A general framework of Riemannian adaptive optimization methods with a convergence analysis

Anonymous authors
Paper under double-blind review

Abstract

This paper proposes a general framework of Riemannian adaptive optimization methods. The framework encapsulates several stochastic optimization algorithms on Riemannian manifolds and incorporates the mini-batch strategy that is often used in deep learning. Within this framework, we also propose AMSGrad on embedded submanifolds of Euclidean space. Moreover, we give convergence analyses valid for both a constant and a diminishing step size. Our analyses also reveal the relationship between the convergence rate and mini-batch size. In numerical experiments, we applied the proposed algorithm to principal component analysis and the low-rank matrix completion problem, which can be considered to be Riemannian optimization problems. Python implementations of the methods used in the numerical experiments are available at `https://anonymous.4open.science/r/202408-adaptive-0BA6/README.md`.

1 Introduction

Riemannian optimization (Absil et al., 2008; Sato, 2021) has received much attention in machine learning. For example, batch normalization (Cho & Lee, 2017), representation learning (Nickel & Kiela, 2017), and the low-rank matrix completion problem (Vandereycken, 2013; Cambier & Absil, 2016; Boumal & Absil, 2015) can be considered optimization problems on Riemannian manifolds. This paper focuses on Riemannian adaptive optimization algorithms for solving stochastic optimization problems on Riemannian manifolds. In particular, we treat Riemannian submanifolds of Euclidean space (e.g., unit spheres and the Stiefel manifold).

In Euclidean settings, adaptive optimization methods are widely used for training deep neural networks. There are many adaptive optimization methods, such as Adaptive gradient (AdaGrad) (Duchi et al., 2011), Adadelta (Zeiler, 2012), Root mean square propagation (RMSProp) (Hinton et al., 2012), Adaptive moment estimation (Adam) (Kingma & Ba, 2015), Yogi (Zaheer et al., 2018), Adaptive mean square gradient (AMSGrad) (Reddi et al., 2018), AdaFom (Chen et al., 2019), AdaBound (Luo et al., 2019), Adam with decoupled weight decay (AdamW) (Loshchilov & Hutter, 2019) and AdaBelief (Zhuang et al., 2020). Reddi et al. (2018) proposed a general framework of adaptive optimization methods that encapsulates many of the popular adaptive methods in Euclidean space.

Bonnabel (2013) proposed Riemannian stochastic gradient descent (RSGD), the most basic Riemannian stochastic optimization algorithm. In particular, Riemannian stochastic variance reduction algorithms, such as Riemannian stochastic variance-reduced gradient (RSVRG) (Zhang et al., 2016), Riemannian stochastic recursive gradient (RSRG) (Kasai et al., 2018), and Riemannian stochastic path-integrated differential estimator (R-SPIDER) (Zhang et al., 2018; Zhou et al., 2019), are based on variance reduction methods in Euclidean space. There are several prior studies on Riemannian adaptive optimization methods for specific Riemannian manifolds. In particular, Kasai et al. (2019) proposed a Riemannian adaptive stochastic gradient algorithm on matrix manifolds (RASA). RASA is an adaptive optimization method on matrix manifolds (e.g., the Stiefel manifold or the Grassmann manifold), with a convergence analysis under the upper-Hessian bounded and retraction L-smooth assumptions (see (Kasai et al., 2019, Section 4) for details). However, RASA is not a direct extension of the adaptive optimization methods commonly used in deep learning, and it works only for diminishing step sizes. On the cartesian product of Riemannian manifolds, RAMSGrad

(Bécigneul & Ganea, 2019) and modified RAMSGrad (Sakai & Iiduka, 2021), direct extensions of AMSGrad, have been proposed as methods that work on Cartesian products of Riemannian manifolds. In particular, Roy et al. (2018) proposed cRAMSProp and applied it to several Riemannian stochastic optimizations. However, they did not provide a convergence analysis of cRAMSProp. More recently, Riemannian stochastic optimization methods, Sharpness-aware minimization on Riemannian manifolds (Riemannian SAM) (Yun & Yang, 2024) and Riemannian natural gradient descent (RNGD) (Hu et al., 2024), were proposed.

1.1 Contributions

Motivated by the above discussion, we propose a framework of adaptive optimization methods on Riemannian submanifolds of Euclidean space (Algorithm 1) that is based on the framework (Reddi et al., 2018, Algorithm 1) proposed by Reddi, Kale and Kumar for Euclidean space. Our framework incorporates the mini-batch strategy that is often used in deep learning. Important examples of Riemannian submanifolds of the Euclidean space include the unit sphere and the Stiefel manifold. Moreover, within this framework, we propose AMSGrad on embedded submanifolds of Euclidean space (Algorithm 2) as a direct extension of AMSGrad. In addition, we give convergence analyses (Theorem 3.7) valid for both a constant step size (Theorem 3.8) and diminishing step size (Theorem 3.9). Our analyses not only ensure that the proposed method converges to the optimal solution, but also reveal the relationship between the convergence rate and mini-batch size. Moreover, we numerically compare the performances of several methods based on Algorithm 1, including Algorithm 2, with the existing methods. In the numerical experiments, we applied the algorithms to principal component analysis (PCA) (Kasai et al., 2018; Roy et al., 2018) and the low-rank matrix completion (LRMC) problem (Boumal & Absil, 2015; Kasai et al., 2019; Hu et al., 2024), which can be considered to be Riemannian optimization problems.

Our first contribution is to propose a general framework of Riemannian adaptive optimization methods (Algorithm 1) and AMSGrad on embedded submanifolds of Euclidean space (Algorithm 2). In particular, the proposed method incorporates the mini-batch strategy. Our second contribution is to give convergence analyses of Algorithms 1 and 2. In particular, we emphasize that the proposed method can use both constant and diminishing step sizes (Theorems 3.8 and 3.9), in contrast to RASA (Kasai et al., 2019), which only uses a diminishing step size. The third contribution is to compare the proposed methods with RSGD and RASA in numerical experiments.

2 Mathematical Preliminaries

Let $\mathbb{R}^d$ be a d-dimensional Euclidean space with inner product $\langle x, y\rangle_2 := x^\top y$, which induces the norm $\|\cdot\|_2$. Let $\mathbb{R}_{++}$ be the set of positive real numbers, i.e., $\mathbb{R}_{++} := \{x \in \mathbb{R} \mid x > 0\}$. I_d denotes a $d \times d$ identity matrix. For square matrices $X, Y \in \mathbb{R}^{d\times d}$, we write $X \prec Y$ (resp. $X \preceq Y$) if $Y - X$ is a positive-definite (resp. positive-semidefinite) matrix. For two matrices X and Y of the same dimension, $X \odot Y$ denotes the Hadamard product, i.e., element-wise product. Let $\max(X, Y)$ be the element-wise maximum. Let $\mathcal{S}^d$ (resp. $\mathcal{S}^d_+$, $\mathcal{S}^d_{++}$) be the set of $d \times d$ symmetric (resp. symmetric positive-semidefinite, symmetric positive-definite) matrices, i.e., $S^d := \{X \in \mathbb{R}^{d\times d} \mid X^\top = X\}$, $\mathcal{S}^d_+ := \{X \in \mathbb{R}^{d\times d} \mid X \succeq O\}$ and $\mathcal{S}^d_{++} := \{X \in \mathbb{R}^{d\times d} \mid X \succ O\}$. Let $\mathcal{D}^d$ be the set of $d \times d$ diagonal matrices. Let $\mathcal{O}_d$ be the orthogonal group, i.e., $\mathcal{O}_d := \{X \in \mathbb{R}^{d\times d} \mid X^\top X = I_d\}$.

Let M be an embedded submanifold of $\mathbb{R}^d$. Moreover, let T_xM be the tangent space at a point $x \in M$ and TM be the tangent bundle of M. Let 0_x be the zero element of T_xM. The inner product $\langle \cdot, \cdot\rangle_2$ of a Euclidean space $\mathbb{R}^d$ induces a Riemannian metric $\langle \cdot, \cdot\rangle_x$ of M at $x \in M$ according to $\langle \xi, \eta\rangle_x = \langle \xi, \eta\rangle_2 = \xi^\top \eta$ for $\xi, \eta \in T_xM \subset T_x\mathbb{R}^d \cong \mathbb{R}^d$. The norm of $\eta \in T_xM$ is defined as $\|\eta\|_x = \sqrt{\eta^\top \eta} = \|\eta\|_2$. Let $P_x : T_x\mathbb{R}^d \cong \mathbb{R}^d \to T_xM$ be the orthogonal projection onto T_xM (see Absil et al. (2008)). For a smooth map $F : M \to N$ between two manifolds M and N, $\mathrm{D}F(x) : T_xM \to T_{F(x)}N$ denotes the derivative of F at $x \in M$. The Riemannian gradient $\operatorname{grad} f(x)$ of a smooth function $f : M \to \mathbb{R}$ at $x \in M$ is defined as a unique tangent vector at x satisfying $\langle \operatorname{grad} f(x), \eta\rangle_x = \mathrm{D}f(x)[\eta]$ for any $\eta \in T_xM$.

Definition 2.1 (Retraction)**.** *Let M be a manifold. Any smooth map $R : TM \to M$ is called a retraction on M if it has the following properties.*

- *$R_x(0_x) = x$ for all $x \in M$;*

- *With the canonical identification $T_{0_x}T_xM \cong T_xM$, $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_xM} : T_xM \to T_xM$ for all $x \in M$,*

where R_x denotes the restriction of R to T_xM.

