

Submitted to *Mathematics of Operations Research*

A Riemannian Alternating Descent Ascent Algorithmic Framework for Nonconvex-Linear Minimax Problems on Riemannian Manifolds

Meng Xu

ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, xumeng22@mails.ucas.ac.cn

Bo Jiang

Corresponding author. Ministry of Education Key Laboratory of NSLSCS, School of Mathematical Sciences, Nanjing Normal University, jiangbo@njnu.edu.cn

Ya-Feng Liu

Corresponding author. Ministry of Education Key Laboratory of Mathematics and Information Networks, School of Mathematical Sciences, Beijing University of Posts and Telecommunications, yafengliu@bupt.edu.cn

Anthony Man-Cho So

Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, manchoso@se.cuhk.edu.hk

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and are not intended to be a true representation of the article's final published form. Use of this template to distribute papers in print or online or to submit papers to another non-INFORM publication is prohibited.

Abstract. In this paper, we consider a class of nonconvex-linear minimax problems on Riemannian manifolds, which find wide applications in machine learning and signal processing. For solving this class of problems, we develop a flexible Riemannian alternating descent ascent (RADA) algorithmic framework. Within this framework, we propose two easy-to-implement yet efficient algorithms that alternately perform one or multiple projected/Riemannian gradient descent steps and a proximal gradient ascent step at each iteration. We show that the proposed RADA algorithmic framework can find both an ε -Riemannian-game-stationary point and an ε -Riemannian-optimization-stationary point within $\mathcal{O}(\varepsilon^{-3})$ iterations, achieving the best-known iteration complexity. We also reveal intriguing similarities and differences between the algorithms developed within our proposed framework and existing algorithms, thus providing important insights into the improved efficiency of the former. Lastly, we present numerical results on sparse principal component analysis (PCA), fair PCA, and sparse spectral clustering to demonstrate the superior performance of the proposed algorithms.

Funding: Meng Xu and Ya-Feng Liu are supported in part by the National Natural Science Foundation of China (NSFC) under Grant 12371314 and Grant 12021001. Bo Jiang is supported by the National Natural Science Foundation of China (NSFC) under Grant 12522116 and Grant 12371314. Anthony Man-Cho So is supported in part by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) project CUHK 14204823.

Key words: Riemannian alternating descent ascent, iteration complexity, nonconvex-linear minimax problem, Riemannian nonsmooth optimization

MSC codes: Primary: 90C26; secondary: 90C47

OR/MS: Nonlinear programming

1. Introduction In this paper, we consider Riemannian nonconvex-linear (NC-L) minimax problems of the form

$$\min_{x \in \mathcal{M}} \max_{y \in \mathcal{E}_2} \{F(x, y) := f(x) + \langle \mathcal{A}(x), y \rangle - h(y)\}, \quad (1)$$

where \mathcal{M} is a Riemannian manifold embedded in a finite-dimensional Euclidean space \mathcal{E}_1 , \mathcal{E}_2 is another finite-dimensional Euclidean space, $f : \mathcal{E}_1 \rightarrow \mathbb{R}$ is a continuously differentiable function, $\mathcal{A} : \mathcal{E}_1 \rightarrow \mathcal{E}_2$ is a smooth mapping, and $h : \mathcal{E}_2 \rightarrow (-\infty, +\infty]$ is a proper closed convex function with a compact domain and a tractable proximal mapping. Many applications in machine learning and signal processing give rise to instances of problem (1). We briefly introduce three such applications below and refer the interested reader to [11, 23, 27, 40, 42, 50, 73] for more applications.

1.1. Motivating Applications In the following, we use $A = [a_1, a_2, \dots, a_N] \in \mathbb{R}^{d \times N}$ to denote a data matrix, where each of the N columns corresponds to a data sample with d attributes.

1. **Sparse principal component analysis (SPCA).** Each principal component of A obtained through the classic PCA is a linear combination of d attributes, making it difficult to interpret the derived principal components as new features [84]. To achieve a good balance between dimension reduction and interpretability, SPCA seeks principal components with very few nonzero components. This motivates the following formulation of SPCA [33]:

$$\min_{X \in \mathcal{S}(d,r)} \left\{ -\langle AA^\top, XX^\top \rangle + \mu \|X\|_1 \right\}. \quad (2)$$

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 ℓ_1 -norm of the matrix X . By Fenchel duality, the SPCA problem (2) can be equivalently reformulated as

$$\min_{X \in \mathcal{S}(d,r)} \max_{Y \in \mathcal{Y}} \left\{ -\langle AA^\top, XX^\top \rangle + \langle Y, X \rangle \right\}, \quad (3)$$

where $\mathcal{Y} = \{Y \in \mathbb{R}^{d \times r} \mid \|Y\|_\infty \leq \mu\}$ with $\|Y\|_\infty = \max_{i,j} |Y_{ij}|$. Problem (3) is an instance of problem (1) with \mathcal{M} being the Stiefel manifold, $\mathcal{A}(X) = X$, and h being the indicator function of the compact set \mathcal{Y} .

