acceleration has not been studied under general retraction and vector transport. To this end, we make the following standard assumptions, which include bounding the deviation between retraction and exponential map as well as between vector transport and parallel transport. In addition, we require the Lipschitz gradient and Hessian to be compatible with retraction and vector transport.

Assumption 5.3. The neighbourhood $\mathcal{X}$ is totally retractive where retraction has a smooth inverse. Function f has retraction Lipschitz gradient and Lipschitz Hessian.

Assumption 5.4. There exists constants $a_0, a_1, a_2, \delta_{a_0,a_1} > 0$ such that for all $x, y, z \in \mathcal{X}$, $\|\mathrm{Retr}_x^{-1}(y)\|_x \leq \delta_{a_0,a_1}$, we have (1) $a_0 d(x,y) \leq \|\mathrm{Retr}_x^{-1}(y)\|_x \leq a_1 d(x,y)$ and (2) $\|\mathrm{Exp}_x^{-1}(z) - \mathrm{Retr}_x^{-1}(z)\|_x \leq a_2 \|\mathrm{Retr}_x^{-1}(z)\|_x^2$.

Assumption 5.5. The vector transport $\mathcal{T}_x^y$ is isometric and there exists a constant $a_3 > 0$ such that for all $x, y \in \mathcal{X}$, $\|\mathcal{T}_x^y u - \Gamma_x^y u\|_y \leq a_3 \|\mathrm{Retr}_x^{-1}(y)\|_x \|u\|_x$.

Assumptions 5.3-5.5 are commonly used for analyzing Riemannian first-order algorithms implemented with retraction and vector transport (Ring & Wirth, 2012; W. Huang, Gallivan, & Absil, 2015; Sato et al., 2019; Kasai et al., 2018b; Han & Gao, 2021).

In this section, we only show convergence under the recursive weighted average computation for extrapolation, i.e.,

$$\bar{x}_{c,x} = \tilde{x}_k, \qquad \tilde{x}_i = \mathrm{Retr}_{\tilde{x}_{i-1}}\Big(\frac{c_i}{\sum_{j=0}^{i} c_j} \mathrm{Retr}_{\tilde{x}_{i-1}}^{-1}(x_i)\Big). \tag{5.5}$$

Similar analysis can be also performed on the alternative two averaging schemes discussed in Section 5.4.

The next theorem shows that asymptotic optimal convergence rate can also be achieved using retraction and vector transport. The proof is similar to the case for exponential map and parallel transport and employs the Assumptions 5.4, 5.5. In particular, both these two assumptions ensure the deviations between retraction and exponential map, vector transport and parallel transport are on the order of $O(d^2(x_0, x^*))$. Thus, the additional error terms $\epsilon_1, \epsilon_2, \epsilon_3 = O(d^2(x_0, x^*))$.

Theorem 5.2 (Convergence under general retraction and vector transport)**.** *Under Assumptions 5.1, 5.3, 5.4, and 5.5, let* $\{x_i\}_{i=0}^k$ *be given by Riemannian gradient descent via retraction, i.e.,* $x_i = \mathrm{Retr}_{x_{i-1}}(-\eta \, \mathrm{grad} f(x_{i-1}))$ *and* $\{\hat{x}_i\}_{i=0}^k$ *be the linearized iterates satisfying* $\mathrm{Retr}_{x^*}^{-1}(\hat{x}_i) = G[\mathrm{Retr}_{x^*}^{-1}(\hat{x}_{i-1})]$ *with* $G = \mathrm{id} - \eta \, \mathrm{Hess} f(x^*)$*, satisfying* $\|G\|_{x^*} \leq \sigma < 1$*. Then, using retraction and vector transport in Algorithm 6 and letting* $\bar{x}_{c,x}$ *be computed from* (5.5)*, the same asymptotic optimal convergence rate (Proposition 5.1) holds under the same choice of* $\lambda = O(d^s(x_0, x^*))$*,* $s \in (2, \frac{8}{3})$*.*

Theorem 5.2 allows Algorithm 7 to be implemented with general retraction and vector transport without affecting the optimal convergence rate achieved asymptotically.

5.6 Experiments

In this section, we evaluate the performance of our Riemannian nonlinear acceleration (RiemNA) strategy on various applications. For RiemNA, we only consider the recursive weighted average in (Avg.1) for the main experiments. The codes are available on https://github.com/andyjm3/RiemNA.

Baselines. We compare the proposed RGD+RiemNA (Algorithm 7) with state-of-the-art Riemannian Nesterov accelerated gradient (RNAG) methods (Kim &

Yang, 2022). We also include RAGD, a variant of Nesterov acceleration on manifolds proposed in H. Zhang & Sra (2018a), and RGD as baselines. In particular, we compare with RNAG-C (Kim & Yang, 2022) (designed for geodesic convex functions) and RNAG-SC (Kim & Yang, 2022) and RAGD (H. Zhang & Sra, 2018a) (designed for geodesic strongly convex functions) regardless of whether the objective is of the particular class. More details of the algorithms are in Appendix 5.A.2.

Parameters. RNAG-C, RNAG-SC, and RAGD require the knowledge of geodesic Lipschitz constant L (Kim & Yang, 2022). Further, RNAG-SC and RAGD require the geodesic strong convexity parameter μ. In particular, the stepsize of RNAG-C, RNAG-SC and RAGD should be set as $1/L$. If such constants are available, we set them accordingly. Otherwise, we tune over the parameters L, μ for RNAG-C, RNAG-SC to obtain the best results and set the same parameters for RAGD for comparability. Following Kim & Yang (2022), the additional parameters ξ, ζ are fixed to be 1 for RNAG-C, RNAG-SC and $\beta = \sqrt{\mu/L}/5$ for RAGD. We set stepsize of RGD to be $1/L$ if available and tune the stepsize otherwise. For the proposed RGD+RiemNA, we fix $\lambda = 10^{-8}$ and choose memory depth $m \in \{5, 10\}$. It should be emphasized that RGD+RiemNA is agnostic to function specific constants.

For fair comparisons, we use exponential map, inverse exponential map, and parallel transport for all the algorithms whenever such operations are properly defined. For other cases, we use retraction, inverse retraction, and vector transport even though the baseline acceleration methods are not analyzed under such general operations. We emphasize that we maintain consistency in the use of these operations across all the algorithms. The experiments are coded in Matlab using Manopt (Boumal et al., 2014). The stopping criterion for all the algorithms is gradient norm reaching below 10^{-6}.

Applications. We consider four applications: leading eigenvector computation (Absil et al., 2007), Fréchet mean of symmetric positive definite (SPD) matrices with the affine-invariant metric (Bhatia, 2009), orthogonal Procrustes problem (Eldén & Park, 1999), and the nonlinear eigenspace problem (Zhao et al., 2015). These applications solve problems on sphere, SPD, Stiefel, and Grassmann manifolds respectively. See Appendix 5.A.1 for detailed introduction of the manifolds, along with the relevant operations required for the experiments. We highlight that except for the task of Fréchet mean which is geodesic strongly convex, other problems are in general nonconvex.

Leading eigenvector computation. The problem computes the leading eigenvector of a symmetric matrix A of size $d \times d$, by solving $\min_{x \in \mathcal{S}^{d-1}} \{ f(x) := -\frac{1}{2} x^\top A x \}$, where $\mathcal{S}^{d-1} := \{ x \in \mathbb{R}^d : \|x\|_2 = 1 \}$ denotes the sphere manifold of intrinsic dimension $d-1$. For the experiment, we generate a positive definite matrix A with condition number 10^3 and exponentially decaying eigenvalues in dimension $d = 10^3$. As shown in (Kim & Yang, 2022, Proposition 7.1), the problem has geodesic L-Lipschitz gradient with L to be the eigengap of matrix A, i.e., the difference between maximum and minimum eigenvalues of A. The optimal solution of the problem is given by $-\frac{1}{2}\lambda_{\max}(A)$, where $\lambda_{\max}$ extracts the largest eigenvalue of A.

The stepsize is thus set as $1/L$ for all methods. For RNAG-SC and RAGD, we set $\mu = 10$. For RiemNA, we set memory depth to be $m = 10$. We use exponential and inverse exponential map as well as projection-type vector transport for all algorithms including RGD+RiemNA.

Fréchet mean of SPD matrices. We consider the problem of computing the Fréchet mean of symmetric positive definite (SPD) matrices $\{A_i\}_{i=1}^N$ of size $d \times d$

under the affine-invariant metric (Bhatia, 2009), i.e.,

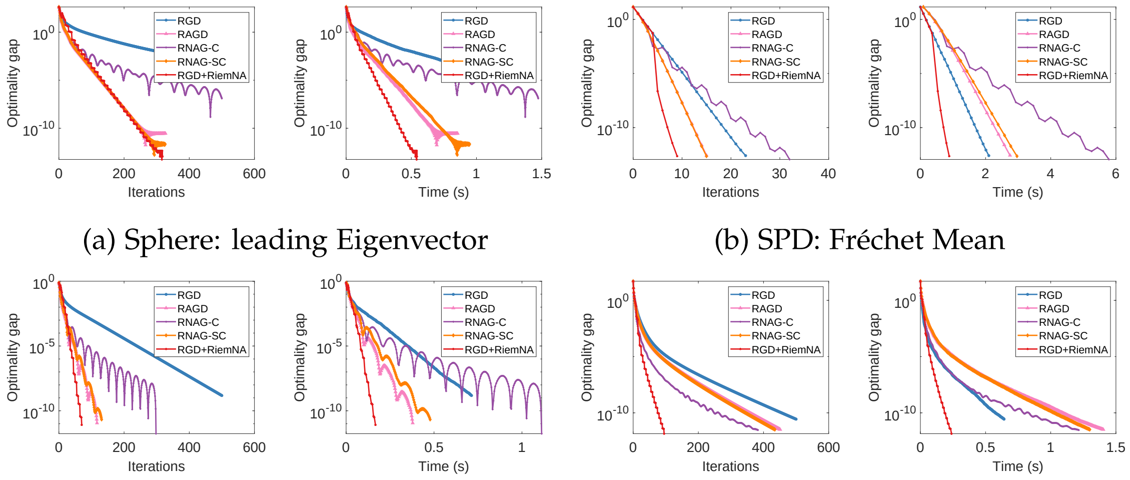
$$\min_{X \in \mathbb{S}^d_{++}} \frac{1}{2N} \sum_{i=1}^{N} \|\mathrm{logm}(X^{-1/2} A_i X^{-1/2})\|_{\mathrm{F}}^2.$$

Here, $\mathbb{S}^d_{++}$ is the set of SPD matrices of size $d \times d$, $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm, and $\mathrm{logm}(\cdot)$ is the matrix logarithm. To trace the optimality gap, we compute the optimal solution of the problem by running R-LBFGS method (W. Huang et al., 2016) until the gradient norm falls below 10^{-10}.

For the experiments, we use exponential map and its inverse as well as the parallel transport for all the algorithms. As commented previously, the geometry is negatively curved, and thus, the Fréchet mean problem is geodesic 1-strongly convex ($\mu = 1$). For this problem, we generate random $N = 100$ SPD matrices of dimension $d = 10$. The stepsize for all methods are tuned and set to be 0.5. For RiemNA, we set memory depth $m = 5$.

Orthogonal Procrustes problem. We also consider the orthogonal Procrustes problem on the Stiefel manifold (Eldén & Park, 1999). Suppose we are given $A \in \mathbb{R}^{r \times r}, B \in \mathbb{R}^{p \times r}$, the objective is $\min_{X \in \mathrm{St}(p,r)} \|XA - B\|_{\mathrm{F}}^2$ where $\mathrm{St}(p,r) := \{X \in \mathbb{R}^{p \times r} : X^\top X = I\}$ is the set of column orthonormal matrices, which forms the so-called Stiefel manifold with the canonical metric. The optimal solution is similarly computed by running R-LBFGS.

To implement the algorithms, we use QR-based retraction and inverse retraction as well as projection-type vector transport. We generate random matrices A, B where the entries are normal distributed. We set $p = 100, r = 5$. For this problem, both L and μ are unknown. Hence, we tune and set stepsize to be 1 for all methods. For RNAG-SC and RAGD, we select $\mu = 0.005$ and for RiemNA we set memory depth $m = 5$.



(a) Sphere: leading Eigenvector

(b) SPD: Fréchet Mean

(c) Stiefel: Orthogonal Procrustes

(d) Grassmann: Nonlinear Eigenspace

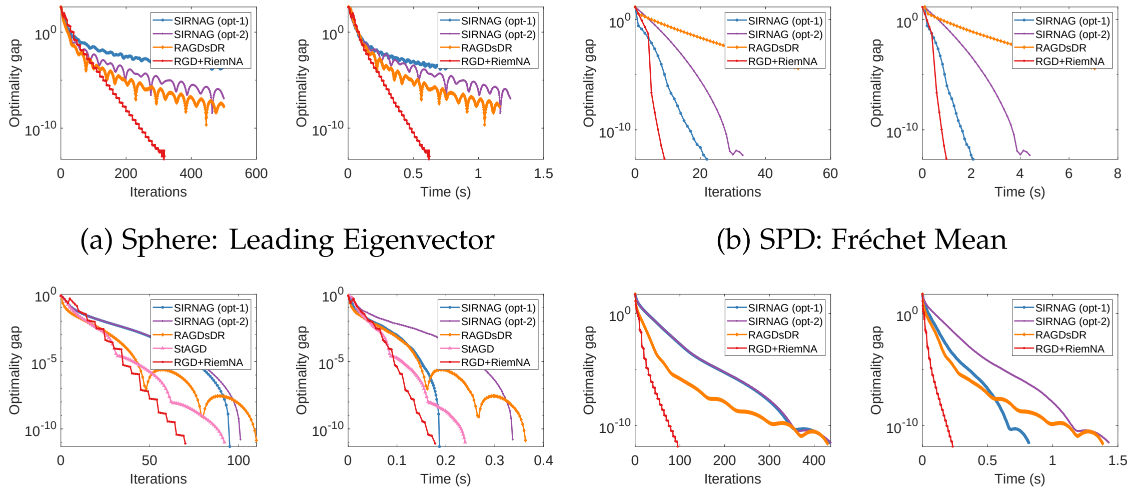
Figure 5.1: Comparing proposed RGD+RiemNA with existing approaches: RGD, RAGD, RNAG-C, and RNAG-SC. We observe that RGD+RiemNA outperforms all the baselines.

Nonlinear eigenspace problem. Finally, the problem of computing nonlinear eigenspace arises as the total energy minimization on the Grassmann manifold (Zhao et al., 2015), i.e., $\min_{X \in \mathrm{Gr}(p,r)} \frac{1}{2}\mathrm{tr}(X^\top L X) + \frac{1}{4}\rho(X)^\top L^{-1}\rho(X)$ where $\rho(X) := \mathrm{diag}(XX^\top)$ and L is a discrete Laplacian operator. The optimal solution is similarly computed by running R-LBFGS.

For experiment, we implement the algorithms with QR-based retraction and inverse retraction as well as projection-based vector transport similar to Stiefel manifold. We generate L as a tridiagonal matrix with main diagonal entries to be 2 and sub- and super-diagonal entries to be -1. The stepsize is tuned and set to be 0.1 for all methods and for RNAG-SC, RAGD, $\mu = 5$ and for RiemNA, $m = 5$.

Results. In Figure 5.1, we plot optimality gap, $f(x_t) - f(x^*)$, against both iteration number and runtime for all the algorithms. We make the following observations:

- Proposed RGD+RiemNA consistently outperforms the baselines in runtime across all the applications.



(a) Sphere: Leading Eigenvector

(b) SPD: Fréchet Mean

(c) Stiefel: Orthogonal Procrustes

(d) Grassmann: Nonlinear Eigenspace

Figure 5.2: Comparing RGD+RiemNA with additional approaches: SIRNAG, RAGDsDR, and StAGD. SIRNAG (opt-1) and (opt-2) represent SIRNAG with two update options. We observe that RGD-RiemNA maintains its superior performance.

- In iteration counts as well, RGD+RiemNA is consistently better than others in all the applications except in the leading eigevector problem, where RGD+RiemNA matches the performance of RAGD and RNAG-SC.
- In Figure 5.1a, RGD+RiemNA is faster than RAGD and RNAG-SC even though the number of iterations needed are similar. This implies that RGD+RiemNA is computationally more efficient. This is in accordance with RGD+RiemNA requiring fewer number of calls to manifold operations like exponential map (or retraction) and parallel transport (or vector transport).
- For the SPD Fréchet mean problem, which is geodesic strongly convex, RGD+RiemNA consistently exhibits faster convergence than others where the extrapolation step leads to significant convergence acceleration.
- RGD+RiemNA does not necessarily ensure descent in the objective for the initial iterations. Only in the later phase the acceleration takes place. This is in accordance with our local convergence analysis.

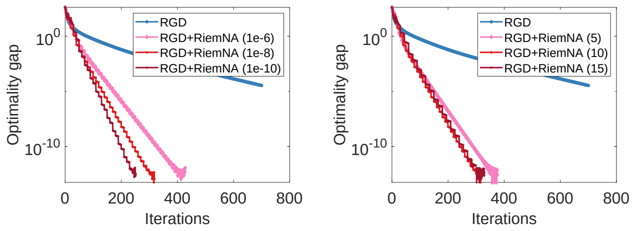


Figure 5.3: Parameter sensitivity on the leading eigenvector problem. Left: we vary λ by fixing $m = 10$. Right: we vary m by fixing $\lambda = 10^{-8}$. Our proposed RGD+RiemNA is robust to parameter changes.

Comparison with additional baselines. We also compare with additional Riemannian acceleration methods in Figure 5.2, including an ODE-based acceleration method SIRNAG (Alimisis et al., 2020), an adaptive momentum-based acceleration method RAGDsDR (Alimisis et al., 2021), and an acceleration method for the Stiefel manifold StAGD (Siegel, 2019). We notice that the curvature parameter $\zeta \geq 1$ is required for both SIRNAG and RAGDsDR, which should be set as 1 if the manifold is positively curved and $\zeta > 1$ when the minimum curvature is negative. For the case of leading eigenvector problem, which is on sphere, manifold of positive curvature, we fix $\zeta = 1$. Otherwise, we first tune ζ for SIRNAG and RAGDsDR. Then the stepsize is tuned accordingly. For StAGD, only the stepsize is tuned. In Figure 5.2, we observe that RGD+RiemNA outperforms the above baselines as well. Even in the Stiefel case, our general RGD+RiemNA is faster than the specialized acceleration method StAGD.

Ablation studies. In Figure 5.3, we test the sensitivity of RGD+RiemNA to the choices of regularization parameter λ (on the left with $m = 10$ fixed) and memory depth m (on the right with $\lambda = 10^{-8}$ fixed). The results show robustness of RiemNA under various choices of regularization parameter λ and memory depth m. Additionally, in Appendix 5.A.3, we also test on alternative averaging schemes where we show that RGD+RiemNA with (Avg.2) performs very similar to the strategy (Avg.1).

5.7 Discussions

In this chapter, we introduce a scheme for accelerating first-order Riemannian optimization algorithms, based on the idea of iterate extrapolation on the manifolds. The extrapolation step is performed via novel intrinsic weighted averaging schemes on manifolds. We show that Riemannian acceleration achieves convergence with asymptotically optimal rates irrespective of function classes. We also show our analysis holds with computationally cheap retraction and vector transport operations. Empirically, we see superior performance of the proposed algorithm RGD+RiemNA against many state-of-the-art Riemannian acceleration algorithms.

Even though the convergence analysis of our proposed acceleration scheme is asymptotic, we empirically observe its good performance against the baselines. It thus raises the question whether non-asymptotic convergence rates can be established. While we have focused on analyzing the RGD, it is also interesting to see whether such an acceleration scheme can be applied to other algorithm classes, such as momentum-based algorithms. An equally rewarding direction is to analyze acceleration in the stochastic settings where gradient information is corrupted by noise.

Appendices

The appendix sections are organized as follows. In Section 5.A, we include detailed introduction of the manifolds considered in the chapter, and of the baseline acceleration methods on Riemannian manifolds. We also include additional experiment results to consolidate the findings in the main text. Section 5.C shows how weighted averaging in the Euclidean space can be computed recursively, which serves as the main motivation for the introduction of averaging schemes on manifolds. Section 5.D presents the proofs for the case of exponential map and parallel transport and Section 5.E proves for the general retraction and vector transport. Section 5.F introduces several extensions to the current algorithm, including the use of line-search schemes and globalization techniques.

5.A Experiment details and additional experiments

5.A.1 Geometry of specific Riemannian manifolds

Sphere manifold. The sphere manifold $\mathcal{S}^{d-1}$ is an embedded submanifold of $\mathbb{R}^d$ with the tangent space identified as $T_x\mathcal{S}^{d-1} = \{u \in \mathbb{R}^d : x^\top u = 0\}$. The Riemannian metric is given by $\langle u, v \rangle = \langle u, v \rangle_2$ for $u, v \in T_x\mathcal{S}^{d-1}$. We use the exponential map derived as $\mathrm{Exp}_x(u) = \cos(\|u\|_2)x + \sin(\|u\|_2)\frac{u}{\|u\|_2}$ and the inverse exponential map as $\mathrm{Exp}_x^{-1}(y) = \arccos(x^\top y)\frac{\mathrm{Proj}_x(y-x)}{\|\mathrm{Proj}_x(y-x)\|_2}$ where $\mathrm{Proj}_x(v) = v - (x^\top v)x$ is the orthogonal projection of any $v \in \mathbb{R}^d$ to the tangent space $T_x\mathcal{S}^{d-1}$. The vector transport is given by the projection operation, i.e., $\mathcal{T}_x^y u = \mathrm{Proj}_y(u)$.

