

# On the Oracle Complexity of a Riemannian Inexact Augmented Lagrangian Method for Nonsmooth Composite Problems over Riemannian Submanifolds

Meng Xu · Bo Jiang · Ya-Feng Liu ·  
Anthony Man-Cho So

Received: date / Accepted: date

**Abstract** In this paper, we establish for the first time the oracle complexity of a Riemannian inexact augmented Lagrangian (RiAL) method with the classical dual update for solving a class of nonsmooth composite problems over Riemannian submanifolds. By using the Riemannian gradient descent method with a specified stopping criterion for solving the inner subproblem, we show that the RiAL method can find an  $\varepsilon$ -stationary point of the considered problem with  $\mathcal{O}(\varepsilon^{-3})$  calls to the first-order oracle. This achieves the best oracle complexity known to date. Numerical results demonstrate that the use of the classical dual stepsize is crucial to the high efficiency of the RiAL method.

**Keywords** Riemannian augmented Lagrangian method, Riemannian nonsmooth optimization, first-order oracle complexity

---

Meng Xu

LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, Beijing, China.

E-mail: xumeng22@mails.ucas.ac.cn

Bo Jiang

Ministry of Education Key Laboratory of NSLSCS, School of Mathematical Sciences, Nanjing Normal University, Nanjing, China.

E-mail: jiangbo@njnu.edu.cn

Ya-Feng Liu

LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China.

E-mail: yafliu@lsec.cc.ac.cn

Anthony Man-Cho So

Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, HKSAR, China.

E-mail: manchoso@se.cuhk.edu.hk

## 1 Introduction

In this paper, we consider the following Riemannian nonsmooth composite problem:

$$\min_{x \in \mathcal{M}} \{\Phi(x) := f(x) + h(\mathcal{A}(x))\}, \quad (1.1)$$

where  $\mathcal{M}$  is a Riemannian submanifold embedded in a finite-dimensional Euclidean space  $\mathcal{E}_1$ ,  $f : \mathcal{E}_1 \rightarrow \mathbb{R}$  is a continuously differentiable function,  $\mathcal{A} : \mathcal{E}_1 \rightarrow \mathcal{E}_2$  is a smooth mapping with  $\mathcal{E}_2$  being another finite-dimensional Euclidean space, and  $h : \mathcal{E}_2 \rightarrow \mathbb{R}$  is a convex Lipschitz continuous function with a tractable proximal mapping. Many problems in machine learning and signal processing can be formulated as problem (1.1), such as sparse principal component analysis (PCA) [24, 50], fair PCA [41, 46, 47], sparse canonical correlation analysis (CCA) [10, 12, 14], sparse spectral clustering [34, 35, 42], and beamforming design [32]. A variety of algorithms can be applied to tackle problem (1.1), including Riemannian subgradient-type methods [5, 16–18, 29], Riemannian proximal gradient-type methods [8, 9, 19, 20, 31, 42], Riemannian smoothing-type algorithms [4, 37, 48], splitting-type methods [12, 13, 26–28, 49], and Riemannian min-max algorithms [45, 46]. Among the previously mentioned algorithms, the Riemannian augmented Lagrangian (AL) method has demonstrated advantages in addressing the general mapping  $\mathcal{A}$  along with possible additional constraints in problem (1.1) [49]. In this paper, we focus on the Riemannian AL method for solving problem (1.1).

As a powerful algorithmic framework for constrained problems, the AL method has been extensively studied since 1960s [15, 38]. At each iteration, it updates the primal variable by (approximately) minimizing the AL function followed by a dual (gradient ascent) step to update the dual variable. Recently, the AL method has been generalized to tackle optimization problems with Riemannian manifold constraints (e.g., problem (1.1)), resulting in various efficient Riemannian AL methods (e.g., [12, 13, 49]). Below, we briefly introduce several such algorithms, which maintain the manifold constraint within the subproblem when solving problem (1.1). For some recent advances in AL methods and their variants, one can refer to [11, 25, 30, 33, 40, 44] and the references therein.

When  $\mathcal{A}$  is a linear mapping, Deng and Peng [13], and Deng et al. [12] proposed two types of Riemannian AL methods, where each subproblem is solved inexactly, for solving problem (1.1) on a compact manifold  $\mathcal{M}$ . The first one has asymptotic convergence, while the second achieves the best-known first-order oracle complexity of  $\mathcal{O}(\varepsilon^{-3})$  to attain an  $\varepsilon$ -stationary point. To establish the convergence or complexity results, additional requirements are imposed on the dual updates therein. Specifically, in [13], the dual update is followed by a projection onto a specified compact set at each iteration, in order to ensure the boundedness of the (Lagrange) multiplier sequence. In contrast, the work [12] employed a damped technique to satisfy the same boundedness requirement for the multiplier sequence. However, as observed in [25], such damped dual stepsizes may slow down the convergence of AL-like methods. When  $\mathcal{A}$  is a

**Table 1** Comparison of state-of-the-art Riemannian AL methods for solving problem (1.1).

Algorithm	Mapping $\mathcal{A}$	Dual Stepsize	Complexity
MIAL [13]	linear	classical but with projection	—
MAL [49]	nonlinear	classical	—
ManIAL [12]	linear	damped	$\mathcal{O}(\varepsilon^{-3})$
RiAL (this paper)	nonlinear	classical	$\mathcal{O}(\varepsilon^{-3})$

general nonlinear mapping, Zhou et al. [49] proposed a manifold-based AL (MAL) method with classical dual updates and established its global convergence. However, the oracle complexity of their approach remains unclear. In summary, the oracle complexity of the Riemannian AL method with the classical dual update for solving problem (1.1) involving a general nonlinear mapping  $\mathcal{A}$  remains unknown. Given this background, *we are motivated to investigate the oracle complexity of such method for solving problem (1.1).*

In this paper, we propose a Riemannian inexact AL (RiAL) method, where each subproblem is solved to a specified accuracy using the Riemannian gradient descent (RGD) method. We establish its first-order oracle complexity of  $\mathcal{O}(\varepsilon^{-3})$  for finding an  $\varepsilon$ -stationary point of problem (1.1). As seen from Table 1, which summarizes the applicability and complexity of different Riemannian AL methods for solving problem (1.1), our proposed approach is able to tackle more general problem settings and achieves the best-known oracle complexity result. Additionally, we present numerical results on sparse PCA, sparse CCA, and sparse spectral clustering to demonstrate that the proposed RiAL method outperforms the ManIAL method in terms of computational efficiency. This suggests that the damped dual stepsize used in ManIAL slows down the convergence of the algorithm.

## 2 Notation and Preliminaries

We begin by introducing the notation and some concepts in Riemannian optimization [1, 6]. Let  $\langle \cdot, \cdot \rangle$  and  $\|\cdot\|$  denote the standard inner product and its induced norm on the Euclidean space  $\mathcal{E}$ , respectively. Let  $\mathcal{M}$  be a Riemannian submanifold embedded in  $\mathcal{E}$  and  $T_x\mathcal{M}$  denote the tangent space to  $\mathcal{M}$  at  $x \in \mathcal{M}$ . Throughout this paper, the Riemannian metric on  $\mathcal{M}$  is induced from the standard Euclidean product. The Riemannian gradient of a smooth function  $f : \mathcal{E} \rightarrow \mathbb{R}$  at a point  $x \in \mathcal{M}$  is given by  $\text{grad } f(x) = \text{proj}_{T_x\mathcal{M}}(\nabla f(x))$ , where  $\nabla f(x)$  is the Euclidean gradient of  $f$  at  $x$  and  $\text{proj}_{T_x\mathcal{M}}(\cdot)$  is the Euclidean projection operator onto  $T_x\mathcal{M}$ . A retraction at  $x \in \mathcal{M}$  is a smooth mapping  $R_x : T_x\mathcal{M} \rightarrow \mathcal{M}$  satisfying (i)  $R_x(\mathbf{0}_x) = x$ , where  $\mathbf{0}_x$  is the zero element in  $T_x\mathcal{M}$ ; (ii)  $\frac{d}{dt}R_x(tv)|_{t=0} = v$  for all  $v \in T_x\mathcal{M}$ . Without loss of generality, we assume that the retraction  $R_x$  is globally defined over  $T_x\mathcal{M}$ .