2.1 Examples

The unite sphere $\mathbb{S}^{d-1} := \{x \in \mathbb{R}^d \mid \|x\|_2 = 1\}$ is an embedded manifold of $\mathbb{R}^d$. The tangent space $T_x\mathbb{S}^{n-1}$ at $x \in \mathbb{S}^{d-1}$ is given by $T_x\mathbb{S}^{n-1} = \{\eta \in \mathbb{R}^d \mid \eta^\top x = 0\}$. The induced Riemannian metric on $\mathbb{S}^{d-1}$ is given by $\langle \xi, \eta \rangle_x = \langle \xi, \eta \rangle_2 := \xi^\top \eta$ for $\xi, \eta \in T_x\mathbb{S}^{d-1}$. The orthogonal projection $P_x : \mathbb{R}^d \to T_x\mathbb{S}^{d-1}$ onto the tangent space $T_x\mathbb{S}^{n-1}$ is given by $P_x(\eta) = (I_d - xx^\top)\eta$ for $x \in \mathbb{S}^{d-1}$ and $\eta \in T_x\mathbb{S}^{d-1}$.

An important example is the Stiefel manifold (Absil et al., 2008, Chapter 3.3.2), which is defined as $\mathrm{St}(p, n) := \{X \in \mathbb{R}^{n\times p} \mid X^\top X = I_p\}$ for $n \geq p$. $\mathrm{St}(p, n)$ is an embedded manifold of $\mathbb{R}^{n\times d}$. The tangent space $T_x \mathrm{St}(p, n)$ at $X \in \mathrm{St}(p, n)$ is given by

$$T_X \mathrm{St}(p, n) = \{\eta \in \mathbb{R}^{n\times p} \mid X^\top \eta + \eta^\top X = O\}.$$

The induced Riemannian metric on $\mathrm{St}(p, n)$ is given by $\langle \xi, \eta \rangle_X = \mathrm{tr}(\xi^\top \eta)$ for $\xi, \eta \in T_X \mathrm{St}(p, n)$. The orthogonal projection onto the tangent space $T_X \mathrm{St}(p, n)$ is given by $P_X(\eta) = \eta - X \,\mathrm{sym}(X^\top \eta)$ for $X \in \mathrm{St}(p, n)$, $\eta \in T_X \mathrm{St}(p, n)$, where $\mathrm{sym}(A) := (A + A^\top)/2$. The Stiefel manifold $\mathrm{St}(p, n)$ reduces to the orthogonal groups when $n = p$, i.e. $\mathrm{St}(p, p) = \mathcal{O}_p$.

Moreover, we will also consider the Grassmann manifold (Absil et al., 2008, Chapter 3.4.4) $\mathrm{Gr}(p, n) := \mathrm{St}(p, n)/\mathcal{O}_p$. Let $X \in \mathrm{St}(p, n)$ be a representative of $[X] := \{XQ \mid Q \in \mathcal{O}_p\} \in \mathrm{Gr}(p, n)$. We denote the horizontal lift of $\eta \in T_{[X]} \mathrm{Gr}(p, n)$ at X by $\bar{\eta}_X \in T_X \mathrm{St}(p, n)$. The Riemannian metric of the Grassmann manifold $\mathrm{Gr}(p, n)$ is endowed with $\langle \xi, \eta \rangle_{[X]} := \langle \bar{\xi}_X, \bar{\eta}_X \rangle_2$ for $\xi, \eta \in T_{[X]} \mathrm{Gr}(p, n)$. The orthogonal projection onto the tangent space $T_{[X]} \mathrm{Gr}(p, n)$ is defined through

$$\overline{P_{[X]}(\eta)} = (I_n - XX^\top)\bar{\eta}_X,$$

for $[X] \in \mathrm{Gr}(p, n)$ and $\eta \in T_{[X]} \mathrm{Gr}(p, n)$.

2.2 Riemannian stochastic optimization problem

We focus on minimizing a objective function $f : M \to \mathbb{R}$ of the form,

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x),$$

where f_i is a smooth function for $i = 1, \dots, N$. We use the mini-batch strategy as follows (see Iiduka (2024) for detail). $s_{k,i}$ is a random variable generated from the i-th sampling at the k-th iteration, and $\boldsymbol{s}_k := (s_{k,1}, \dots, s_{k,b})^\top$ is independent of $(x_k)_{k=1}^{\infty}$, where b $(\leq N)$ is the batch size. To simplify the notation, we denote the expectation $\mathbb{E}_{\boldsymbol{s}_k}$ with respect to $\boldsymbol{s}_k$ by $\mathbb{E}_k$. From the independence of $\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_k$, we can define the total expectation $\mathbb{E}$ by $\mathbb{E}_1\mathbb{E}_2 \cdots \mathbb{E}_k$. We define the mini-batch stochastic gradient $\mathrm{grad} f_{B_k}(x_k)$ of f at the k-th iteration by

$$\mathrm{grad} f_{B_k}(x_k) := \frac{1}{b} \sum_{i=1}^{b} \mathrm{grad} f_{s_{k,i}}(x_k). \tag{1}$$

Our main objective is to find a local minimizer of f, i.e., a stationary point $x_\star \in M$ satisfying $\mathrm{grad} f(x_\star) = 0_{x_\star}$.

2.3 Proposed general framework of Riemannian adaptive methods

Reddi et al. (2018) provided a general framework of adaptive gradient methods in Euclidean space. We devised Algorithm 1 by generalizing that framework to an embedded manifold of $\mathbb{R}^d$. The main difference from the Euclidean setting is computing the projection of $H_k^{-1} m_k$ onto the tangent space $T_{x_k}M$ by the orthogonal projection P_{x_k}. Algorithm 1 requires sequences of maps, $(\phi_k)_{k=1}^{\infty}$ and $(\psi_k)_{k=1}^{\infty}$, such that $\phi_k : T_{x_1}M \times \cdots \times T_{x_k}M \to \mathbb{R}^d$ and $\psi_k : T_{x_1}M \times \cdots \times T_{x_k}M \to \mathcal{D}^d \cap \mathcal{S}_{++}^d$, respectively. Note that Algorithm 1 is still abstract because the maps $(\phi_k)_{k=1}^{\infty}$ and $(\psi_k)_{k=1}^{\infty}$ are not specified. Algorithm 1 is the extension of a general framework in Euclidean space proposed by Reddi, Kale and Kumar (Reddi et al., 2018). In the Euclidean setting (i.e., $M = \mathbb{R}^d$), the orthogonal projection P_{x_k} yields an identity map and this corresponds to the Euclidean version of the general framework.

Algorithm 1 The general framework of Riemannian adaptive optimization methods on an embedded submanifold of $\mathbb{R}^d$.

Require: Initial point $x_1 \in M$, retraction $R : TM \to M$, step sizes $(\alpha_k)_{k=1}^{\infty} \subset \mathbb{R}_{++}$, sequences of maps $(\phi_k)_{k=1}^{\infty}$, $(\psi_k)_{k=1}^{\infty}$.
Ensure: Sequence $(x_k)_{k=1}^{\infty} \subset M$.

1: $k \leftarrow 1$.
2: **loop**
3: $\quad g_k = \operatorname{grad} f_{B_k}(x_k)$.
4: $\quad m_k = \phi_k(g_1, \dots, g_k) \in \mathbb{R}^d$.
5: $\quad H_k = \psi_k(g_1, \dots, g_k) \in \mathcal{D}^d \cap \mathcal{S}_{++}^d$.
6: $\quad x_{k+1} = R_{x_k}(-\alpha_k P_{x_k}(H_k^{-1} m_k))$.
7: $\quad k \leftarrow k + 1$.
8: **end loop**

Although Algorithm 1 is an optimization method on Riemannian manifold M, since $g_k \in T_{x_k}M \subset \mathbb{R}^d$, $m_k \in \mathbb{R}^d$ and $H_k \in \mathcal{D}^d \cap \mathcal{S}_{++}^d \subset \mathbb{R}^{d \times d}$, we can directly use $(\phi_n)_{n=1}^{\infty}$ and $(\psi_n)_{n=1}^{\infty}$ to extend the Euclidean adaptive gradient methods.

Here, SGD is the most basic method; it uses

$$\phi_k(g_1, \dots, g_k) = g_k, \quad \psi_k(g_1, \dots, g_k) = I_d.$$

Algorithm 1 with these maps corresponds to RSGD (Bonnabel, 2013) in the Riemannian setting. AdaGrad (Duchi et al., 2011), the first adaptive gradient method in Euclidean space that propelled research on adaptive methods, uses the sequences of maps $\phi_k(g_1, \dots, g_k) = g_k$ and

$$\begin{aligned} v_k &= v_{k-1} + g_k \odot g_k, \\ \psi_k(g_1, \dots, g_k) &= \operatorname{diag}(\sqrt{v_{k,1}}, \dots, \sqrt{v_{k,d}}) + \epsilon I_d, \end{aligned}$$

where $v_0 = 0 \in \mathbb{R}^d$ and $\epsilon > 0$. Here, we will denote the i-th component of v_k by $v_{k,i}$. The exponential moving average variant of AdaGrad is often used in deep-learning training. The most basic variant is RMSProp (Hinton et al., 2012), which uses the sequences of maps $\phi_k(g_1, \dots, g_k) = g_k$ and

$$\begin{aligned} v_k &= \beta_2 v_{k-1} + (1 - \beta_2) g_k \odot g_k, \\ \psi_k(g_1, \dots, g_k) &= \operatorname{diag}(\sqrt{v_{k,1}}, \dots, \sqrt{v_{k,d}}) + \epsilon I_d, \end{aligned}$$

where $v_0 = 0 \in \mathbb{R}^d$ and $\epsilon > 0$. Both Algorithm 1 with these maps and cRMSProp (Roy et al., 2018) can be considered extensions of RMSProp to Riemannian manifolds. They differ from each other in that parallel transport is needed to compute the search direction of cRMSProp, but it is not needed in our method.