2. **Fair principal component analysis (FPCA).** Suppose that the N data samples belong to m groups according to certain clustering, and each group i corresponds to a $d \times n_i$ submatrix A_i with $\sum_{i=1}^m n_i = N$. Recall that classic PCA aims to find a subspace with dimension $r < d$ to minimize the total reconstruction error, or equivalently, to maximize the total variance. As such, certain groups may suffer a higher reconstruction error than the others [62]. To reduce such disparity among different groups, FPCA minimizes the maximum reconstruction error among the m groups, which is equivalent to maximizing the minimum variance among the m groups. Mathematically, the FPCA problem can be formulated as [62, 64, 74, 79]

$$\min_{X \in \mathcal{S}(d,r)} \max_{i \in \{1,2,\dots,m\}} -\langle A_i A_i^\top, X X^\top \rangle,$$

which is equivalent to

$$\min_{X \in \mathcal{S}(d,r)} \max_{y \in \Delta_m} -\sum_{i=1}^m y_i \langle A_i A_i^\top, X X^\top \rangle. \quad (4)$$

Here, $\Delta_m = \{y \in \mathbb{R}^m \mid \sum_{i=1}^m y_i = 1, y_i \geq 0, i = 1, 2, \dots, m\}$ is the standard simplex in \mathbb{R}^m . Note that problem (4) is an instance of problem (1), where \mathcal{M} is the Stiefel manifold, $\mathcal{A} : \mathbb{R}^{d \times r} \rightarrow \mathbb{R}^m$ is given by $\mathcal{A}(X)_i = -\langle A_i A_i^\top, X X^\top \rangle$ for $i = 1, 2, \dots, m$, and h is the indicator function of Δ_m .

3. Sparse spectral clustering (SSC). This task aims to divide N data samples into m groups, each of which consists of similar data points. Spectral clustering constructs a symmetric affinity matrix $W = [W_{ij}]_{N \times N}$, where $W_{ij} \geq 0$ measures the pairwise similarity between two samples a_i and a_j . To promote the sparsity and interpretability of spectral clustering, the works [51, 52, 70] proposed SSC, which entails solving the following problem:

$$\min_{Q \in \mathcal{G}(N, m)} \{ \langle L, Q \rangle + \mu \|Q\|_1 \}. \quad (5)$$

Here, $\mathcal{G}(N, m) = \{XX^T \in \mathbb{R}^{N \times N} \mid X \in \mathcal{S}(N, m)\}$ is the Grassmann manifold [6], which can be regarded as a submanifold in the Euclidean space [63]; $L = I_N - S^{-1/2}WS^{-1/2}$ 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. By Fenchel duality, problem (5) can be equivalently reformulated as

$$\min_{Q \in \mathcal{G}(N, m)} \max_{Y \in \mathcal{Y}} \{ \langle L, Q \rangle + \langle Y, Q \rangle \}. \quad (6)$$

Again, problem (6) is an instance of problem (1) with \mathcal{M} being the Grassmann manifold, $\mathcal{A}(Q) = Q$, and h being the indicator function of the compact set $\mathcal{Y} := \{Y \in \mathbb{R}^{N \times N} \mid \|Y\|_\infty \leq \mu\}$.

1.2. Related Works Minimax problems in the Euclidean space—i.e., the setting where $\mathcal{M} \subseteq \mathcal{E}_1$ is an arbitrary closed convex set in problem (1)—have been the focus of many recent studies. For general smooth nonconvex-concave (NC-C) minimax problems, many nested-loop algorithms have been developed and analyzed [36, 46, 56, 57, 60, 68, 78]. Nevertheless, single-loop algorithms have gained growing interest due to their simplicity. A classic single-loop algorithm is the gradient descent ascent (GDA) method, which performs a (projected) gradient descent step on x and a (projected) gradient ascent step on y at each iteration. For smooth nonconvex-strongly concave (NC-SC) minimax problems, GDA can find an ε -stationary point within $\mathcal{O}(\varepsilon^{-2})$ iterations [45]. However, when $F(x, \cdot)$ is not strongly concave for some x , GDA may encounter oscillations even for bilinear games. Many improved GDA variants have been proposed to overcome this issue [32, 41, 45, 53, 72, 75–77, 81]. One powerful approach is to design an appropriate regularized objective function. Such an approach was used by Xu et al. [76] and Zhang et al. [81] to develop the alternating gradient projection algorithm and the smoothed GDA algorithm, respectively. Both algorithms can find an ε -stationary point of a general smooth NC-C minimax problem within $\mathcal{O}(\varepsilon^{-4})$ iterations. Later, Li et al. [41] extended the techniques in [81] and developed the smoothed proximal linear descent ascent algorithm. The algorithm enjoys, among other things, an iteration complexity of $\mathcal{O}(\varepsilon^{-4})$ for finding an ε -stationary point of certain nonsmooth NC-C minimax problem. For the special case of NC-L minimax problems, Pan et al. [58], Shen et al. [65], and He et al. [20] used the regularization approach to develop algorithms that can reach ε -stationarity within $\mathcal{O}(\varepsilon^{-3})$ iterations, which is the best complexity known to date.

Recently, there has been growing interest in minimax problems on Riemannian manifolds, but existing works on the topic are few. These works can be divided into two groups based on the nature of the constraints: (i) The constraints on x and y are manifolds [18, 19, 34, 82] and (ii) the constraint on x is a manifold while the constraint on y is a closed convex set [26, 74]. Among the works in the first group, Zhang et al. [82] generalized the classic Sion’s minimax theorem to geodesic metric spaces and proposed a Riemannian corrected extragradient (RCEG) algorithm for solving geodesically convex-geodesically concave (GC-GC) minimax problems. Jordan et al. [34]

analyzed both the last-iterate and average-iterate convergence of RCEG and a Riemannian gradient descent ascent (RGDA) algorithm in various GC-GC settings. For problems without geodesic convexity, Han et al. [19] extended the Hamiltonian gradient method to the Riemannian setting and established its global linear convergence under the assumption that the Riemannian Hamiltonian of F satisfies the Riemannian Polyak-Łojasiewicz (PŁ) condition. In addition, Han et al. [18] proposed several second-order methods that are proven to asymptotically converge to local minimax points of nonconvex-nonconcave minimax problems under strong assumptions. As for the works in the second group, Huang and Gao [26] proposed a RGDA algorithm different from the one in [34] and established its iteration complexity of $\mathcal{O}(\varepsilon^{-2})$ for finding an ε -stationary point of an NC-SC minimax problem. Later, the work [74] presented an alternating Riemannian/projected gradient descent ascent (ARPGDA) algorithm with an iteration complexity of $\mathcal{O}(\varepsilon^{-3})$ for finding an ε -stationary point of the Riemannian NC-L minimax problem (1) when h is the indicator function of a convex compact set.