Symmetric positive definite (SPD) manifold. The SPD manifold of dimension d is denoted as $\mathbb{S}^d_{++} := \{X \in \mathbb{R}^{d\times d} : X^\top = X, X \succ 0\}$. The tangent space $T_X\mathcal{M}$ is the set of symmetric matrices. The affine-invariant Riemannian metric is given by $\langle U, V\rangle_X = \mathrm{tr}(X^{-1}UX^{-1}V)$ for any $U, V \in T_X\mathbb{S}^d_{++}$. We make use of the exponential map, which is $\mathrm{Exp}_X(U) = X\mathrm{expm}(X^{-1}U)$ where $\mathrm{expm}(\cdot)$ is the matrix exponential. The inverse exponential map is derived as $\mathrm{Exp}_X^{-1}(Y) = X\mathrm{logm}(X^{-1}Y)$ for any $X, Y \in \mathbb{S}^d_{++}$. We consider the parallel transport given by $\Gamma_X^Y U = EUE^\top$ with $E = (YX^{-1})^{1/2}$.

Stiefel manifold. The Stiefel manifold of dimension $p \times r$ is written as $\mathrm{St}(p, r) := \{X \in \mathbb{R}^{p\times r} : X^\top X = I\}$. The Riemannian metric is the Euclidean inner product defined as $\langle U, V\rangle_X = \langle U, V\rangle_2$. We consider the QR-based retraction $\mathrm{Retr}_X(U) = \mathrm{qf}(X + U)$ where $\mathrm{qf}(\cdot)$ returns the Q-factor from the QR decomposition. The inverse retraction is derived as for $X, Y \in \mathcal{O}(d)$ $\mathrm{Retr}_X^{-1}(Y) = YR - X$, where R is solved from the system $X^\top YR + R^\top Y^\top X = 2I$. The vector transport is given by the orthogonal projection, which is $\mathcal{T}_X^Y = U - Y\{Y^\top U\}_S$ where $\{A\}_S := (A + A^\top)/2$.

Grassmann manifold. The Grassmann manifold of dimension $p \times r$, denoted as $\mathrm{Gr}(p, r)$, is the set of all r dimensional subspaces in $\mathbb{R}^p$ $(p \geq r)$. Each point on the Grassmann manifold can be identified as a column orthonormal matrices $X \in \mathbb{R}^{p\times r}, X^\top X = I$ and two points $X, Y \in \mathrm{Gr}(p, r)$ are equivalent if $X = YO$ for some $O \in \mathcal{O}(r)$, the $r \times r$ orthogonal matrix. Hence Grassmann manifold is a quotient manifold of the Stiefel manifold. We consider the popular QR-based retraction, i.e. $R_X(U) = \mathrm{qf}(X + U)$ where for simplicity, we let X to represent the equivalence class and U represents the horizontal lift of the tangent vector. The inverse retraction is also based on QR factorization, i.e. $R_X^{-1}(Y) = Y(X^\top Y)^{-1} - X$. Vector transport is $\mathcal{T}_X^Y U = U - XX^\top U$.

Algorithm 8: RAGD (H. Zhang & Sra, 2018a)

```
Input: Initialization $x_0$, parameter $\beta > 0$, stepsize $h \leq \frac{1}{L}$, strong convexity parameter $\mu > 0$.
Initialize $v_0 = x_0$.
Set $\alpha = \frac{\sqrt{\beta^2+4(1+\beta)\mu h}-\beta}{2}$, $\gamma = \frac{\sqrt{\beta^2+4(1+\beta)\mu h}-\beta}{\sqrt{\beta^2+4(1+\beta)\mu h}+\beta}\mu$, $\bar{\gamma} = (1+\beta)\gamma$.
for $k = 0, ..., K-1$ do
    Compute $\alpha_k \in (0,1)$ from the equation $\alpha_k^2 = h_k((1-\alpha_k)\gamma_k + \alpha_k\mu)$.
    $y_k = \mathrm{Exp}_{x_k}(\frac{\alpha\gamma}{\gamma+\alpha\mu}\mathrm{Exp}_{x_k}^{-1}(v_k))$
    $x_{k+1} = \mathrm{Exp}_{y_k}(-h\,\mathrm{grad} f(y_k))$
    $v_{k+1} = \mathrm{Exp}_{y_k}(\frac{(1-\alpha)\gamma}{\bar{\gamma}}\mathrm{Exp}_{y_k}^{-1}(v_k) - \frac{\alpha}{\bar{\gamma}}\mathrm{grad} f(y_k))$
end for
Output: $x_K$
```

5.A.2 Baseline Riemannian acceleration methods

Here, we include the implementation details of the Riemannian Nesterov accelerated gradient methods presented in H. Zhang & Sra (2018a); Kim & Yang (2022); Alimisis et al. (2020, 2021); Siegel (2019). It is worth noting that those algorithms have been analyzed under the exponential map, inverse exponential map, and parallel transport. In contrast, the proposed RGD+RiemNA works with general retraction and vector transport.

We first present the (constant-stepsize) RAGD method in (H. Zhang & Sra, 2018a, Algorithm 2), which is included in Algorithm 8. We see the algorithm requires three times evaluation of the exponential map and two times the inverse exponential map at every iteration.

Below we present RNAG-C (Algorithm 9), which is designed for geodesic convex functions and RNAG-SC (Algorithm 10) which is for geodesic strongly convex functions in Kim & Yang (2022). We observe the algorithms require two times evaluation of the exponential map, inverse exponential map as well as parallel transport.

We also include SIRNAG (Alimisis et al., 2020), RAGDsDR (Alimisis et al., 2021) and StAGD (Siegel, 2019). We have included the detailed steps in Algo-

Algorithm 9: RNAG-C (Kim & Yang, 2022)

```
Input: Initialization $x_0$, parameters $\xi, T > 0$, stepsize $s \leq \frac{1}{L}$.
Initialize $\bar{v}_0 = 0 \in T_{x_0}\mathcal{M}$.
Set $\lambda_k = \frac{k+2\xi+T}{2}$.
for $k = 0, ..., K-1$ do
    $y_k = \mathrm{Exp}_{x_k}(\frac{\xi}{\lambda_k+\xi-1}\bar{v}_k)$
    $x_{k+1} = \mathrm{Exp}_{y_k}(-s\,\mathrm{grad} f(y_k))$
    $v_k = \Gamma_{x_k}^{y_k}(\bar{v}_k - \mathrm{Exp}_{x_k}^{-1}(y_k))$
    $\bar{\bar{v}}_{k+1} = v_k - \frac{s\lambda_k}{\xi}\mathrm{grad} f(y_k)$
    $\bar{v}_{k+1} = \Gamma_{y_k}^{x_{k+1}}(\bar{\bar{v}}_{k+1} - \mathrm{Exp}_{y_k}^{-1}(x_{k+1}))$
end for
Output: $x_K$
```

Algorithm 10: RNAG-SC (Kim & Yang, 2022)

```
Input: Initialization $x_0$, parameter $\xi$, stepsize $s \leq \frac{1}{L}$, strong convexity
    parameter $\mu$.
Set $q = \mu s$.
for $k = 0, ..., K-1$ do
    $y_k = \mathrm{Exp}_{x_k}(\frac{\sqrt{\xi q}}{1+\sqrt{\xi q}}\bar{v}_k)$
    $x_{k+1} = \mathrm{Exp}_{y_k}(-s\,\mathrm{grad} f(y_k))$
    $v_k = \Gamma_{x_k}^{y_k}(\bar{v}_k - \mathrm{Exp}_{x_k}^{-1}(y_k))$
    $\bar{\bar{v}}_{k+1} = (1 - \sqrt{\frac{q}{\xi}})v_k + \sqrt{\frac{q}{\xi}}(-\frac{1}{\mu}\mathrm{grad} f(y_k))$
    $\bar{v}_{k+1} = \Gamma_{y_k}^{x_{k+1}}(\bar{\bar{v}}_{k+1} - \mathrm{Exp}_{y_k}^{-1}(x_{k+1}))$
end for
Output: $x_K$
```

rithm 11 and 12 respectively. Specifically, SIRNAG is the discretization of an ODE on manifolds that achieves acceleration. For the purpose of experiments, we only consider the version for geodesic convex functions. This is because the version for geodesic strongly convex functions only differs in one parameter setting. SIRNAG involves two update options, SIRNAG (opt-1) and SIRNAG (opt-2), which correspond to two strategies of discretization.

RAGDsDR, accelerates the convergence for both geodesic convex and weakly-quasi-convex functions by exploiting momentum. For experiments, we only consider the convex version. We follow the empirical choice of β_k suggested in the

Algorithm 11: SIRNAG (Alimisis et al., 2020)

```
Input: Initialization $x_0$. Integration stepsize $h$. curvature parameter $\zeta$.
for $k = 0, ..., K-1$ do
    $\beta_k = \frac{k-1}{k+2\zeta}$.
    Option I: $a_k = \beta_k v_k - h\,\mathrm{grad} f(x_k)$.
    Option II: $a_k = \beta_k v_k - h\,\mathrm{grad} f\big(\mathrm{Exp}_{x_k}(h\beta_k v_k)\big)$.
    $x_{k+1} = \mathrm{Exp}_{x_k}(h\,a_k)$.
    $v_{k+1} = \Gamma_{x_k}^{x_{k+1}} a_k$.
end for
Output: $x_K$.
```

Algorithm 12: RAGDsDR (Alimisis et al., 2021)

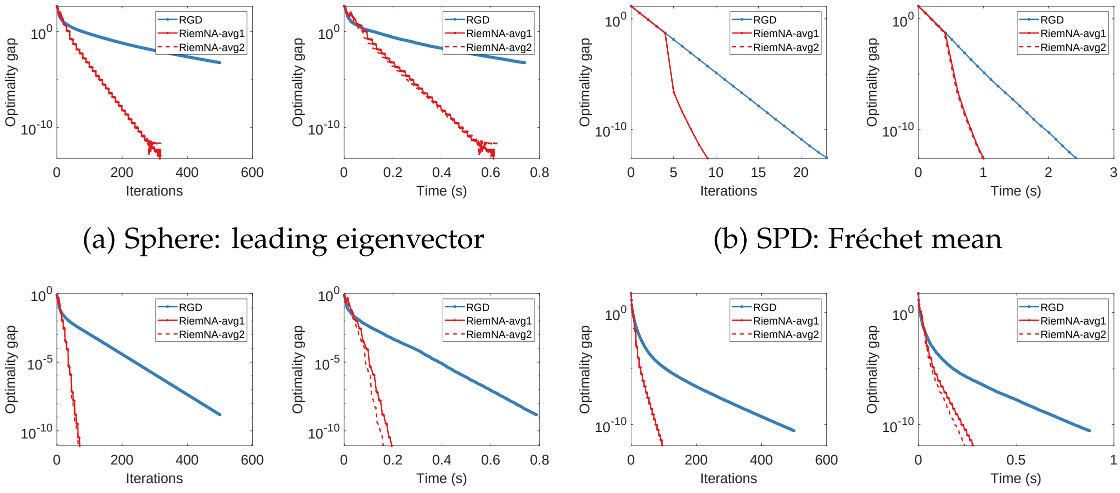
```
Input: Initialization $x_0$. Smoothness parameter $L$. curvature parameter $\zeta$.
$v_0 = x_0$, $A_0 = 0$.
for $k = 0, ..., K-1$ do
    $\beta_k = \frac{k}{k+2}$.
    $y_k = \mathrm{Exp}_{v_k}\big(\beta_k \mathrm{Exp}_{v_k}^{-1}(x_k)\big)$
    $x_{k+1} = \mathrm{Exp}_{y_k}(-\frac{1}{L}\mathrm{grad} f(y_k))$
    Solve $a_{k+1} > 0$ from the equation $\frac{\zeta a_{k+1}^2}{A_k + a_{k+1}} = \frac{1}{L}$.
    $A_{k+1} = A_k + a_{k+1}$.
    $v_{k+1} = \mathrm{Exp}_{v_k}(-a_{k+1}\Gamma_{y_k}^{v_k}\mathrm{grad} f(y_k))$.
end for
Output: $x_K$.
```

paper.

Finally, specifically for the orthogonal Procrustes problem, we include the acceleration method (Siegel, 2019) designed for the Stiefel manifold as another baseline, which we call StAGD. In particular, we implement the version with function restart (Siegel, 2019, Algorithm 4.1) and without using linesearch for comparability. It is worth noticing that Siegel (2019) applies the Cayley-based retraction and canonical Riemannian metric (Edelman et al., 1998) for the implementation.

5.A.3 Ablation study: use of alternative averaging schemes

We next evaluate the numerical performance of RiemNA when using alternative averaging scheme, i.e. (Avg.2). Specifically, the average is given by $\bar{x}_{c,x} = \mathrm{Retr}_{x_k}\big(- \sum_{i=0}^{k-1} \theta_i \Gamma_{x_i}^{x_k} \mathrm{Retr}_{x_i}^{-1}(x_{i+1})\big) = \mathrm{Retr}_{x_k}\big(- \sum_{i=0}^{k-1} \theta_i r_i \big)$ where we use the general retraction. It is worth mentioning that (Avg.2) is more efficient by avoiding k times evaluation of inverse retraction map. We compare the use of two averaging schemes in Figure 5.A.1 where we observe almost identical convergence behaviour when measured against the iteration. For runtime, (Avg.2) can further reduce computational cost compared to (Avg.1), especially for the Stiefel manifold and Grassmann manifold where the inverse retraction is expensive. Even though for SPD manifold, the inverse exponential map is expensive, because the number of iteration to convergence is small, we do not observe a significant reduction in runtime.



(a) Sphere: leading eigenvector

(b) SPD: Fréchet mean

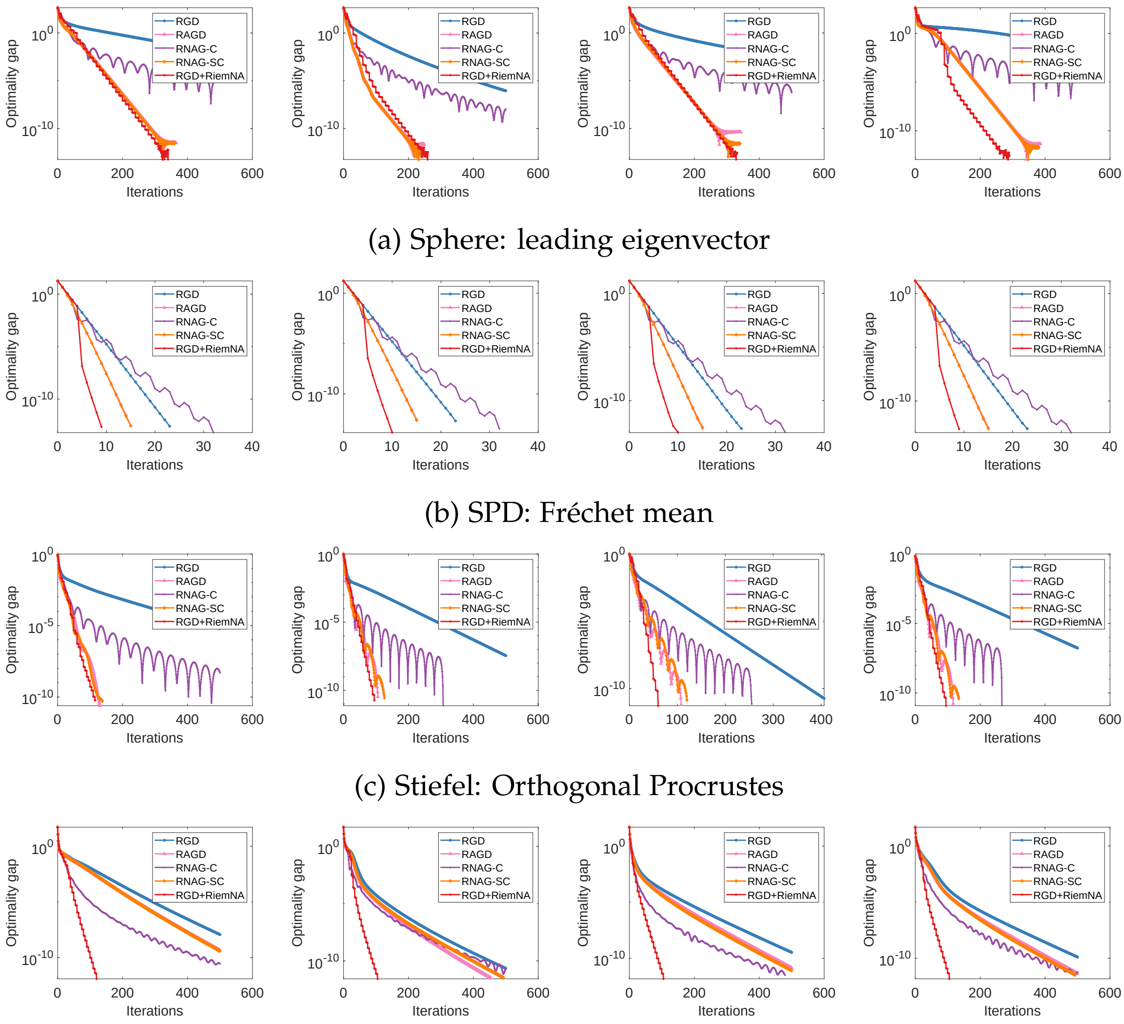
(c) Stiefel: Orthogonal Procrustes

(d) Grassmann: Nonlinear eigenspace

Figure 5.A.1: Comparison of different averaging schemes, i.e., (Avg.1) (used in the main text) and (Avg.2). We observe almost identical convergence in terms of iterations. (Avg.2) is more efficient, particularly for the case Stiefel and Grassmann manifold where the inverse retraction is costly.

5.A.4 Sensitivity to data generation and initialization

Here, we provide additional independent experiment runs to test the model sensitivity to randomness in data generation and initialization. Each column in Figure 5.A.2 corresponds to a run with a fixed random seed. From Figure 5.A.2, we observe the proposed RGD+RiemNA maintains its outperformance against all baselines with good stability.



(a) Sphere: leading eigenvector

(b) SPD: Fréchet mean

(c) Stiefel: Orthogonal Procrustes

(d) Grassmann: Nonlinear eigenspace

Figure 5.A.2: Additional experiment runs with different data and initialization. Each column corresponds to an independent run. We observe the better performance of RGD+RiemNA in all the runs.

5.B Alternative averaging scheme via weighted Fréchet mean

We also consider the weighted Fréchet mean for computing the weighted average on manifolds, defined as

$$\bar{x}_{c,x} = \arg\min_{x \in \mathcal{X}} \sum_{i=0}^{k} c_i d^2(x, x_i). \tag{Avg.3}$$

Nevertheless, for general manifolds, it is not guaranteed the existence and uniqueness of the solution. In fact, one can ensure the uniqueness of the solution when the function $\frac{1}{2}d^2(x, x')$ is geodesic τ-strongly convex in x. From (Alimisis et al., 2020, Lemma 2), we see that the geodesic strong convexity of problem (Avg.3) holds for sufficiently small $\mathcal{X}$ on any manifold as well as for any non-positively curved manifold. Specifically, when $\mathcal{M}$ is non-positively curved, we have $\tau = 1$. While for other manifolds, let D be the diameter of $\mathcal{X}$ and $\kappa^+ > 0$ be the upper curvature bound. Then, geodesic strong convexity is satisfied with $\tau < 1$ when $D < \frac{\pi}{2\sqrt{\kappa^+}}$.

Lemma 5.10. *Under Assumption 5.1, suppose $x \mapsto \frac{1}{2}d^2(x, x')$ is geodesic τ-strongly convex in x for any $x' \in \mathcal{X}$. Consider $\bar{x}_{c,x} = \arg\min_{x \in \mathcal{X}} \sum_{i=0}^{k} c_i d^2(x, x_i)$. Then $d(\bar{x}_{c,x}, x^*) \leq \tau \| \sum_{i=0}^{k} c_i \Delta_{x_i} \|_{x^*}$ and $\|\Delta_{\bar{x}_{c,x}} - \sum_{i=0}^{k} c_i \Delta_{x_i} \|_{x^*} = O(d^3(x_0, x^*))$.*

Under the additional assumption of geodesic strong convexity, Lemma 5.10 shows an extra tighter bound on $d(\bar{x}_{c,x}, x^*)$, i.e., $d(\bar{x}_{c,x}, x^*) \leq \tau \| \sum_{i=0}^{k} c_i \Delta_{x_i} \|_{x^*}$. Thus, we see the error from the linear term does not suffer from metric distortion ($\epsilon_1 = 0$). The error bound from coefficient stability and nonlinearity terms however, still incur additional errors as the previous two averaging schemes. Lemma 5.10 allows convergence under the two averaging schemes to be established by exactly following the same steps as before. This is sufficient to show the same convergence bound holds (i.e., Theorem 5.1 and Proposition 5.1).