Adam (Kingma & Ba, 2015) is one of the most common variants; it uses the sequence of maps,

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k, \quad \phi_k(g_1, \dots, g_k) = \frac{m_k}{1 - \beta_1^{k+1}}, \tag{2}$$

and

$$\begin{aligned} v_k &= \beta_2 v_{k-1} + (1 - \beta_2) g_k \odot g_k, \quad \hat{v}_k = \frac{v_k}{1 - \beta_2^{k+1}}, \\ \psi_k(g_1, \dots, g_k) &= \operatorname{diag}(\sqrt{\hat{v}_{k,1}}, \dots, \sqrt{\hat{v}_{k,d}}) + \epsilon I_d, \end{aligned} \tag{3}$$

where $m_0 = 0 \in \mathbb{R}^d$ and $v_0 = 0 \in \mathbb{R}^d$. $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ are typically recommended values. Moreover, within the general framework (Algorithm 1), we propose the following algorithm as an extension of AMSGrad (Reddi et al., 2018) in Euclidean space.

Algorithm 2 AMSGrad on an embedded submanifold of $\mathbb{R}^d$.

Require: Initial point $x_1 \in M$, retraction $R : TM \to M$, step sizes $(\alpha_k)_{k=1}^{\infty} \subset \mathbb{R}_{++}$, hyperparameters $\beta_1, \beta_2 \in [0, 1)$, $\epsilon > 0$.
Ensure: Sequence $(x_k)_{k=1}^{\infty} \subset M$.

```
Set m_0 = 0, v_0 = 0 and v̂_0 = 0.
k ← 1.
loop
    g_k = grad f_{B_k}(x_k).
    m_k = β_1 m_{k−1} + (1 − β_1) g_k.
    v_k = β_2 v_{k−1} + (1 − β_2) g_k ⊙ g_k.
    v̂_k = max(v̂_{k−1}, v_k).
    H_k = diag(√v̂_{k,1}, ..., √v̂_{k,d}) + ϵ I_d.
    x_{k+1} = R_{x_k}(−α_k P_{x_k}(H_k^{−1} m_k)).
    k ← k + 1.
end loop
```

3 Convergence analysis

3.1 Assumptions and useful lemmas

We make the following Assumptions 3.1 (A1)–(A4). (A1) and (A2) include the standard conditions. (A3) assumes the boundedness of the gradient. (A4) is an assumption on the Lipschitz continuity of the gradient. (A5) assumes that a lower bound exists.

Assumption 3.1. *Let $(x_k)_{k=1}^{\infty}$ be a sequence generated by Algorithm 1.*

(A1) $\mathbb{E}_k[\operatorname{grad} f_{s_{k,i}}(x_k)] = \operatorname{grad} f(x_k)$ *for all* $k \geq 1$ *and* $i = 1, \dots, b$.

(A2) There exists $\sigma^2 > 0$ *such that*

$$\mathbb{E}_k\left[\left\|\operatorname{grad} f_{s_{k,i}}(x_k) - \operatorname{grad} f(x_k)\right\|_2^2\right] \leq \sigma^2,$$

for all $k \geq 1$ *and* $i = 1, \dots, b$.

(A3) There exists $G, B > 0$ *such that* $\|\operatorname{grad} f(x_k)\|_2 \leq G$ *and* $\|\operatorname{grad} f_{B_k}(x_k)\|_2 \leq B$ *for all* $k \geq 1$.

(A4) There exists a constant $L > 0$ *such that*

$$|\mathrm{D}(f \circ R_x)(\eta)[\eta] - \mathrm{D}f(x)[\eta]| \leq L \|\eta\|_2^2,$$

for all $x \in M$, $\eta \in T_x M$.

(A5) f *is bounded below by* $f_\star \in \mathbb{R}$.

Lemma 3.2. *Suppose that Assumption 3.1 (A1) holds. Let $(x_k)_{k=1}^{\infty}$ be a sequence generated by Algorithm 1. Then,*

$$\mathbb{E}_k\left[\operatorname{grad} f_{B_k}(x_k)\right] = \operatorname{grad} f(x_k),$$

for all $k \geq 1$.

Proof. See Appendix A. □

Lemma 3.3. *Suppose that Assumptions 3.1 (A1) and (A2) hold. Let* $(x_k)_{k=1}^{\infty}$ *be a sequence generated by Algorithm 1. Then,*

$$\mathbb{E}_k\left[\|\operatorname{grad} f_{B_k}(x_k)\|_2^2\right] \leq \frac{\sigma^2}{b} + \|\operatorname{grad} f(x_k)\|_2^2$$

for all $k \geq 1$.

Proof. See Appendix B. □

It is known that if Assumption 3.1 (A4) holds, so does the following Proposition 3.4. This property is known as retraction L-smooth (see Huang et al. (2015); Kasai et al. (2018) for details).

Proposition 3.4. *Suppose that Assumption 3.1 (A4) holds. Then,*

$$f(R_x(\eta)) \leq f(x) + \langle \operatorname{grad} f(x), \eta \rangle_2 + \frac{L}{2}\|\eta\|_2^2,$$

for all $x \in M$ *and* $\eta \in T_x M$.

3.2 Convergence analysis of Algorithm 1

The main difficulty in analyzing the convergence of adaptive gradient methods is due to the stochastic momentum $m_k = \phi_k(g_1, \ldots, g_k)$. As a way to overcome this challenge in Euclidean space, Zhou et al. (2024); Yan et al. (2018); Chen et al. (2019) defined a new sequence z_k,

$$z_k = x_k + \frac{\beta_1}{1-\beta_1}(x_k - x_{k-1}).$$

However, this strategy does not work in the Riemannian setting. Therefore, by following the policy of Zaheer et al. (2018), let us analyze the case in which $\phi_k(g_1, \ldots, g_k) = g_k$. To simplify the notation, we denote the i-th component of g_k (resp. v_k, $\hat{v}_k$) by $g_{k,i}$ (resp. $v_{k,i}$, $\hat{v}_{k,i}$).

Lemma 3.5. *Suppose that Assumption 3.1 (A3) holds. Then, the sequence* $(x_k)_{k=1}^{\infty} \subset M$ *generated by Algorithm 2 satisfies*

$$\hat{v}_{k,i} \leq B^2,$$

for all $k \geq 1$ *and* $i = 1, \ldots, d$.

Proof. See Appendix C. □

Lemma 3.6. *Suppose that Assumption 3.1 (A4) holds. If* $\phi_k(g_1, \ldots, g_k) = g_k$ *and* $H_k^{-1} \preceq \nu I_d$ *for all* $k \geq 1$ *and some* $\nu > 0$*, then the sequence* $(x_k)_{k=1}^{\infty} \subset M$ *generated by Algorithm 1 satisfies*

$$f(x_{k+1}) \leq f(x_k) + \left\langle \operatorname{grad} f(x_k), -\alpha_k H_k^{-1} g_k \right\rangle_2 + \frac{L\alpha_k^2 \nu^2}{2}\|g_k\|_2^2,$$

for all $k \geq 1$.

Proof. See Appendix D. □

Theorem 3.7. *Suppose that Assumptions 3.1 (A1)–(A5) hold. Moreover, let us assume that* $\alpha_{k+1} \leq \alpha_k$, $\phi_k(g_1, \ldots, g_k) = g_k$, $\alpha_k H_k^{-1} \succeq \alpha_{k+1} H_{k+1}^{-1}$ *and there exist* $\mu, \nu > 0$ *such that* $\mu I_d \preceq H_k^{-1} \preceq \nu I_d$ *for all* $k \geq 1$. *Then, the sequence* $(x_k)_{k=1}^{\infty} \subset M$ *generated by Algorithm 1 satisfies*

$$\sum_{k=1}^{K} \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2}\right) \mathbb{E}\left[\|\operatorname{grad} f(x_k)\|_2^2\right] \leq C_1 + \frac{C_2}{b}\sum_{k=1}^{K} \alpha_k^2,$$

for some constant $C_1, C_2 > 0$.

Remark: Since Algorithm 2 satisfies $\hat{v}_{k+1,i} := \max(\hat{v}_{k,i}, v_{k+1,i}) \geq \hat{v}_{k,i}$, it together with $\alpha_{k+1} \leq \alpha_k$, leads to $\alpha_k/(\sqrt{\hat{v}_{k,i}} + \epsilon) \geq \alpha_{k+1}/(\sqrt{\hat{v}_{k+1,i}} + \epsilon)$. Moreover, from Lemma 3.5,

$$\frac{1}{B+\epsilon} \leq \frac{1}{\sqrt{\hat{v}_{k,i}} + \epsilon} \leq \frac{1}{\epsilon},$$

which implies $(B+\epsilon)^{-1} I_d \preceq H_k^{-1} \preceq \epsilon^{-1} I_d$. Therefore, Algorithm 2 satisfies the assumption $\alpha_k H_k^{-1} \succeq \alpha_{k+1} H_{k+1}^{-1}$ and $\mu I_d \preceq H_k^{-1} \preceq \nu I_d$ with $\mu = (B+\epsilon)^{-1}$ and $\nu = \epsilon^{-1}$.