From the above discussion, we observe that existing works on minimax problems over Riemannian manifolds do not address the settings of problem (1) that we are interested in. Indeed, neither the geodesic convexity assumption in [34, 82] nor the strong concavity assumption in [26] apply to the motivating applications introduced in Section 1.1. Additionally, the Riemannian PŁ condition of the Hamiltonian function in [19] is difficult to verify and is not known to hold for these applications. On the other hand, the second-order methods in [18] cannot be used to tackle problem (1) since the Hessian of $F(x, \cdot)$ is singular. Given the above background, *we are motivated to develop simple yet efficient single-loop first-order methods with strong theoretical guarantees for solving the Riemannian NC-L minimax problem (1).*

One possible approach is based on the following reformulation of (1) as a Riemannian nonsmooth composite problem:

$$\min_{x \in \mathcal{M}} \{f(x) + h^*(\mathcal{A}(x))\}. \quad (7)$$

Here, h^* is the conjugate function of h . Indeed, since h is assumed to have a compact domain, we know from [4, Theorem 4.23] that h^* is Lipschitz continuous. Then, we can apply a host of algorithms to tackle problem (7), including Riemannian subgradient-type methods [7, 21, 22, 25, 42], Riemannian proximal gradient-type methods [11, 12, 27, 28, 49, 70], Riemannian smoothing-type algorithms [5, 59, 80], and splitting-type methods [13, 14, 38–40, 47, 83]. In particular, when \mathcal{A} is the identity map \mathcal{I} and h^* is a real-valued weakly convex function, Li et al. [42] proposed a family of Riemannian subgradient (RSG) methods and showed that all of them have an iteration complexity of $\mathcal{O}(\varepsilon^{-4})$ for getting certain stationarity measure of problem (7) below ε . When $\mathcal{A} = \mathcal{I}$, Chen et al. [11] proposed the manifold proximal gradient (ManPG) method with an outer iteration complexity of $\mathcal{O}(\varepsilon^{-2})$ for finding an ε -stationary point of problem (7). Here, the outer iteration complexity refers to the number of subproblems that need to be solved in the nested-loop algorithms. In particular, the complexity of solving each subproblem is not counted in the outer iteration complexity. When \mathcal{A} is a linear mapping, Beck and Rosset [5] proposed a dynamic smoothing gradient descent method (DSGM) with an iteration complexity of $\mathcal{O}(\varepsilon^{-3})$ for finding an ε -stationary point of problem (7). Using the splitting technique, Li et al. [40] proposed a Riemannian ADMM (RADMM) with an iteration complexity of $\mathcal{O}(\varepsilon^{-4})$ and Deng et al. [13] proposed a manifold inexact augmented Lagrangian (ManIAL) method with an iteration complexity of $\mathcal{O}(\varepsilon^{-3})$ for finding an ε -stationary point of problem (7). When \mathcal{A} is a nonlinear mapping, Wang et al. [70] proposed a manifold proximal linear (ManPL) method and established its outer iteration complexity of $\mathcal{O}(\varepsilon^{-2})$ for finding an ε -stationary point of problem (7).

In this paper, we propose an alternative approach, which is based on exploiting the minimax structure of problem (1) and can also be used to solve problem (7). As seen from Table 1, which

TABLE 1. A summary of methods with iteration complexity results for solving problem (1) or (7), where “ \star ” denotes the outer iteration complexity. The results for HiBSA (developed for general Euclidean NC-C minimax problems) and AGP (developed for Euclidean NC-L minimax problems) are stated for the Euclidean NC-L minimax setting.

Algorithm	Mapping \mathcal{A}	Function h^*	Complexity	Type
RSG [42]	\mathcal{I}	real-valued weakly convex	$\mathcal{O}(\varepsilon^{-4})$	single-loop
ManPG [11]	\mathcal{I}	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-2})^\star$	nested-loop
ManIAL [13]	linear	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-3})$	nested-loop
DSGM [5]	linear	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-3})$	single-loop
RADMM [40]	linear	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-4})$	single-loop
ManPL [70]	nonlinear	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-2})^\star$	nested-loop
ARPGDA [74]	nonlinear	support function of a convex compact set	$\mathcal{O}(\varepsilon^{-3})$	single-loop
HiBSA [53]	nonlinear	support function of a convex compact set	$\mathcal{O}(\varepsilon^{-4})$	single-loop
AGP [58]	nonlinear	support function of a convex compact set	$\mathcal{O}(\varepsilon^{-3})$	single-loop
RADA-PGD/RGD [this paper]	nonlinear	convex, Lipschitz continuous	$\mathcal{O}(\varepsilon^{-3})$	single-loop

summarizes the applicability and complexity of different methods, our approach is able to address more general problem settings and has the best complexity known to date among single-loop algorithms for solving problem (1). Moreover, as we shall see in Section 5, our approach reveals intriguing similarities and differences between our proposed algorithms for problem (1) and existing algorithms for problem (7).

1.3. Our Contributions

We now summarize the main contributions of this paper as follows.