Next, we introduce some necessary notions in convex analysis [3, 39]. For a subset  $\mathcal{X}$  in  $\mathcal{E}$ , we use  $\text{conv } \mathcal{X}$  to denote the convex hull of  $\mathcal{X}$ . Let  $h : \mathcal{E} \rightarrow \mathbb{R}$  be a convex function. For a given constant  $\lambda > 0$ , the proximal mapping and

the Moreau envelope of  $h$  are defined as

$$\text{prox}_{\lambda h}(x) = \arg \min_{u \in \mathcal{E}} \left\{ h(u) + \frac{1}{2\lambda} \|u - x\|^2 \right\},$$

and

$$M_{\lambda h}(x) = \min_{u \in \mathcal{E}} \left\{ h(u) + \frac{1}{2\lambda} \|u - x\|^2 \right\}, \quad (2.1)$$

respectively. The following theorem characterizes several useful properties related to the subgradient and the Moreau envelope.

**Theorem 2.1** ([3, Theorems 3.61, 6.39, and 6.67]) Let  $h : \mathcal{E} \rightarrow \mathbb{R}$  be a convex  $L_h$ -Lipschitz continuous function and  $h^*$  be its conjugate function. Then,  $\|g\| \leq L_h$  for any  $g \in \partial h(x)$  and  $x \in \mathcal{E}$ . Moreover, for a given constant  $\lambda > 0$  and any  $x \in \mathcal{E}$ ,

$$M_{\lambda h}(x) = \max_{z \in \mathcal{E}} \left\{ \langle x, z \rangle - h^*(z) - \frac{\lambda}{2} \|z\|^2 \right\}$$

and

$$\nabla M_{\lambda h}(x) = \frac{1}{\lambda} (x - \text{prox}_{\lambda h}(x)) \in \partial h(\text{prox}_{\lambda h}(x)).$$

### 3 The RiAL Method and Its Oracle Complexity

In this section, we introduce the RiAL method for solving problem (1.1) and establish its first-order oracle complexity.

#### 3.1 The RiAL Method

The key challenge in solving problem (1.1) stems from the presence of both a manifold constraint and a nonsmooth objective function. To tackle this, existing Riemannian AL methods split these two difficulties. Specifically, by introducing an auxiliary variable  $y$ , problem (1.1) can be equivalently reformulated as

$$\min_{x \in \mathcal{M}, y \in \mathcal{E}_2} f(x) + h(y) \quad \text{s.t.} \quad \mathcal{A}(x) - y = 0. \quad (3.1)$$

The augmented Lagrangian function associated with problem (3.1) is defined as

$$\mathcal{L}_\sigma(x, y; z) := f(x) + h(y) + \langle z, \mathcal{A}(x) - y \rangle + \frac{\sigma}{2} \|\mathcal{A}(x) - y\|^2,$$

where  $z$  is the Lagrange multiplier (also called dual variable) corresponding to the constraint  $\mathcal{A}(x) - y = 0$  and  $\sigma > 0$  is the penalty parameter. At the  $k$ -th iteration, an ordinary Riemannian AL method updates the next point as

$$(x_{k+1}, y_{k+1}) \approx \arg \min_{x \in \mathcal{M}, y \in \mathcal{E}_2} \mathcal{L}_{\sigma_k}(x, y; z_k).$$

Observe that for any fixed  $x \in \mathcal{M}$ , the optimal  $y$  in the above minimization problem can be expressed as  $y^* = \text{prox}_{h/\sigma_k}(\mathcal{A}(x) + z_k/\sigma_k)$ . Similar to the Riemannian AL methods in [12, 13, 49], we update  $x_{k+1}$  and  $y_{k+1}$  via the following scheme:

$$x_{k+1} \approx \arg \min_{x \in \mathcal{M}} \mathcal{L}_k(x), \quad (3.2a)$$

$$y_{k+1} = \text{prox}_{h/\sigma_k} \left( \mathcal{A}(x_{k+1}) + \frac{z_k}{\sigma_k} \right). \quad (3.2b)$$

Here,

$$\mathcal{L}_k(x) := \min_{y \in \mathcal{E}_2} \mathcal{L}_{\sigma_k}(x, y; z_k) = f(x) + M_{h/\sigma_k} \left( \mathcal{A}(x) + \frac{z_k}{\sigma_k} \right), \quad (3.3)$$

where  $M_{h/\sigma_k}(\cdot)$  is defined in (2.1). Let  $\nabla \mathcal{A}^\top(x)$  be the adjoint mapping of  $\nabla \mathcal{A}(x)$ . We know from Theorem 2.1 that  $\mathcal{L}_k$  is differentiable and

$$\nabla \mathcal{L}_k(x) = \nabla f(x) + \sigma_k \nabla \mathcal{A}(x)^\top \left[ \mathcal{A}(x) + \frac{z_k}{\sigma_k} - \text{prox}_{h/\sigma_k} \left( \mathcal{A}(x) + \frac{z_k}{\sigma_k} \right) \right]. \quad (3.4)$$

Therefore, we propose to use the RGD method to solve subproblem (3.2a). Specifically, starting from an initial point  $x_{k,1} = x_k$ , the iteration of RGD for  $t \geq 1$  is given by

$$x_{k,t+1} = R_{x_{k,t}}(-\zeta_{k,t} \text{grad } \mathcal{L}_k(x_{k,t})), \quad (3.5)$$

where  $\zeta_{k,t} > 0$  is the stepsize determined later in Theorem 3.1. The RiAL method with RGD (RiAL-RGD) for solving problem (1.1) is formally presented in Algorithm 1.

Here are some important remarks on Algorithm 1. The subproblem (3.2a) is solved inexactly, with the tolerance  $\{\varepsilon_k\}$  in (3.6) gradually converging to 0 to enhance practical efficiency. The specific convergence rate of  $\{\varepsilon_k\}$  is crucial for both theoretical guarantee and practical efficiency of Algorithm 1. Intuitively, a faster decreasing rate of  $\{\varepsilon_k\}$  accelerates the outer convergence rate of RiAL, whereas a larger  $\varepsilon_k$  leads to a reduced complexity of solving the subproblem. To balance this trade-off and achieve a better overall oracle complexity result, we set  $\varepsilon_k$  as in (3.8). Further details on this choice are provided in the following subsection.

### 3.2 Oracle Complexity

In this subsection, we establish the first-order oracle complexity of RiAL-RGD. Before presenting our results, we first introduce the definitions of the first-order oracle and a commonly used notion of  $\varepsilon$ -stationary point of problem (1.1) (see, e.g., [12, 28, 45]).

**Definition 1** For problem (1.1), given  $x \in \mathcal{M}$  and  $y \in \mathcal{E}_2$ , the first-order oracle returns  $f(x)$ ,  $\nabla f(x)$ ,  $\mathcal{A}(x)$ ,  $\nabla \mathcal{A}(x)$ , and  $\text{prox}_{\lambda h}(y)$  for any  $\lambda > 0$ .