5.C From Euclidean averaging to Riemannian averaging

To extend the idea of weighted average to manifolds, we first rewrite the weighted average on the Euclidean space as follows.

Lemma 5.11 (Weighted average recursion)**.** *Given a set of coefficients $\{c_i\}_{i=0}^k$ with $\sum_{i=0}^k c_i = 1$ and a set of iterates $\{x_i\}_{i=0}^k$. Let the streaming weighted average be defined as $\tilde{x}_i = \tilde{x}_{i-1} + \gamma_i(x_i - \tilde{x}_{i-1})$ where $\gamma_i = \frac{c_i}{\sum_{j=0}^i c_j}$ for $i = 0, ..., k$ and $\tilde{x}_{-1} = x_0$. Then $\tilde{x}_k = \sum_{i=0}^k c_i x_i$.*

Proof. For some $\gamma_1, ..., \gamma_k$, the streaming weighted average is defined as $\tilde{x}_i = \tilde{x}_{i-1} + \gamma_i(x_i - \tilde{x}_{i-1})$ for $i \in [k]$. We first show the streaming weighted average has the form

$$\tilde{x}_i = \prod_{j=1}^{i}(1-\gamma_j)x_0 + \gamma_1 \prod_{j=2}^{i}(1-\gamma_j)x_1 + \cdots + \gamma_i x_i, \quad \forall i \in [k].$$

We prove such argument by induction. For $i = 1$, it is clear that $\tilde{x}_1 = (1-\gamma_1)x_0 + \gamma_1 x_1$ and satisfies the form. Suppose at $i = k'$, the equality is satisfied, then for $i = k'+1$, we have

$$\tilde{x}_{k'+1} = \tilde{x}_{k'} + \gamma_{k'+1}(x_{k'+1} - \tilde{x}_{k'}) = (1-\gamma_{k'+1})\tilde{x}_{k'} + \gamma_{k'+1}x_{k'+1}$$

which satisfies the equality. Hence this argument holds for all $i \in [k]$. Finally, at $i = k$, we see that the choice that $\gamma_i = \frac{c_i}{\sum_{j=0}^i c_i}$ leads to the matching coefficients.

□

5.D Main proofs

Before we proceed with the proofs of the results in the main text, we introduce a lemma that is used often in the course of the proof.

Lemma 5.12. *Under Assumption 5.1, for any $w, x, y, z \in \mathcal{X}$, we have*

$$\begin{aligned}
&\|\Gamma_w^x \Gamma_y^w \mathrm{Exp}_y^{-1}(z) - \big(\mathrm{Exp}_x^{-1}(z) - \mathrm{Exp}_x^{-1}(y)\big)\|_x \\
&\quad \leq C_0 d(y,w)d(w,x)d(y,z) + C_2 \min\{d(y,z), d(x,y)\} C_\kappa\big(d(y,z) + d(x,y)\big).
\end{aligned}$$

Proof of Lemma 5.12.

$$\begin{aligned}
&\|\Gamma_w^x \Gamma_y^w \mathrm{Exp}_y^{-1}(z) - \big(\mathrm{Exp}_x^{-1}(z) - \mathrm{Exp}_x^{-1}(y)\big)\|_x \\
&\leq \|\Gamma_w^x \Gamma_y^w \mathrm{Exp}_y^{-1}(z) - \Gamma_y^x \mathrm{Exp}_y^{-1}(z)\|_x + \|\Gamma_y^x \mathrm{Exp}_y^{-1}(z) - \big(\mathrm{Exp}_x^{-1}(z) - \mathrm{Exp}_x^{-1}(y)\big)\|_x \\
&\leq C_0 d(y,w)d(w,x)d(y,z) + C_2 d\Big(\mathrm{Exp}_x\big(\Gamma_y^x \mathrm{Exp}_y^{-1}(z) + \mathrm{Exp}_x^{-1}(y)\big), z\Big) \\
&\leq C_0 d(y,w)d(w,x)d(y,z) + C_2 d\Big(\mathrm{Exp}_x\big(\Gamma_y^x \mathrm{Exp}_y^{-1}(z) + \mathrm{Exp}_x^{-1}(y)\big), \mathrm{Exp}_y(\mathrm{Exp}_y^{-1}(z))\Big) \\
&\leq C_0 d(y,w)d(w,x)d(y,z) + C_2 \min\{d(y,z), d(x,y)\} C_\kappa\big(d(y,z) + d(x,y)\big).
\end{aligned}$$

where we apply Lemma 5.1 and 5.2. □

5.D.1 Proof of Proposition 5.2

We show in Proposition 5.2 that the optimal coefficients c^* has a closed-form solution.

Proposition 5.2. *Let $R = [\langle r_i, r_j \rangle_{x_k}]_{i,j} \in \mathbb{R}^{(k+1)\times(k+1)}$ collects all pairwise inner products. Then the solution $c^* = \arg\min_{c \in \mathbb{R}^{k+1} : c^\top 1 = 1} \|\sum_{i=0}^k c_i r_i\|_{x_k}^2 + \lambda \|c\|_2^2$ is explicitly derived as $c^* = \frac{(R+\lambda I)^{-1} 1}{1^\top (R+\lambda I)^{-1} 1}$.*

Proof of Proposition 5.2. Let $\mu \in \mathbb{R}$ be the dual variable. Then we have c^*, μ^* satisfy the KKT system:

$$\begin{bmatrix} 2(R+\lambda I) & 1 \\ 1^\top & 0 \end{bmatrix} \begin{bmatrix} c^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Solving the system yields the desired result. □

5.D.2 Proof of Lemma 5.3

Proof of Lemma 5.3. First, we consider the pushforward operator $\mathrm{Exp}_x^y : T_x\mathcal{M} \to T_y\mathcal{M}$ for any $x, y \in \mathcal{M}$, defined as $\mathrm{Exp}_x^y(u) := \mathrm{Exp}_y^{-1}(\mathrm{Exp}_x(u))$ for any $u \in T_x\mathcal{M}$. The differential of Exp_x^y at 0 along $u \in T_x\mathcal{M}$ is derived as

$$\begin{aligned} \mathrm{DExp}_x^y(0)[u] = \mathrm{DExp}_y^{-1}(\mathrm{Exp}_x(0))[\mathrm{DExp}_x(0)[u]] &= \mathrm{DExp}_y^{-1}(x)[u] \\ &= [\mathrm{DExp}_y(\mathrm{Exp}_y^{-1}(x))]^{-1}[u] \\ &= (T_y^x)^{-1}[u] \end{aligned}$$

where we denote $T_x^y(v) = \mathrm{DExp}_x(\mathrm{Exp}_x^{-1}(y))[v] \in T_y\mathcal{M}$ for $v \in T_x\mathcal{M}$. The second equality is due to $\mathrm{Exp}_x(0) = 0, \mathrm{DExp}_x(0) = \mathrm{id}$ and the third equality follows from the inverse function theorem. Then by Taylor's theorem for $\mathrm{Exp}_{x_i}^{x^*}$ around 0, we have

$$\begin{aligned} &\mathrm{Exp}_{x_*}^{-1}(x_{i+1}) = \mathrm{Exp}_{x_i}^{x^*}(\mathrm{Exp}_{x_i}^{-1}(x_{i+1})) \\ &= \mathrm{Exp}_{x_i}^{x^*}(0) + \mathrm{DExp}_{x_i}^{x^*}(0)[\mathrm{Exp}_{x_i}^{-1}(x_{i+1})] + \frac{1}{2}\mathrm{D}^2\mathrm{Exp}_{x_i}^{x^*}(\zeta_i)[\mathrm{Exp}_{x_i}^{-1}(x_{i+1}), \mathrm{Exp}_{x_i}^{-1}(x_{i+1})] \\ &= \mathrm{Exp}_{x^*}^{-1}(x_i) - \eta(T_{x^*}^{x_i})^{-1}[\mathrm{grad}f(x_i)] + \frac{\eta^2}{2}\mathrm{D}^2\mathrm{Exp}_{x_i}^{x^*}(\zeta_i)[\mathrm{grad}f(x_i), \mathrm{grad}f(x_i)] \\ &= \mathrm{Exp}_{x^*}^{-1}(x_i) - \eta(T_{x^*}^{x_i})^{-1}[\mathrm{grad}f(x_i)] + \frac{\eta^2}{2}\epsilon_i \qquad (5.6) \end{aligned}$$

for some $\zeta_i = s\mathrm{Exp}_{x_i}^{-1}(x_{i+1}), s \in (0,1)$. Let $\epsilon_i := \mathrm{D}^2\mathrm{Exp}_{x_i}^{x^*}(\zeta_i)[\mathrm{grad}f(x_i), \mathrm{grad}f(x_i)]$ with $\|\epsilon_i\|_{x^*} = O\big(\|\mathrm{grad}f(x_i)\|_{x_i}^2\big)$. Then by Hessian Lipschitzness (Lemma 2.1), we have around x^*

$$e_i := \Gamma_{x_i}^{x^*}\mathrm{grad}f(x_i) - \mathrm{Hess}f(x^*)[\mathrm{Exp}_{x^*}^{-1}(x_i)] \leq \frac{\rho}{2}\|\mathrm{Exp}_{x^*}^{-1}(x_i)\|_{x^*}^2. \qquad (5.7)$$

Combining (5.6) with (5.7) yields

$$\mathrm{Exp}_{x_*}^{-1}(x_{i+1}) - \mathrm{Exp}_{x^*}^{-1}(x_i) = -\eta(\Gamma_{x_i}^{x^*}T_{x^*}^{x_i})^{-1}[\Gamma_{x_i}^{x^*}\mathrm{grad}f(x_i)] + \frac{\eta^2}{2}\epsilon_i$$

$$= -\eta(\Gamma_{x_i}^{x^*} T_{x^*}^{x_i})^{-1}[\mathrm{Hess} f(x^*)[\mathrm{Exp}_{x^*}^{-1}(x_i)] + e_i] + \frac{\eta^2}{2}\epsilon_i. \tag{5.8}$$

To show the desired result, it remains to show the operator $(\Gamma_{x_i}^{x^*} T_{x^*}^{x_i})^{-1}$ is locally identity. This is verified in (Tripuraneni et al., 2018, Lemma 6) for general retraction. We restate here and adapt to the case of exponential map.

Consider the function $H(u) := (\Gamma_x^{\mathrm{Exp}_x(u)})^{-1} T_x^{\mathrm{Exp}_x(u)} : T_x\mathcal{M} \to L(T_x\mathcal{M})$, where $L(T_x\mathcal{M})$ denotes the set of linear maps on $T_x\mathcal{M}$. Let $\gamma(t) = \mathrm{Exp}_x(tu)$. Then we have

$$\begin{aligned}\frac{d}{dt}H(tu)|_{t=0} = \frac{d}{dt}(\Gamma_x^{\gamma(t)})^{-1}T_x^{\gamma(t)}|_{t=0} &= \Big((\Gamma_x^{\gamma(t)})^{-1}\frac{D}{dt}T_x^{\gamma(t)}\Big)|_{t=0} \\ &= (\frac{D}{dt}\mathrm{DExp}_x(tu))|_{t=0} \\ &= \frac{D^2}{dt^2}\mathrm{Exp}_x(tu)|_{t=0} = 0.\end{aligned}$$

where the second equality is due to the property of parallel transport (see for example (Boumal, 2023, Proposition 10.37)). In addition, from (Waldmann, 2012, Theorem A.2.9), we see the second order derivative of H is given by $\frac{d^2}{dt^2}H(tu)|_{t=0} = \frac{1}{6}\mathrm{Riem}_x(u,\cdot)u$ where we denote Riem_x as the Riemann curvature tensor evaluated at x. We notice that $\mathrm{Riem}_x(u,\cdot)u : T_x\mathcal{M} \to T_x\mathcal{M}$ is symmetric with respect to the Riemannian metric (see for example Andrews & Hopper (2010)).

For any $v \in T_x\mathcal{M}$, $H(u)[v] \in T_x\mathcal{M}$, we apply the Taylor's theorem for H up to second order, which yields

$$H(u)[v] = v + \frac{1}{6}\mathrm{Riem}_x(u,v)u + O(\|u\|^3),$$

Let $x = x^*$ and $u = \mathrm{Exp}_{x^*}^{-1}(x_i) = \Delta_{x_i}$. Then we obtain for any $v \in T_{x^*}\mathcal{M}$,

$H(u)[v] \in T_{x^*}\mathcal{M}$

$$\Gamma_{x_i}^{x^*} T_{x^*}^{x_i}[v] = v + \frac{1}{6}\mathrm{Riem}_{x^*}(\Delta_{x_i}, v)\Delta_{x_i} + O(\|\Delta_{x_i}\|^3). \tag{5.9}$$

It satisfies that $(\Gamma_{x_i}^{x^*} T_{x^*}^{x_i})^{-1} = \mathrm{id} - \frac{1}{6}\mathrm{Riem}_{x^*}(\Delta_{x_i}, \cdot)\Delta_{x_i} + O(\|\Delta_{x_i}\|^3)$. Substituting this result into (5.8), we obtain

$$\begin{aligned}
&\Delta_{x_{i+1}} - \Delta_{x_i} \\
&= -\eta\Big(\mathrm{id} - \frac{1}{6}\mathrm{Riem}_{x^*}(\Delta_{x_i}, \cdot)\Delta_{x_i} + O(\|\Delta_{x_i}\|^3)\Big)\left[\mathrm{Hess} f(x^*)[\Delta_{x_i}] + e_i\right] + \frac{\eta^2}{2}\epsilon_i \\
&= -\eta\,\mathrm{Hess} f(x^*)[\Delta_{x_i}] - \eta e_i + \frac{\eta}{6}\mathrm{Riem}_{x^*}(\Delta_{x_i}, \mathrm{Hess} f(x^*)[\Delta_{x_i}] + e_i)\Delta_{x_i} \\
&\quad + \frac{\eta^2}{2}\epsilon_i + O(\|\Delta_{x_i}\|^3).
\end{aligned}$$

Let $\varepsilon_i = -\eta e_i + \frac{\eta}{6}\mathrm{Riem}_{x^*}(\Delta_{x_i}, \mathrm{Hess} f(x^*)[\Delta_{x_i}] + e_i)\Delta_{x_i} + \frac{\eta^2}{2}\epsilon_i + O(\|\Delta_{x_i}\|^3)$. We can bound the error term as follows.

$$\|\varepsilon_i\|_{x^*}^2 = O(\|e_i\|_{x^*}^2 + \|\Delta_{x_i}\|_{x^*}^4\|\mathrm{grad} f(x_i)\|_{x_i}^2 + \|\epsilon_i\|_{x^*}^2 + \|\Delta_{x_i}\|_{x^*}^6) = O(\|\Delta_{x_i}\|^4),$$

where we use the bounds on $\|e_i\|_{x^*}, \|\epsilon_i\|_{x^*}$ as well as $\mathrm{Hess} f(x^*)[\Delta_{x_i}] + e_i = \Gamma_{x_i}^{x^*}\mathrm{grad} f(x_i)$ and geodesic gradient Lipschitzness such that $\|\mathrm{grad} f(x_i)\|^2 \leq L\|\Delta_i\|_{x^*}^2$. □

5.D.3 Proof of Lemma 5.4

Proof of Lemma 5.4. The proof is by induction. Let $\gamma_i = \frac{c_i}{\sum_{j=0}^{i} c_j}$ and first we rewrite the averaging on tangent space as following the recursion defined as $\widetilde{\Delta}_{x_i} = \widetilde{\Delta}_{x_{i-1}} + \gamma_i(\Delta_{x_i} - \widetilde{\Delta}_{x_{i-1}})$. As we have shown in Lemma 5.11, $\sum_{i=0}^{k} c_i \Delta_{x_i} = \widetilde{\Delta}_{x_k}$. To show the difference between $\Delta_{\bar{x}_{c,x}}$, based on Lemma 5.2, it suffices to show the distance between $\tilde{x}_k = \bar{x}_{c,x}$ and $\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_k})$ is bounded.

To this end, we first notice that $\tilde{x}_0 = x_0 = \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_0})$ and we consider bounding the difference between $\tilde{x}_1$ and $\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_1})$. To derive the bound, we first observe that by Lemma 5.1,

$$\begin{aligned} & d\Big(\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_1}), \mathrm{Exp}_{x_0}\big(\Gamma_{x^*}^{x_0}\gamma_1(\Delta_{x_1} - \widetilde{\Delta}_{x_0})\big)\Big) \\ & = d\Big(\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_0} + \gamma_1(\Delta_{x_1} - \widetilde{\Delta}_{x_0})), \mathrm{Exp}_{x_0}\big(\Gamma_{x^*}^{x_0}\gamma_1(\Delta_{x_1} - \widetilde{\Delta}_{x_0})\big)\Big) \\ & \leq d(x_0, x^*) C_\kappa\Big(\|\widetilde{\Delta}_{x_0}\|_{x^*} + \gamma_1\|\Delta_{x_1} - \widetilde{\Delta}_{x_0}\|_{x^*}\Big), \end{aligned} \tag{5.10}$$

where we see $x_0 = \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_0})$ with $\widetilde{\Delta}_{x_0} = \Delta_{x_0}$. In addition,

$$\begin{aligned} & d\Big(\tilde{x}_1, \mathrm{Exp}_{x_0}\big(\Gamma_{x^*}^{x_0}\gamma_1(\Delta_{x_1} - \widetilde{\Delta}_{x_0})\big)\Big) \\ & \quad = d\Big(\mathrm{Exp}_{x_0}\big(\gamma_1 \mathrm{Exp}_{x_0}^{-1}(x_1)\big), \mathrm{Exp}_{x_0}\big(\Gamma_{x^*}^{x_0}\gamma_1(\Delta_{x_1} - \widetilde{\Delta}_{x_0})\big)\Big) \\ & \quad \leq \gamma_1 C_1 \|\mathrm{Exp}_{x_0}^{-1}(x_1) - \Gamma_{x^*}^{x_0}(\Delta_{x_1} - \Delta_{x_0})\|_{x_0} \\ & \quad \leq \gamma_1 C_1 C_2 d(x_0, x^*) C_\kappa\big(d(x_0, x_1) + d(x_0, x^*)\big). \end{aligned} \tag{5.11}$$

where the last inequality is from the proof of Lemma 5.12. Thus combining (5.10), (5.11) leads to

$$\begin{aligned} d\big(\tilde{x}_1, \mathrm{Exp}_{x^*}(\tilde{\Delta}_{x_1})\big) \leq d(x_0, x^*) C_\kappa & \Big(\|\widetilde{\Delta}_{x_0}\|_{x^*} + \gamma_1\|\Delta_{x_1} - \widetilde{\Delta}_{x_0}\|_{x^*}\Big) \\ & + \gamma_1 C_1 C_2 d(x_0, x^*) C_\kappa\big(d(x_0, x_1) + d(x_0, x^*)\big). \end{aligned}$$

By noticing $C_\kappa(x) = O(x^2)$, we see $d(\tilde{x}_1, \mathrm{Exp}_{x^*}(\tilde{\Delta}_{x_1})) = O(d^3(x_0, x^*))$.

Now suppose at $i \leq k-1$, we have $d(\tilde{x}_i, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})) = O(d^3(x_0, x^*))$ and we wish to show $d(\tilde{x}_{i+1}, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_{i+1}})) = O(d^3(x_0, x^*))$. To this end, we first see $\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_{i+1}}) = \mathrm{Exp}_{x^*}\big(\widetilde{\Delta}_{x_i} + \gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)$ and by Lemma 5.1

$$\begin{aligned} & d\Big(\mathrm{Exp}_{x^*}\big(\widetilde{\Delta}_{x_i} + \gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big), \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\big(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)\Big) \\ & \leq \|\widetilde{\Delta}_{x_i}\|_{x^*} C_\kappa\big(\|\widetilde{\Delta}_{x_i}\|_{x^*} + \gamma_{i+1}\|\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}\|_{x^*}\big) \\ & = O(d^3(x_0, x^*)), \end{aligned}$$

where the last equality is due to $C_\kappa(x) = O(x^2)$ and $\|\widetilde{\Delta}_{x_i}\|_{x^*} = O(d(x_0, x^*))$, which can be shown by induction.