- We propose a flexible Riemannian alternating descent ascent (RADA) algorithmic framework for solving problem (1). At the heart of our framework is the observation that the value function associated with the maximization of certain regularized version of the objective function F with respect to y is smooth and has an easily computable gradient; see (11) and (18). At each iteration, the proposed RADA algorithmic framework finds an approximate minimizer of the said value function to update x and performs one proximal gradient ascent step to update y . The value function contains information about the best response of y , namely, the maximization of the objective function $F(x, y)$ with respect to y for any given x . Therefore, minimizing the proposed value function can be regarded as approximately solving a surrogate of problem (1). It should be noted that our approach to updating x is different from simply finding an (approximate) minimizer of some surrogate of $F(\cdot, y)$ for a fixed y , as is done in, e.g., [20, 53, 58, 76], and is more advantageous in practice. Within the RADA algorithmic framework, we provide two customized simple yet efficient single-loop first-order algorithms, namely, RADA-PGD and RADA-RGD, which perform a fixed number of projected gradient descent steps and Riemannian gradient descent steps to update x at each iteration, respectively. Interestingly, even when adapted to the setting where \mathcal{M} is simply a closed convex set in a Euclidean space, our proposed RADA algorithmic framework, along with the two customized single-loop algorithms, is different from existing methods for solving Euclidean NC-L minimax problems (such as those in [20, 53, 58, 76]) and is, to the best of our knowledge, new.

- We prove that our proposed RADA algorithmic framework can find an ε -Riemannian-game-stationary (ε -RGS) point and an ε -Riemannian-optimization-stationary (ε -ROS) point of problem (1) within $\mathcal{O}(\varepsilon^{-3})$ iterations. This iteration complexity result matches the best-known complexity result for general NC-L minimax problems where the variable x is constrained on a nonempty compact convex set instead of a Riemannian manifold [20, 58]. It also matches the best-known complexity result for the Riemannian nonsmooth composite problem (7) with a linear mapping \mathcal{A}

[5, 13]. Furthermore, while one typically needs extra computation to convert between an ε -game stationary point and an ε -optimization stationary point in the Euclidean setting (see, e.g., [45, 77]), it is worth noting that using our RADA algorithmic framework, the complexity of finding an ε -RGS point is of the same order as that of finding an ε -ROS point.

- We elaborate on the relationship between (i) the RADA algorithmic framework and RADA-RGD in particular, which are designed for problem (1), and (ii) the Riemannian augmented Lagrangian method (RALM) [13, 14, 83] and RADMM [40], which are designed for problem (7). Roughly speaking, the subproblem for updating x in RALM is the same as that in RADA when certain regularization parameter in the aforementioned value function is zero. However, the stopping criterion for the subproblem in RADA is less stringent than that in RALM. Moreover, we show that RADA-RGD with certain parameter setting gives rise to a new Riemannian symmetric Gauss-Seidel ADMM (sGS-ADMM). These similarities and differences provide important insights into understanding the superior performance of our proposed algorithms over existing ones [40, 83].

- We present numerical results on SPCA, FPCA, and SSC to demonstrate the advantages of our proposed algorithms over existing state-of-the-art algorithms for the corresponding problems. Our numerical results on SPCA show that when compared against the nested-loop algorithms ManPG [11] and RALM [83], which employ second-order methods to solve their respective subproblems, our proposed RADA-RGD can return a solution of comparable quality with much lower computational cost. In addition, our numerical results on SSC and FPCA show that our proposed algorithms outperform several state-of-the-art single-loop first-order algorithms [5, 40, 51, 74] in terms of both solution quality and speed.

1.4. Organization The rest of the paper is organized as follows. We first introduce the notation and cover some preliminaries on Riemannian optimization and convex analysis in Section 2. We then present our proposed algorithmic framework and establish its iteration complexity in Section 3. Subsequently, we develop two customized algorithms within the proposed framework in Section 4. We reveal the connections between the proposed algorithms and some existing ones in Section 5. Then, we present numerical results on SPCA, FPCA, and SSC to illustrate the efficiency of the proposed algorithms in Section 6. Finally, we draw some conclusions in Section 7.

2. Notation and Preliminaries We first introduce the notation and some basic concepts in Riemannian optimization [1, 8]. We use $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to denote the standard inner product and its induced norm on the Euclidean space \mathcal{E} , respectively. Let \mathcal{X} be a subset of \mathcal{E} . We use $\text{dist}(y, \mathcal{X}) = \inf_{x \in \mathcal{X}} \|y - x\|$ to denote the distance from $y \in \mathcal{E}$ to \mathcal{X} and $\text{conv } \mathcal{X}$ to denote the convex hull of \mathcal{X} . For a linear mapping $\mathcal{A} : \mathcal{E}_1 \rightarrow \mathcal{E}_2$, we use $\mathcal{A}^\top : \mathcal{E}_2 \rightarrow \mathcal{E}_1$ to denote its adjoint mapping. Let \mathcal{M} be a submanifold embedded in \mathcal{E} and $\text{T}_x \mathcal{M}$ denote the tangent space to \mathcal{M} at $x \in \mathcal{M}$. Throughout this paper, we take the standard inner product $\langle \cdot, \cdot \rangle$ on the Euclidean space \mathcal{E} as the Riemannian metric on \mathcal{M} . Then, for a smooth function $f : \mathcal{E} \rightarrow \mathbb{R}$ and a point $x \in \mathcal{M}$, the Riemannian gradient of f at x is given by $\text{grad } f(x) = \text{proj}_{\text{T}_x \mathcal{M}}(\nabla f(x))$, where $\text{proj}_{\mathcal{X}}(\cdot)$ is the Euclidean projection operator onto a closed set \mathcal{X} and $\nabla f(x)$ is the Euclidean gradient of f at x . A retraction at $x \in \mathcal{M}$ is a smooth mapping $R_x : \text{T}_x \mathcal{M} \rightarrow \mathcal{M}$ satisfying (i) $R_x(\mathbf{0}_x) = x$, where $\mathbf{0}_x$ is the zero element in $\text{T}_x \mathcal{M}$; (ii) $\frac{d}{dt} R_x(tv)|_{t=0} = v$ for all $v \in \text{T}_x \mathcal{M}$.