**Algorithm 1:** RiAL-RGD for solving problem (1.1)

---

```

1 Input  $x_1 \in \mathcal{M}$ ,  $y_1 = z_1 = \mathbf{0}$ ,  $\varepsilon_1, \sigma_1 > 0$ ,  $b > 1$ .
2 for  $k = 1, 2, \dots$  do
3   Apply the RGD method (3.5) with  $x_{k,1} = x_k$  and stepsize  $\zeta_{k,t}$  for
      $t = 1, 2, \dots$  until
            $\|\text{grad } \mathcal{L}_k(x_{k,t_k})\| \leq \varepsilon_k$  (3.6)
           holds for some positive integer  $t_k$  and set  $x_{k+1} = x_{k,t_k}$ .
4   Update the auxiliary variable via (3.2b).
5   Update the dual variable:
            $z_{k+1} = z_k + \sigma_k(\mathcal{A}(x_{k+1}) - y_{k+1})$ . (3.7)
6   Set
            $\sigma_{k+1} = b\sigma_k$  and  $\varepsilon_{k+1} = \varepsilon_k/b$ . (3.8)

```

---

**Definition 2** For any given  $\varepsilon > 0$ , we say that  $x \in \mathcal{M}$  is an  $\varepsilon$ -stationary point of problem (1.1) if there exist  $y \in \mathcal{E}_2$  and  $z \in \partial h(y)$  such that

$$\max \left\{ \|\text{proj}_{\mathcal{T}_x \mathcal{M}}(\nabla f(x) + \nabla \mathcal{A}(x)^\top z)\|, \|\mathcal{A}(x) - y\| \right\} \leq \varepsilon.$$

We make the following assumptions for our analysis (see, e.g., [7, 8, 45, 49]).

**Assumption 3.1** *The level set*

$$\Omega_{x_1} := \{x \in \mathcal{M} \mid \Phi(x) \leq \Phi(x_1) + \Upsilon\}$$

is compact, where  $\Phi$  is defined in (1.1),  $x_1$  is the initial point of Algorithm 1, and

$$\Upsilon := \sum_{k=1}^{+\infty} \frac{1}{\sigma_k} = \frac{b}{\sigma_1(b-1)}. \quad (3.9)$$

**Assumption 3.2** *The function  $\Phi$  defined in (1.1) is bounded from below on  $\mathcal{M}$ , namely,  $\Phi^* := \inf_{x \in \mathcal{M}} \Phi(x) > -\infty$ .*

**Assumption 3.3** *The functions  $f$  and  $h$  and the smooth mapping  $\mathcal{A}$  satisfy the following conditions:*

(i) *The function  $f : \mathcal{E}_1 \rightarrow \mathbb{R}$  is continuously differentiable and satisfies the descent property over  $\mathcal{M}$  and  $h$  is  $L_h$ -continuous, i.e.,*

$$f(x') \leq f(x) + \langle \nabla f(x), x' - x \rangle + \frac{L_f}{2} \|x' - x\|^2, \quad \forall x, x' \in \mathcal{M}, \quad (3.10)$$

$$\|h(x) - h(x')\| \leq L_h \|x - x'\|, \quad \forall x, x' \in \mathcal{E}_1. \quad (3.11)$$

(ii) *The mapping  $\mathcal{A} : \mathcal{E}_1 \rightarrow \mathcal{E}_2$  and its Jacobian mapping  $\nabla \mathcal{A}$  are  $L_{\mathcal{A}}^0$ -Lipschitz and  $L_{\mathcal{A}}^1$ -Lipschitz continuous over  $\text{conv } \mathcal{M}$ , respectively. In other words, for any  $x, x' \in \text{conv } \mathcal{M}$ , there hold that*

$$\|\mathcal{A}(x) - \mathcal{A}(x')\| \leq L_{\mathcal{A}}^0 \|x - x'\|, \quad (3.12a)$$

$$\|\nabla \mathcal{A}(x) - \nabla \mathcal{A}(x')\| \leq L_{\mathcal{A}}^1 \|x - x'\|. \quad (3.12b)$$

(iii) The Jacobian mapping  $\nabla \mathcal{A}$  is bounded over  $\text{conv } \mathcal{M}$ , i.e.,

$$\rho_{\mathcal{A}} := \max_{x \in \text{conv } \mathcal{M}} \|\nabla \mathcal{A}(x)\| < +\infty. \quad (3.13)$$

The following lemma, which is extracted from [7, Appendix B], shows that the retraction satisfies the first- and second-order boundedness conditions.

**Lemma 3.1** Suppose that Assumption 3.1 holds. Then, there exist constants  $\alpha_1, \alpha_2 > 0$  such that

$$\|\mathbf{R}_x(v) - x\| \leq \alpha_1 \|v\| \quad \text{and} \quad \|\mathbf{R}_x(v) - x - v\| \leq \alpha_2 \|v\|^2$$

for any  $x \in \Omega_{x_1}$  and  $v \in \mathbb{T}_x \mathcal{M}$ .

We next establish the descent property of  $\mathcal{L}_k$  in (3.3) as follows.

**Lemma 3.2** Suppose that Assumptions 3.1 and 3.3 hold. Then, the function  $\mathcal{L}_k$  in (3.3) satisfies the following properties:

(i) **Euclidean Descent.** For any  $x, x' \in \mathcal{M}$ , we have

$$\mathcal{L}_k(x') \leq \mathcal{L}_k(x) + \langle \nabla \mathcal{L}_k(x), x' - x \rangle + \frac{\ell_k}{2} \|x' - x\|^2, \quad (3.14)$$

where  $\ell_k = L_f + L_h L_{\mathcal{A}}^1 + \sigma_k \rho_{\mathcal{A}} L_{\mathcal{A}}^0$ .

(ii) **Riemannian Descent.** For any  $x \in \Omega_{x_1}$  and  $v \in \mathbb{T}_x \mathcal{M}$ , we have

$$\mathcal{L}_k(\mathbf{R}_x(v)) \leq \mathcal{L}_k(x) + \langle \text{grad } \mathcal{L}_k(x), v \rangle + \frac{L_k(x)}{2} \|v\|^2, \quad (3.15)$$

where

$$L_k(x) = \ell_k \alpha_1^2 + 2(\|\nabla f(x)\| + \rho_{\mathcal{A}} L_h) \alpha_2. \quad (3.16)$$

*Proof* The proof is similar to that in [45, Lemma 4.2]. We include the proof here for completeness. For simplicity of notation, denote

$$\begin{aligned} \mathcal{B}_k(x) &:= \sigma_k \mathcal{A}(x) + z_k - \sigma_k \text{prox}_{h/\sigma_k} \left( \mathcal{A}(x) + \frac{z_k}{\sigma_k} \right), \\ \psi_k(x) &:= M_{h/\sigma_k} \left( \mathcal{A}(x) + \frac{z_k}{\sigma_k} \right). \end{aligned} \quad (3.17)$$

From Theorem 2.1, we know that

$$\nabla \psi_k(x) = \nabla \mathcal{A}(x)^\top \mathcal{B}(x) \quad \text{and} \quad \mathcal{B}_k(x) \in \partial h \left( \text{prox}_{h/\sigma_k} \left( \mathcal{A}(x) + \frac{z_k}{\sigma_k} \right) \right).$$

Since  $h$  is  $L_h$ -Lipschitz continuous as assumed in (3.11), from Theorem 2.1, we have

$$\|\mathcal{B}_k(x)\| \leq L_h, \quad \forall x \in \mathcal{M}, k \geq 1. \quad (3.18)$$