Proof. We denote $\operatorname{grad} f(x_k)$ by $g(x_k)$. First, let us consider the case of $k = 1$. From Lemma 3.6, we have

$$f(x_2) \leq f(x_1) + \left\langle g(x_1), -\alpha_1 H_1^{-1} g_1 \right\rangle_2 + \frac{L\alpha_1^2 \nu^2}{2} \|g_1\|_2^2 .$$

By taking $\mathbb{E}_1[\cdot]$ of both sides, we obtain

$$\begin{aligned}
\mathbb{E}_1[f(x_2)] &\leq f(x_1) + \left\langle g(x_1), -\alpha_1 \mathbb{E}_1[H_1^{-1} g_1] \right\rangle_2 + \frac{L\alpha_1^2 \nu^2}{2} \mathbb{E}_1 \left[\|g_1\|_2^2 \right] \\
&\leq f(x_1) + \left\langle g(x_1), -\alpha_1 \mathbb{E}_1[H_1^{-1} g_1] \right\rangle_2 + \frac{L\alpha_1^2 \nu^2}{2} \left(\frac{\sigma^2}{b} + \|g(x_1)\|_2^2 \right)
\end{aligned}$$

where the second inequality comes from Lemma 3.3. By taking $\mathbb{E}[\cdot]$ of both sides and rearranging terms, we get

$$-\frac{L\alpha_1 \nu^2}{2} \mathbb{E} \left[\|g(x_1)\|_2^2 \right] \leq f(x_1) - \mathbb{E}[f(x_2)] + \left\langle g(x_1), -\alpha_1 \mathbb{E}[H_1^{-1} g_1] \right\rangle_2 + \frac{L\alpha_1^2 \sigma^2 \nu^2}{2b}.$$

By adding $\alpha_1 \mu G^2$ to both sides, we obtain

$$\alpha_1 \mu G^2 - \frac{L\alpha_1 \nu^2}{2} \mathbb{E} \left[\|g(x_1)\|_2^2 \right] \leq f(x_1) - \mathbb{E}[f(x_2)] + \frac{L\alpha_1^2 \sigma^2 \nu^2}{2b} + \underbrace{\left\langle g(x_1), -\alpha_1 \mathbb{E}[H_1^{-1} g_1] \right\rangle_2 + \alpha_1 \mu G^2}_{C_0}.$$

Here, we note that

$$\alpha_1 \mu \mathbb{E} \left[\|g(x_1)\|_2^2 \right] \leq \alpha_1 \mu G^2.$$

Therefore, we have

$$\alpha_1 \left(\mu - \frac{L\alpha_1 \nu^2}{2} \right) \mathbb{E} \left[\|g(x_1)\|_2^2 \right] \leq f(x_1) - \mathbb{E}[f(x_2)] + \frac{L\alpha_1^2 \sigma^2 \nu^2}{2b} + C_0. \tag{4}$$

Next, let us consider the case of $k \geq 2$. From Lemma 3.6, we have

$$f(x_{k+1}) \leq f(x_k) + \left\langle g(x_k), -\alpha_{k-1} H_{k-1}^{-1} g_k \right\rangle_2 + \left\langle g(x_k), (\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1}) g_k \right\rangle_2 + \frac{L\alpha_k^2 \nu^2}{2} \|g_k\|_2^2$$

for all $k \geq 2$. From Assumption 3.1 (A3), Lemma E.1, and $\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1} \succeq O$, we have

$$f(x_{k+1}) \leq f(x_k) + \left\langle g(x_k), -\alpha_{k-1} H_{k-1}^{-1} g_k \right\rangle_2 + GB \operatorname{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1}) + \frac{L\alpha_k^2 \nu^2}{2} \|g_k\|_2^2 .$$

By taking $\mathbb{E}_k[\cdot]$ of both sides, we obtain

$$\begin{aligned}
\mathbb{E}_k[f(x_{k+1})] &\leq f(x_k) + \left\langle g(x_k), -\alpha_{k-1} H_{k-1}^{-1} \mathbb{E}_k[g_k] \right\rangle_2 \\
&\quad + GB \mathbb{E}_k[\operatorname{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1})] + \frac{L\alpha_k^2 \nu^2}{2} \mathbb{E}_k \left[\|g_k\|_2^2 \right] \\
&\leq f(x_k) - \alpha_{k-1} \left\langle g(x_k), H_{k-1}^{-1} g(x_k) \right\rangle_2 \\
&\quad + GB \mathbb{E}_k[\operatorname{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1})] + \frac{L\alpha_k^2 \nu^2}{2} \left(\frac{\sigma^2}{b} + \|g(x_k)\|_2^2 \right),
\end{aligned}$$

where the first inequality comes from the independence of H_{k-1}^{-1} for $\boldsymbol{s}_k$ and the second inequality comes from Lemmas 3.2 and 3.3. Here, since $H_{k-1}^{-1} \succeq \mu I_d$ and $\alpha_k \leq \alpha_{k-1}$, it follows that

$$-\alpha_{k-1} \left\langle g(x_k), H_{k-1}^{-1} g(x_k) \right\rangle_2 \leq -\alpha_k \mu \left\| g(x_k) \right\|_2^2 ,$$

which implies

$$\mathbb{E}_k[f(x_{k+1})] \leq f(x_k) - \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \|g(x_k)\|_2^2 + GB\mathbb{E}_k[\mathrm{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1})] + \frac{L\alpha_k^2 \sigma^2 \nu^2}{2b}.$$

By taking $\mathbb{E}[\cdot]$ of both sides, we have

$$\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] - \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] + GB\mathbb{E}[\mathrm{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1})] + \frac{L\alpha_k^2 \sigma^2 \nu^2}{2b}.$$

By rearranging the above inequality gives us

$$\alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] \leq \mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})] + GB\mathbb{E}[\mathrm{tr}(\alpha_{k-1} H_{k-1}^{-1} - \alpha_k H_k^{-1})] + \frac{L\alpha_k^2 \sigma^2 \nu^2}{2b}. \quad (5)$$

By summing (5) from $k = 2$ to $k = K$, we have

$$\begin{aligned} &\sum_{k=2}^{K} \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] \\ &\quad \leq \mathbb{E}[f(x_2)] - \mathbb{E}[f(x_{K+1})] + GB\mathbb{E}[\mathrm{tr}(\alpha_1 H_1^{-1} - \alpha_K H_K^{-1})] + \sum_{k=2}^{K} \frac{L\alpha_k^2 \sigma^2 \nu^2}{2b}. \end{aligned}$$

Since $\mu I_d \preceq H_K^{-1} \preceq \nu I_d$ for all $k \geq 1$, it follows that $\mathrm{tr}(\alpha_1 H_1^{-1}) \leq \alpha_1 \nu d$ and $\mathrm{tr}(\alpha_K H_K^{-1}) \geq 0$. Here, we note that $\mathbb{E}[f(x_{K+1})] \geq f_\star$, from Assumption 3.1 (A3). Therefore, we have

$$\sum_{k=2}^{K} \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] \leq \mathbb{E}[f(x_2)] - f_\star + GB\alpha_1 \nu d + \sum_{k=2}^{K} \frac{L\alpha_k^2 \sigma^2 \nu^2}{2b}. \quad (6)$$

Here, by adding both sides of (4) and (6), we have

$$\sum_{k=1}^{K} \alpha_k \left(\mu - \frac{L\alpha_k \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] \leq \underbrace{f(x_1) - f_\star + C_0 + GB\alpha_1 \nu d}_{C_1} + \underbrace{\frac{L\sigma^2 \nu^2}{2}}_{C_2} \cdot \frac{1}{b} \sum_{k=1}^{K} \alpha_k^2.$$

This completes the proof. □

Our convergence analysis (Theorem 3.7) allows the proposed framework (Algorithm 1) to use both constant and diminishing steps sizes. Theorems 3.8 and 3.9 are convergence analyses of Algorithm 1 with constant and diminishing steps sizes, respectively.

Theorem 3.8. *Under the assumptions in Theorem 3.7 and assuming that the constant step size $\alpha_k := \alpha$ satisfies $0 < \alpha < 2\mu L^{-1} \nu^{-2}$, the sequence $(x_k)_{k=1}^\infty \subset M$ generated by Algorithm 1 satisfies*

$$\frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\|\mathrm{grad}\, f(x_k)\|_2^2 \right] = \mathcal{O}\left(\frac{1}{K} + \frac{1}{b} \right).$$

Proof. We denote $\mathrm{grad}\, f(x_k)$ by $g(x_k)$. From Theorem 3.7, we obtain

$$\frac{1}{K} \sum_{k=1}^{K} \alpha \left(\mu - \frac{L\alpha \nu^2}{2} \right) \mathbb{E}\left[\|g(x_k)\|_2^2 \right] \leq \frac{C_1}{K} + \frac{C_2 \alpha^2}{b}. \quad (7)$$

Since $0 < \alpha < 2\mu L^{-1}\nu^{-2}$, it follows that $(2\alpha\mu - L\alpha^2\nu^2)/2 > 0$. Therefore, dividing both sides of (7) by $(2\alpha\mu - L\alpha^2\nu^2)/2$ gives

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|g(x_k)\|_2^2\right] \leq \frac{2C_1}{2\alpha\mu - L\alpha^2\nu^2}\cdot\frac{1}{K} + \frac{2C_2\alpha^2}{2\alpha\mu - L\alpha^2\nu^2}\cdot\frac{1}{b}.$$

This completes the proof. □

Theorem 3.9. *Under the assumptions in Theorem 3.7 and assuming that the diminishing step size $\alpha_k := \alpha/\sqrt{k}$ satisfies $\alpha \in (0,1]$, the sequence $(x_k)_{k=1}^{\infty} \subset M$ generated by Algorithm 1 satisfies*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathrm{grad}\, f(x_k)\|_2^2\right] = \mathcal{O}\left(\left(1+\frac{1}{b}\right)\frac{\log K}{\sqrt{K}}\right).$$

Proof. We denote $\mathrm{grad}\, f(x_k)$ by $g(x_k)$. Since $(\alpha_k)_{k=1}^{\infty}$ satisfies $\alpha_k \to 0$ $(k \to \infty)$, there exists a natural number $k_0 \geq 1$ such that, for all $k \geq 1$, if $k \geq k_0$, then $0 < \alpha_k < 2\mu L^{-1}\nu^{-2}$. Therefore, we obtain

$$0 < \mu - \frac{L\alpha_k\nu^2}{2} < \mu,$$

for all $k \geq k_0$. From Theorem 3.7, we have

$$\sum_{k=k_0}^{K}\alpha_k\left(\mu - \frac{L\alpha_k\nu^2}{2}\right)\mathbb{E}\left[\|g(x_k)\|_2^2\right] \leq C_1 + \frac{C_2}{b}\sum_{k=1}^{K}\alpha_k^2 - \sum_{k=1}^{k_0-1}\alpha_k\left(\mu - \frac{L\alpha_k\nu^2}{2}\right)\mathbb{E}\left[\|g(x_k)\|_2^2\right],$$

for all $K \geq k_0$. Since $(\alpha_k)_{k=1}^{\infty}$ is monotone decreasing and $\alpha_k > 0$, we obtain

$$\alpha_K\left(\mu - \frac{L\alpha_{k_0}\nu^2}{2}\right)\sum_{k=k_0}^{K}\mathbb{E}\left[\|g(x_k)\|_2^2\right] \leq C_1 + \frac{C_2}{b}\sum_{k=1}^{K}\alpha_k^2 + \sum_{k=1}^{k_0-1}L\alpha_k^2\nu^2\mathbb{E}\left[\|g(x_k)\|_2^2\right].$$