Now, we review some basic notions in convex analysis [4, 61]. Let $g : \mathcal{E} \rightarrow (-\infty, +\infty]$ be a proper closed convex function whose domain is given by $\text{dom } g := \{x \in \mathcal{E} \mid g(x) < +\infty\}$. Its

conjugate function is defined as $g^*(y) := \max_{x \in \mathcal{E}} \{\langle y, x \rangle - g(x)\}$. The function g is called λ -strongly convex if $g(\cdot) - \frac{\lambda}{2} \|\cdot\|^2$ is convex. For a given constant $\lambda > 0$, the proximal mapping and the Moreau envelope of g are defined as

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

and

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

respectively. For any $x \in \mathcal{E}$, we have the following so-called Moreau decomposition and Moreau envelope decomposition:

$$x = \text{prox}_{\lambda g}(x) + \lambda \text{prox}_{g^*/\lambda} \left(\frac{x}{\lambda} \right), \quad (8)$$

$$\frac{1}{2\lambda} \|x\|^2 = M_{\lambda g}(x) + M_{g^*/\lambda} \left(\frac{x}{\lambda} \right). \quad (9)$$

The following theorem characterizes the smoothness of the Moreau envelope.

THEOREM 1. ([4, Theorems 6.39 and 6.60]) *Let $g : \mathcal{E} \rightarrow (-\infty, +\infty]$ be a proper closed convex function and $\lambda > 0$. Then, for any $x \in \mathcal{E}$,*

$$\nabla M_{\lambda g}(x) = \frac{1}{\lambda} (x - \text{prox}_{\lambda g}(x)) = \text{prox}_{g^*/\lambda} \left(\frac{x}{\lambda} \right) \in \partial g(\text{prox}_{\lambda g}(x)). \quad (10)$$

Moreover, $\nabla M_{\lambda g}$ is λ^{-1} -Lipschitz continuous.

3. Proposed Algorithmic Framework In this section, we propose a flexible algorithmic framework for solving the Riemannian NC-L minimax problem (1) and establish its iteration complexity.

3.1. Proposed RADA Algorithmic Framework A widely used idea for tackling the minimax problem (1) is to alternately update the variables x and y by solving an appropriate minimization problem for x and an appropriate maximization problem for y . However, even a small perturbation in x may cause a drastic change in the solution to the problem $\max_{y \in \mathcal{E}_2} F(x, y)$ [67]. Inspired in part by [20, 53, 58, 76], we address this issue by adding quadratic regularization terms to the objective function and defining the following value function at the k -th iteration:

$$\Phi_k(x) := \max_{y \in \mathcal{E}_2} \left\{ F(x, y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 \right\}. \quad (11)$$

Here, $\lambda > 0$ is the regularization parameter, $\beta_k \geq 0$ is the proximal parameter, and $y_k \in \text{dom } h$ is the proximal center. Intuitively, the quadratic regularization term $\frac{\lambda}{2} \|y\|^2$ ensures the smoothness of Φ_k (as shown later in (18)), while the proximal term $\frac{\beta_k}{2} \|y - y_k\|^2$ stabilizes the update of y .

At the k -th iteration, our algorithmic framework computes the update x_{k+1} by approximately solving the subproblem

$$\min_{x \in \mathcal{M}} \Phi_k(x). \quad (12)$$

Specifically, we need x_{k+1} to satisfy the sufficient decrease condition

$$\Phi_k(x_{k+1}) - \Phi_k(x_k) \leq -\min \{c_\lambda \|\text{grad } \Phi_k(x_k)\|, c'\} \|\text{grad } \Phi_k(x_k)\| + \nu_k, \quad (13)$$

Algorithm 1: RADA algorithmic framework for problem (1)

- 1 Input $x_1 \in \mathcal{M}$, $y_1 \in \text{dom } h$, $\lambda > 0$, $\beta_1 \geq 0$, $\rho > 1$, $c_\lambda > 0$, and $\{\nu_k\} \in \mathbb{S}$.
 - 2 **for** $k = 1, 2, \dots$ **do**
 - 3 Approximately solve subproblem (12) to find a point $x_{k+1} \in \mathcal{M}$ that satisfies the sufficient decrease condition (13).
 - 4 Calculate y_{k+1} via (14).
 - 5 Update β_{k+1} such that $0 \leq \beta_{k+1} \leq \beta_1/(k+1)^\rho$.
-

where $c_\lambda > 0$ is a predetermined constant on the order of λ (i.e., $c_\lambda = \mathcal{O}(\lambda)$), $c' > 0$ is another predetermined constant, and

$$\{\nu_k\} \in \mathbb{S} := \left\{ \{\nu_k\} \left| \sum_{k=1}^{+\infty} \nu_k < +\infty, \nu_k \geq 0, k = 1, 2, \dots \right. \right\}$$

is a nonnegative summable sequence. Then, we compute the update

$$\begin{aligned} y_{k+1} &= \arg \max_{y \in \mathcal{E}_2} \left\{ F(x_{k+1}, y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 \right\} \\ &= \text{prox}_{h/(\lambda+\beta_k)} \left(\frac{\mathcal{A}(x_{k+1}) + \beta_k y_k}{\lambda + \beta_k} \right). \end{aligned} \quad (14)$$

The above update of y can also be regarded as performing a proximal gradient ascent step of the regularized function $F(x_{k+1}, \cdot) - \frac{\lambda}{2} \|\cdot\|^2$. The proposed RADA algorithmic framework is formally presented in Algorithm 1.