Further, noticing $\tilde{x}_{i+1} = \mathrm{Exp}_{\tilde{x}_i}\big(\gamma_{i+1}\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1})\big)$, we can show

$$\begin{aligned}
&d\Big(\tilde{x}_{i+1}, \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\big(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)\Big) \\
&\;\le d\Big(\mathrm{Exp}_{\tilde{x}_i}\big(\gamma_{i+1}\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1})\big), \mathrm{Exp}_{\tilde{x}_i}\big(\Gamma_{x^*}^{\tilde{x}_i}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)\Big) \\
&\quad + d\Big(\mathrm{Exp}_{\tilde{x}_i}\big(\Gamma_{x^*}^{\tilde{x}_i}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big), \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\big(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)\Big).
\end{aligned} \tag{5.12}$$

The first term on the right of (5.12) can be bounded as

$$\begin{aligned}
&d\Big(\mathrm{Exp}_{\tilde{x}_i}\big(\gamma_{i+1}\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1})\big), \mathrm{Exp}_{\tilde{x}_i}\big(\Gamma_{x^*}^{\tilde{x}_i}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\big)\Big) \\
&\le \gamma_{i+1}\|\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1}) - \Gamma_{x^*}^{\tilde{x}_i}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})\|_{\tilde{x}_i} \\
&= \gamma_{i+1}\|\Gamma_{\tilde{x}_i}^{x^*}\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1}) - (\Delta_{x_{i+1}} - \Delta_{\tilde{x}_i}) + (\widetilde{\Delta}_{x_i} - \Delta_{\tilde{x}_i})\|_{x^*} \\
&\le \gamma_{i+1}\|\Gamma_{\tilde{x}_i}^{x^*}\mathrm{Exp}_{\tilde{x}_i}^{-1}(x_{i+1}) - (\Delta_{x_{i+1}} - \Delta_{\tilde{x}_i})\|_{x^*} + \gamma_{i+1}\|\widetilde{\Delta}_{x_i} - \Delta_{\tilde{x}_i}\|_{x^*} \\
&\le \gamma_{i+1}C_2 d(\tilde{x}_i, x^*)C_\kappa\big(d(x_{i+1}, \tilde{x}_i) + d(\tilde{x}_i, x^*)\big) + \gamma_{i+1}C_2 d(\tilde{x}_i, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})),
\end{aligned} \tag{5.13}$$

where we again use the result from the proof of Lemma 5.12. To see (5.13) is on the order of $O(d^3(x_0, x^*))$, we only need to show $\|\Delta_{\tilde{x}_i}\|_{x^*}^2 = d^2(\tilde{x}_i, x^*) = O(d^2(x_0, x^*))$, which can be seen by a simple induction argument. First, it is clear that $\|\Delta_{\tilde{x}_0}\|_{x^*}^2 = d^2(x_0, x^*)$. Then suppose for any $i < k$, we have $d(\tilde{x}_i, x^*) = O(d(x_0, x^*))$. Then from Lemma 5.2, we have

$$\begin{aligned}
d(\tilde{x}_{i+1}, x^*) \le C_1\|\mathrm{Exp}_{\tilde{x}_i}^{-1}(\tilde{x}_{i+1}) - \mathrm{Exp}_{\tilde{x}_i}^{-1}(x^*)\|_{\tilde{x}_i} &\le \frac{C_1 c_{i+1}}{\sum_{j=0}^{i+1} c_j} d(\tilde{x}_i, x_{i+1}) + d(\tilde{x}_i, x^*) \\
&\le \Big(\frac{C_1 c_{i+1}}{\sum_{j=0}^{i+1} c_j} + 1\Big) d(\tilde{x}_i, x^*) + d(x_{i+1}, x^*) = O(d(x_0, x^*)).
\end{aligned}$$

Thus, using $d(\tilde{x}_i, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})) = O(d^3(x_0, x^*))$, we see (5.13) is on the order of $O(d^3(x_0, x^*))$.

Now we bound the second term on the right of (5.12). Particularly,

$$\begin{aligned}
&d\Big(\mathrm{Exp}_{\tilde{x}_i}(\Gamma_{x^*}^{\tilde{x}_i}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})), \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}))\Big) \\
&\le d\Big(\mathrm{Exp}_{\tilde{x}_i}(\Gamma_{x^*}^{\tilde{x}_i}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})), \mathrm{Exp}_{\tilde{x}_i}(\Gamma_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}^{\tilde{x}_i}\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}))\Big) \\
&+ d\Big(\mathrm{Exp}_{\tilde{x}_i}(\Gamma_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}^{\tilde{x}_i}\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})), \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}))\Big) \\
&\le \gamma_{i+1}C_1C_0\|\widetilde{\Delta}_{x_i}\|_{x^*}d(\tilde{x}_i, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i}))\|\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}\|_{x^*} + C_3 d(\tilde{x}_i, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})) \\
&= O(d^3(x_0, x^*)),
\end{aligned}$$

where we apply Lemma 5.2 multiple times. Combining the previous results, we see

$$\begin{aligned}
&d(\tilde{x}_{i+1}, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_{i+1}})) \\
&\le d\Big(\tilde{x}_{i+1}, \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}))\Big) \\
&\quad + d\Big(\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i} + \gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i})), \mathrm{Exp}_{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}(\Gamma_{x^*}^{\mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_i})}\gamma_{i+1}(\Delta_{x_{i+1}} - \widetilde{\Delta}_{x_i}))\Big) \\
&= O(d^3(x_0, x^*))
\end{aligned}$$

Now applying Lemma 5.2, we obtain

$$\|\Delta_{\tilde{x}_{i+1}} - \widetilde{\Delta}_{x_{i+1}}\|_{x^*} \le C_2 d(\tilde{x}_{i+1}, \mathrm{Exp}_{x^*}(\widetilde{\Delta}_{x_{i+1}})) = O(d^3(x_0, x^*))$$

for all $i \le k-1$. Let $i = k-1$ we have $\|\Delta_{\tilde{x}_k} - \widetilde{\Delta}_{x_k}\|_{x^*} = \|\Delta_{\bar{x}_{c,x}} - \sum_{i=0}^k c_i \Delta_{x_i}\|_{x^*} = O(d^3(x_0, x^*))$. Thus the proof is complete. □

5.D.4 Proof of Lemma 5.5

Proof of Lemma 5.5. Directly combining Lemma 5.13 and Lemma 5.4 gives the result. □

Lemma 5.13 (Convergence of the linearized iterates)**.** *Consider the linearized iterates* $\{\hat{x}_i\}_{i=0}^k$ *satisfying* (5.4) *for some* $G \succeq 0$ *with* $\|G\|_{x^*} \le \sigma < 1$*. Let* $\hat{r}_i = \Delta_{\hat{x}_{i+1}} - \Delta_{\hat{x}_i}$*,*

$\hat{c}^* = \arg\min_{c^\top 1 = 1} \| \sum_{i=0}^k c_i \hat{r}_i \|_{x^*}^2 + \lambda \|c\|_2^2$. *Then*

$$\| \sum_{i=0}^k \hat{c}_i^* \Delta_{\hat{x}_i} \|_{x^*} \leq \frac{d(x_0, x^*)}{1-\sigma} \sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{d^2(x_0, x^*)} \|\hat{c}^*\|_2^2}$$

Proof of Lemma 5.13. The proof follows from (Scieur et al., 2020, Proposition 3.4) and we include it here for completeness. Denote $\mathcal{P}_k^1 := \{p \in \mathbb{R}[x] : \deg(p) = k, p(1) = 1\}$ as the set of polynomials of degree k with coefficients summing to 1. Noticing that $\hat{r}_i = \Delta_{\hat{x}_{i+1}} - \Delta_{\hat{x}_i} = (G - \mathrm{id})[\Delta_{\hat{x}_i}] = (G - \mathrm{id}) G^i [\Delta_{x_0}]$, we have $\| \sum_{i=0}^k c_i \hat{r}_i \|_{x^*}^2 = \| (G - \mathrm{id}) p(G) [\Delta_{x_0}] \|_{x^*}^2$ where $p \in \mathcal{P}_k^1$ and $\{c_i\}_{i=0}^k$ are the corresponding coefficients. Then we obtain

$$\begin{aligned}
&\min_{p \in \mathcal{P}_k^1} \left\{ \| (G - \mathrm{id}) p(G) [\Delta_{x_0}] \|_{x^*}^2 + \lambda \|c\|_2^2 \right\} \\
&\quad \leq d^2(x_0, x^*) \min_{p \in \mathcal{P}_k^1} \left\{ \|p(G)\|_{x^*}^2 + \frac{\lambda}{d^2(x_0, x^*)} \|p\|_2^2 \right\} \\
&\quad \leq d^2(x_0, x^*) \min_{p \in \mathcal{P}_k^1} \max_{M : 0 \preceq M \preceq \sigma \mathrm{id}} \left\{ \|p(M)\|_{x^*}^2 + \frac{\lambda}{d^2(x_0, x^*)} \|p\|_2^2 \right\} \\
&\quad = d^2(x_0, x^*) \min_{p \in \mathcal{P}_k^1} \max_{x \in [0,\sigma]} \left\{ p^2(x) + \frac{\lambda}{d^2(x_0, x^*)} \|p\|_2^2 \right\} \\
&\quad = (S_{k,\bar{\lambda}}^{[0,\sigma]})^2 d^2(x_0, x^*),
\end{aligned}$$

where $\bar{\lambda} = \lambda / d^2(x_0, x^*)$ and we use the fact that $\|G - \mathrm{id}\|_{x^*} \leq 1$. Then

$$\begin{aligned}
\| \sum_{i=0}^k \hat{c}_i^* \Delta_{\hat{x}_i} \|_{x^*}^2 &= \| \sum_{i=0}^k \hat{c}_i^* (G - \mathrm{id})^{-1} \hat{r}_i \|_{x^*}^2 \\
&\leq \|(G - \mathrm{id})^{-1}\|_{x^*}^2 \Big(\| \sum_{i=0}^k \hat{c}_i^* \hat{r}_i \|_{x^*}^2 + \lambda \|\hat{c}^*\|_2^2 - \lambda \|\hat{c}^*\|_2^2 \Big) \\
&\leq \frac{d^2(x_0, x^*)}{(1-\sigma)^2} \Big((S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{d^2(x_0, x^*)} \|\hat{c}^*\|_2^2 \Big),
\end{aligned}$$

where we see that $\|(G - \mathrm{id})^{-1}\|_{x^*} \leq \frac{1}{1-\sigma}$. □

5.D.5 Proof of Lemma 5.6

Proof of Lemma 5.6. From Proposition 5.2 and following (Scieur et al., 2020, Proposition 3.2), we obtain

$$\|c^*\| \leq \sqrt{\frac{\|R\|_2 + \lambda}{(k+1)\lambda}}.$$

Now we bound $\|R\|_2$. First we see R can be rewritten as $\mathcal{R}^\top \mathcal{G}_{x_k} \mathcal{R}$, where $\mathcal{G}_{x_k} \in \mathbb{R}^{r \times r}$ is the positive definite metric tensor at x_k and $\mathcal{R} = [\mathrm{vec} r_i] \in \mathbb{R}^{r \times k}$ is the collection of tangent vector in an orthonormal basis and r is the intrinsic dimension of the manifold. Thus we can write Riemannian inner product as $\langle r_i, r_j \rangle_{x_k} = \mathrm{vec} r_i^\top \mathcal{G}_{x_k} \mathrm{vec} r_j$ and

$$\|R\|_2 = \|\mathcal{G}_{x_k}^{1/2} \mathcal{R}\|_2^2 \leq \|\mathcal{G}_{x_k}^{1/2} R\|_{\mathrm{F}}^2 = \sum_{i=0}^{k} \mathrm{vec} r_i^\top \mathcal{G}_{x_k} \mathrm{vec} r_i = \sum_{i=0}^{k} \|r_i\|_{x_k}^2 = \sum_{i=0}^{k} d^2(x_i, x_{i+1}).$$

On the other hand, denote the perturbation matrix $P = R - \hat{R}$. Then from Proposition 5.2 and following (Scieur et al., 2020, Proposition 3.2), we have

$$\|\delta^c\|_2 \leq \frac{\|P\|_2}{\lambda} \|\hat{c}^*\|_2.$$

Now we need to bound $\|P\|_2$. Let $\mathcal{E}_i = \Delta_{x_i} - \Delta_{\hat{x}_i}$. Then we have

$$\begin{aligned} \|\Gamma_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*} &= \|\Gamma_{x_k}^{x^*} r_i - (\Delta_{x_{i+1}} - \Delta_{x_i}) + (\Delta_{x_{i+1}} - \Delta_{x_i}) - \hat{r}_i\|_{x^*} \\ &\leq \|\Gamma_{x_k}^{x^*} r_i - (\Delta_{x_{i+1}} - \Delta_{x_i})\|_{x^*} + \|(\Delta_{x_{i+1}} - \Delta_{x_i}) - \hat{r}_i\|_{x^*} \\ &= \|\Gamma_{x_k}^{x^*} r_i - (\Delta_{x_{i+1}} - \Delta_{x_i})\|_{x^*} + \|\mathcal{E}_{i+1} - \mathcal{E}_i\|_{x^*} \qquad (5.14) \end{aligned}$$

where we use Lemma 5.2. Now we respectively bound each of the two terms on the right. First we see from Lemma 5.12,

$$\|\Gamma_{x_k}^{x^*} r_i - (\Delta_{x_{i+1}} - \Delta_{x_i})\|_{x^*} \leq C_0 d(x_i, x_k) d(x_k, x^*) d(x_i, x_{i+1})$$

$$+ C_2 d(x_i, x^*) C_\kappa\big(d(x_i, x^*) + d(x_i, x_{i+1})\big) \tag{5.15}$$

Further, we bound $\|\mathcal{E}_{i+1} - \mathcal{E}_i\|_{x^*}$. From Lemma 5.3, we have $\mathcal{E}_i = G[\mathcal{E}_{i-1}] + \varepsilon_i, \mathcal{E}_0 = 0$ and

$$\|\mathcal{E}_{i+1} - \mathcal{E}_i\|_{x^*} = \|(G - \mathrm{id})\mathcal{E}_i + \varepsilon_{i+1}\|_{x^*} = \|(G - \mathrm{id})\sum_{j=1}^{i} G^{i-j}\varepsilon_j + \varepsilon_{i+1}\|_{x^*} \le \sum_{j=1}^{i+1} \|\varepsilon_j\|_{x^*}. \tag{5.16}$$

Combining (5.16), (5.15), (5.14) leads to

$$\begin{aligned}\|\Gamma_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*} \le C_0 d(x_i, x_k) d(x_k, x^*) d(x_i, x_{i+1}) + C_2 d(x_i, x^*) C_\kappa\big(d(x_i, x^*) \\ + d(x_i, x_{i+1})\big) + \sum_{j=1}^{i+1} \|\varepsilon_j\|_{x^*}.\end{aligned}$$

Finally, recall we can write $R = \mathcal{R}^\top \mathcal{G}_{x_k} \mathcal{R}$ and similarly for $\hat{R} = \hat{\mathcal{R}}^\top \mathcal{G}_{x^*} \hat{\mathcal{R}}$ where $\hat{\mathcal{R}} = [\mathrm{vec}\hat{r}_i]$. By isometry of parallel transport, we have $R = \mathcal{R}_{x^*}^\top \mathcal{G}_{x^*} \mathcal{R}_{x^*}$ where $\mathcal{R}_{x^*} = [\overrightarrow{\Gamma_{x_k}^{x^*} r_i}]$. Let $E = \mathcal{G}_{x^*}^{1/2}(\mathcal{R}_{x^*} - \hat{\mathcal{R}})$. Then

$$\|P\|_2 = \|\mathcal{R}_{x^*}^\top \mathcal{G}_{x^*} \mathcal{R}_{x^*} - \hat{\mathcal{R}}^\top \mathcal{G}_{x^*} \hat{\mathcal{R}}\|_2 \le 2\|E\|_2 \|\mathcal{G}_{x^*}^{1/2} \hat{\mathcal{R}}\|_2 + \|E\|_2^2.$$

Notice that

$$\begin{aligned}\|\mathcal{G}_{x^*}^{1/2} \hat{R}\|_2 \le \|\mathcal{G}_{x^*}^{1/2} \hat{R}\|_{\mathrm{F}} \le \sum_{i=0}^{k} \|\hat{r}_i\|_{x^*} \le \sum_{i=0}^{k} \|(G - \mathrm{id}) G^i \hat{r}_0\|_{x^*} \le \sum_{i=0}^{k} \sigma^i \|\hat{r}_0\|_{x^*} \\ \le \frac{1 - \sigma^{k+1}}{1 - \sigma} d(x_0, x^*),\end{aligned}$$

Also

$$\begin{aligned}\|E\|_2 = \|\mathcal{G}_{x^*}^{1/2}(\mathcal{R}_{x^*} - \hat{\mathcal{R}})\|_2 \le \sum_{i=0}^{k} \|\Gamma_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*} \\ \le d(x_k, x^*) C_0 \sum_{i=0}^{k} d(x_i, x_k) d(x_i, x_{i+1}) + C_2 \sum_{i=0}^{k} d(x_i, x^*) C_\kappa\big(d(x_i, x^*) + d(x_i, x_{i+1})\big)\end{aligned}$$

$$\begin{aligned}&+\sum_{i=0}^{k}\sum_{j=1}^{i+1}\|\varepsilon_j\|_{x^*}\\ &= O(d^2(x_0,x^*)),\end{aligned}$$

where we notice that $C_\kappa(d(x_i,x^*)+d(x_i,x_{i+1})) = O(d^2(x_i,x^*))$ and recall that $\|\varepsilon_j\|_{x^*} = O(d^2(x_j,x^*)) = O(d^2(x_0,x^*))$. Thus $\|P\|_2 \leq 2\psi\frac{1-\sigma^{k+1}}{1-\sigma}d(x_0,x^*)+(\psi)^2$ where $\psi = O(d^2(x_0,x^*))$. □

5.D.6 Proof of Lemma 5.7

Proof of Lemma 5.7. From Lemma 5.2, we first observe $d(\bar{x}_{\hat{c}^*,\hat{x}},\bar{x}_{c^*,\hat{x}}) \leq C_1\|\Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}}\|_{x^*}$. Now we derive a bound on the term $\|\Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}}\|_{x^*}$. Notice that from Lemma 5.4, we have

$$\begin{aligned}\|\Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}}\|_{x^*} &= \|\sum_{i=0}^{k}(\hat{c}_i^* - c_i^*)\Delta_{\hat{x}_i} + \hat{\epsilon}\|_{x^*} \leq \|\delta^c\|_2\big(\sum_{i=0}^{k}\|\Delta_{\hat{x}_i}\|_{x^*}^2\big)^{1/2} + \|\hat{\epsilon}\|_{x^*}\\ &\leq \|\delta^c\|_2(\sum_{i=0}^{k}\|\Delta_{\hat{x}_i}\|_{x^*}) + \|\hat{\epsilon}\|_{x^*}\\ &\leq \|\delta^c\|_2(\sum_{i=0}^{k}\|G\|^i\|\Delta_{x_0}\|_{x^*}) + \|\hat{\epsilon}\|_{x^*}\\ &\leq \frac{1-\sigma^{k+1}}{1-\sigma}d(x_0,x^*)\|\delta^c\|_2 + \|\hat{\epsilon}\|_{x^*}\\ &\leq \frac{1}{1-\sigma}\frac{d(x_0,x^*)}{\lambda}\Big(\frac{1}{1-\sigma}2\psi d(x_0,x^*) + (\psi)^2\Big)\|\hat{c}^*\|_2 + \|\hat{\epsilon}\|_{x^*}\end{aligned}$$

for some $\|\hat{\epsilon}\|_{x^*} = O(d^3(x_0,x^*))$ and we denote $\delta^c = c^* - \hat{c}^*$. The bound on $\|\delta^c\|_2$ is from Lemma 5.6. □

5.D.7 Proof of Lemma 5.8

Proof of Lemma 5.8. Similarly to Lemma 5.7, we see $d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x}) \leq C_1 \|\Delta_{\bar{x}_{c^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,x}}\|_{x^*}$ due to Lemma 5.2. Again using Lemma 5.4, we see

$$\begin{aligned}\|\Delta_{\bar{x}_{c^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,x}}\|_{x^*} = \| \sum_{i=0}^{k} c_i^* (\Delta_{x_i} - \Delta_{\hat{x}_i}) + \hat{\epsilon}\|_{x^*} &\leq \|c^*\|_2 (\sum_{i=0}^{k} \|\mathcal{E}_i\|_{x^*}^2)^{1/2} + \|\hat{\epsilon}\|_{x^*} \\ &\leq \|c^*\|_2 (\sum_{i=0}^{k} \|\mathcal{E}_i\|_{x^*}) + \|\hat{\epsilon}\|_{x^*}\end{aligned}$$

where $\|\hat{\epsilon}\|_{x^*} = O(d^3(x_0, x^*))$ and $\mathcal{E}_i = \Delta_{x_i} - \Delta_{\hat{x}_i}$. From Lemma 5.3, we have $\mathcal{E}_i = G[\mathcal{E}_{i-1}] + \varepsilon_i, \mathcal{E}_0 = 0$. Thus we can bound

$$\|\mathcal{E}_i\|_{x^*} = \| \sum_{j=1}^{i} G^{i-j} \varepsilon_j \|_{x^*} \leq \sum_{j=1}^{i} \|\varepsilon_j\|_{x^*}.$$

Then using Lemma 5.6 to bound $\|c^*\|_2$, we obtain

$$\|\Delta_{\bar{x}_{c^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,x}}\|_{x^*} \leq \sqrt{\frac{\sum_{i=0}^{k} d^2(x_i, x_{i+1}) + \lambda}{(k+1)\lambda}} \Big(\sum_{i=0}^{k} \sum_{j=0}^{i} \|\varepsilon_j\|_{x^*} \Big) + \epsilon_3,$$

where $\epsilon_3 = O(d^3(x_0, x^*))$. □

5.D.8 Proof of Theorem 5.1

Proof of Theorem 5.1. Following the decomposition of error, we show

$$\begin{aligned}&d(\bar{x}_{c^*,x}, x^*) \\ &\leq d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*) + d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}}) + d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x}) \\ &\leq \frac{d(x_0, x^*)}{1-\sigma} \sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{d^2(x_0, x^*)} \|\hat{c}^*\|_2^2} + \frac{C_1 d(x_0, x^*)}{\lambda(1-\sigma)} \Big(\frac{2d(x_0, x^*)}{1-\sigma} \psi + (\psi)^2 \Big) \|\hat{c}^*\|_2 \\ &\quad + C_1 \sqrt{\frac{\sum_{i=0}^{k} d^2(x_i, x_{i+1}) + \lambda}{(k+1)\lambda}} \Big(\sum_{i=0}^{k} \sum_{j=0}^{i} \|\varepsilon_j\|_{x^*} \Big) + \epsilon_1 + \epsilon_2 + \epsilon_3.\end{aligned}$$