We now show that  $\nabla\psi_k$  is Lipschitz continuous. Specifically, for any  $x, x' \in \text{conv } \mathcal{M}$ , it holds that

$$\begin{aligned}
& \|\nabla\psi_k(x) - \nabla\psi_k(x')\| \\
& \leq \|(\nabla\mathcal{A}(x) - \nabla\mathcal{A}(x'))^\top \mathcal{B}_k(x)\| + \|\nabla\mathcal{A}(x')^\top (\mathcal{B}_k(x) - \mathcal{B}_k(x'))\| \\
& \stackrel{(a)}{\leq} L_{\mathcal{A}}^1 \|\mathcal{B}_k(x)\| \cdot \|x - x'\| + \rho_{\mathcal{A}} \|\mathcal{B}_k(x) - \mathcal{B}_k(x')\| \\
& \stackrel{(b)}{\leq} L_{\mathcal{A}}^1 \|\mathcal{B}_k(x)\| \cdot \|x - x'\| + \rho_{\mathcal{A}} \sigma_k \|\mathcal{A}(x) - \mathcal{A}(x')\| \\
& \stackrel{(c)}{\leq} (L_h L_{\mathcal{A}}^1 + \sigma_k \rho_{\mathcal{A}} L_{\mathcal{A}}^0) \|x - x'\|,
\end{aligned}$$

where (a) is due to (3.12b) and (3.13), (b) is due to the firm nonexpansiveness of the proximal operator (see [3, Theorem 6.42]), and (c) is due to (3.12a) and (3.18). Hence, for any  $x, x' \in \text{conv } \mathcal{M}$ , it follows from [36, Lemma 1.2.3] that

$$\psi_k(x') \leq \psi_k(x) + \langle \nabla\psi_k(x), x' - x \rangle + \frac{L_h L_{\mathcal{A}}^1 + \sigma_k \rho_{\mathcal{A}} L_{\mathcal{A}}^0}{2} \|x - x'\|^2.$$

This, together with the definition of  $\mathcal{L}_k$  in (3.3) and the definition of  $\psi_k$  in (3.17), implies the desired Euclidean descent property in (3.14) over  $\mathcal{M}$ .

Moreover, following the similar analysis in [7, Appendix B], we know that  $\mathcal{L}_k$  in (3.3) also satisfies the Riemannian descent property in (3.15).  $\blacksquare$

Next, we present some important inequalities related to  $\mathcal{L}_k$ .

**Lemma 3.3** Suppose that Assumption 3.3 holds. Then, for any  $x \in \mathcal{M}$  and  $k \geq 1$ ,  $\mathcal{L}_k$  defined in (3.3) satisfies

$$\mathcal{L}_k(x) \geq \Phi^* - \frac{2L_h^2}{\sigma_1}, \quad (3.19)$$

$$\mathcal{L}_k(x) \leq \Phi(x) + \frac{L_h^2}{\sigma_k}, \quad (3.20)$$

$$\mathcal{L}_{k+1}(x) \leq \mathcal{L}_k(x) + \frac{2L_h^2}{\sigma_k}. \quad (3.21)$$

*Proof* By the optimality of  $y_{k+1}$  in (3.2b), we have

$$z_k + \sigma_k (\mathcal{A}(x_{k+1}) - y_{k+1}) \in \partial h(y_{k+1}),$$

which, together with (3.7), implies

$$z_{k+1} \in \partial h(y_{k+1}). \quad (3.22)$$

Since  $h$  is  $L_h$ -Lipschitz continuous as assumed in (3.11), by using Theorem 2.1 and noting that  $z_1 = \mathbf{0}$ , we have

$$\|z_k\| \leq L_h, \quad \forall k \geq 1. \quad (3.23)$$

By (3.23), the  $L_h$ -Lipschitz continuity of  $h$ , the definition of  $\Phi$  in (1.1), and [3, Theorem 10.51], for any  $x \in \mathcal{M}$ , we have the following inequalities:

$$\mathcal{L}_k(x) \geq f(x) + h\left(\mathcal{A}(x) + \frac{z_k}{\sigma_k}\right) - \frac{L_h^2}{2\sigma_k} \geq \Phi(x) - \frac{3L_h^2}{2\sigma_k} \quad (3.24)$$

and

$$\mathcal{L}_k(x) \leq f(x) + h\left(\mathcal{A}(x) + \frac{z_k}{\sigma_k}\right) \leq \Phi(x) + \frac{L_h^2}{\sigma_k}. \quad (3.25)$$

With the update of  $\sigma_k$  in (3.8) and the definition of  $\Phi^*$  in Assumption 3.2, we immediately obtain (3.19) and (3.20) from (3.24) and (3.25), respectively.

It remains to prove (3.21). First, we show that, for any  $\lambda_1, \lambda_2 > 0$  and  $w, w' \in \mathcal{E}$ , we have

$$\begin{aligned} & M_{\lambda_1 h}(w) - M_{\lambda_2 h}(w') \\ & \stackrel{(a)}{=} \max_{z \in \mathcal{E}} \left\{ \langle w, z \rangle - h^*(z) - \frac{\lambda_1}{2} \|z\|^2 \right\} - \max_{z \in \mathcal{E}} \left\{ \langle w', z \rangle - h^*(z) - \frac{\lambda_2}{2} \|z\|^2 \right\} \\ & \stackrel{(b)}{\leq} \langle w, z^* \rangle - h^*(z^*) - \frac{\lambda_1}{2} \|z^*\|^2 - \langle w', z^* \rangle + h^*(z^*) + \frac{\lambda_2}{2} \|z^*\|^2 \\ & = \langle w - w', z^* \rangle + \frac{\lambda_2 - \lambda_1}{2} \|z^*\|^2 \stackrel{(c)}{\leq} L_h \|w - w'\| + \frac{\lambda_2 - \lambda_1}{2} L_h^2, \end{aligned} \quad (3.26)$$

where (a) follows from Theorem 2.1, (b) holds by the optimality of the maximization problems and  $z^* := \text{prox}_{h^*/\lambda_1}(w/\lambda_1)$ , and (c) comes from the fact  $\|z^*\| \leq L_h$ , as shown in [3, Theorem 4.23]. Based on the two ends of (3.26), and noting  $\sigma_{k+1} = b\sigma_k$  with  $b > 1$  as in (3.8), we have

$$\begin{aligned} \mathcal{L}_{k+1}(x) - \mathcal{L}_k(x) &= M_{h/\sigma_{k+1}}\left(\mathcal{A}(x) + \frac{z_{k+1}}{\sigma_{k+1}}\right) - M_{h/\sigma_k}\left(\mathcal{A}(x) + \frac{z_k}{\sigma_k}\right) \\ &\leq \frac{1}{2} \left( \frac{1}{\sigma_k} - \frac{1}{\sigma_{k+1}} \right) L_h^2 + L_h \left\| \frac{z_{k+1}}{\sigma_{k+1}} - \frac{z_k}{\sigma_k} \right\| \\ &\leq \frac{1}{2} \left( \frac{1}{\sigma_k} - \frac{1}{\sigma_{k+1}} \right) L_h^2 + \frac{L_h^2}{\sigma_{k+1}} + \frac{L_h^2}{\sigma_k} \leq \frac{2L_h^2}{\sigma_k}, \end{aligned}$$

which is the desired (3.21).  $\blacksquare$

Denote

$$\Delta := \mathcal{L}_1(x_1) + 2L_h^2\Upsilon - \Phi^* + \frac{2L_h^2}{\sigma_1}, \quad (3.27)$$

which serves as an upper bound of  $\mathcal{L}_k(x_k) - \mathcal{L}_k(x_{k+1})$  for any  $k \geq 1$ , as shown in the following lemma. Additionally, define

$$c_1 := \rho_{\mathcal{A}} L_{\mathcal{A}}^0 \alpha_1^2, \quad c_2 := (L_f + L_h L_{\mathcal{A}}^1) \alpha_1^2 + 2 \left( \max_{x \in \Omega_{x_1}} \|\nabla f(x)\| + \rho_{\mathcal{A}} L_h \right) \alpha_2.$$

Recalling the definition of  $L_k(x)$  in (3.16), we derive a universal upper bound for  $L_k(x_{k,t})$  as

$$L_k := \max_{x \in \Omega_{x_1}} L_k(x) = c_1 \sigma_k + c_2. \quad (3.28)$$

With the above results at hand, we start characterizing the inner iteration complexity of Algorithm 1.