Dividing both sides of this inequality by $2^{-1}K\alpha_K(2\mu - L\alpha_{k_0}\nu^2) > 0$ yields

$$\begin{aligned}\frac{1}{K}\sum_{k=k_0}^{K}\mathbb{E}\left[\|g(x_k)\|_2^2\right] &\leq \frac{2}{K\alpha_K(2\mu - L\alpha_{k_0}\nu^2)}\left(C_1 + \frac{C_2}{b}\sum_{k=1}^{K}\alpha_k^2 + \sum_{k=1}^{k_0-1}L\alpha_k^2\nu^2\mathbb{E}\left[\|g(x_k)\|_2^2\right]\right)\\ &= \frac{1}{K\alpha_K}\cdot\underbrace{\frac{2}{2\mu - L\alpha_{k_0}\nu^2}\left(C_1 + \sum_{k=1}^{k_0-1}L\alpha_k^2\nu^2\mathbb{E}\left[\|g(x_k)\|_2^2\right]\right)}_{C_3}\\ &\quad + \frac{1}{bK\alpha_K}\cdot\underbrace{\frac{2C_2}{2\mu - L\alpha_{k_0}\nu^2}}_{C_4}\sum_{k=1}^{K}\alpha_k^2.\end{aligned}$$

From this and $\alpha_K := \alpha/\sqrt{K} < 1$, we obtain

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|g(x_k)\|_2^2\right] &\leq \frac{1}{K\alpha_K}\left(C_3 + \frac{C_4}{b}\sum_{k=1}^{K}\alpha_k^2\right) + \frac{1}{K\alpha_K}\sum_{k=1}^{k_0-1}\mathbb{E}\left[\|g(x_k)\|_2^2\right]\\ &= \frac{1}{\alpha\sqrt{K}}\left(C_3 + \sum_{k=1}^{k_0-1}\mathbb{E}\left[\|g(x_k)\|_2^2\right] + \frac{C_4}{b}\sum_{k=1}^{K}\alpha_k^2\right).\end{aligned}$$

From $\alpha \in (0,1]$, we have that

$$\sum_{k=1}^{K}\alpha_k^2 = \sum_{k=1}^{K}\frac{\alpha^2}{k} \leq \sum_{k=1}^{K}\frac{1}{k} \leq 1 + \int_1^K\frac{dt}{t} = 1 + \log K.$$

Therefore,

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|g(x_k)\|_2^2\right] \leq \frac{1}{\alpha\sqrt{K}}\left(C_3 + \sum_{k=1}^{k_0-1}\mathbb{E}\left[\|g(x_k)\|_2^2\right] + \frac{C_4}{b} + \frac{C_4}{b}\log K\right).$$

This completes the proof. □

4 Numerical Experiments

We experimentally compared our general framework of Riemannian adaptive optimization methods (Algorithms 1) with several choices of $(\phi_n)_{n=1}^{\infty}$ and $(\psi_n)_{n=1}^{\infty}$ with the following algorithms:

- RSGD (Bonnabel, 2013): Algorithm 1 with $\phi_k(g_1, \dots, g_k) = g_k$ and $\psi_k(g_1, \dots, g_k) = I_d$.
- RASA-LR, RASA-L, RASA-R (Kasai et al., 2019, Algorithm 1): $\beta = 0.99$.
- RAdam: Algorithm 1 with $(\phi_n)_{n=1}^{\infty}$ defined by (2), $(\psi_n)_{n=1}^{\infty}$ defined by (3), $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.
- RAMSGrad: Algorithm 2 with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

We experimented with both constant and diminishing step sizes. For each algorithm, we searched in the set $\{10^{-1}, 10^{-2}, \dots, 10^{-8}\}$ for the best initial step size α (both constant and diminishing). Note that the constant (resp. diminishing) step size was determined to be $\alpha_k = \alpha$ (resp. $\alpha_k = \alpha/\sqrt{k}$) for all $k \geq 1$. The experiments used a MacBook Air (M1, 2020) and the macOS Monterey version 12.2 operating system. The algorithms were written in Python 3.12.1 with the NumPy 1.26.0 package and the Matplotlib 3.9.1 package. The Python implementations of the methods used in the numerical experiments are available at `https://anonymous.4open.science/r/202408-adaptive-0BA6/README.md`.

4.1 Principal component analysis

We applied the algorithms to a principal component analysis (PCA) problem (Kasai et al., 2018; Roy et al., 2018). For N given data points $x_1, \dots, x_N \in \mathbb{R}^n$ and p $(\leq n)$, the PCA problem is equivalent to minimizing

$$f(U) := \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - UU^{\top}x_i\right\|_2^2, \tag{8}$$

on the Stiefel manifold $\mathrm{St}(p,n)$. Therefore, the PCA problem can be considered to be optimization problem on the Stiefel manifold.

In the experiments, we set p to 10 and the batch size b to 2^{10}. We used the QR-based retraction on the Stiefel manifold $\mathrm{St}(p,n)$ (Absil et al., 2008, Example 4.1.3), which is defined by

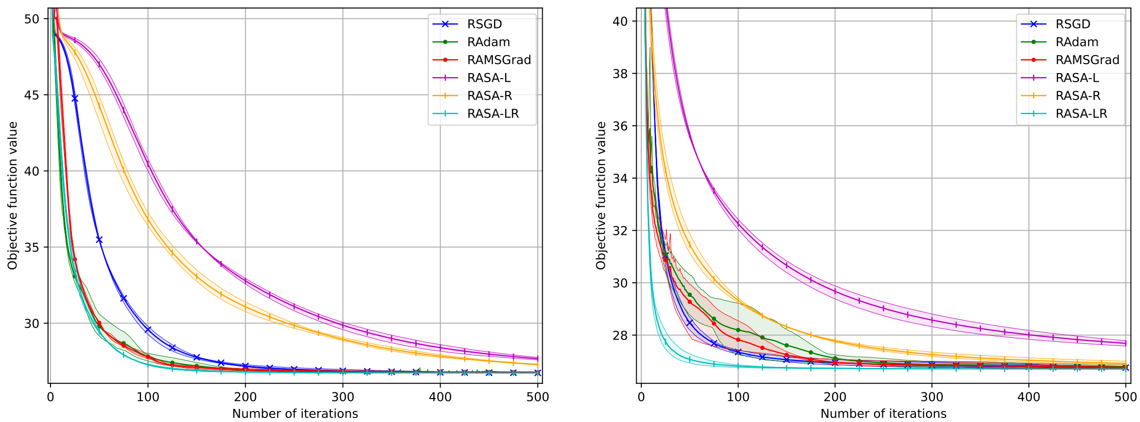
$$R_X(\eta) := \mathrm{qf}(X + \eta),$$

for $X \in \mathrm{St}(p,n)$ and $\eta \in T_X\,\mathrm{St}(p,n)$, where $\mathrm{qf}(\cdot)$ returns the Q-factor of the QR decomposition.

We evaluated the algorithms on training images of the MNIST dataset (LeCun et al., 1998) and the COIL100 dataset (Nene et al., 1996). The MNIST dataset contains 60,000 28×28 gray-scale images of handwritten digits. We transformed every image into a 784-dimensional vector and normalized its pixel values to lie in the range of $[0,1]$. Thus, we set $N = 60000$ and $n = 784$. The COIL100 dataset contains 7,200 normalized color camera images of the 100 objects taken from different angles. As in the previous study (Kasai et al., 2019), we resized them to 32×32 pixels. Thus, we set $N = 7200$ and $n = 1024$.

Figure 1(a) (resp. Figure 1(b)) shows the performances of the algorithms with a constant (resp. diminishing) step size for the objective function values defined by (8) with respect to the number of iterations on the

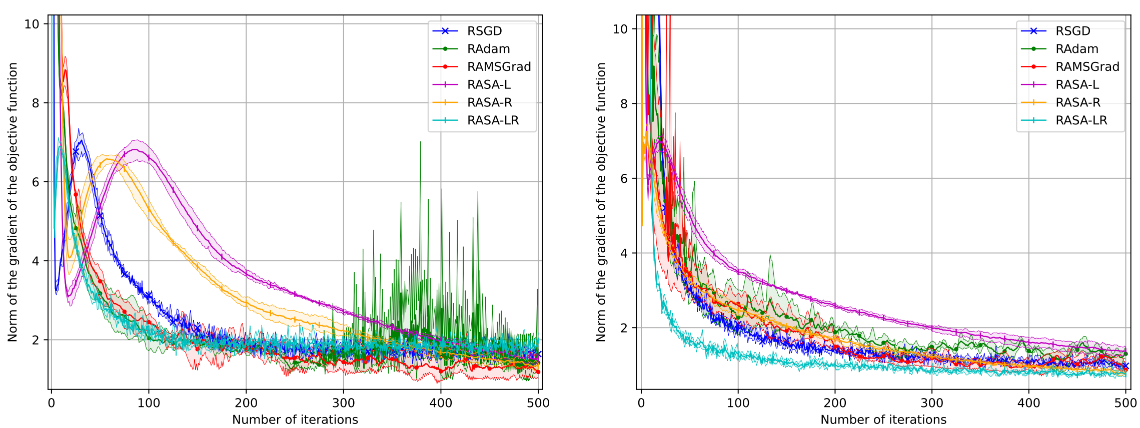
MNIST dataset, while Figure 3(a) and 3(b) present those on the COIL100 dataset. Figure 2(a) (resp. 2(b)) presents the performances of the algorithms with a constant (resp. diminishing) step size for the norm of the gradient of objective function defined by (8) with respect to the number of iterations on the MNIST dataset, while Figure 4(a) and 4(b) present those on the COIL100 dataset. The experiments were performed for three random initial points, and the thick line plots the average of all experiments. The area bounded by the maximum and minimum values is painted the same color as the corresponding line.