Now we maximize the bound over $\|\hat{c}^*\|$. From (Scieur et al., 2020, Proposition A.1), we see the maximum of a function $g(x) = c\sqrt{a - \bar{\lambda}x^2} + bx$ is $\sqrt{a}\sqrt{c^2 + \frac{b^2}{\bar{\lambda}}}$ where $\bar{\lambda} = \lambda / d^2(x_0, x^*)$. Let $a = (S_{k,\bar{\lambda}}^{[0,\sigma]})^2$, $b = \frac{C_1 d(x_0,x^*)}{\lambda(1-\sigma)}\Big(\frac{2d(x_0,x^*)}{1-\sigma}\psi + (\psi)^2\Big)$, $c = \frac{d(x_0,x^*)}{1-\sigma}$. We then obtain

$$
\begin{aligned}
d(\bar{x}_{c^*,x}, x^*) \leq S_{k,\bar{\lambda}}^{[0,\sigma]} & \sqrt{\frac{d^2(x_0,x^*)}{(1-\sigma)^2} + \frac{C_1^2 d^4(x_0,x^*)\Big(\frac{2d(x_0,x^*)}{1-\sigma}\psi + (\psi)^2\Big)^2}{\lambda^3(1-\sigma)^2}} \\
& + C_1\sqrt{\frac{\sum_{i=0}^k d^2(x_i,x_{i+1}) + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^{k}\sum_{j=0}^{i}\|\varepsilon_j\|_{x^*}\Big) + \epsilon_1 + \epsilon_2 + \epsilon_3,
\end{aligned}
$$

which completes the proof. □

5.D.9 Proof of Proposition 5.1

Proof of Proposition 5.1. Dividing the bound from Theorem 5.1 by $d(x_0, x^*)$ gives

$$
\begin{aligned}
\frac{d(\bar{x}_{c^*,x}, x^*)}{d(x_0,x^*)} \leq \frac{S_{k,\bar{\lambda}}^{[0,\sigma]}}{1-\sigma} & \sqrt{1 + O(d^{(2-3s)}(x_0,x^*)\Big(\frac{2d(x_0,x^*)}{1-\sigma}\psi + (\psi)^2\Big)^2} \\
& + C_1\sqrt{\frac{\sum_{i=0}^k d^2(x_i,x_{i+1})}{(k+1)O(d^s(x_0,x^*))} + \frac{1}{k+1}}\Big(\sum_{i=0}^{k}\sum_{j=0}^{i}\|\varepsilon_j\|_{x^*}\Big) \\
& + \frac{1}{d(x_0,x^*)}(\epsilon_1 + \epsilon_2 + \epsilon_3).
\end{aligned}
$$

By setting $\psi = O(d^2(x_0,x^*))$, the first term of RHS of the bound simplifies to $\frac{S_{k,\bar{\lambda}}^{[0,\sigma]}}{1-\sigma}\sqrt{1 + O(d^{(8-3s)}(x_0,x^*))}$, and similarly because $d(x_i, x_{i+1}) = O(d(x_0,x^*))$, $\|\varepsilon_j\|_{x^*} = O(d^2(x_0,x^*))$ under Assumption 5.1, the second term simplifies to $O(\sqrt{d^2(x_0,x^*) + d^{(4-s)}(x_0,x^*)})$ and the last term reduces to $O(d^2(x_0,x^*))$ as $\epsilon_1, \epsilon_2, \epsilon_3 = O(d^3(x_0,x^*))$. Hence we obtain

$$
\begin{aligned}
\frac{d(\bar{x}_{c^*,x}, x^*)}{d(x_0,x^*)} \leq \frac{S_{k,\bar{\lambda}}^{[0,\sigma]}}{1-\sigma} & \sqrt{1 + O(d^{(8-3s)}(x_0,x^*))} \\
& + O(\sqrt{d^2(x_0,x^*) + d^{(4-s)}(x_0,x^*)}) + O(d^2(x_0,x^*)).
\end{aligned}
$$

Finally we notice that the last two terms vanish when $d(x_0, x^*) \to 0$ for the choice of s. For the first term, given that when $d(x_0, x^*) \to 0$, $\bar{\lambda} = O(d^{(s-2)}(x_0, x^*)) \to 0$ and $O(d^{(8-3s)}(x_0, x^*)) \to 0$ for $s \in (2, \frac{8}{3})$, then

$$\lim_{d(x_0,x^*)\to 0} \frac{S^{[0,\sigma]}_{k,\bar{\lambda}}}{1-\sigma}\sqrt{1 + O(d^{(2-3s)}(x_0, x^*))} = \frac{S^{[0,\sigma]}_{k,0}}{1-\sigma} = \frac{1}{1-\sigma}\frac{2}{\beta^{-k}+\beta^k}$$

where $\beta = \frac{1-\sqrt{1-\sigma}}{1+\sqrt{1-\sigma}}$. This follows because without regularization, $S^{[0,\sigma]}_{k,0}$ reduces to the rescaled and shifted Chebyshev polynomial. See for example d'Aspremont et al. (2021). □

5.D.10 Proof of Lemma 5.9

Proof of Lemma 5.9. First, we write

$$\sum_{i=0}^{k} c_i \Delta_{x_i} = \Delta_{x_k} - \sum_{i=0}^{k-1} \theta_i (\Delta_{x_{i+1}} - \Delta_{x_i}).$$

By Lemma 5.1, we obtain

$$\begin{aligned} &d\Big(\mathrm{Exp}_{x^*}\big(\sum_{i=0}^{k} c_i\Delta_{x_i}\big), \mathrm{Exp}_{x_k}\big(-\Gamma_{x^*}^{x_k}\sum_{i=0}^{k-1}\theta_i(\Delta_{x_{i+1}} - \Delta_{x_i})\big)\Big) \\ &\quad\le d(x_k, x^*)C_\kappa\Big(d(x_k, x^*) + \|\sum_{i=0}^{k-1}\theta_i(\Delta_{x_{i+1}} - \Delta_{x_i})\|_{x^*}\Big) \\ &\quad\le d(x_k, x^*)C_\kappa\Big(d(x_k, x^*) + \sum_{i=0}^{k-1}\theta_i(d(x_{i+1}, x^*) + d(x_i, x^*))\Big), \end{aligned}$$

where we use the fact that $C_\kappa(x)$ is increasing for $x > 0$. In addition, from Lemma 5.2,

$$\begin{aligned} &d\Big(\bar{x}_{c,x}, \mathrm{Exp}_{x_k}\big(-\Gamma_{x^*}^{x_k}\sum_{i=0}^{k-1}\theta_i(\Delta_{x_{i+1}} - \Delta_{x_i})\big)\Big) \\ &\le C_1\|\sum_{i=0}^{k-1}\theta_i\big(\Gamma_{x^*}^{x_k}(\Delta_{x_{i+1}} - \Delta_{x_i}) - \Gamma_{x_i}^{x_k}\mathrm{Exp}_{x_i}^{-1}(x_{i+1})\big)\|_{x_k} \end{aligned}$$

$$\leq C_1 \sum_{i=0}^{k-1} \theta_i \|\Delta_{x_{i+1}} - \Delta_{x_i} - \Gamma_{x_k}^{x^*}\Gamma_{x_i}^{x_k}\mathrm{Exp}_{x_i}^{-1}(x_{i+1})\|_{x^*}.$$

Using Lemma 5.12, we obtain

$$\begin{aligned}\|\Delta_{x_{i+1}} - \Delta_{x_i} - \Gamma_{x_k}^{x^*}\Gamma_{x_i}^{x_k}\mathrm{Exp}_{x_i}^{-1}(x_{i+1})\|_{x^*} &\leq C_0 d(x_i, x_k) d(x_k, x^*) d(x_i, x_{i+1}) \\ &\quad + C_2 d(x_i, x^*) C_\kappa\big(d(x_i, x^*) + d(x_i, x_{i+1})\big).\end{aligned}$$

Let $e = \Delta_{\bar{x}_{c,x}} - \sum_{i=0}^k c_i \Delta_{x_i}$. Now combining the above results gives

$$\begin{aligned}&\|e\|_{x^*} = \|\Delta_{\bar{x}_{c,x}} - \sum_{i=0}^k c_i \Delta_{x_i}\|_{x^*} \\ &\leq C_2 d\Big(\bar{x}_{c,x}, \mathrm{Exp}_{x^*}\big(\sum_{i=0}^k c_i \Delta_{x_i}\big)\Big) \\ &\leq C_2 d\Big(\bar{x}_{c,x}, \mathrm{Exp}_{x_k}\big(-\Gamma_{x^*}^{x_k}\sum_{i=0}^{k-1}\theta_i(\Delta_{x_{i+1}} - \Delta_{x_i})\big)\Big) \\ &\qquad + C_2 d\Big(\mathrm{Exp}_{x^*}\big(\sum_{i=0}^k c_i \Delta_{x_i}\big), \mathrm{Exp}_{x_k}\big(-\Gamma_{x^*}^{x_k}\sum_{i=0}^{k-1}\theta_i(\Delta_{x_{i+1}} - \Delta_{x_i})\big)\Big) \\ &\leq C_2 C_1 \sum_{i=0}^{k-1}\theta_i\Big(C_0 d(x_i, x_k) d(x_k, x^*) d(x_i, x_{i+1}) + C_2 d(x_i, x^*) C_\kappa\big(d(x_i, x^*) + d(x_i, x_{i+1})\big)\Big) \\ &\qquad + C_2 d(x_k, x^*) C_\kappa\Big(d(x_k, x^*) + \sum_{i=0}^{k-1}\theta_i\big(d(x_{i+1}, x^*) + d(x_i, x^*)\big)\Big).\end{aligned}$$

Under Assumption 5.1 and $C_\kappa(x) = O(x^2)$, we see $\|e\|_{x^*} = O(d^3(x_0, x^*))$. □

5.D.11 Proof of Lemma 5.10

Proof of Lemma 5.10. Let $D(x) := \frac{1}{2}\sum_{i=0}^k c_i d^2(x, x_i)$. Then we can show $\mathrm{grad} D(x) = -\sum_{i=0}^k c_i \mathrm{Exp}_x^{-1}(x_i)$. See for example Alimisis et al. (2020). By the first-order stationarity,

$$\mathrm{grad} D(\bar{x}_{c,x}) = -\sum_{i=0}^k c_i \mathrm{Exp}_{\bar{x}_{c,x}}^{-1}(x_i) = 0$$

and $\mathrm{grad} D(x^*) = -\sum_{i=0}^k c_i \mathrm{Exp}_{x^*}^{-1}(x_i)$.

The first claim that $d(\bar{x}_{c,x}, x^*) \leq \|\sum_{i=0}^k c_i \Delta_{x_i}\|_{x^*}$ follows from (Tripuraneni et al., 2018, Lemma 10) and we include here for completeness. Define a real-valued function $g(t) := D\big(\mathrm{Exp}_{x^*}(t\eta)\big)$ with $\eta = \frac{\Delta_{\bar{x}_{c,x}}}{\|\Delta_{\bar{x}_{c,x}}\|_{x^*}}$. Under the assumption and definition of geodesic μ-strongly convex, we see $g(t)$ is μ-strongly convex in t. Thus, we have $g'(t_0) - g'(0) \geq \mu t_0$ for any t_0. Let $t_0 = \|\Delta_{\bar{x}_{c,x}}\|_{x^*}$ and denote the geodesic $\gamma(t) := \mathrm{Exp}_{x^*}(t\eta)$. We derive that $g'(t) = \langle \mathrm{grad} D(\mathrm{Exp}_{x^*}(t\eta)), \gamma'(t)\rangle$ by chain rule. Then $g'(t_0) = \langle \mathrm{grad} D(\bar{x}_{c,x}), \gamma'(t_0)\rangle_{\bar{x}_{c,x}} = 0$ and $g'(0) = \langle \mathrm{grad} D(x^*), \eta\rangle$. Finally, we see

$$\|\mathrm{grad} D(x^*)\|_{x^*}^2 \geq (g'(0))^2 = (g'(t_0) - g'(0))^2 \geq \mu^2 t_0^2 = \mu^2 \|\Delta_{\bar{x}_{c,x}}\|_{x^*}^2,$$

where the first inequality is due to Cauchy–Schwarz inequality. The first claim is proved by noticing $\|\mathrm{grad} D(x^*)\|_{x^*} = \|\sum_{i=0}^k c_i \Delta_{x_i}\|_{x^*}$ and $\|\Delta_{\bar{x}_{c,x}}\|_{x^*} = d(\bar{x}_{c,x}, x^*)$.

For the second claim, we first observe from the proof of Lemma 5.12 that

$$\begin{aligned}
&\|\mathrm{Exp}_{\bar{x}_{c,x}}^{-1}(x_i) - \Gamma_{x^*}^{\bar{x}_{c,x}}\big(\mathrm{Exp}_{x^*}^{-1}(x_i) - \mathrm{Exp}_{x^*}^{-1}(\bar{x}_{c,x})\big)\|_{\bar{x}_{c,x}} \\
&\leq C_2 d(\bar{x}_{c,x}, x^*) C_\kappa\big(d(\bar{x}_{c,x}, x^*) + d(\bar{x}_{c,x}, x_i)\big) \\
&= O(d^3(x_0, x^*)),
\end{aligned}$$

where the order is based on that $d(\bar{x}_{c,x}, x^*) \leq \frac{1}{\mu}\sum_{i=0}^k c_i d(x_i, x^*) = O(d(x_0, x^*))$ from the first claim. Letting $\bar{\varepsilon} := \mathrm{Exp}_{\bar{x}_{c,x}}^{-1}(x_i) - \Gamma_{x^*}^{\bar{x}_{c,x}}\big(\mathrm{Exp}_{x^*}^{-1}(x_i) - \mathrm{Exp}_{x^*}^{-1}(\bar{x}_{c,x})\big)$, we obtain $\|\bar{\varepsilon}\|_{\bar{x}_{c,x}} = O(d^3(x_0, x^*))$. From the first order stationarity, we see

$$\begin{aligned}
0 = \sum_{i=0}^k c_i \mathrm{Exp}_{\bar{x}_{c,x}}^{-1}(x_i) &= \sum_{i=0}^k c_i \Big(\Gamma_{x^*}^{\bar{x}_{c,x}}\big(\mathrm{Exp}_{x^*}^{-1}(x_i) - \mathrm{Exp}_{x^*}^{-1}(\bar{x}_{c,x})\big) + \bar{\varepsilon}\Big) \\
&= \Gamma_{x^*}^{\bar{x}_{c,x}}\Big(\sum_{i=0}^k c_i \Delta_{x_i} - \Delta_{\bar{x}_{c,x}}\Big) + \bar{\varepsilon}.
\end{aligned}$$

Taking the norm and using the isometry of parallel transport, we obtain the desired result. □

5.E Proofs under general retraction and vector transport

Discussions on the assumptions. Before we prove the results, we discuss the assumptions made for the general case. In particular, Assumption 5.4 is required to bound the deviation from the retraction to the exponential map, which can be considered natural given retraction approximates the exponential map to the first-order. In fact, Assumption 5.4 has been commonly used in Sato et al. (2019); Kasai et al. (2018b); Han & Gao (2021) for analyzing Riemannian first-order algorithms using retraction and can be satisfied for a sufficiently small neighbourhood (see for example Ring & Wirth (2012); W. Huang, Absil, & Gallivan (2015)). Similarly, Assumption 5.5 is used to bound the deviation between the vector transport to parallel transport, which is also standard in W. Huang, Gallivan, & Absil (2015); Kasai et al. (2018b); Han & Gao (2021). One can follow the procedures in W. Huang, Gallivan, & Absil (2015) to construct isometric vector transport that satisfies such condition for common manifolds like SPD manifold (W. Huang, Gallivan, & Absil, 2015), Stiefel and Grassmann manifold (W. Huang, 2013).

Here we show that when we use general retraction Retr in place of the exponential map Exp, thus invalidating the lemmas on metric distortion (Lemma 5.1, 5.2), we can still show a similar result as Lemma 5.4 but with an error on the order of $O(d^2(x_0, x^*))$ instead of $O(d^3(x_0, x^*))$ as for the case of exponential map. The main idea of proof follows from Tripuraneni et al. (2018). The next proposition formalizes such claim. For this section, we denote $\Delta_x = \mathrm{Retr}_{x^*}^{-1}(x)$ for any $x \in \mathcal{X}$ where the retraction has a smooth inverse. For general retraction, the deviation is on the order of $O(\|\Delta_{x_0}\|_{x^*}^2) = O(d^2(x_0, x^*))$ where we use the fact that retraction approximates the exponential map to the first order.

Proposition 5.3. *Suppose all iterates $x_i \in \mathcal{X}$, a neighbourhood where retraction has* *a smooth inverse. Consider the weighted average* $\bar{x}_{c,x} = \tilde{x}_k$ *given by* (Avg.1) *with retraction. Assume the sequence of iterates is non-divergent, i.e.* $\|\Delta_{x_i}\|_{x^*}, \|\Delta_{\tilde{x}_i}\|_{x^*} = O(\|\Delta_{x_0}\|_{x^*})$*. Then we have* $\Delta_{\bar{x}_{c,x}} = \sum_{i=0}^{k} c_i \Delta_{x_i} + e$*, with* $\|e\|_{x^*} = O(\|\Delta_{x_0}\|_{x^*}^2)$*,*

Proof. The proof generalize the proof of (Tripuraneni et al., 2018, Lemma 12). First denote $\mathrm{Retr}_x^y := \mathrm{Retr}_y^{-1} \circ \mathrm{Retr}_x$ and we notice that

$$\begin{aligned}
\Delta_{\tilde{x}_{i+1}} = \mathrm{Retr}_{x^*}^{-1}(\tilde{x}_{i+1}) &= \mathrm{Retr}_{x^*}^{-1}\Big(\mathrm{Retr}_{\tilde{x}_i}\big(\gamma_{i+1}\mathrm{Retr}_{\tilde{x}_i}^{-1}(x_{i+1})\big)\Big) \\
&= \mathrm{Retr}_{\tilde{x}_i}^{x^*}\Big(\gamma_{i+1}\mathrm{Retr}_{\tilde{x}_i}^{-1}\big(\mathrm{Retr}_{x^*}(\Delta_{x_{i+1}})\big)\Big) \\
&= \mathrm{Retr}_{\tilde{x}_i}^{x^*}\Big(\gamma_{i+1}\big(\mathrm{Retr}_{\tilde{x}_i}^{x^*}\big)^{-1}(\Delta_{x_{i+1}})\Big) \\
&= F(\Delta_{x_{i+1}}),
\end{aligned}$$

where we denote $\gamma_i = \frac{c_i}{\sum_{j=0}^{i} c_j}$ and $F : T_{x^*}\mathcal{M} \to T_{x^*}\mathcal{M}$ defined as

$$F(u) = \mathrm{Retr}_{\tilde{x}_i}^{x^*}\Big(\gamma_{i+1}\big(\mathrm{Retr}_{\tilde{x}_i}^{x^*}\big)^{-1}(u)\Big)$$

In addition, it can be verified that $F(\Delta_{\tilde{x}_i}) = \Delta_{\tilde{x}_i}$.