**Lemma 3.4** Suppose that Assumptions 3.1, 3.2, and 3.3 hold. Then, the inner RGD of Algorithm 1 with  $\zeta_{k,t} = 1/L_k(x_{k,t})$  stops within at most  $\lceil 2L_k\Delta/\varepsilon_k^2 \rceil$  iterations.

*Proof* First, we show that if  $x_{k,t} \in \Omega_{x_1}$ , then

$$0 \leq \frac{\|\text{grad } \mathcal{L}_k(x_{k,t})\|^2}{2L_k(x_{k,t})} \leq \mathcal{L}_k(x_{k,t}) - \mathcal{L}_k(x_{k,t+1}). \quad (3.29)$$

This follows directly by substituting  $v = \text{grad } \mathcal{L}_k(x_{k,t})/L_k(x_{k,t})$  into (3.15). Based on (3.20) and (3.21) and following the similar argument in [45, Proposition 4.1], we can further show that (3.29) holds for any  $k \geq 1$  and  $t \geq 1$ . Summing both sides of (3.29) over  $t = 1, 2, \dots, T_k$  gives

$$\sum_{t=1}^{T_k} \frac{\|\text{grad } \mathcal{L}_k(x_{k,t})\|^2}{2L_k(x_{k,t})} \leq \mathcal{L}_k(x_{k,1}) - \mathcal{L}_k(x_{k,T_k+1}) \stackrel{(a)}{\leq} \mathcal{L}_k(x_k) - \Phi^* + \frac{2L_h^2}{\sigma_1}, \quad (3.30)$$

where (a) follows from (3.19). From (3.30), we know that the inner RGD method must stop within a finite number of steps. With a slight abuse of notation, we assume that it stops at the  $T_k$ -th iteration. In other words, we have

$$\|\text{grad } \mathcal{L}_k(x_{k,t})\| > \varepsilon_k, \quad t = 1, 2, \dots, T_k - 1, \quad \|\text{grad } \mathcal{L}_k(x_{k,T_k})\| \leq \varepsilon_k.$$

Using  $L_k(x_k) \leq L_k$  as shown in (3.28), we can derive from (3.30) that

$$T_k \leq \left\lceil \frac{2L_k(\mathcal{L}_k(x_k) - \Phi^* + 2L_h^2/\sigma_1)}{\varepsilon_k^2} \right\rceil. \quad (3.31)$$

Next, we prove that  $T_k \leq \lceil 2L_k\Delta/\varepsilon_k^2 \rceil$ . Since the inner RGD method stops at the  $T_k$ -th iteration, we have  $x_{k+1} = x_{k,T_k}$ . Using  $x_k = x_{k,1}$  and (3.29), we find that  $\mathcal{L}_k(x_{k+1}) \leq \mathcal{L}_k(x_k)$ . This, together with (3.21), implies that

$$\mathcal{L}_{k+1}(x_{k+1}) \leq \mathcal{L}_k(x_{k+1}) + \frac{2L_h^2}{\sigma_k} \leq \mathcal{L}_k(x_k) + \frac{2L_h^2}{\sigma_k}, \quad \forall k \geq 1.$$

Therefore, with (3.9), we obtain

$$\mathcal{L}_{k+1}(x_{k+1}) \leq \mathcal{L}_1(x_1) + 2L_h^2 \sum_{i=1}^k \frac{1}{\sigma_i} \leq \mathcal{L}_1(x_1) + 2L_h^2\mathcal{Y}, \quad \forall k \geq 1.$$

Combining this with (3.27) and (3.31) gives the desired result.  $\blacksquare$

Next, we present the outer iteration complexity of Algorithm 1 for finding an  $\varepsilon$ -stationary point of problem (1.1).

**Lemma 3.5** Suppose that Assumption 3.3 holds. Let  $\{x_k\}$  be the sequence generated by Algorithm 1. Then, the point  $x_{K+1}$  is an  $\varepsilon$ -stationary point of problem (1.1) with

$$K := 1 + \left\lceil \log_b \left( \frac{\max\{2L_h\sigma_1^{-1}, \varepsilon_1\}}{\varepsilon} \right) \right\rceil. \quad (3.32)$$

*Proof* By the updates of  $z_{k+1}$  in (3.7) and  $y_{k+1}$  in (3.2b), we have

$$z_{k+1} = z_k + \sigma_k \left( \mathcal{A}(x_{k+1}) - \text{prox}_{h/\sigma_k} \left( \mathcal{A}(x_{k+1}) + \frac{z_k}{\sigma_k} \right) \right). \quad (3.33)$$

It follows from (3.4), (3.33), and the choice of  $\varepsilon_k$  in (3.8) that

$$\begin{aligned} & \left\| \text{proj}_{\Gamma_{x_{k+1}} \mathcal{M}} (\nabla f(x_{k+1}) + \nabla \mathcal{A}(x_{k+1})^\top z_{k+1}) \right\| \\ &= \left\| \text{proj}_{\Gamma_{x_{k+1}} \mathcal{M}} (\nabla \mathcal{L}_k(x_{k+1})) \right\| = \|\text{grad } \mathcal{L}_k(x_{k+1})\| \stackrel{(a)}{\leq} \varepsilon_k \stackrel{(b)}{=} \frac{\varepsilon_1}{b^{k-1}}, \end{aligned} \quad (3.34)$$

where (a) follows from (3.6) and  $x_{k+1} = x_{k,t_k}$  and (b) is due to the update of  $\varepsilon_k$  in (3.8). From (3.23) and the update of  $z_{k+1}$  in (3.7), we have the following bound on  $\|\mathcal{A}(x_{k+1}) - y_{k+1}\|$ :

$$\|\mathcal{A}(x_{k+1}) - y_{k+1}\| = \frac{\|z_{k+1} - z_k\|}{\sigma_k} \leq \frac{\|z_{k+1}\| + \|z_k\|}{\sigma_k} \leq \frac{2L_h}{\sigma_k} = \frac{2L_h}{\sigma_1 b^{k-1}}, \quad (3.35)$$

where the last equality is due to the update of  $\sigma_k$  in (3.8). Combining (3.34), (3.22), and (3.35), and the definition of  $K$  in (3.32), we conclude that  $x_{K+1}$  is an  $\varepsilon$ -stationary point of problem (1.1).  $\blacksquare$

We are now ready to establish the first-order oracle complexity of Algorithm 1 in finding an  $\varepsilon$ -stationary point of problem (1.1).

**Theorem 3.1** Suppose that Assumptions 3.1, 3.2, and 3.3 hold. Then, for any given  $\varepsilon > 0$ , Algorithm 1 with  $\zeta_{k,t} = 1/L_k(x_{k,t})$  can find an  $\varepsilon$ -stationary point of problem (1.1) with at most  $\mathcal{O}(\varepsilon^{-3})$  first-order oracle calls.