(a) constant learning rate

(b) diminishing learning rate

Figure 1: Objective function value defined by (8) versus number of iterations on the MNIST datasets.

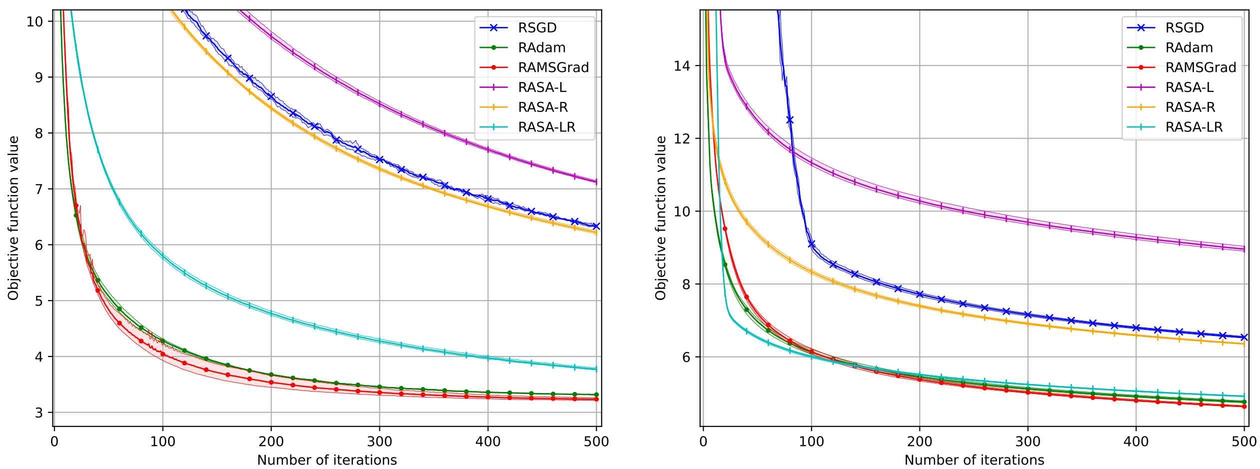


(a) constant learning rate

(b) diminishing learning rate

Figure 2: Norm of the gradient of objective function defined by (8) versus number of iterations on the MNIST datasets.

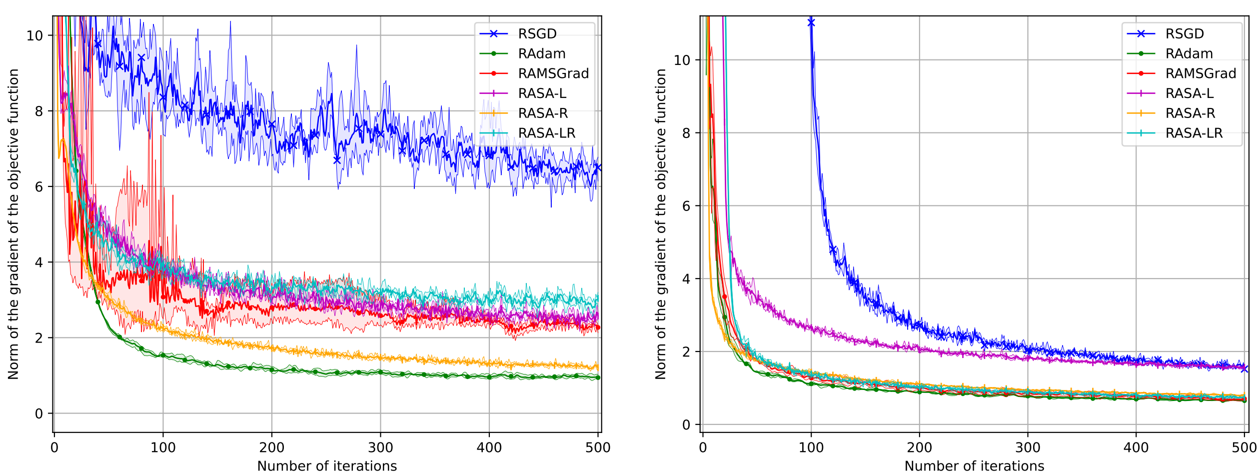
Figure 1(a) indicates that RAdam and RAMSGrad (Algorithm 2) performed comparably to RASA-LR in the sense of minimizing the objective function value. Figure 1(b) indicates that RAdam and RAMSGrad (Algorithm 2) outperformed RASA-L and RASA-R. Figure 2(a) shows that RAMSGrad (Algorithm 2) performed better than RASA-LR in the sense of minimizing the full gradient norm of the objective function. Figure 3(a) indicates that RAdam and RAMSGrad (Algorithm 2) had the best performance in the sense of minimizing the objective function value. Figure 3(b) indicates that RAdam and RAMSGrad (Algorithm 2) performed comparably to RASA-LR. Figure 4(a) shows that RAdam had the best performance in the sense of minimizing the full gradient norm of the objective function. Figure 4(b) indicates that RAdam and RAMSGrad (Algorithm 2) performed comparably to RASA-R and RASA-LR.



(a) constant learning rate

(b) diminishing learning rate

Figure 3: Objective function value defined by (8) versus number of iterations on the COIL100 datasets.



(a) constant learning rate

(b) diminishing learning rate

Figure 4: Norm of the gradient of objective function defined by (8) versus number of iterations on the COIL100 datasets.

4.2 Low-rank matrix completion

We applied the algorithms to the low-rank matrix completion (LRMC) problem (Boumal & Absil, 2015; Kasai et al., 2019; Hu et al., 2024). The LRMC problem aims to recover a low-rank matrix from an incomplete matrix $X = (X_{ij}) \in \mathbb{R}^{n\times N}$. We denote the set of observed entries by $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, N\}$, i.e., $(i, j) \in \Omega$ if and only if X_{ij} is known. Here, we defined the orthogonal projection $P_{\Omega_i} : \mathbb{R}^n \to \mathbb{R}^n : a \mapsto P_{\Omega_i}(a)$ such that the j-th element of $P_{\Omega_i}(a)$ is a_j if $(i, j) \in \Omega$, and 0 otherwise. Moreover, we defined $q_i : \mathbb{R}^{n\times p} \times \mathbb{R}^n \to \mathbb{R}^p$ as

$$q_i(U, x) := \operatorname*{arg\,min}_{a\in\mathbb{R}^p} \left\| P_{\Omega_i}(Ua - x) \right\|_2, \tag{9}$$

for $i \geq 1$. By partitioning $X = (x_1, \ldots, x_N)$, the rank-p LRMC problem is equivalent to minimizing

$$f(U) := \frac{1}{2N} \sum_{i=1}^{N} \left\| P_{\Omega_i}(Uq_i(U, x_i) - x_i) \right\|_2^2, \tag{10}$$

on the Grassmann manifold $\mathrm{Gr}(p, n)$. Therefore, the rank-p LRMC problem can be considered to be an optimization problem on the Grassmann manifold (see Hu et al. (2024, Section 1) or Kasai et al. (2019, Section 6.3) for details).

We evaluated the algorithms on the MovieLens-1M[1] datasets (Harper & Konstan, 2015) and the Jester[2] datasets for recommender systems. The MovieLens-1M datasets contains 1,000,209 ratings given by 6,040 users on 3,952 movies. Thus, we set $N = 3952$ and $n = 6040$. The Jester datasets contains ratings of 100 jokes given by 24,983 users with scores from -10 to 10. Thus, we set $N = 24983$ and $n = 100$.

In the experiments, we set p to 10 and the batch size b to 2^8. We used `numpy.linalg.lstsq`[3] to solve the least squares problem (9). We used a retraction based on a polar decomposition on the Grassmann manifold $\mathrm{Gr}(p, n)$ (Absil et al., 2008, Example 4.1.3), which is defined through

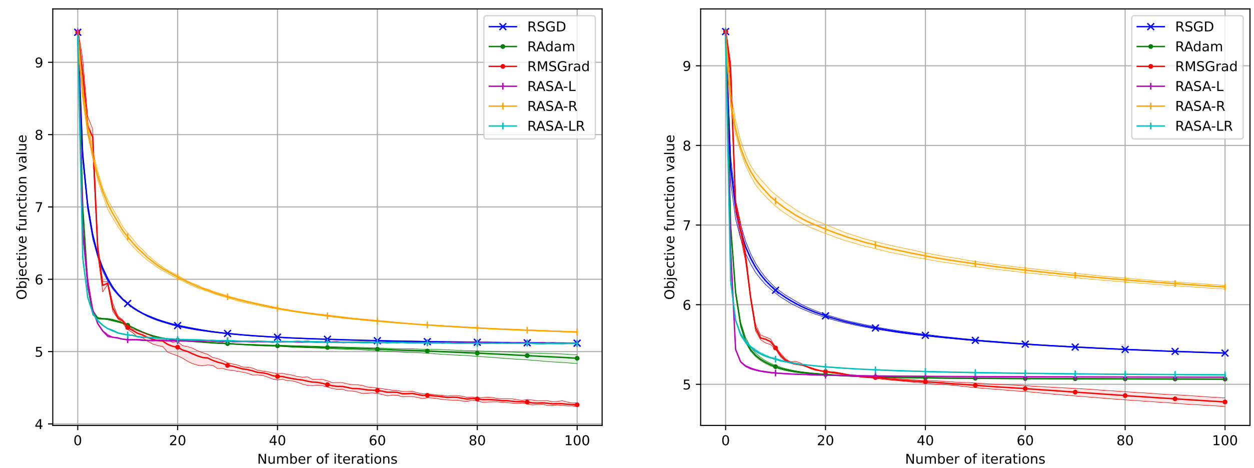
$$\overline{R_{[X]}(\eta)} := (X + \bar{\eta}_X)(I_p + \bar{\eta}_X^\top \bar{\eta}_X)^{-\frac{1}{2}},$$

for $[X] \in \mathrm{Gr}(p, n)$ and $\eta \in T_{[X]} \mathrm{Gr}(p, n)$.