Several remarks on the proposed Algorithm 1 are in order. First, the requirement on the update β_{k+1} is quite flexible. A simple approach is to set $\beta_k = \beta_1/k^\rho$ for all $k \geq 1$. Alternatively, we can adaptively update β_k for improved practical efficiency. For instance, given constants $\tau_1 \in (0, 1)$ and $\tau_2 \in (0, 1)$, we may set

$$\beta_{k+1} = \frac{\beta_1^{(k+1)}}{(k+1)^\rho} \quad \text{with} \quad \beta_1^{(k+1)} = \begin{cases} \tau_2 \beta_1^{(k)}, & \text{if } \delta_{k+1} \geq \tau_1 \delta_k; \\ \beta_1^{(k)}, & \text{if } \delta_{k+1} < \tau_1 \delta_k, \end{cases} \quad (15)$$

where $\beta_1^{(1)} = \beta_1$ and δ_{k+1} is defined as

$$\delta_{k+1} := \|(\lambda + \beta_k)y_{k+1} - \beta_k y_k\|_\infty, \quad \forall k \geq 0. \quad (16)$$

Here, we adopt the convention that $\beta_0 = \beta_1$ and $y_0 = y_1$ when computing δ_1 . It is simple to see that the update β_{k+1} in (15) satisfies $0 \leq \beta_{k+1} \leq \beta_1/(k+1)^\rho$ for all $k \geq 1$. As will be discussed at the end of Section 5.1, our proposed Algorithm 1 is closely related to RALM. In particular, the parameter δ_{k+1} in (16) can be viewed as a primal residual in RALM.

Second, the sufficient decrease condition (13) is easy to satisfy, making the proposed algorithmic framework computationally efficient. For instance, we can find the desired $x_{k+1} \in \mathcal{M}$ by performing a simple Riemannian gradient descent step on x with a suitable stepsize. To do this, we need to establish the smoothness of Φ_k . By the fact that

$$\langle \mathcal{A}(x), y \rangle - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 + \frac{\beta_k}{2} \|y_k\|^2$$

$$= \frac{1}{2(\lambda + \beta_k)} \|\mathcal{A}(x) + \beta_k y_k\|^2 - \frac{\lambda + \beta_k}{2} \left\| y - \frac{\mathcal{A}(x) + \beta_k y_k}{\lambda + \beta_k} \right\|^2,$$

we have from (11) and (9) that

$$\begin{aligned} \Phi_k(x) &= f(x) + \max_{y \in \mathcal{E}_2} \left\{ \langle \mathcal{A}(x), y \rangle - h(y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 \right\} \\ &= f(x) + \frac{1}{2(\lambda + \beta_k)} \|\mathcal{A}(x) + \beta_k y_k\|^2 - M_{h/(\lambda + \beta_k)} \left(\frac{\mathcal{A}(x) + \beta_k y_k}{\lambda + \beta_k} \right) - \frac{\beta_k}{2} \|y_k\|^2 \\ &= f(x) + M_{(\lambda + \beta_k)h^*}(\mathcal{A}(x) + \beta_k y_k) - \frac{\beta_k}{2} \|y_k\|^2. \end{aligned} \quad (17)$$

Therefore, by Theorem 1, the value function Φ_k is differentiable and

$$\nabla \Phi_k(x) = \nabla f(x) + \nabla \mathcal{A}(x)^\top \text{prox}_{h/(\lambda + \beta_k)} \left(\frac{\mathcal{A}(x) + \beta_k y_k}{\lambda + \beta_k} \right), \quad (18)$$

where $\nabla \mathcal{A}$ denotes the Jacobian mapping of \mathcal{A} . More approaches for approximately solving the subproblem (12) are presented in Section 4. The choice of the approach mainly depends on the structure of the underlying problem. The update y_{k+1} in (14) is also simple since h is assumed to have a tractable proximal mapping. Therefore, our proposed algorithmic framework enjoys a low per-iteration computational cost.

Finally, we highlight the difference between our proposed RADA algorithmic framework and some related algorithms in the literature. The works [20, 53, 58, 76] for solving Euclidean NC-L minimax problems minimize certain surrogate of $F(\cdot, y)$ with a fixed y to update x , i.e.,

$$x_{k+1} \approx \arg \min_{x \in \mathcal{M}} \left\{ F(x, y_k) - \frac{\lambda}{2} \|y_k\|^2 \right\},$$

where \mathcal{M} is a compact convex set. By contrast, the proposed value function (11) at x is the maximization of a regularized $F(x, \cdot)$, and the subproblem (12) used to update x_{k+1} is itself a minimax problem, i.e.,

$$x_{k+1} \approx \arg \min_{x \in \mathcal{M}} \left\{ \max_{y \in \mathcal{E}_2} \left\{ F(x, y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 \right\} \right\},$$

which can be viewed as a surrogate of problem (1). Thus, our proposed RADA is fundamentally different from the approaches in [20, 53, 58, 76]. To provide more insights into why the former could outperform the latter, we elaborate on the difference between RADA and the ARPGDA algorithm developed in [74], which can be regarded as a Riemannian counterpart of the methods in [20, 53, 58, 76]. ARPGDA computes the update

$$x_{k+1} = R_{x_k}(-\zeta_k \text{grad}_x F(x_k, y_k)),$$

where $\text{grad}_x F(x, y)$ denotes the Riemannian gradient of $F(x, y)$ with respect to x . Then, it computes the update y_{k+1} according to (14). Similar to the derivation of $\nabla \Phi_k(x)$, we have $\nabla_x F(x_k, y_k) = \nabla \Phi_{k-1}(x_k)$ and hence

$$\text{grad}_x F(x_k, y_k) = \text{grad} \Phi_{k-1}(x_k).$$

Therefore, ARPGDA uses the Riemannian gradient of the value function Φ_{k-1} at x_k to compute the update x_{k+1} . By contrast, our proposed framework uses the Riemannian gradient of the value function Φ_k at x_k to compute the update x_{k+1} . In particular, our proposed framework makes use of the most recent information about y (since Φ_k involves the latest iterate y_k) to compute the update x_{k+1} . This suggests that RADA could outperform ARPGDA.