*Proof* Combining the inner and outer iteration complexity in Lemmas 3.4 and 3.5, respectively, the total number of RGD updates can be bounded by

$$\begin{aligned} \sum_{k=1}^K \left( \frac{2L_k \Delta}{\varepsilon_k^2} + 1 \right) &= K + 2\Delta \sum_{k=1}^K \frac{c_1 \sigma_k + c_2}{\varepsilon_k^2} \\ &\stackrel{(a)}{=} K + \frac{2\Delta}{\varepsilon_1^2 b^2} \left( \frac{c_1 \sigma_1}{b} \sum_{k=1}^K b^{3k} + c_2 \sum_{k=1}^K b^{2k} \right) \\ &\leq K + \frac{2\Delta}{\varepsilon_1^2 b^2} \left( \frac{c_1 \sigma_1 b^2}{b^3 - 1} b^{3K} + \frac{c_2 b^2}{b^2 - 1} b^{2K} \right) \\ &\stackrel{(b)}{\leq} K + \frac{2\Delta(c_1 \sigma_1 + c_2)}{\varepsilon_1^2 (b^2 - 1)} b^{3K} \\ &\stackrel{(c)}{=} \mathcal{O}(\varepsilon^{-3}), \end{aligned} \quad (3.36)$$

where (a) is due to the definition of  $\varepsilon_k$  in (3.8), (b) uses  $b > 1$ , and (c) follows from (3.32). This completes the proof. ■

Theorem 3.1 shows that the first-order oracle complexity of Algorithm 1 for finding an  $\varepsilon$ -stationary point of problem (1.1) is  $\mathcal{O}(\varepsilon^{-3})$ , which matches the best-known complexity results for problem (1.1) in [4, 12]. Some remarks are listed below. First, it should be emphasized that the boundedness of the dual variable, as stated in (3.23), is crucial for establishing the oracle complexity result. While the dual update in RiAL follows a classical approach, the ManIAL method in [12] computes  $z_{k+1}$  using a damped stepsize as follows:

$$z_{k+1} = z_k + \beta_0 \min \left\{ \frac{\|\mathcal{A}(x_1) - y_1\| \log^2 2}{\|\mathcal{A}(x_{k+1}) - y_{k+1}\| (k+1)^2 \log(k+2)}, 1 \right\} (\mathcal{A}(x_{k+1}) - y_{k+1}). \quad (3.37)$$

Here,  $\beta_0 > 0$  is a preset constant. Our result demonstrates that this damped dual stepsize in (3.37), as well as the additional projection onto a compact set required in the MIAL method [13] in order to bound the dual variable, seems unnecessary. In fact, the derivation of (3.23) demonstrates that the boundedness of the dual variable can be guaranteed by the Lipschitz continuity of  $h$ . Second, our results work for a general nonlinear mapping  $\mathcal{A}$ , whereas the results in [4, 12] apply only to the case where  $\mathcal{A}$  is a linear mapping. Third, our analysis requires that the level set  $\Omega_{x_1}$  is bounded (i.e., Assumption 3.1), which is much weaker than the boundedness assumptions on the manifold  $\mathcal{M}$  made in [12, 13]. Finally, as shown in (3.34), (3.35), and (3.36), the specific choices of  $\{\sigma_k\}$  and  $\{\varepsilon_k\}$  in (3.8) effectively balance the complexity of solving the inner subproblem and the outer convergence rate, which is critical for achieving the desired overall oracle complexity result.

## 4 Numerical Results

In this section, we report the numerical results of RiAL-RGD and ManIAL [12] for solving sparse PCA, sparse CCA, and sparse spectral clustering problems. All tests are implemented in MATLAB 2023b and evaluated on Apple M2 Pro CPU. In all tests, both algorithms are terminated when they return an  $\varepsilon$ -stationary point or hit the maximum outer iteration number 100. The maximum number of inner iterations for RGD when solving each subproblem is set to 5000. We set  $\varepsilon = 10^{-5}$  and  $\varepsilon_1 = \sigma_1 = b = 1.5$ . To enhance the performance of the inner RGD method, we utilize the Riemannian Barzilai-Borwein stepsize [2, 21–23, 43] as the initial stepsize and perform a backtracking line search to find a suitable stepsize. The key difference between RiAL-RGD and ManIAL lies in the stepsize used to update the dual variable; see (3.7) and (3.37). For ManIAL, the damped dual stepsize in (3.37) with  $\beta_0 = 1$  is used.

**Table 2** Average performance comparison on sparse PCA.

$\mu$	ManIAL, [12]					RiAL-RGD				
	$-\Phi$	spar	cpu	outer	total	$-\Phi$	spar	cpu	outer	total
$d = 500, N = 50, r = 10$										
0.5	410.3	31.2	5.9	34	31658	410.4	31.6	<b>2.9</b>	<b>22</b>	<b>18725</b>
0.75	377.0	39.1	7.0	36	37236	377.2	39.3	<b>3.4</b>	<b>24</b>	<b>22286</b>
1	346.3	45.5	13.3	43	81793	346.5	45.7	<b>4.3</b>	<b>24</b>	<b>27344</b>
1.25	318.0	50.8	8.9	37	50867	318.5	51.2	<b>3.5</b>	<b>24</b>	<b>22572</b>
1.5	292.2	55.1	9.2	37	49510	292.3	55.4	<b>3.3</b>	<b>25</b>	<b>21563</b>
$d = 1000, N = 50, \mu = 5$										
4	327.2	41.6	13.2	40	53399	327.5	41.7	<b>3.5</b>	<b>28</b>	<b>20950</b>
5	341.5	56.6	20.1	40	53001	342.2	56.7	<b>3.5</b>	<b>27</b>	<b>14891</b>
6	324.7	58.8	22.2	40	52699	325.0	58.9	<b>5.8</b>	<b>28</b>	<b>23586</b>
7	362.0	67.4	24.3	40	50866	364.2	67.8	<b>5.9</b>	<b>27</b>	<b>20328</b>
8	315.9	70.9	26.2	40	52356	316.2	71.1	<b>5.0</b>	<b>28</b>	<b>17899</b>

#### 4.1 Results on Sparse PCA

Given a data matrix  $A \in \mathbb{R}^{d \times N}$ , where each of the  $N$  columns corresponds to a data sample with  $d$  attributes, the sparse PCA problem can be mathematically formulated as [50]

$$\min_{X \in \mathcal{S}(d,r)} \{ \Phi(X) := -\langle AA^\top, XX^\top \rangle + \mu \|X\|_1 \}. \quad (4.1)$$

Here,  $\mathcal{S}(d,r) = \{X \in \mathbb{R}^{d \times r} \mid X^\top X = I_r\}$  is the Stiefel manifold with  $I_r$  being the  $r$ -by- $r$  identity matrix,  $\mu > 0$  is the weighting parameter, and  $\|X\|_1 = \sum_{i,j} |X_{ij}|$  is the  $\ell_1$ -norm of the matrix  $X$ . In our tests, we randomly generate the data matrix  $A$  as described in [49].

The average results over 20 runs with different randomly generated data matrices and initial points are presented in Table 2. In this table,  $\Phi$  is the objective value of problem (4.1), “cpu” represents the cpu time in seconds, “outer” denotes the number of outer iterations, and “total” denotes the total number of Riemannian gradient descent steps. We also compare the sparsity of  $X$  (denoted by “spar”), which is defined as the percentage of entries with the absolute value less than  $10^{-5}$ . From Table 2, we observe that compared with ManIAL, RiAL-RGD can generally find higher-quality solutions in terms of both the value of  $\Phi$  and the sparsity. Moreover, RiAL is significantly faster than ManIAL, requiring much fewer outer and total iterations. The higher efficiency of RiAL-RGD mainly benefits from its use of the classical full stepsize for the dual update, as opposed to the damped stepsize employed by ManIAL.