Now by chain rule, we have

$$\begin{aligned}
\mathrm{D}F(u) &= \mathrm{D}\mathrm{Retr}_{\tilde{x}_i}^{x^*}\Big(\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big)\Big[\mathrm{D}\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big] \\
&= \gamma_{i+1}\mathrm{D}\Big(\frac{1}{\gamma_{i+1}}\mathrm{Retr}_{\tilde{x}_i}^{x^*}\Big)\Big(\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big)\Big[\mathrm{D}\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big] \\
&= \gamma_{i+1}\Big(\mathrm{D}\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big)^{-1}\Big[\mathrm{D}\gamma_{i+1}(\mathrm{Retr}_{\tilde{x}_i}^{x^*})^{-1}(u)\Big] = \gamma_{i+1}\mathrm{id},
\end{aligned}$$

where the third inequality uses the inverse function theorem. Hence the Taylor expansion of F at $\Delta_{\tilde{x}_i}$ up to second order gives

$$\begin{aligned}
\Delta_{\tilde{x}_{i+1}} = F(\Delta_{x_{i+1}}) &= F(\Delta_{\tilde{x}_i}) + \gamma_{i+1}(\Delta_{x_{i+1}} - \Delta_{\tilde{x}_i}) + \tilde{e}_i \\
&= (1-\gamma_{i+1})\Delta_{\tilde{x}_i} + \gamma_{i+1}\Delta_{x_{i+1}} + \tilde{e}_i.
\end{aligned}$$

where we let $\tilde{\epsilon}_i = O(\|\Delta_{x_{i+1}} - \Delta_{\tilde{x}_i}\|^2_{x^*})$. From the expansion, it follows that $\Delta_{\tilde{x}_{i+1}} = \frac{\sum_{j=0}^{i} c_i}{\sum_{j=0}^{i+1} c_j}\Delta_{\tilde{x}_i} + \frac{c_{i+1}}{\sum_{j=0}^{i+1} c_j}\Delta_{x_{i+1}} + \tilde{\epsilon}_i$, which yields

$$(\sum_{j=0}^{i+1} c_j)\Delta_{\tilde{x}_{i+1}} = (\sum_{j=0}^{i} c_j)\Delta_{\tilde{x}_i} + c_{i+1}\Delta_{x_{i+1}} + (\sum_{j=0}^{i} c_j)\tilde{\epsilon}_i = \sum_{j=0}^{i+1} c_j \Delta_{x_j} + \sum_{j=0}^{i}(\sum_{\ell=0}^{j} c_\ell)\tilde{\epsilon}_j,$$

where the second equality follows by expanding the first equality. Let $i = k-1$, this leads to

$$\Delta_{\bar{x}_{c,x}} = \Delta_{\tilde{x}_k} = \sum_{j=0}^{k} c_j \Delta_{x_j} + e,$$

where we let $e = \sum_{j=0}^{k-1}(\sum_{\ell=0}^{j} c_\ell)\tilde{\epsilon}_j = O\big(\sum_{j=0}^{k-1}(\sum_{\ell=0}^{j} c_\ell)(\|\Delta_{x_{j+1}}\|^2_{x^*} + \|\Delta_{\tilde{x}_j}\|^2_{x^*})\big)$. We observe that $\|\Delta_{x_{i+1}}\|^2_{x^*} = O(\|\Delta_{x_0}\|^2_{x^*})$ and $\|\Delta_{\tilde{x}_j}\|^2_{x^*} = O(\|\Delta_{x_0}\|^2_{x^*})$ due to the non-divergent assumption. The proof is complete. □

5.E.1 Proof of Theorem 5.2

Theorem 5.2 (Restatement). Under Assumption 5.1, 5.3, 5.4 and 5.5, let $\{x_i\}_{i=0}^k$ be given by Riemannian gradient descent implemented via retraction, i.e., $x_i = \mathrm{Retr}_{x_{i-1}}(-\eta\,\mathrm{grad} f(x_{i-1}))$ and $\{\hat{x}_i\}_{i=0}^k$ be the linearized sequence that satisfies $\mathrm{Retr}_{x^*}^{-1}(\hat{x}_i) = G[\mathrm{Retr}_{x^*}^{-1}(\hat{x}_{i-1})]$ with $G = \mathrm{id} - \eta\,\mathrm{Hess} f(x^*)$, satisfying $\|G\|_{x^*} \leq \sigma < 1$. Then, using retraction and vector transport in Algorithm 6 and letting $\bar{x}_{c,x}$ be computed from (5.5), it satisfies that

$$\begin{aligned} & d(\bar{x}_{c^*,x}, x^*) \\ & \leq \|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} \frac{S_{k,\bar{\lambda}}^{[0,\sigma]}}{1-\sigma}\sqrt{\frac{1}{a_0^2} + \frac{C_1^2\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|^2_{x^*}\big(\frac{2\psi}{1-\sigma}\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} + \psi^2\big)^2}{\lambda^3}} \\ & \quad + C_1\sqrt{\frac{\sum_{i=0}^{k}\|\mathrm{Retr}_{x_i}^{-1}(x_{i+1})\|^2_{x_i} + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^{k}\sum_{j=0}^{i}\|\varepsilon_j\|_{x^*}\Big) + \epsilon_1 + \epsilon_2 + \epsilon_3, \end{aligned}$$

where $\psi = O(d^2(x_0, x^*))$, $\epsilon_1, \epsilon_2, \epsilon_3 = O(d^2(x_0, x^*))$ and $\varepsilon_i = O(d^2(x_i, x^*))$. Under the same choice of $\lambda = O(d^s(x_0, x^*))$, $s \in (2, \frac{8}{3})$, the same asymptotic optimal convergence rate (Proposition 5.1) holds.

Proof of Theorem 5.2. Here we only provide a sketch of proof because the main idea is exactly the same as the case of exponential map.

Under general retraction and vector transport, an analogue of Lemma 5.3 holds. That is,

$$\mathrm{Retr}_{x^*}^{-1}(x_i) = (\mathrm{id} - \eta \mathrm{Hess} f(x^*))[\mathrm{Retr}_{x^*}^{-1}(x_{i-1})] + \varepsilon_i, \tag{5.17}$$

where $\|\varepsilon_i\|_{x^*} = O(d^2(x_i, x^*))$. To show (5.17), we follow the exact same steps as the proof for Lemma 5.3 where we replace exponential map with retraction. The only difference is that the second order derivative is no longer the Riemann curvature tensor. In addition, we have shown in Proposition 5.3 that for retraction, we also have

$$\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c,x}) = \sum_{i=0}^{k} c_i \mathrm{Retr}_{x^*}^{-1}(x_i) + e \tag{5.18}$$

with $\|e\|_{x^*} = O(d^2(x_0, x^*))$.

Further, we still consider the same error bound decomposition, i.e.,

$$d(\bar{x}_{c^*,x}, x^*) \le d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*) + d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}}) + d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x}).$$

(I). For the linear term $d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*)$, we first see the linearized iterates $\hat{x}_i$ enjoys the same convergence as in Lemma 5.13 that

$$\| \sum_{i=0}^{k} \hat{c}_i^* \mathrm{Retr}_{x^*}^{-1}(\hat{x}_i) \|_{x^*} \le \frac{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{1-\sigma} \sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^2} \|\hat{c}^*\|_2^2}, \tag{5.19}$$

where $\bar{\lambda} := \lambda / \|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^2$ and we use Assumption 5.4. Combining (5.19) with (5.18) yields

$$\begin{aligned} d(\bar{x}_{\hat{c}^*,\hat{x}}, x^*) &\leq \frac{1}{a_0}\|\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{\hat{c}^*,\hat{x}})\|_{x^*} \\ &\leq \|\sum_{i=0}^{k} \hat{c}_i^* \mathrm{Retr}_{x^*}^{-1}(\hat{x}_i)\|_{x^*} + \epsilon_1, \\ &\leq \frac{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{a_0(1-\sigma)} \sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^2}\|\hat{c}^*\|_2^2} + \epsilon_1, \end{aligned}$$

with $\epsilon_1 = O(d^2(x_0, x^*))$.

(II). For the stability term $d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}})$, we first use Assumption 5.4 to show

$$\begin{aligned} &\|\Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}} - \big(\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{\hat{c}^*,\hat{x}}) - \mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,\hat{x}})\big)\|_{x^*} \\ &\leq a_2\|\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{\hat{c}^*,\hat{x}})\|_{x^*}^2 + a_2\|\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,\hat{x}}))\|_{x^*}^2 \\ &\leq a_2 a_1^2\big(d^2(\bar{x}_{\hat{c}^*,\hat{x}}, x^*) + d^2(\bar{x}_{c^*,\hat{x}}, x^*)\big) \\ &= O(d^2(x_0, x^*)). \end{aligned}$$

Let $\epsilon_r := \Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}} - \big(\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{\hat{c}^*,\hat{x}}) - \mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,\hat{x}})\big)$. Then we have $\|\epsilon_r\|_{x^*} = O(d^2(x_0, x^*))$. In addition, based on Assumption 5.5, we show

$$\begin{aligned} &\|\mathcal{T}_{x_k}^{x^*} r_i - \big(\mathrm{Retr}_{x^*}^{-1}(x_{i+1}) - \mathrm{Retr}_{x^*}^{-1}(x_i)\big) - \Gamma_{x_k}^{x^*} r_i + \big(\Delta_{x_{i+1}} - \Delta_{x_i}\big)\|_{x^*} \\ &\leq \|\mathcal{T}_{x_k}^{x^*} r_i - \Gamma_{x_k}^{x^*} r_i\|_{x^*} + O(d^2(x_0, x^*)) = O(d^2(x_0, x^*)), \end{aligned}$$

where we use Assumption 5.4, 5.5 and notice $\|r_i\|_{x_i} = \|\mathrm{Retr}_{x_i}^{-1}(x_{i+1})\|_{x_i} \leq a_1 d(x_i, x_{i+1}) = O(d(x_0, x^*))$. Let $\epsilon_v := \mathcal{T}_{x_k}^{x^*} r_i - \big(\mathrm{Retr}_{x^*}^{-1}(x_{i+1}) - \mathrm{Retr}_{x^*}^{-1}(x_i)\big) - \Gamma_{x_k}^{x^*} r_i + \big(\Delta_{x_{i+1}} - \Delta_{x_i}\big)$, we have $\|\epsilon_v\|_{x^*} = O(d^2(x_0, x^*))$.

Using Lemma 5.2, we then obtain

$$\begin{aligned} d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}}) &\leq C_1\|\Delta_{\bar{x}_{\hat{c}^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,\hat{x}}}\|_{x^*} \\ &\leq C_1\|\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{\hat{c}^*,\hat{x}}) - \mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,\hat{x}})\|_{x^*} + C_1\|\epsilon_r\|_{x^*} \end{aligned}$$

$$\leq \frac{C_1\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{1-\sigma}\|c^* - \hat{c}^*\|_2 + O(d^2(x_0, x^*)),$$

where we apply (5.18). Now we proceed to bound $\|c^* - \hat{c}^*\|_2 \leq \frac{\|P\|_2}{\lambda}\|\hat{c}^*\|_2$ in a similar manner as Lemma 5.6 where $P = R - \hat{R}$. From the proof of Lemma 5.6, we have

$$\|P\|_2 \leq \frac{2}{1-\sigma}\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}\|E\|_2 + \|E\|_2^2,$$

where $\|E\|_2 \leq \sum_{i=0}^{k}\|\mathcal{T}_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*}$. Thus it remains to bound $\|\mathcal{T}_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*}$. Similarly, we can show

$$\begin{aligned}\|\mathcal{T}_{x_k}^{x^*} r_i - \hat{r}_i\|_{x^*} &\leq \|\mathcal{T}_{x_k}^{x^*} r_i - \big(\mathrm{Retr}_{x^*}^{-1}(x_{i+1}) - \mathrm{Retr}_{x^*}^{-1}(x_i)\big)\|_{x^*} + \sum_{j=1}^{i+1}\|\varepsilon_j\|_{x^*} \\ &\leq \|\Gamma_{x_k}^{x^*} r_i - \big(\Delta_{x_{i+1}} - \Delta_{x_i}\big)\|_{x^*} + \|\epsilon_v\|_{x^*} + \sum_{j=1}^{i+1}\|\varepsilon_j\|_{x^*} = O(d^2(x_0, x^*)),\end{aligned}$$

where ε_j is defined in (5.17) and we use Lemma 5.12 for the exponential map. Thus $\|P\|_2 \leq 2\psi\frac{a_1}{1-\sigma}d(x_0, x^*) + \psi^2$ where $\psi = O(d^2(x_0, x^*))$. This leads to

$$d(\bar{x}_{\hat{c}^*,\hat{x}}, \bar{x}_{c^*,\hat{x}}) \leq \frac{C_1\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{\lambda(1-\sigma)}\Big(\frac{2\psi}{1-\sigma}\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} + \psi^2\Big)\|\hat{c}^*\|_2 + \epsilon_2,$$

where $\epsilon_2 = O(d^2(x_0, x^*))$.

(III). Finally for the nonlinearity term $d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x})$, we show

$$\begin{aligned}d(\bar{x}_{c^*,\hat{x}}, \bar{x}_{c^*,x}) &\leq C_1\|\Delta_{\bar{x}_{c^*,\hat{x}}} - \Delta_{\bar{x}_{c^*,x}}\|_{x^*} \\ &\leq C_1\|\mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,\hat{x}}) - \mathrm{Retr}_{x^*}^{-1}(\bar{x}_{c^*,x})\|_{x^*} + O(d^2(x_0, x^*)) \\ &\leq C_1\|c^*\|_2(\sum_{i=0}^{k}\|\mathrm{Retr}_{x^*}^{-1}(x_i) - \mathrm{Retr}_{x^*}^{-1}(\hat{x}_i)\|_{x^*}) + O(d^2(x_0, x^*)) \\ &\leq C_1\sqrt{\frac{\sum_{i=0}^{k}\|\mathrm{Retr}_{x_i}^{-1}(x_{i+1})\|_{x_i}^2 + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^{k}\sum_{j=0}^{i}\|\varepsilon_j\|_{x^*}\Big) + \epsilon_3,\end{aligned}$$

where $\epsilon_3 = O(d^2(x_0, x^*))$ and we follow similar steps as in Lemma 5.6.

Finally, combining results from (I), (II), (III), we have

$$
\begin{aligned}
d(\bar{x}_{c^*,x}, x^*) \leq{} & \frac{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{a_0(1-\sigma)} \sqrt{(S_{k,\bar{\lambda}}^{[0,\sigma]})^2 - \frac{\lambda}{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^2}\|\hat{c}^*\|_2^2} \\
& + \frac{C_1\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}}{\lambda(1-\sigma)}\Big(\frac{2\psi}{1-\sigma}\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} + \psi^2\Big)\|\hat{c}^*\|_2 \\
& + C_1\sqrt{\frac{\sum_{i=0}^k \|\mathrm{Retr}_{x_i}^{-1}(x_{i+1})\|_{x_i}^2 + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^k\sum_{j=0}^i \|\varepsilon_j\|_{x^*}\Big) + \epsilon_1 + \epsilon_2 + \epsilon_3.
\end{aligned}
$$

Maximizing the bound over $\|\hat{c}^*\|_2$ yields

$$
\begin{aligned}
& d(\bar{x}_{c^*,x}, x^*) \\
& \leq S_{k,\bar{\lambda}}^{[0,\sigma]}\sqrt{\frac{\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^2}{a_0^2(1-\sigma)^2} + \frac{C_1^2\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*}^4\big(\frac{2\psi}{1-\sigma}\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} + \psi^2\big)^2}{\lambda^3(1-\sigma)^2}} \\
& \quad + C_1\sqrt{\frac{\sum_{i=0}^k \|\mathrm{Retr}_{x_i}^{-1}(x_{i+1})\|_{x_i}^2 + \lambda}{(k+1)\lambda}}\Big(\sum_{i=0}^k\sum_{j=0}^i \|\varepsilon_j\|_{x^*}\Big) + \epsilon_1 + \epsilon_2 + \epsilon_3.
\end{aligned}
$$

Finally, to see the asymptotic convergence rate, we notice that $\|\mathrm{Retr}_{x^*}^{-1}(x_0)\|_{x^*} = O(d(x_0, x^*))$ and $\lim_{d(x_0,x^*)\to 0}\frac{1}{d(x_0,x^*)}(\epsilon_1+\epsilon_2+\epsilon_3) = 0$. □

5.F Extensions

In this section, we consider various extensions to the proposed nonlinear acceleration on manifolds.

5.F.1 Online Riemannian nonlinear acceleration

Following Scieur et al. (2018); Bollapragada et al. (2022), we can extend Algorithm 6 to the online setting, where the extrapolated point $\bar{x}_{c,x}$ is used to update the iterate sequence. The idea is to add a mixing step by updating $\bar{x}_{c,x}$ in the direction of the weighted average of the gradients, i.e., $\overline{\mathrm{grad}} f(\bar{x}_{c,x}) = \sum_{i=0}^k c_i \Gamma_{x_i}^{\bar{x}_{c,x}} \mathrm{grad} f(x_i)$. For the averaging schemes (Avg.1), (Avg.3), the next it-

Algorithm 13: Riemannian nonlinear acceleration (RiemNA-online)

```
Input: Initialization $x_0$. Regularization parameter $\lambda$. Mixing parameter $\delta$.
for $k = 0, ..., K-1$ do
    Compute $r_i = \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) \in T_{x_k}\mathcal{M}, i = 0, ..., k$
    Solve $c^* = \arg\min_{c \in \mathbb{R}^{k+1}: c^\top 1 = 1} \| \sum_{i=0}^k c_i r_i \|_{x_k}^2 + \lambda \|c\|_2^2$.
    Compute $x_{k+1} =$
    $\mathrm{Exp}_{x_k}\big( -\delta c_k^* \mathrm{grad} f(x_k) - \sum_{i=0}^{k-1} \Gamma_{x_i}^{x_k} \big(\theta_i^* \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) + \delta c_i^* \mathrm{grad} f(x_i)\big)\big)$,
    where $\theta_i^* = \sum_{j=0}^i c_j^*$.
end for
Output: $x_K$.
```

eration starts with $\mathrm{Exp}_{\bar{x}_{c,x}}(-\delta \overline{\mathrm{grad}} f(\bar{x}_{c,x}))$ for some mixing parameter $\delta > 0$. Particularly for the tangent space averaging scheme (Avg.2), we show a more efficient strategy of mixing, which we focus in this work. The averaging and mixing steps are both performed on the same tangent space. Specifically, let $x_{-1} = x_0$, we define the following progression of the online nonlinear acceleration on manifolds.

$$
\begin{aligned}
x_{k+1} &= \mathrm{Exp}_{x_k}\Big(-\sum_{i=0}^{k-1} \theta_i \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) - \delta \sum_{i=0}^{k} c_i \Gamma_{x_i}^{x_k} \mathrm{grad} f(x_i) \Big) \\
&= \mathrm{Exp}_{x_k}\Big(-\delta c_k \mathrm{grad} f(x_k) - \sum_{i=0}^{k-1} \Gamma_{x_i}^{x_k} \big(\theta_i \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) + \delta c_i \mathrm{grad} f(x_i)\big) \Big).
\end{aligned}
$$

The complete procedures are presented in Algorithm 13.

5.F.2 Practical considerations

Here are some practical considerations to use nonlinear acceleration on manifolds.

Iterates from Riemannian gradient descent with line-search Suppose the iterates $\{x_i\}_{i=0}^k$ are generated from $x_i = \mathrm{Exp}_{x_i}(-\eta_i \mathrm{grad} f(x_{i-1}))$ where the stepsize is determined from a line-search procedure (such as backtracking line-search in Boumal et al. (2019b)) and thus varies across iterations. Nevertheless, Lemma

Algorithm 14: Adaptive regularized Riemannian nonlinear acceleration (AdaRiemNA)

```
Input: A sequence of iterates $x_0, ..., x_{k+1}$. Tentative regularization parameters $\{\lambda_j\}_{j=1}^k$.
Compute $r_i = \Gamma_{x_i}^{x_k} \mathrm{Exp}_{x_i}^{-1}(x_{i+1}) \in T_{x_k}\mathcal{M}, i = 0, ..., k$
for $j = 1, ..., k$ do
    Solve $c^*(\lambda_j) = \arg\min_{c \in \mathbb{R}^{k+1}: c^\top 1 = 1} \| \sum_{i=0}^k c_i r_i \|_{x_k}^2 + \lambda_j \|c\|_2^2$.
    Compute $\bar{x}(\lambda_j) = \bar{x}_{c,x}$ using $c^*(\lambda_j)$.
end for
Set $\bar{x}^* = \arg\min_{j=1,...,k} f(\bar{x}(\lambda_j))$.
Compute $u = \mathrm{Exp}_{x_0}^{-1}(\bar{x}^*)$ and set $t = 1$.
while $f(\mathrm{Exp}_{x_0}(2tu)) < f(\mathrm{Exp}_{x_0}(tu))$ do
    Update $t = 2t$.
end while
Output: $\mathrm{Exp}_{x_0}(tu)$.
```

5.3 still holds with $G_i = \mathrm{id} - \eta_i \mathrm{Hess} f(x^*)$. Suppose the stepsize is chosen such that $\|G_i\| \leq \sigma < 1$. Then the analysis still holds under this setting.

Safeguarding decrease Due to the curved geometry of the manifold and nonlinearity of the objective function, it is not guaranteed that $f(\bar{x}_{c,x})$ will decrease. In the main text, we only show local convergence of the acceleration strategy. A typical globalization technique is to only keep the extrapolated point if it shows sufficient decrease compared to previous iterates, i.e., $f(\bar{x}_{c,x}) \leq \tau \min_{i=0,...,k} f(x_i)$ for some $\tau < 1$. In Scieur et al. (2020), an adaptive regularization strategy has been proposed to select regularization parameter λ. Here we adapt the same strategy on manifolds, which we show in Algorithm 14. As noticed in Scieur et al. (2020), a higher value of λ pushes the weights close to uniform and thus stays closer to x_0. Thus the line-search over t tries to enhance the progress compared to the initialization. In addition, for online Riemannian nonlinear acceleration specifically, we may consider performing a line-search over the parameter δ to ensure a sufficient descent condition is met.

Limited-memory and extrapolation frequency Rather than keeping all the previous iterates for extrapolation, we can set a memory depth of m and using only the most recent m iterates to compute the extrapolated point. In practice, m is usually set to be less than 10. In addition, we notice that compared to the Euclidean version, the computational cost for the Riemannian nonlinear acceleration can be high due to the use of parallel transport. Hence to mitigate this issue, we may only compute the extrapolated point every m iteration.