Figure 5(a) (resp. Figure 5(b)) shows the performances of the algorithms with a constant (resp. diminishing) step size for objective function values defined by (10) with respect to the number of iterations on the MovieLens-1M dataset, while Figure 7(a) and 7(b) present those on the Jester dataset. Figure 6(a) (resp. 6(b)) shows the performances of the algorithms with a constant (resp. diminishing) step size for the norm of the gradient of the objective function defined by (8) with respect to the number of iterations on the MovieLens-1M dataset, while Figure 7(a) and 7(b) present those on the Jester dataset. The experiments were performed for three random initial points, and the thick line plots the average results of all experiments. The area bounded by the maximum and minimum values is painted the same color as the corresponding line.

Figure 5(a) indicates that RAMSGrad (Algorithm 2) performed better than RASA-L and RASA-LR in the sense of minimizing the objective function value. Figure 5(b) shows that RAdam and RAMGRad (Algorithm 2) performed comparably to RASA-L and RASA-LR. Figure 6(a) indicates that RAdam performed comparably to RASA-R in the sense of minimizing the full gradient norm of the objective function. Figure 6(b) shows that RAdam outperformed RASA-R. Figure 7(a) and 7(b) indicate that RAMSGrad (Algorithm 2) performed better than RASA-LR in the sense of minimizing the objective function value. Figure 8(a) indicates that RAdam performed comparably to RASA-L and RASA-R in the sense of minimizing the full gradient norm of the objective function. Figure 8(b) shows that RAMSGrad (Algorithm 2) outperformed RASA-L and RASA-R.

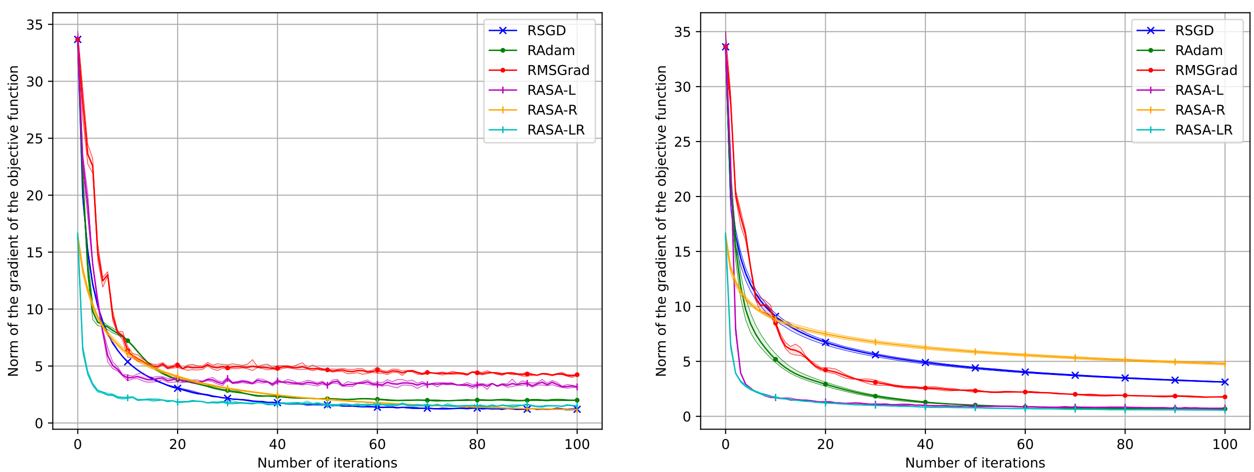
[1]`https://grouplens.org/datasets/movielens/`
[2]`https://grouplens.org/datasets/jester`
[3]`https://numpy.org/doc/1.26/reference/generated/numpy.linalg.lstsq.html`



(a) constant learning rate

(b) diminishing learning rate

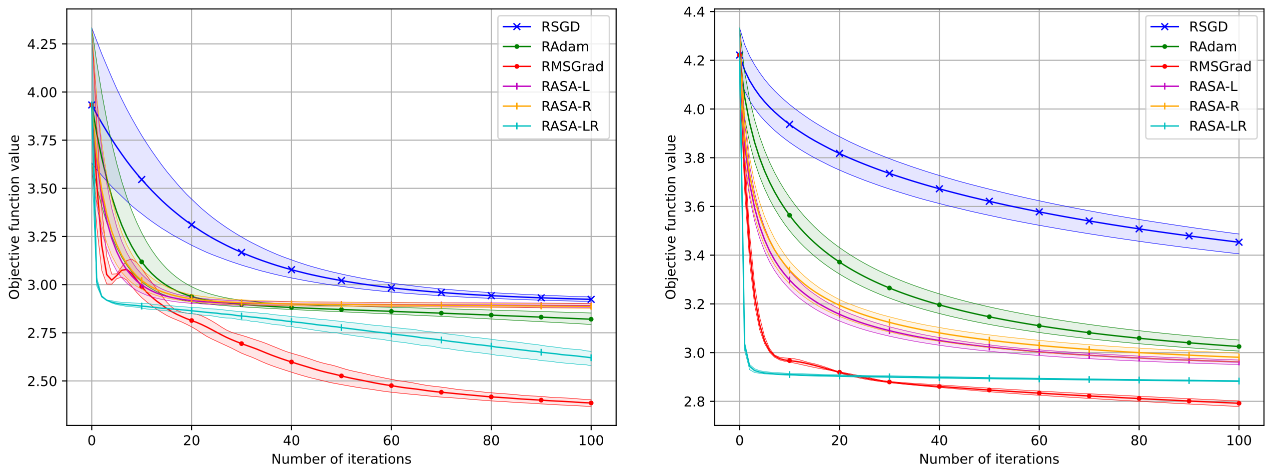
Figure 5: Objective function value defined by (10) versus number of epochs (iterations for the entire dataset) on the MovieLens-1M datasets.



(a) constant learning rate

(b) diminishing learning rate

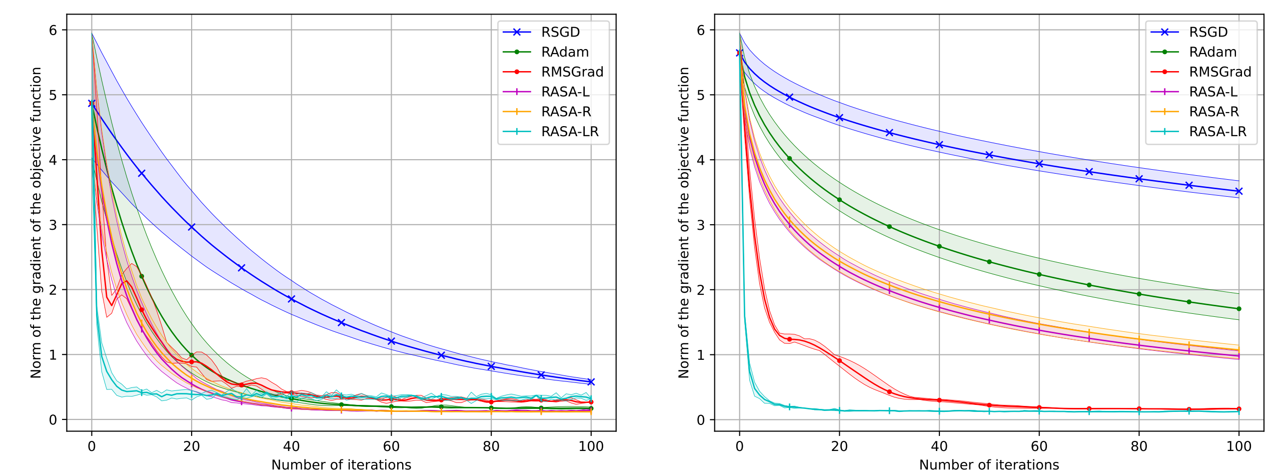
Figure 6: Norm of the gradient of objective function defined by (10) versus number of epochs (iterations for the entire dataset) on the MovieLens-1M datasets.



(a) constant learning rate

(b) diminishing learning rate

Figure 7: Objective function value defined by (10) versus number of epochs (iterations for the entire dataset) on the Jester datasets.



(a) constant learning rate

(b) diminishing learning rate

Figure 8: Norm of the gradient of objective function defined by (10) versus number of epochs (iterations for the entire dataset) on the Jester datasets.

5 Conclusion

This paper proposed a general framework of Riemannian adaptive optimization methods, which encapsulates several stochastic optimization algorithms on Riemannian manifolds. The framework incorporates the mini-batch strategy often used in deep learning. We also proposed AMSGrad that works on embedded submanifolds of Euclidean space within our framework. In addition, we gave convergence analyses that are valid for both a constant and diminishing step size. The analyses also revealed the relationship between the convergence rate and mini-batch size. We numerically compared the AMSGrad with the existing algorithms by applying them to the principal component analysis and the low-rank matrix completion problem, which can be considered to be Riemannian optimization problems. Numerical experiments showed that the proposed method performs well against PCA. RAdam and RAMSGrad performed well for constant and diminishing step sizes especially on the COIL100 dataset.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *Proceedings of The International Conference on Learning Representations*, 2019.

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

Nicolas Boumal and P-A Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.

Léopold Cambier and P-A Absil. Robust low-rank matrix completion by Riemannian optimization. *SIAM Journal on Scientific Computing*, 38(5):S440–S460, 2016.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *Proceedings of The International Conference on Learning Representations*, 2019.

Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30, 2017.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, pp. 2121–2159, 2011.

F Maxwell Harper and Joseph A Konstan. The MovieLens datasets: History and context. *Acm transactions on interactive intelligent systems*, 5(4):1–19, 2015.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *COURSERA: Neural Networks for Machine Learning*, pp. 26–31, 2012.