**Table 3** Average performance comparison on sparse CCA.

	ManIAL [12]						RiAL-RGD					
	$-\Phi$	sparu	sparv	cpu	outer	total	$-\Phi$	sparu	sparv	cpu	outer	total
$\mu$	$r = 5, d = 1000, p = q = 200$											
0.05	3.011	32.7	33.0	30.7	28	53253	3.013	32.7	33.7	<b>14.7</b>	<b>18</b>	<b>26941</b>
0.07	2.448	42.6	42.2	49.4	36	98864	2.459	44.0	43.8	<b>14.7</b>	<b>18</b>	<b>27098</b>
0.10	1.700	51.1	50.9	72.7	51	169284	1.745	58.9	58.5	<b>14.6</b>	<b>19</b>	<b>27566</b>
0.12	1.313	54.4	54.8	101.2	66	246947	1.363	67.1	67.1	<b>14.0</b>	<b>20</b>	<b>26600</b>
0.15	0.822	66.5	66.7	81.4	59	200863	0.826	78.7	78.4	<b>20.2</b>	<b>21</b>	<b>37009</b>
$r$	$\mu = 0.05, d = 1000, p = q = 200$											
2	1.252	30.7	31.1	15.2	28	44210	1.256	31.0	31.1	<b>3.5</b>	<b>18</b>	<b>10904</b>
3	1.816	31.1	31.2	49.9	40	114272	1.821	31.9	31.8	<b>9.2</b>	<b>18</b>	<b>18601</b>
4	2.410	32.7	32.2	22.2	28	55946	2.418	32.9	32.6	<b>7.2</b>	<b>18</b>	<b>19530</b>
5	3.011	32.7	33.0	31.4	28	53253	3.013	32.7	33.7	<b>15.0</b>	<b>18</b>	<b>26941</b>
6	3.600	32.4	32.9	52.4	36	100133	3.611	33.2	33.6	<b>16.3</b>	<b>17</b>	<b>28998</b>

#### 4.2 Results on Sparse CCA

Given two data matrices  $A \in \mathbb{R}^{d \times p}$  and  $B \in \mathbb{R}^{d \times q}$ , let  $\hat{\Sigma}_{aa} = \frac{1}{d}A^\top A$  and  $\hat{\Sigma}_{bb} = \frac{1}{d}B^\top B$  be the sample covariance matrices of  $X$  and  $Y$ , respectively, and  $\hat{\Sigma}_{ab} = \frac{1}{d}A^\top B$  be the sample cross-covariance matrix. The sparse CCA can be formulated as [10, 12]

$$\min_{U \in \mathcal{S}_{\Sigma_{aa}}(p,r), V \in \mathcal{S}_{\Sigma_{bb}}(p,r)} \left\{ \Phi(U, V) := -\text{tr}(U^\top \hat{\Sigma}_{ab} V) + \mu_1 \|U\|_1 + \mu_2 \|V\|_1 \right\}. \quad (4.2)$$

Here,  $\mathcal{S}_G(p, r) = \{X \in \mathbb{R}^{p \times r} \mid X^\top G X = I_r\}$  with  $G \in \mathbb{R}^{p \times p}$  being a positive definite matrix is the generalized Stiefel manifold and  $\mu_1, \mu_2 > 0$  are the weighting parameters. In our tests, the data matrices are randomly generated as in [12] and we set  $\mu_1 = \mu_2 = \mu$ .

The average results over 20 runs with different randomly generated data matrices and initial points are presented in Table 3, where ‘‘sparu’’ and ‘‘sparv’’ denote the sparsity of  $U$  and  $V$ , respectively. We can observe from Table 3 that RiAL-RGD always return better solutions in terms of both the value of  $\Phi$  in (4.2) and the sparsity of  $U$  and  $V$ . Moreover, RiAL-RGD is much more efficient than ManIAL, requiring less CPU time and fewer outer and total iterations. These results again validate the necessity of using the classical full dual stepsize in Riemannian AL methods.

#### 4.3 Results on Sparse Spectral Clustering

Let  $W = [W_{ij}]_{N \times N}$  be a symmetric affinity matrix of a data matrix  $A = [a_1, a_2, \dots, a_N] \in \mathbb{R}^{d \times N}$ , where  $W_{ij} \geq 0$  measures the pairwise similarity between two samples  $a_i$  and  $a_j$ . The sparse spectral clustering can be formulated

**Table 4** Average performance comparison on sparse spectral clustering.

	ManIAL [12]					RiAL-RGD				
	$\Phi$	spar	cpu	outer	total	$\Phi$	spar	cpu	outer	total
$\mu \times 10^3$	$d = N = 200, m = 10$									
4.2	9.8191	29.1	0.7	16	934	9.8191	29.1	<b>0.6</b>	<b>11</b>	<b>872</b>
4.4	9.8572	40.4	0.8	16	1116	9.8572	40.4	<b>0.7</b>	<b>10</b>	<b>1032</b>
4.5	9.8768	46.6	1.3	16	1994	9.8768	46.6	<b>1.0</b>	<b>11</b>	<b>1520</b>
4.6	9.8938	54.8	1.7	16	2690	9.8938	54.7	<b>1.3</b>	<b>11</b>	<b>2118</b>
4.8	9.9299	63.9	15.6	14	25944	9.9299	64.2	<b>2.6</b>	<b>8</b>	<b>4275</b>
$m$	$d = N = 200, \mu = 0.005$									
2	1.9768	67.5	4.6	16	10891	1.9768	67.5	<b>1.0</b>	<b>11</b>	<b>2171</b>
3	2.9735	73.6	7.6	40	17045	2.9735	73.6	<b>1.8</b>	<b>12</b>	<b>4051</b>
4	3.9727	75.0	13.8	16	32483	4.9725	48.3	<b>4.0</b>	<b>12</b>	<b>9394</b>
5	4.9709	71.6	16.0	16	37964	4.9709	71.9	<b>4.9</b>	<b>12</b>	<b>11588</b>
6	5.9687	69.0	26.4	17	45124	5.9687	68.9	<b>7.6</b>	<b>12</b>	<b>12343</b>

as [34, 35, 42]

$$\min_{X \in \mathcal{S}(N, m)} \{ \Phi(X) := \langle L, XX^\top \rangle + \mu \|XX^\top\|_1 \}, \quad (4.3)$$

where  $L = I_N - S^{-1/2}WS^{-1/2}$  is the normalized Laplacian matrix with  $S^{1/2}$  being the diagonal matrix with diagonal elements  $\sqrt{s_1}, \sqrt{s_2}, \dots, \sqrt{s_N}$  and  $s_i = \sum_j W_{ij}$ , and  $\mu > 0$  is the weighting parameter. Note that problem (4.3) is an instance of problem (1.1) with a *nonlinear* operator  $\mathcal{A}$  and the existing convergence guarantees for ManIAL do not apply to this case. We set  $\beta_0 = 10$  for ManIAL and  $b = 10$  for RiAL-RGD. The initial point  $X_1$  consists of the  $m$  eigenvectors associated to the  $m$  smallest eigenvalues of  $L$ .

Table 4 presents the results on sparse spectral clustering averaged over 20 runs with randomly generated data matrices. From the table, we observe that the solutions returned by RiAL-RGD and ManIAL are comparable in terms of the objective value  $\Phi$  as well as the sparsity of the matrix  $XX^\top$ . However, RiAL-RGD requires fewer outer and total iterations and is more efficient than ManIAL, especially when  $\mu$  is large.

## 5 Concluding Remarks

In this paper, we established the first-order oracle complexity of a Riemannian AL method called RiAL-RGD which utilizes the classical dual update for solving the Riemannian nonsmooth composite problem in (1.1). We proved that RiAL-RGD can find an  $\varepsilon$ -stationary point of the considered problem with at most  $\mathcal{O}(\varepsilon^{-3})$  calls to the first-order oracle, thereby achieving the best-known oracle complexity. Numerical results on sparse PCA, sparse CCA, and sparse spectral clustering validate the superiority of RiAL compared to an existing Riemannian AL method in [12].

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest to this work.

**Data availability** The datasets are generated randomly, with details and citations provided in the corresponding sections.