Efficient update of the residual matrix R Recall for each application of Riemannian nonlinear acceleration, we need to compute $R = [\langle r_i, r_j \rangle_{x_k}]_{0 \leq i,j \leq k}$, where $r_i = \mathcal{T}_{x_i}^{x_k} \mathrm{Retr}_{x_i}^{-1}(x_{i+1})$, where we write using (isometric) vector transport and general retraction. This includes parallel transport and exponential map as special cases. By isometry, in the next iteration when we receive r_{k+1}, the update of R only requires computing $\langle \Gamma_{x_k}^{x_{k+1}} r_i, r_{k+1} \rangle_{x_{k+1}}$, $i = 0, ..., k+1$. Denote the vector $r_+ := [\langle \Gamma_{x_k}^{x_{k+1}} r_i, r_{k+1} \rangle_{x_{k+1}}]_{0 \leq i \leq k}$. Then the updated residual matrix is

$$R_+ = \begin{bmatrix} R & r_+ \\ r_+^\top & \|r_{k+1}\|_{x_{k+1}}^2 \end{bmatrix}.$$

Chapter 6

Conclusions

The thesis advances the developments of Riemannian optimization, in terms of understanding Riemannian geometry in algorithmic performance, improving and unifying variance reduction methods, as well as designing a generic acceleration strategy on manifolds. This chapter recaps and summarizes the achievements of the three main chapters within a larger context. Then the chapter concludes by providing some perspectives into the future research directions.

6.1 Summary of the thesis

Recent years have seen many great developments in the field of optimization on manifolds. This includes improvements on algorithmic designs, exploration of general problem settings and introduction of a range of software and packages, thus expanding the potential of Riemannian optimization for both researchers and practitioners.

A majority of research efforts focus on proposing more advanced numerical algorithms on manifolds by generalizing the ideas in the Euclidean space. Successful generalizations often exploit the function structure (e.g., geodesic convexity), gradient sequence (e.g., variance reduction methods) and iterate sequence (e.g., averaging and interpolation), which are primarily characteristics of the problem instances.

While the contributions are fruitful and promising, the most commonly used Riemannian metric is often taken for granted when implementing the proposed algorithms, despite that different choices may result in fundamentally different search spaces. Chapter 3 aims to study how the choices of Riemannian metric affects the numerical performance of optimization algorithms on manifolds. We particularly focus on the SPD manifold, for which we identify two aspects where the metric impacts the algorithmic performance, i.e., the curvature and the conditioning of Riemannian Hessian. The former plays a crucial role in bounding the side lengths of a geodesic triangle, and thus naturally appears in the convergence rate for first-order algorithms. The latter determines the shape of level curves, which regulate both the directions of gradient and Newton steps. Among the many available metrics for SPD matrices, we show the Bures-Wasserstein metric is favourable than the default Affine-Invariant metric in both of the two aspects, for a wide range of function classes. In many cases, we observe the empirical speedup on the Bures-Wasserstein geometry is significant, which suggests an equal attention should be placed on choosing or designing proper Riemannian metrics, orthogonal to the algorithmic developments.

The theory of variance reduction has been central for classic statistical estimation, particularly for Monte Carlo methods. This serves as the building block for the rise of variance reduction methods for stochastic optimization in the last decade. Indeed, most variance reduced gradient methods, such as SVRG (Johnson & Zhang, 2013) and SAGA (Defazio et al., 2014) are inspired by the idea of control variate, a key technique to reduce variance of a random variable for Monte Carlo methods. Looking back through the history of variance reduction for optimization, a majority of methods are originally designed for (strongly) convex problems. The journey to understand the behaviours of variance reduction methods in the nonconvex regime starts with the analysis of SVRG (Reddi, Hefny, et al., 2016), showing an improved gradient complexity compared to gradient descent and stochastic gradient descent. It is later discovered that the complexity can be further improved in SPIDER (Fang et al., 2018) if the stochastic gradient correction is performed by consecutive iterates instead of anchored iterates each epoch. Such scheme is proved to be optimal in gradient complexity for both finite-sum and online settings.

Parallel to the above developments, variance reduction has also been considered for optimization on manifolds, with the resulting algorithms matching the complexity established in the Euclidean space, up to some manifold specific constants. However, a consistent and unified framework is absent for both analyzing and improving the Riemannian variance reduction methods. Chapter 4 offers such a framework for both SVRG-type and SPIDER-type methods, unifying the convergence analysis via gradient estimation bound and improving the complexity via batch size adaptation. This allows principled comparisons between different styles of variance reduction on manifolds, in terms of the use of general retraction and vector transport, the choice of stepsize and batch size, and the impacts of manifold curvature. Further, Chapter 4 also provides the necessary analysis toolkit for the many prospective developments of Riemannian variance reduction methods in more general contexts and problem settings.

Another long-lasting puzzle in the field of Riemannian optimization is acceleration, i.e., the possibility of generating accelerated sequences on manifolds using only first-order information. In the Euclidean space, Nesterov's accelerated gradient methods (Y. E. Nesterov, 1983; Y. Nesterov, 2003) provide an affirmative and inspiring answer to the above question, achieving optimal convergence rates among first-order methods. Nevertheless, the original analysis requires intricate proof strategies as well as algebraic tricks to select the parameters, obscuring the underlying intuition of the acceleration. Later many research efforts have approached the Nesterov acceleration from diverse perspectives, including a linear coupling framework (Allen-Zhu & Orecchia, 2017), continuous dynamics (Su et al., 2014), variational framework (Wibisono et al., 2016), proximal point method

(Defazio, 2019; Ahn & Sra, 2022), and continuized formulation (Even et al., 2021).

Generalizing Nesterov acceleration to Riemannian manifold presents significant challenges, because most of the analysis in the Euclidean space relies heavily on the vector space structure and careful choice of parameters. The curvature of the search space has been found to be the main culprit obstructing the global acceleration (Hamilton & Moitra, 2021; Criscitiello & Boumal, 2022b). Nonetheless, there are continuing efforts in contributing to the understanding of acceleration on manifolds, particularly in terms of Nesterov's type of acceleration. Such an understanding is crucial for both Riemannian optimization and Euclidean optimization as the insights generated may promote better design of accelerated algorithms on both domains.

In Chapter 5, we depart from the existing endeavors to develop Riemannian Nesterov accelerated gradient methods, and pursue a generic acceleration strategy by extrapolating the iterates produced from Riemannian first-order methods. We have proved an optimal asymptotic convergence rate of the proposed scheme, coupled with Riemannian gradient descent method. While our convergence guarantees are weaker than the existing works (with mostly non-asymptotic optimal rates), our method is simple to implement, efficient, and allows the use of more general retraction and vector transport. Despite the significance of Nesterov acceleration, the Chapter suggests new possibilities and promises for Riemannian acceleration by exploring alternative acceleration schemes.

6.2 Future research

Following on the developments of the thesis, there are many open questions and new directions that remain unexplored. Motivated by Chapter 5, it is worth extending the idea and analysis of extrapolation to other solvers, including momentum-based solvers and stochastic solvers. Also it is equally interesting to study alternative acceleration schemes on manifolds apart from the well-celebrated Nesterov acceleration, such as by taking inspiration from classical dynamical systems and control theory. The main aim is to improve the empirical efficiency, while showing non-asymptotic convergence guarantees. Another challenging but rewarding direction is on (global) complexity lower bounds of Riemannian optimization. This requires to construct "hard" functions on manifolds, which may be out of reach for arbitrary Riemannian manifolds. Hamilton & Moitra (2021); Criscitiello & Boumal (2022b) are successful to build such "hard" geodesic (strongly) convex functions on hyperbolic and more general Hadamard manifolds, respectively. The functions constructed largely deviate from the classic ones used in the Euclidean space for proving lower bounds. Some natural questions arise, such as whether the analysis can be adapted for establishing lower bounds for general nonconvex functions and min-max problems on manifolds. In addition, many settings of Riemannian optimization are currently underdeveloped, including min-max problems, compositional problems, bilevel problems, distributed settings, among many others. All the above are exciting new directions that worth investigation, which may lead to new applications of Riemannian optimization. Finally, given the versatility of the framework of Riemannian optimization and its ever-growing applications in data science, some imperative issues require special attention before deployment in practice, such as privacy, fairness, and interpretability. Future research efforts shall also build awareness on trustworthiness and transparency when learning and optimizing over Riemannian manifolds, by integrating existing frameworks established in the Euclidean space.

References

Absil, P.-A., Baker, C. G., & Gallivan, K. A. (2007). Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, *7*(3), 303–330.

Absil, P.-A., & Gallivan, K. A. (2006). Joint diagonalization on the oblique manifold for independent component analysis. In *International Conference on Acoustics Speech and Signal Processing Proceedings* (Vol. 5, pp. V–V).

Absil, P.-A., Mahony, R., & Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Agarwal, A., & Bottou, L. (2015). A lower bound for the optimization of finite sums. In *International Conference on Machine Learning* (pp. 78–86).

Agarwal, N., Boumal, N., Bullins, B., & Cartis, C. (2021). Adaptive regularization with cubics on manifolds. *Mathematical Programming*, *188*(1), 85–134.

Ahn, K., & Sra, S. (2020). From Nesterov's estimate sequence to Riemannian acceleration. In *Conference on Learning Theory* (pp. 84–118).

Ahn, K., & Sra, S. (2022). Understanding Nesterov's acceleration via proximal point method. In *Symposium on Simplicity in Algorithms (SOSA)* (pp. 117–130).

Aitken, A. C. (1927). On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, *46*, 289–305. doi: 10.1017/S0370164600022070

Alcalá-Fdez, J., Sánchez, L., Garcia, S., del Jesus, M. J., Ventura, S., Garrell, J. M., ... Rivas, V. M. (2009). KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, *13*(3), 307–318.

Alimisis, F., Orvieto, A., Bécigneul, G., & Lucchi, A. (2020). A continuous-time perspective for modeling acceleration in Riemannian optimization. In *International Conference on Artificial Intelligence and Statistics* (pp. 1297–1307).

Alimisis, F., Orvieto, A., Bécigneul, G., & Lucchi, A. (2021). Momentum improves optimization on Riemannian manifolds. In *International Conference on Artificial Intelligence and Statistics* (pp. 1351–1359).

Allen-Zhu, Z., & Orecchia, L. (2017). Linear coupling: An ultimate unification of gradient and mirror descent. In *Innovations in Theoretical Computer Science Conference (ITCS 2017).*

Al-Mohy, A. H., & Higham, N. J. (2009). Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM Journal on Matrix Analysis and Applications*, *30*(4), 1639–1657.

Ambrosio, L., Gigli, N., & Savaré, G. (2008). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

Andrews, B., & Hopper, C. (2010). *The Ricci flow in Riemannian geometry: a complete proof of the differentiable* $1/4$*-pinching sphere theorem*. Springer.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., & Woodworth, B. (2023). Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, *199*(1-2), 165–214.

Arsigny, V., Fillard, P., Pennec, X., & Ayache, N. (2007). Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, *29*(1), 328–347.

Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding* (Tech. Rep.). Stanford, CA: Stanford.

Babanezhad, R., Laradji, I. H., Shafaei, A., & Schmidt, M. (2018). MASAGA: A linearly-convergent stochastic first-order method for optimization on manifolds. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 344–359).

Balles, L., Romero, J., & Hennig, P. (2017). Coupling adaptive batch sizes with learning rates. In *Uncertainty in Artificial Intelligence (UAI)* (pp. 675–684).

Batmanghelich, N. K., Saeedi, A., Narasimhan, K. R., & Gershman, S. J. (2016). Nonparametric spherical topic modeling with word embeddings. In *Annual Meeting of the Association for Computational Linguistics* (pp. 537–542).

Becigneul, G., & Ganea, O.-E. (2019). Riemannian adaptive optimization methods. In *International Conference on Learning Representations.*

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in Neural Information Processing Systems*, *13*.

Bennett, J., & Lanning, S. (2007). The Netflix prize..

Bergmann, R. (2019). *Optimisation on Manifolds in Julia.* GitHub. (https://github.com/kellertuer/Manopt.jl)

Bhatia, R. (2009). *Positive definite matrices*. Princeton university press.

Bhatia, R., Jain, T., & Lim, Y. (2019). On the Bures-Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, *37*(2), 165–191.

Bhutani, M., Jawanpuria, P., Kasai, H., & Mishra, B. (2018). *Low-rank geometric mean metric learning.* ICML workshop on Geometry in Machine Learning (GiMLi).

Bini, D. A., & Iannazzo, B. (2013). Computing the karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, *438*(4), 1700–1710.

Bollapragada, R., Scieur, D., & d'Aspremont, A. (2022). Nonlinear acceleration of momentum and primal-dual algorithms. *Mathematical Programming*, 1–38.

Bonnabel, S. (2013). Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, *58*(9), 2217–2229.

Boothby, W. M. (1986). *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press.

Boumal, N. (2015). Riemannian trust regions with finite-difference Hessian approximations are globally convergent. In *International Conference on Geometric Science of Information* (pp. 467–475).

Boumal, N. (2023). *An introduction to optimization on smooth manifolds*. Cambridge University Press.

Boumal, N., & Absil, P.-A. (2011a). A discrete regression method on manifolds and its application to data on $SO(n)$. *IFAC Proceedings Volumes*, *44*(1), 2284–2289.

Boumal, N., & Absil, P.-a. (2011b). RTRMC: A Riemannian trust-region method for low-rank matrix completion. *Advances in Neural Information Processing Systems*, 406–414.

Boumal, N., & Absil, P.-A. (2015). Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, *475*, 200–239.

Boumal, N., Absil, P.-A., & Cartis, C. (2019a). Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, *39*(1), 1–33.

Boumal, N., Absil, P.-A., & Cartis, C. (2019b). Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, *39*(1), 1–33.

Boumal, N., Mishra, B., Absil, P.-A., & Sepulchre, R. (2014). Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, *15*(1), 1455–1459.

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brezinski, C., Redivo-Zaglia, M., & Saad, Y. (2018). Shanks sequence transformations and Anderson acceleration. *SIAM Review*, *60*(3), 646–669.

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv:2104.13478*.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18–42.

Brooks, D. A., Schwander, O., Barbaresco, F., Schneider, J.-Y., & Cord, M. (2019). Exploring complex time-series representations for Riemannian machine learning of radar data. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3672–3676).

Cai, T. T., & Zhang, A. (2015). ROP: Matrix recovery via rank-one projections. *Annals of Statistics*, *43*(1), 102–138.

Cannon, J. W., Floyd, W. J., Kenyon, R., & Parry, W. R. (1997). Hyperbolic geometry. *Flavors of geometry*, *31*(59-115), 2.

Chamberlain, B. P., Clough, J., & Deisenroth, M. P. (2017). Neural embeddings of graphs in hyperbolic space. *arXiv:1705.10359*.

Chami, I., Wolf, A., Juan, D.-C., Sala, F., Ravi, S., & Ré, C. (2020). Low-dimensional hyperbolic knowledge graph embeddings. In *Annual Meeting of the Association for Computational Linguistics* (pp. 6901–6914).

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 1–27.

Chebbi, Z., & Moakher, M. (2012). Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function. *Linear Algebra and its Applications*, *436*(7), 1872–1889.

Chen, S., Garcia, A., Hong, M., & Shahrampour, S. (2021). Decentralized Riemannian gradient descent on the Stiefel manifold. In *International Conference on Machine Learning* (pp. 1594–1605).

Chen, S., Ma, S., Man-Cho So, A., & Zhang, T. (2020). Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM Journal on Optimization*, *30*(1), 210–239.

Cherian, A., & Sra, S. (2016). Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 2859–2871.

Criscitiello, C., & Boumal, N. (2019). Efficiently escaping saddle points on manifolds. *Advances in Neural Information Processing Systems*, *32*.

Criscitiello, C., & Boumal, N. (2022a). An accelerated first-order method for non-convex optimization on manifolds. *Foundations of Computational Mathematics*, 1–77.

Criscitiello, C., & Boumal, N. (2022b). Negative curvature obstructs acceleration for strongly geodesically convex optimization, even with exact first-order oracles. In *Conference on Learning Theory* (pp. 496–542).

Cruceru, C., Becigneul, G., & Ganea, O.-E. (2021). Computationally tractable Riemannian manifolds for graph embeddings. In *AAAI Conference on Artificial Intelligence* (pp. 7133–7141).

De, S., Yadav, A., Jacobs, D., & Goldstein, T. (2017). Automated inference with adaptive batches. In *Artificial Intelligence and Statistics* (pp. 1504–1513).

Defazio, A. (2019). On the curved geometry of accelerated optimization. *Advances in Neural Information Processing Systems*, *32*.

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems*, 1646–1654.

Di Giovanni, F., Luise, G., & Bronstein, M. M. (2022). Heterogeneous manifolds for curvature-aware graph embedding. In *ICLR Workshop on Geometrical and Topological Representation Learning.*

Douik, A., & Hassibi, B. (2019). Manifold optimization over the set of doubly stochastic matrices: A second-order geometry. *IEEE Transactions on Signal Processing*, *67*(22), 5761–5774.

Dryden, I. L., Koloydenko, A., & Zhou, D. (2009). Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, *3*(3), 1102–1123.

Duruisseaux, V., & Leok, M. (2022). A variational formulation of accelerated optimization on Riemannian manifolds. *SIAM Journal on Mathematics of Data Science*, *4*(2), 649–674.

d'Aspremont, A., Scieur, D., & Taylor, A. (2021). Acceleration methods. *Foundations and Trends® in Optimization*, *5*(1-2), 1–245.

Edelman, A., Arias, T. A., & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, *20*(2), 303–353.

Eldén, L., & Park, H. (1999). A procrustes problem on the stiefel manifold. *Numerische Mathematik*, *82*(4), 599–619.

Even, M., Berthier, R., Bach, F., Flammarion, N., Gaillard, P., Hendrikx, H., ... Taylor, A. (2021). A continuized view on Nesterov acceleration for stochastic gradient descent and randomized gossip. *Advances in Neural Information Processing Systems*, 1–32.

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

Friedlander, M. P., & Schmidt, M. (2012). Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, *34*(3), A1380–A1405.

Gabay, D. (1982). Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, *37*(2), 177–219.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4*(2), 133–151.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., ... He, K. (2017). Accurate, large minibatch SGD: Training imagenet in 1 hour. *arXiv:1706.02677*.

Gu, A., Sala, F., Gunel, B., & Ré, C. (2018). Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations.*

Guillaumin, M., Verbeek, J., & Schmid, C. (2009). Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision* (pp. 498–505).

Hamilton, L., & Moitra, A. (2021). No-go theorem for acceleration in the hyperbolic plane. *arXiv:2101.05657*.

Hamm, J., & Lee, D. D. (2008). Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International Conference on Machine Learning* (pp. 376–383).

Han, A., & Gao, J. (2021). Improved variance reduction methods for Riemannian non-convex optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Han, A., Mishra, B., Jawanpuria, P., & Gao, J. (2022). Riemannian block SPD coupling manifold and its application to optimal transport. *Machine Learning*, 1–28.

Han, A., Mishra, B., Jawanpuria, P., & Gao, J. (2023). Learning with symmetric positive definite matrices via generalized Bures-Wasserstein geometry. In *International Conference on Geometric Science of Information.*

Han, A., Mishra, B., Jawanpuria, P. K., & Gao, J. (2021). On Riemannian optimization over positive definite matrices with the Bures-Wasserstein geometry. *Advances in Neural Information Processing Systems*, *34*, 8940–8953.

Harandi, M., Sanderson, C., Shen, C., & Lovell, B. C. (2013). Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *International Conference on Computer Vision* (pp. 3120–3127).

Harandi, M. T., Salzmann, M., & Hartley, R. (2014). From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *European Conference on Computer Vision* (pp. 17–32).

Hardy, Y., & Steeb, W.-H. (2019). *Matrix calculus, kronecker product and tensor product: A practical approach to linear algebra, multilinear algebra and tensor calculus with software implementations*. World Scientific.

Harikandeh, R. B., Ahmed, M. O., Virani, A., Schmidt, M., Konečný, J., & Sallinen, S. (2015). Stopwasting my gradients: Practical SVRG. *Advances in Neural Information Processing Systems*, 2251–2259.

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *5*(4), 1–19.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Horváth, S., Lei, L., Richtárik, P., & Jordan, M. I. (2022). Adaptivity of stochastic gradient methods for nonconvex optimization. *SIAM Journal on Mathematics of Data Science*, *4*(2), 634–648.

Hosseini, R., & Mash'al, M. (2015). MixEst: An estimation toolbox for mixture models. *arXiv:1507.06065*.

Hosseini, R., & Sra, S. (2020). An alternative to EM for Gaussian mixture models: batch and stochastic Riemannian optimization. *Mathematical Programming*, *181*(1), 187–223.

Hosseini, S., Huang, W., & Yousefpour, R. (2018). Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM Journal on Optimization*, *28*(1), 596–619.

Huang, F., & Gao, S. (2023). Gradient descent ascent for minimax problems on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, W. (2013). *Optimization algorithms on Riemannian manifolds with applications* (Unpublished doctoral dissertation). The Florida State University.

Huang, W., Absil, P.-A., & Gallivan, K. A. (2015). A Riemannian symmetric rank-one trust-region method. *Mathematical Programming*, *150*(2), 179–216.

Huang, W., Absil, P.-A., & Gallivan, K. A. (2016). A Riemannian BFGS method for nonconvex optimization problems. In *Numerical Mathematics and Advanced Applications* (pp. 627–634). Springer.