3.2. Iteration Complexity In this subsection, we study the convergence behavior of the proposed RADA algorithmic framework. Towards that end, let us first introduce the stationarity measures that we are interested in. The following definition is motivated by the stationarity measures used in Euclidean minimax problems [41, 76] and (Riemannian) nonsmooth composite problems [5, 40, 69].

DEFINITION 1. Let $\varepsilon > 0$ be some given constant.

(a) The point $(x, y) \in \mathcal{M} \times \text{dom } h$ is an ε -Riemannian-game-stationary (ε -RGS) point of problem (1) if there exists a constant $\gamma > 0$ such that

$$\max \left\{ \|\text{grad}_x F(x, y)\|, \frac{1}{\gamma} \|y - \text{prox}_{\gamma h}(y + \gamma \mathcal{A}(x))\| \right\} \leq \varepsilon.$$

(b) The point $x \in \mathcal{M}$ is an ε -Riemannian-optimization-stationary (ε -ROS) point of problem (1) if there exists a point $p \in \mathcal{E}_2$ such that

$$\max \left\{ \text{dist} \left(0, \text{grad } f(x) + \text{proj}_{T_x \mathcal{M}} \left(\nabla \mathcal{A}(x)^\top \partial h^*(p) \right) \right), \|p - \mathcal{A}(x)\| \right\} \leq \varepsilon,$$

where ∂h^* is the subdifferential of h^* .

For simplicity, denote

$$F_k(x, y) := F(x, y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2, \quad (19)$$

$$y_{k+\frac{1}{2}} := \arg \max_{y \in \mathcal{E}_2} F_k(x_k, y) = \text{prox}_{h/(\lambda+\beta_k)} \left(\frac{\mathcal{A}(x_k) + \beta_k y_k}{\lambda + \beta_k} \right), \quad (20)$$

$$\Phi(x) := \max_{y \in \mathcal{E}_2} F(x, y), \quad (21)$$

$$R := \max_{y \in \text{dom } h} \|y\| < +\infty. \quad (22)$$

Recalling that $\rho > 1$, $0 \leq \beta_k \leq \beta_1/k^\rho$, and $\{\nu_k\} \in \mathbb{S}$ in Algorithm 1, we have

$$\Upsilon := \frac{\lambda R^2}{2} + \sum_{k=1}^{+\infty} (\nu_k + 2\beta_k R^2) < +\infty. \quad (23)$$

We make the following blanket assumption.

ASSUMPTION 1. The function Φ defined in (21) is bounded from below on \mathcal{M} , namely, $\Phi^* := \inf_{x \in \mathcal{M}} \Phi(x) > -\infty$.

Now, we are ready to establish the iteration complexity of the proposed RADA algorithmic framework for returning an ε -RGS point and an ε -ROS point of problem (1). We start with the following estimate:

LEMMA 1. Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then, we have

$$\Phi_{k+1}(x_{k+1}) - \Phi_k(x_k) \leq -\min \{c_\lambda \|\text{grad } \Phi_k(x_k)\|, c'\} \|\text{grad } \Phi_k(x_k)\| + \nu_k + 2\beta_k R^2, \quad (24)$$

where the constant c_λ is given after (13).

Proof By the definition of Φ_k in (11) and the update formula for y_{k+1} in (14), we have $\Phi_k(x_{k+1}) = F_k(x_{k+1}, y_{k+1})$. Similarly, by the definition of $y_{k+\frac{1}{2}}$ in (20), we have $\Phi_{k+1}(x_{k+1}) = F_{k+1}(x_{k+1}, y_{k+\frac{3}{2}})$, where we use $y_{k+\frac{3}{2}}$ as a shorthand for $y_{(k+1)+\frac{1}{2}}$. We now compute

$$\begin{aligned} \Phi_{k+1}(x_{k+1}) - \Phi_k(x_{k+1}) &= F_{k+1}(x_{k+1}, y_{k+\frac{3}{2}}) - F_k(x_{k+1}, y_{k+1}) \\ &\stackrel{(a)}{\leq} F_{k+1}(x_{k+1}, y_{k+\frac{3}{2}}) - F_k(x_{k+1}, y_{k+\frac{3}{2}}) \\ &\stackrel{(b)}{=} F(x_{k+1}, y_{k+\frac{3}{2}}) - \frac{\lambda}{2} \|y_{k+\frac{3}{2}}\|^2 - \frac{\beta_{k+1}}{2} \|y_{k+\frac{3}{2}} - y_{k+1}\|^2 \\ &\quad - F(x_{k+1}, y_{k+\frac{3}{2}}) + \frac{\lambda}{2} \|y_{k+\frac{3}{2}}\|^2 + \frac{\beta_k}{2} \|y_{k+\frac{3}{2}} - y_k\|^2 \\ &= \frac{\beta_k}{2} \|y_{k+\frac{3}{2}} - y_k\|^2 - \frac{\beta_{k+1}}{2} \|y_{k+\frac{3}{2}} - y_{k+1}\|^2 \stackrel{(c)}{\leq} 2\beta_k R^2, \end{aligned} \quad (25)$$

where (a) comes from the optimality of y_{k+1} in (14), (b) holds by the definition of F_{k+1} in (19), and (c) is due to (22) and the fact that $y_k, y_{k+\frac{3}{2}} \in \text{dom } h$. By combining (25) and (13), we obtain (24). ■

Based on Lemma 1, we can establish the iteration complexity of Algorithm 1 for returning an ε -RGS point of problem (1).

THEOREM 2. *Given a constant $\varepsilon > 0$, let $\{(x_k, y_k)\}$ be the sequence generated by Algorithm 1 with $\lambda = \varepsilon/(2R)$. Suppose that Assumption 1 holds. Then, there exists a $k \leq K$ with*

$$K := \left\lceil \left(\frac{4\beta_1 R}{\varepsilon} \right)^{\frac{1}{\rho}} \right\rceil + \left\lceil \frac{\Phi_1(x_1) - \Phi_* + \Upsilon}{\min\{c_\lambda \varepsilon, c'\} \varepsilon} \right\rceil \quad (26)$$

such that $(x_k, y_{k+\frac{1}{2}})$ is an ε -RGS point of problem (1).