Jiang Hu, Ruicheng Ao, Anthony Man-Cho So, Minghan Yang, and Zaiwen Wen. Riemannian natural gradient methods. *SIAM Journal on Scientific Computing*, 46(1):A204–A231, 2024.

Wen Huang, Kyle A Gallivan, and P-A Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

Hideaki Iiduka. Theoretical analysis of Adam using hyperparameters close to one without Lipschitz smoothness. *Numerical Algorithms*, 95(1):383–421, 2024.

Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian stochastic recursive gradient algorithm. In *International Conference on Machine Learning*, pp. 2516–2524. PMLR, 2018.

Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *International Conference on Machine Learning*, pp. 3262–3271. PMLR, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceedings of The International Conference on Learning Representations*, pp. 1–15, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (COIL-100). *Technical Report CUCS-006-96*, 1996.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pp. 6338–6347, 2017.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. *Proceedings of The International Conference on Learning Representations*, pp. 1–23, 2018.

Soumava Kumar Roy, Zakaria Mhammedi, and Mehrtash Harandi. Geometry aware constrained optimization techniques for deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4460–4469, 2018.

Hiroyuki Sakai and Hideaki Iiduka. Riemannian adaptive optimization algorithm and its application to natural language processing. *IEEE Transactions on Cybernetics*, 52(8):7328–7339, 2021.

Hiroyuki Sato. *Riemannian Optimization and Its Applications*, volume 670. Springer, 2021.

Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 2955–2961, 2018.

Jihun Yun and Eunho Yang. Riemannian SAM: Sharpness-aware minimization on riemannian manifolds. *Advances in Neural Information Processing Systems*, 36, 2024.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016.

Jingzhao Zhang, Hongyi Zhang, and Suvrit Sra. R-SPIDER: A fast Riemannian stochastic optimization algorithm with curvature independent rate. *arXiv preprint arXiv:1811.04194*, 2018.

Dongruo Zhou, Jinghui Chen, Yuan Cao, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *Transactions on Machine Learning Research*, 2024.

Pan Zhou, Xiao-Tong Yuan, and Jiashi Feng. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 138–147. PMLR, 2019.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33:18795–18806, 2020.

A Proof of Lemma 3.2

Proof. From (1), Assumption 3.1 (A1) and the linearity of $\mathbb{E}_k[\cdot]$, we have

$$\mathbb{E}_k\left[\operatorname{grad} f_{B_k}(x_k)\right] = \frac{1}{b}\sum_{i=1}^{b} \mathbb{E}_k\left[\operatorname{grad} f_{s_{k,i}}(x_k)\right] = \operatorname{grad} f(x_k).$$

This completes the proof. $\square$

B Proof of Lemma 3.3

Proof. From $\|a+b\|_2^2 = \|a\|_2^2 + 2\langle a, b\rangle_2 + \|b\|_2^2$, we obtain

$$\begin{aligned}\mathbb{E}_k\left[\|\operatorname{grad} f_{B_k}(x_k)\|_2^2\right] &= \mathbb{E}_k\left[\|\operatorname{grad} f_{B_k}(x_k) - \operatorname{grad} f(x_k)\|_2^2\right] \\ &\quad + 2\mathbb{E}_k\left[\langle \operatorname{grad} f_{B_k}(x_k) - \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\rangle_2\right] + \mathbb{E}_k\left[\|\operatorname{grad} f(x_k)\|_2^2\right], \end{aligned} \tag{11}$$

for all $k \geq 1$. From (1) and Assumption 3.1 (A1), the first term on the right-hand side of (11) yields

$$\begin{aligned}\mathbb{E}_k\left[\|\operatorname{grad} f_{B_k}(x_k) - \operatorname{grad} f(x_k)\|_2\right] &= \mathbb{E}_k\left[\left\|\frac{1}{b}\sum_{i=1}^{b} \operatorname{grad} f_{s_{k,i}}(x_k) - \operatorname{grad} f(x_k)\right\|_2^2\right] \\ &= \frac{1}{b^2}\mathbb{E}_k\left[\sum_{i=1}^{b}\left\|\operatorname{grad} f_{s_{k,i}}(x_k) - \operatorname{grad} f(x_k)\right\|_2^2\right] \\ &\leq \frac{\sigma^2}{b},\end{aligned}$$

where the second equality comes from Assumption 3.1 (A2). From Lemma 3.2, the second term on the right-hand side of (11) yields

$$\begin{aligned}2\mathbb{E}_k\left[\langle \operatorname{grad} f_{B_k}(x_k) - \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\rangle_2\right] &= 2\left\langle \mathbb{E}_k[\operatorname{grad} f_{B_k}(x_k)] - \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\right\rangle_2 \\ &= 2\left\langle \operatorname{grad} f(x_k) - \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\right\rangle_2 \\ &= 0.\end{aligned}$$

Therefore, we obtain

$$\mathbb{E}_k\left[\|\operatorname{grad} f_{B_k}(x_k)\|_2^2\right] \leq \frac{\sigma^2}{b} + \|\operatorname{grad} f(x_k)\|_2^2,$$

for all $k \geq 1$. This completes the proof. $\square$

C Proof of Lemma 3.5

Proof. Note that from Assumption 3.1 (A3), we have

$$g_{k,i}^2 \leq g_{k,1}^2 + \cdots + g_{k,d}^2 = \|g_k\|_2^2 \leq B^2$$

for all $k \geq 1$ and $i = 1, \dots, d$. The proof is by induction. For $k = 1$, from $0 \leq \beta_2 < 1$, we have

$$\hat{v}_{1,i} = v_{1,i} := \beta_2 v_{0,i} + (1-\beta_2) g_{1,i}^2 = (1-\beta_2) g_{1,i}^2 \leq g_{1,i}^2 \leq B^2.$$

Suppose that $\hat{v}_{k-1,i} \leq B^2$. From $v_{k-1,i} \leq \hat{v}_{k-1,i} \leq B^2$, we have

$$v_{k,i} = \beta_2 v_{k-1,i} + (1-\beta_2) g_{k,i}^2 \leq \beta_2 B^2 + (1-\beta_2) B^2 = B^2.$$

Thus, induction ensures that $v_{k,i} \leq B^2$ for all $k \geq 1$. $\square$

D Proof of Lemma 3.6

Proof. We denote $\operatorname{grad} f(x_k)$ by $g(x_k)$. From Proposition 3.4, we have

$$f(x_{k+1}) \leq f(x_k) + \left\langle g(x_k), -\alpha_k P_{x_k}(H_k^{-1} g_k) \right\rangle_2 + \frac{L}{2} \left\| -\alpha_k P_{x_k}(H_k^{-1} g_k) \right\|_2^2,$$

for all $k \geq 1$. From the linearity and symmetry of P_{x_k}, we obtain

$$\left\langle g(x_k), -\alpha_k P_{x_k}(H_k^{-1} g_k) \right\rangle_2 = \left\langle P_{x_k}(g(x_k)), -\alpha_k H_k^{-1} g_k \right\rangle_2 = \left\langle g(x_k), -\alpha_k H_k^{-1} g_k \right\rangle_2.$$

From the symmetry of P_{x_k} and $P_{x_k} \circ P_{x_k} = P_{x_k}$, we have

$$\begin{aligned}
\left\| -\alpha_k P_{x_k}(H_k^{-1} g_k) \right\|_2^2 &= \alpha_k^2 \left\| P_{x_k}(H_k^{-1} g_k) \right\|_2^2 \\
&= \alpha_k^2 \left\langle P_{x_k}(H_k^{-1} g_k), P_{x_k}(H_k^{-1} g_k) \right\rangle_2 \\
&= \alpha_k^2 \left\langle H_k^{-1} g_k, P_{x_k}(P_{x_k}(H_k^{-1} g_k)) \right\rangle_2 \\
&= \alpha_k^2 \left\langle H_k^{-1} g_k, P_{x_k}(H_k^{-1} g_k) \right\rangle_2 \\
&\leq \alpha_k^2 \left\| H_k^{-1} g_k \right\|_2 \left\| P_{x_k}(H_k^{-1} g_k) \right\|_2.
\end{aligned}$$

Here, when $P_{x_k}(H_k^{-1} g_k) \neq 0 \in \mathbb{R}^d$, it follows that

$$\left\| -\alpha_k P_{x_k}(H_k^{-1} g_k) \right\|_2^2 \leq \alpha_k^2 \left\| H_k^{-1} g_k \right\|_2 \leq \alpha_k^2 \nu^2 \left\| g_k \right\|_2^2,$$

where the second inequality comes from $H_k^{-1} \preceq \nu I_d$. On the other hand, this inequality clearly holds if $P_{x_k}(H_k^{-1} g_k) = 0 \in \mathbb{R}^d$. Therefore, we obtain

$$f(x_{k+1}) \leq f(x_k) + \left\langle g(x_k), -\alpha_k H_k^{-1} g_k \right\rangle_2 + \frac{L \alpha_k^2 \nu^2}{2} \left\| g_k \right\|_2^2,$$

for all $k \geq 1$. This completes the proof. □

E Linear algebra lemma

Lemma E.1. *Let* $a = (a_1, \dots, a_n)^\top \in \mathbb{R}^n$, $b = (b_1, \dots, b_n)^\top \in \mathbb{R}^n$ *and* $D = \operatorname{diag}(d_1, \dots, d_n) \in \mathcal{S}_+^n \cap \mathcal{D}^n$. *If* $\|a\|_2 \leq A$ *and* $\|b\|_2 \leq B$, *then*

$$a^\top D b \leq AB \operatorname{tr}(D).$$

Proof. From $\|a\|_2 \leq A$ and $\|b\|_2 \leq A$, we have $|a_i| \leq A$ and $|b_i| \leq B$ for all $i = 1, \dots, n$. Therefore, we obtain

$$a^\top D b = \sum_{i=1}^n a_i d_i b_i \leq \sum_{i=1}^n |a_i| \cdot |b_i| \, d_i \leq AB \sum_{i=1}^n d_i \leq AB \operatorname{tr}(D).$$

This completes the proof. □