Huang, W., Absil, P.-A., Gallivan, K. A., & Hand, P. (2018). ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds. *ACM Transactions on Mathematical Software (TOMS)*, *44*(4), 1–21.

Huang, W., Gallivan, K. A., & Absil, P.-A. (2015). A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM Journal on Optimization*, *25*(3), 1660–1685.

Huang, W., & Wei, K. (2022). Riemannian proximal gradient methods. *Mathematical Programming*, *194*(1), 371–413.

Huang, Z., & Gool, L. V. (2017). A Riemannian network for SPD matrix learning. In *AAAI Conference on Artificial Intelligence.*

Huang, Z., Wang, R., Li, X., Liu, W., Shan, S., Van Gool, L., & Chen, X. (2017). Geometry-aware similarity learning on SPD manifolds for visual recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 2513–2523.

Huang, Z., Wang, R., Shan, S., Li, X., & Chen, X. (2015). Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *International Conference on Machine Learning* (pp. 720–729).

Hutchinson, M., Terenin, A., Borovitskiy, V., Takao, S., Teh, Y., & Deisenroth, M. (2021). Vector-valued Gaussian processes on Riemannian manifolds via gauge independent projected kernels. *Advances in Neural Information Processing Systems*, *34*, 17160–17169.

Jawanpuria, P., Balgovind, A., Kunchukuttan, A., & Mishra, B. (2019). Learning multilingual word embeddings in latent metric space: A geometric approach. *Transactions of the Association for Computational Linguistics*, *7*, 107–120.

Jawanpuria, P., Meghwanshi, M., & Mishra, B. (2020a). Geometry-aware domain adaptation for unsupervised alignment of word embeddings. In *Annual Meeting of the Association for Computational Linguistics.*

Jawanpuria, P., Meghwanshi, M., & Mishra, B. (2020b). A simple approach to learning unsupervised multilingual embeddings. In *Conference on Empirical Methods in Natural Language Processing.*

Jawanpuria, P., & Mishra, B. (2018). A unified framework for structured low-rank matrix learning. In *International Conference on Machine Learning.*

Jawanpuria, P., Satya Dev, N. T. V., & Mishra, B. (2021). Efficient robust optimal transport: formulations and algorithms. In *IEEE Conference on Decision and Control.*

Jawanpuria, P. K., Lapin, M., Hein, M., & Schiele, B. (2015). Efficient output kernel learning for multiple tasks. *Advances in Neural Information Processing Systems*, *28*.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., & Harandi, M. (2013). Kernel methods on the Riemannian manifold of symmetric positive definite matrices. In *Computer Vision and Pattern Recognition* (pp. 73–80).

Jeuris, B., Vandebril, R., & Vandereycken, B. (2012). A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis*, *39*, 379–402.

Ji, K., Wang, Z., Weng, B., Zhou, Y., Zhang, W., & Liang, Y. (2020). History-gradient aided batch size adaptation for variance reduced algorithms. In *International Conference on Machine Learning* (pp. 4762–4772).

Ji, K., Wang, Z., Zhou, Y., & Liang, Y. (2019). Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization. In *International conference on machine learning* (pp. 3100–3109).

Jin, J., & Sra, S. (2022). Understanding Riemannian acceleration via a proximal extragradient framework. In *Conference on Learning Theory* (pp. 2924–2962).

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 315–323.

Journée, M., Bach, F., Absil, P.-A., & Sepulchre, R. (2010). Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, *20*(5), 2327–2351.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... Potapenko, A. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, *30*(5), 509–541.

Karimi, H., Nutini, J., & Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 795–811).

Kasai, H., Jawanpuria, P., & Mishra, B. (2019). Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *International Conference on Machine Learning* (pp. 3262–3271).

Kasai, H., Sato, H., & Mishra, B. (2018a). Riemannian stochastic quasi-Newton algorithm with variance reduction and its convergence analysis. In *International Conference on Artificial Intelligence and Statistics* (pp. 269–278).

Kasai, H., Sato, H., & Mishra, B. (2018b). Riemannian stochastic recursive gradient algorithm. In *International Conference on Machine Learning* (pp. 2516–2524).

Keshavan, R. H., & Oh, S. (2009). A gradient descent algorithm on the Grassman manifold for matrix completion. *arXiv:0910.5260*.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv:1609.04836*.

Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., & Lempitsky, V. (2020). Hyperbolic image embeddings. In *Computer Vision and Pattern Recognition* (pp. 6418–6428).

Kim, J., & Yang, I. (2022). Accelerated gradient methods for geodesically convex optimization: Tractable algorithms and convergence analysis. In *International Conference on Machine Learning* (pp. 11255–11282).

Kumar Roy, S., Mhammedi, Z., & Harandi, M. (2018). Geometry aware constrained optimization techniques for deep learning. In *Computer Vision and Pattern Recognition* (pp. 4460–4469).

Kylberg, G. (2011, September). *The kylberg texture dataset v. 1.0* (External report (Blue series) No. 35). Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden. Retrieved from http://www.cb.uu.se/~gustaf/texture/

Lancaster, P. (1970). Explicit solutions of linear matrix equations. *SIAM Review*, *12*(4), 544–566.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, J. (2012). *Introduction to smooth manifolds* (Vol. 218). Springer Science & Business Media.

Lee, J. M. (2018). *Introduction to Riemannian manifolds* (Vol. 176). Springer.

Lei, L., Ju, C., Chen, J., & Jordan, M. I. (2017). Non-convex finite-sum optimization via SCSG methods. *Advances in Neural Information Processing Systems*, 2348–2358.

Li, J., Balasubramanian, K., & Ma, S. (2022). Stochastic zeroth-order Riemannian derivative estimation and optimization. *Mathematics of Operations Research*.

Li, J., & Ma, S. (2022). Federated learning on Riemannian manifolds. *arXiv:2206.05668*.

Li, J., Ma, S., & Srivastava, T. (2022). A Riemannian ADMM. *arXiv:2211.02163*.

Li, Z., Bao, H., Zhang, X., & Richtárik, P. (2021). PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning* (pp. 6286–6295).

Li, Z., & Li, J. (2018). A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 5564–5574.

Lian, X., Wang, M., & Liu, J. (2017). Finite-sum composition optimization via variance reduced gradient descent. In *Artificial Intelligence and Statistics* (pp. 1159–1167).

Lin, Y., & Simoncini, V. (2015). A new subspace iteration method for the algebraic Riccati equation. *Numerical Linear Algebra with Applications*, *22*(1), 26–47.

Lin, Z. (2019). Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, *40*(4), 1353–1370.

Linial, N., London, E., & Rabinovich, Y. (1995). The geometry of graphs and some of its algorithmic applications. *Combinatorica*, *15*(2), 215–245.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., & Amini, L. (2018). Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 3727–3737.

Liu, Y., Shang, F., Cheng, J., Cheng, H., & Jiao, L. (2017). Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, *30*.

Luchnikov, I. A., Ryzhov, A., Filippov, S. N., & Ouerdane, H. (2021). QGOpt: Riemannian optimization for quantum technologies. *SciPost Phys.*, *10*, 79.

Luenberger, D. G. (1972). The gradient projection method along geodesics. *Management Science*, *18*(11), 620–631.

Mahadevan, S., Mishra, B., & Ghosh, S. (2019). A unified framework for domain adaptation using metric learning on manifolds. In *ECML-PKDD*.

Malagò, L., Montrucchio, L., & Pistone, G. (2018). Wasserstein Riemannian geometry of Gaussian densities. *Information Geometry*, *1*(2), 137–179.

Mangoubi, O., & Smith, A. (2018). Rapid mixing of geodesic walks on manifolds with positive curvature. *The Annals of Applied Probability*, *28*(4), 2501–2543.

Mardia, K. V., Jupp, P. E., & Mardia, K. (2000). *Directional statistics* (Vol. 2). Wiley Online Library.

Martínez-Rubio, D. (2022). Global Riemannian acceleration in hyperbolic and spherical spaces. In *International Conference on Algorithmic Learning Theory* (pp. 768–826).

Meghwanshi, M., Jawanpuria, P., Kunchukuttan, A., Kasai, H., & Mishra, B. (2018). Mctorch, a manifold optimization library for deep learning. *arXiv:1810.01811*.

Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L., & Han, J. (2019). Spherical text embedding. *Advances in Neural Information Processing Systems*, *32*.

Merikoski, J. K., & Kumar, R. (2004). Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes*, *4*, 150–159.

Meyer, G., Bonnabel, S., & Sepulchre, R. (2011). Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *The Journal of Machine Learning Research*, *12*, 593–625.

Meyer, W. (1989). Toponogov's theorem and applications. *Lecture Notes, Trieste*.

Mishra, B., Kasai, H., & Jawanpuria, P. (2020). *Riemannian optimization on the simplex of positive definite matrices.* NeurIPS workshop on Optimization for Machine Learning.

Mishra, B., Satyadev, N., Kasai, H., & Jawanpuria, P. (2021). Manifold optimization for non-linear optimal transport problems. *arXiv:2103.00902*.

Moritz, P., Nishihara, R., & Jordan, M. (2016). A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics* (pp. 249–258).

Nesterov, Y. (2003). *Introductory lectures on convex optimization: A basic course* (Vol. 87). Springer Science & Business Media.

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. akad. nauk sssr* (Vol. 269, pp. 543–547).

Nguyen, L. M., Liu, J., Scheinberg, K., & Takáč, M. (2017a). SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning* (pp. 2613–2621).

Nguyen, L. M., Liu, J., Scheinberg, K., & Takáč, M. (2017b). Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv:1705.07261*.

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems, 30*.

Nimishakavi, M., Jawanpuria, P., & Mishra, B. (2018). A dual framework for low-rank tensor completion. *Advances in Neural Information Processing Systems*.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. Springer.

Pang, Y., Yuan, Y., & Li, X. (2008). Gabor-based region covariance matrices for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 18*(7), 989–993.

Pennec, X. (2020). Manifold-valued image processing with SPD matrices. In *Riemannian Geometric Statistics in Medical Image Analysis* (pp. 75–134). Elsevier.

Pennec, X., Fillard, P., & Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision, 66*(1), 41–66.

Pewsey, A., & García-Portugués, E. (2021). Recent advances in directional statistics. *Test, 30*(1), 1–58.

Pham, N. H., Nguyen, L. M., Phan, D. T., & Tran-Dinh, Q. (2020). ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research, 21*(110), 1–48.

Polyak, B. T. (1963). Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, *3*(4), 643–653.

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855.

Quang, M. H., San Biagio, M., & Murino, V. (2014). Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces. *Advances in Neural Information Processing Systems*, 388–396.

Reddi, S. J., Hefny, A., Sra, S., Póczos, B., & Smola, A. (2016). Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning* (pp. 314–323).

Reddi, S. J., Sra, S., Póczos, B., & Smola, A. (2016). Stochastic Frank-Wolfe methods for nonconvex optimization. In *Annual Allerton Conference on Communication, Control, and Computing* (pp. 1244–1251).

Reddi, S. J., Sra, S., Poczos, B., & Smola, A. J. (2016). Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advanced in Neural Information Processing Systems*, 1153–1161.

Ring, W., & Wirth, B. (2012). Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, *22*(2), 596–627.

Rothschild, D., & Jameson, A. (1970). Comparison of four numerical algorithms for solving the Liapunov matrix equation. *International Journal of Control*, *11*(2), 181–198.

Sakai, H., & Iiduka, H. (2021). Riemannian adaptive optimization algorithm and its application to natural language processing. *IEEE Transactions on Cybernetics*, *52*(8), 7328–7339.

Sala, F., De Sa, C., Gu, A., & Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning* (pp. 4460–4469).

Sato, H., Kasai, H., & Mishra, B. (2019). Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, *29*(2), 1444–1472.

Savas, B., & Lim, L.-H. (2010). Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing*, *32*(6), 3352–3393.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Scieur, D., d'Aspremont, A., & Bach, F. (2020). Regularized nonlinear acceleration. *Mathematical Programming*, *179*(1), 47–83.

Scieur, D., Oyallon, E., d'Aspremont, A., & Bach, F. (2018). Online regularized nonlinear acceleration. *arXiv:1805.09639*.

Shanks, D. (1955). Non-linear transformations of divergent and slowly convergent sequences. *Journal of Mathematics and Physics*, *34*(1-4), 1–42.

Shen, Z., Zhou, P., Fang, C., & Ribeiro, A. (2019). A stochastic trust region method for non-convex minimization. *arXiv:1903.01540*.

Shi, D., Gao, J., Hong, X., Boris Choy, S., & Wang, Z. (2021). Coupling matrix manifolds assisted optimization for optimal transport problems. *Machine Learning*, *110*(3), 533–558.

Shinohara, Y., Masuko, T., & Akamine, M. (2010). Covariance clustering on Riemannian manifolds for acoustic model compression. In *International Conference on Acoustics, Speech and Signal Processing* (pp. 4326–4329).

Sidi, A., Ford, W. F., & Smith, D. A. (1986). Acceleration of convergence of vector sequences. *SIAM Journal on Numerical Analysis*, *23*(1), 178–196.

Siegel, J. W. (2019). Accelerated optimization with orthogonality constraints. *arXiv:1903.05204*.

Sievert, S., & Charles, Z. (2019). Improving the convergence of sgd through adaptive batch sizes. *arXiv:1910.08222*.

Sim, A., Wiatrak, M. L., Brayne, A., Creed, P., & Paliwal, S. (2021). Directed graph embeddings in pseudo-Riemannian manifolds. In *International Conference on Machine Learning* (pp. 9681–9690).

Slawski, M., Li, P., & Hein, M. (2015). Regularization-free estimation in trace regression with symmetric positive semidefinite matrices. *Advances in Neural Information Processing Systems*.

Smith, R. L. (2008). The positive definite completion problem revisited. *Linear Algebra and its Applications*, *429*(7), 1442–1452.

Smith, S. L., Kindermans, P.-J., Ying, C., & Le, Q. V. (2018). Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*.

Sra, S., & Hosseini, R. (2015). Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, *25*(1), 713–739.

Su, W., Boyd, S., & Candes, E. (2014). A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Advances in Neural Information Processing Systems*, *27*.

Suárez, J. L., García, S., & Herrera, F. (2021). A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing*, *425*, 300–322.

Sun, Y., Flammarion, N., & Fazel, M. (2019). Escaping from saddle points on Riemannian manifolds. *Advances in Neural Information Processing Systems*, *32*.

Sun, Y., Gao, J., Hong, X., Mishra, B., & Yin, B. (2015). Heterogeneous tensor decomposition for clustering via manifold optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(3), 476–489.

Takatsu, A. (2008). On Wasserstein geometry of the space of Gaussian measures. *arXiv:0801.2250*.

Thanwerdas, Y., & Pennec, X. (2019). Is affine-invariance well defined on SPD matrices? a principled continuum of metrics. In *International Conference on Geometric Science of Information* (pp. 502–510).

Theis, F. J., Cason, T. P., & Absil, P.-A. (2009). Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. In *International Conference on Independent Component Analysis and Signal Separation* (pp. 354–361).

Townsend, J., Koep, N., & Weichwald, S. (2016). Pymanopt: A Python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research*, *17*(1), 4755–4759.

Tripuraneni, N., Flammarion, N., Bach, F., & Jordan, M. I. (2018). Averaging stochastic gradient descent on Riemannian manifolds. In *Conference on Learning Theory* (pp. 650–687).

Tsuda, K., Rätsch, G., & Warmuth, M. K. (2005). Matrix exponentiated gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, *6*(34), 995-1018.

Turaga, P., Veeraraghavan, A., & Chellappa, R. (2008). Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In *Computer Vision and Pattern Recognition* (pp. 1–8).

Turaga, P., Veeraraghavan, A., Srivastava, A., & Chellappa, R. (2011). Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2273–2286.

Tuzel, O., Porikli, F., & Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision* (pp. 589–600).

Tuzel, O., Porikli, F., & Meer, P. (2008). Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(10), 1713–1727.

Udriste, C. (1994). *Convex functions and optimization methods on Riemannian manifolds* (Vol. 297). Springer Science & Business Media.

Vandereycken, B. (2013). Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, *23*(2), 1214–1236.

Vandereycken, B., & Vandewalle, S. (2010). A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, *31*(5), 2553–2579.

van Oostrum, J. (2020). Bures-Wasserstein geometry. *arXiv:2001.08056*.

Vishnoi, N. K. (2018). Geodesic convex optimization: Differentiation on manifolds, geodesics, and convexity. *arXiv:1806.06373*.

Vorontsov, E., Trabelsi, C., Kadoury, S., & Pal, C. (2017). On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning* (pp. 3570–3578).

Waldmann, S. (2012). Geometric wave equations. *arXiv:1208.4706*.

Walker, H. F., & Ni, P. (2011). Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, *49*(4), 1715–1735.

Wang, B., Ma, S., & Xue, L. (2022). Riemannian stochastic proximal gradient methods for nonsmooth optimization over the Stiefel manifold. *The Journal of Machine Learning Research*, *23*(1), 4599–4631.

Wang, C., Sun, D., & Toh, K.-C. (2010). Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM Journal on Optimization*, *20*(6), 2994–3013.

Wang, J., Chen, Y., Chakraborty, R., & Yu, S. X. (2020). Orthogonal convolutional neural networks. In *Computer Vision and Pattern Recognition* (pp. 11505–11515).

Wang, L., & Liu, X. (2022). Decentralized optimization over the Stiefel manifold by an approximate augmented Lagrangian function. *IEEE Transactions on Signal Processing*, *70*, 3029–3041.

Wang, X., Ma, S., Goldfarb, D., & Liu, W. (2017). Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, *27*(2), 927–956.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., & Tarokh, V. (2019). Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, *32*, 2406–2416.

Wang, Z., Zhou, Y., Liang, Y., & Lan, G. (2019). Stochastic variance-reduced cubic regularization for nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics* (pp. 2731–2740).

Weber, M., & Sra, S. (2019). Nonconvex stochastic optimization on manifolds via Riemannian Frank-Wolfe methods. *arXiv:1910.04194*.

Weber, M., & Sra, S. (2022a). Projection-free nonconvex stochastic optimization on Riemannian manifolds. *IMA Journal of Numerical Analysis*, *42*(4), 3241–3271.

Weber, M., & Sra, S. (2022b). Riemannian optimization via Frank-Wolfe methods. *Mathematical Programming*, 1–32.

Wibisono, A., Wilson, A. C., & Jordan, M. I. (2016). A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, *113*(47), E7351–E7358.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.

Wynn, P. (1956). On a device for computing the e m (s n) transformation. *Mathematical Tables and Other Aids to Computation*, 91–96.

Xiong, B., Zhu, S., Potyka, N., Pan, S., Zhou, C., & Staab, S. (2022). Pseudo-Riemannian graph convolutional networks. *Advances in Neural Information Processing Systems*, *35*.

Yu, Y., & Huang, L. (2017). Fast stochastic variance reduced ADMM for stochastic composition optimization. In *International joint conference on artificial intelligence* (pp. 3364–3370).

Yuan, X., Huang, W., Absil, P.-A., & Gallivan, K. A. (2016). A Riemannian limited-memory BFGS algorithm for computing the matrix geometric mean. *Procedia Computer Science*, *80*, 2147–2157.

Yurtsever, A., Sra, S., & Cevher, V. (2019). Conditional gradient methods via stochastic path-integrated differential estimator. In *International Conference on Machine Learning* (pp. 7282–7291).

Zhang, D., & Tajbakhsh, S. D. (2020). Riemannian stochastic variance-reduced cubic regularized Newton method. *arXiv:2010.03785*.

Zhang, H., J Reddi, S., & Sra, S. (2016). Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, *29*.

Zhang, H., & Sra, S. (2016). First-order methods for geodesically convex optimization. In *Conference on Learning Theory* (pp. 1617–1638).

Zhang, H., & Sra, S. (2018a). An estimate sequence for geodesically convex optimization. In *Conference on Learning Theory* (pp. 1703–1723).

Zhang, H., & Sra, S. (2018b). Towards Riemannian accelerated gradient methods. *arXiv:1806.02812*.

Zhang, J., Zhang, H., & Sra, S. (2018). R-SPIDER: A fast Riemannian stochastic optimization algorithm with curvature independent rate. *arXiv:1811.04194*.

Zhang, P., Zhang, J., & Sra, S. (2022). Minimax in geodesic metric spaces: Sion's theorem and algorithms. *arXiv:2202.06950*.

Zhao, Z., Bai, Z.-J., & Jin, X.-Q. (2015). A Riemannian Newton algorithm for nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, *36*(2), 752–774.

Zhou, B., Zheng, X., Wang, Y. G., Li, M., & Gao, J. (2022). Embedding graphs on Grassmann manifold. *Neural Networks*, *152*, 322–331.

Zhou, D., Xu, P., & Gu, Q. (2020). Stochastic nested variance reduction for nonconvex optimization. *The Journal of Machine Learning Research*, *21*(1), 4130–4192.

Zhou, P., Yuan, X., & Feng, J. (2018). New insight into hybrid stochastic gradient descent: Beyond with-replacement sampling and convexity. *Advances in Neural Information Processing Systems*, 1234–1243.

Zhou, P., Yuan, X., Yan, S., & Feng, J. (2019). Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, P., Yuan, X.-T., & Feng, J. (2019). Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. In *International Conference on Artificial Intelligence and Statistics* (pp. 138–147).