## References

1. P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton Univ. Press, 2008.
2. J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.
3. A. Beck. *First-Order Methods in Optimization*. SIAM, 2017.
4. A. Beck and I. Rosset. A dynamic smoothing technique for a class of nonsmooth optimization problems on manifolds. *SIAM J. Optim.*, 33(3):1473–1493, 2023.
5. P. B. Borckmans, S. E. Selvan, N. Boumal, and P.-A. Absil. A Riemannian subgradient algorithm for economic dispatch with valve-point effect. *J. Comput. Appl. Math.*, 255:848–866, 2014.
6. N. Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge Univ. Press, 2023.
7. N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.*, 39(1):1–33, 2019.
8. S. Chen, S. Ma, A. M.-C. So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM J. Optim.*, 30(1):210–239, 2020.
9. S. Chen, S. Ma, A. M.-C. So, and T. Zhang. Nonsmooth optimization over the Stiefel manifold and beyond: Proximal gradient method and recent variants. *SIAM Rev.*, 66(2):319–352, 2024.
10. S. Chen, S. Ma, L. Xue, and H. Zou. An alternating manifold proximal gradient method for sparse principal component analysis and sparse canonical correlation analysis. *INFORMS J. Optim.*, 2(3):192–208, 2020.
11. H. Dahal, W. Liu, and Y. Xu. Damped proximal augmented Lagrangian method for weakly-convex problems with convex constraints. *arXiv preprint arXiv:2311.09065*, 2023.
12. K. Deng, J. Hu, J. Wu, and Z. Wen. Oracle complexities of augmented Lagrangian methods for nonsmooth manifold optimization. *arXiv Preprint arXiv:2404.05121*, 2024.
13. K. Deng and Z. Peng. A manifold inexact augmented Lagrangian method for nonsmooth optimization on Riemannian submanifolds in Euclidean space. *IMA J. Numer. Anal.*, 43(3):1653–1684, 2023.
14. D. R. Hardoon and J. Shawe-Taylor. Sparse canonical correlation analysis. *Mach. Learn.*, 83:331–353, 2011.
15. M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4(5):303–320, 1969.
16. S. Hosseini, W. Huang, and R. Yousefpour. Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM J. Optim.*, 28(1):596–619, 2018.
17. S. Hosseini and A. Uschmajew. A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM J. Optim.*, 27(1):173–189, 2017.
18. X. Hu, N. Xiao, X. Liu, and K.-C. Toh. A constraint dissolving approach for nonsmooth optimization over the Stiefel manifold. *IMA J. Numer. Anal.*, 44(6):3717–3748, 2024.
19. W. Huang and K. Wei. Riemannian proximal gradient methods. *Math. Program.*, 194(1):371–413, 2022.
20. W. Huang and K. Wei. An inexact Riemannian proximal gradient method. *Comput. Optim. Appl.*, 85(1):1–32, 2023.
21. B. Iannazzo and M. Porcelli. The Riemannian Barzilai-Borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA J. Numer. Anal.*, 38(1):495–517, 2018.

22. B. Jiang and Y.-H. Dai. A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Math. Program.*, 153(2):535–575, 2015.
23. B. Jiang and Y.-F. Liu. A Riemannian exponential augmented Lagrangian method for computing the projection robust Wasserstein distance. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, pages 79999–80023, 2023.
24. I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *J. Comput. Graph. Stat.*, 12(3):531–547, 2003.
25. W. Kong, J. G. Melo, and R. D. C. Monteiro. Iteration complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical Lagrangian function. *SIAM J. Optim.*, 33(1):181–210, 2023.
26. A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. Comput. Vis. ECCV*, pages 680–696. Springer, 2016.
27. R. Lai and S. Osher. A splitting method for orthogonality constrained problems. *J. Sci. Comput.*, 58:431–449, 2014.
28. J. Li, S. Ma, and T. Srivastava. A Riemannian alternating direction method of multipliers. *Mathematics of Operations Research*, 2024. doi: [10.1287/moor.2023.0068](https://doi.org/10.1287/moor.2023.0068).
29. X. Li, S. Chen, Z. Deng, Q. Qu, Z. Zhu, and A. M.-C. So. Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM J. Optim.*, 31(3):1605–1634, 2021.
30. Z. Li, P.-Y. Chen, S. Liu, S. Lu, and Y. Xu. Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization. In *Proc. Int. Conf. Artif. Intell. Stat.*, pages 2170–2178. PMLR, 2021.
31. X. Liu, N. Xiao, and Y.-X. Yuan. A penalty-free infeasible approach for a class of nonsmooth optimization problems over the Stiefel manifold. *J. Sci. Comput.*, 99(2):30, 2024.
32. Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. M.-C. So, E. A. Jorswieck, and W. Yu. A survey of recent advances in optimization methods for wireless communications. *IEEE J. Sel. Areas Commun.*, 42(11):2992–3031, 2024.
33. Y.-F. Liu, X. Liu, and S. Ma. On the nonergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming. *Math. Oper. Res.*, 44(2):632–650, 2019.
34. C. Lu, J. Feng, Z. Lin, and S. Yan. Nonconvex sparse spectral clustering by alternating direction method of multipliers and its convergence analysis. In *Proc. AAAI Conf. Artif. Intell.*, volume 32, 2018.
35. C. Lu, S. Yan, and Z. Lin. Convex sparse spectral clustering: Single-view to multi-view. *IEEE Trans. Image Process.*, 25(6):2833–2843, 2016.
36. Y. Nesterov. *Lectures on Convex Optimization*, volume 137. Springer, 2018.
37. Z. Peng, W. Wu, J. Hu, and K. Deng. Riemannian smoothing gradient type algorithms for nonsmooth optimization problem on compact Riemannian submanifold embedded in Euclidean space. *Appl. Math. Optim.*, 88(3):85, 2023.
38. M. J. D. Powell. A method for nonlinear constraints in minimization problems. *Optim.*, pages 283–298, 1969.
39. R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317. Springer Sci. Business Media, 2009.
40. M. F. Sahin, A. Alacaoglu, F. Latorre, and V. Cevher. An inexact augmented Lagrangian framework for nonconvex optimization with nonlinear constraints. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 32, pages 13966–13978, 2019.
41. S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala. The price of fair PCA: One extra dimension. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 31, pages 10999–11010, 2018.
42. Z. Wang, B. Liu, S. Chen, S. Ma, L. Xue, and H. Zhao. A manifold proximal linear method for sparse spectral clustering with application to single-cell RNA sequencing data analysis. *INFORMS J. Optim.*, 4(2):200–214, 2022.
43. Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Math. Program.*, 142(1):397–434, 2013.
44. J. Wu, B. Jiang, X. Li, Y.-F. Liu, and J. Yuan. A new adaptive balanced augmented Lagrangian method with application to ISAC beamforming design. *arXiv preprint arXiv:2410.15358*, 2024.

45. M. Xu, B. Jiang, Y.-F. Liu, and A. M.-C. So. A Riemannian alternating descent ascent algorithmic framework for nonconvex-linear minimax problems on Riemannian manifolds. <https://arxiv.org/abs/2409.19588>, 2024.
46. M. Xu, B. Jiang, W. Pu, Y.-F. Liu, and A. M.-C. So. An efficient alternating Riemannian/projected gradient descent ascent algorithm for fair principal component analysis. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 7195–7199, 2024.
47. G. Zalcberg and A. Wiesel. Fair principal component analysis and filter design. *IEEE Trans. Signal Process.*, 69:4835–4842, 2021.
48. C. Zhang, X. Chen, and S. Ma. A Riemannian smoothing steepest descent method for non-Lipschitz optimization on embedded submanifolds of  $\mathbb{R}^n$ . *Math. Oper. Res.*, 49(3):1710–1733, 2023.
49. Y. Zhou, C. Bao, C. Ding, and J. Zhu. A semismooth Newton based augmented Lagrangian method for nonsmooth optimization on matrix manifolds. *Math. Program.*, 201(1):1–61, 2023.
50. H. Zou and L. Xue. A selective overview of sparse principal component analysis. *Proc. IEEE*, 106(8):1311–1320, 2018.