Proof Given any $\gamma > 0$, we first show that $y_{k+\frac{1}{2}}$ satisfies

$$\frac{1}{\gamma} \left\| y_{k+\frac{1}{2}} - \text{prox}_{\gamma h}(y_{k+\frac{1}{2}} + \gamma \mathcal{A}(x_k)) \right\| \leq \varepsilon, \quad \forall k \geq K_1 := \left\lceil \left(\frac{4\beta_1 R}{\varepsilon} \right)^{\frac{1}{\rho}} \right\rceil. \quad (27)$$

For brevity, denote $y'_{k+\frac{1}{2}} = \text{prox}_{\gamma h}(y_{k+\frac{1}{2}} + \gamma \mathcal{A}(x_k))$. It is easy to see that

$$y'_{k+\frac{1}{2}} = \arg \max_{y \in \mathcal{E}_2} \left\{ F(x_k, y) - \frac{1}{2\gamma} \|y - y_{k+\frac{1}{2}}\|^2 \right\}. \quad (28)$$

Due to the optimality of $y_{k+\frac{1}{2}}$ in (20), we have

$$0 \leq F_k(x_k, y_{k+\frac{1}{2}}) - F_k(x_k, y'_{k+\frac{1}{2}}). \quad (29)$$

Since $F(x_k, \cdot) - \frac{1}{2\gamma} \|\cdot - y_{k+\frac{1}{2}}\|^2$ is γ^{-1} -strongly concave, it follows from the optimality of $y'_{k+\frac{1}{2}}$ in (28) that

$$\frac{1}{2\gamma} \|y_{k+\frac{1}{2}} - y'_{k+\frac{1}{2}}\|^2 \leq F(x_k, y'_{k+\frac{1}{2}}) - \frac{1}{2\gamma} \|y'_{k+\frac{1}{2}} - y_{k+\frac{1}{2}}\|^2 - F(x_k, y_{k+\frac{1}{2}}).$$

This, together with (29), further implies that

$$\begin{aligned}
\frac{1}{\gamma} \|y_{k+\frac{1}{2}} - y'_{k+\frac{1}{2}}\|^2 &\leq \left(F_k(x_k, y_{k+\frac{1}{2}}) - F(x_k, y_{k+\frac{1}{2}}) \right) - \left(F_k(x_k, y'_{k+\frac{1}{2}}) - F(x_k, y'_{k+\frac{1}{2}}) \right) \\
&\stackrel{(a)}{=} \frac{\lambda}{2} \left(\|y'_{k+\frac{1}{2}}\|^2 - \|y_{k+\frac{1}{2}}\|^2 \right) + \frac{\beta_k}{2} \left(\|y'_{k+\frac{1}{2}} - y_k\|^2 - \|y_{k+\frac{1}{2}} - y_k\|^2 \right) \\
&= \frac{\lambda}{2} \left\langle y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}}, y'_{k+\frac{1}{2}} - y_{k+\frac{1}{2}} \right\rangle + \frac{\beta_k}{2} \left\langle y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}} - 2y_k, y'_{k+\frac{1}{2}} - y_{k+\frac{1}{2}} \right\rangle \\
&\leq \left(\frac{\lambda}{2} \|y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}}\| + \frac{\beta_k}{2} \|y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}} - 2y_k\| \right) \|y_{k+\frac{1}{2}} - y'_{k+\frac{1}{2}}\|,
\end{aligned}$$

where (a) uses the definition of F_k in (19). Cancelling the term $\|y_{k+\frac{1}{2}} - y'_{k+\frac{1}{2}}\|$ on the two ends of the above inequalities yields

$$\frac{1}{\gamma} \|y_{k+\frac{1}{2}} - y'_{k+\frac{1}{2}}\| \leq \frac{\lambda}{2} \|y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}}\| + \frac{\beta_k}{2} \|y'_{k+\frac{1}{2}} + y_{k+\frac{1}{2}} - 2y_k\| \stackrel{(a)}{\leq} \lambda R + 2\beta_k R \stackrel{(b)}{\leq} \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,$$

where (a) is due to (22) and the fact that $y_k, y_{k+\frac{1}{2}}, y'_{k+\frac{1}{2}} \in \text{dom } h$, and (b) uses $\lambda = \varepsilon/(2R)$, $0 \leq \beta_k \leq \beta_1/k^\rho$, and $k \geq K_1 = \lceil (4\beta_1 R/\varepsilon)^{\frac{1}{\rho}} \rceil$.

Next, we prove the existence of some $\hat{k} \in \{K_1, K_1 + 1, \dots, K\}$ such that $x_{\hat{k}}$ satisfies

$$\|\text{grad}_x F(x_{\hat{k}}, y_{\hat{k}+\frac{1}{2}})\| \leq \varepsilon. \tag{30}$$

Note that for any $x \in \mathcal{M}$, we deduce from (11) and the expression of $F(x, y)$ in (1) that

$$\begin{aligned}
\Phi_k(x) &= \max_{y \in \mathcal{E}_2} \left\{ f(x) + \langle \mathcal{A}(x), y \rangle - h(y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_k}{2} \|y - y_k\|^2 \right\} \\
&\geq \max_{y \in \mathcal{E}_2} \left\{ f(x) + \langle \mathcal{A}(x), y \rangle - h(y) - \frac{\lambda}{2} R^2 - 2\beta_k R^2 \right\} \\
&= \Phi(x) - \frac{\lambda}{2} R^2 - 2\beta_k R^2,
\end{aligned} \tag{31}$$

where the inequality is due to (22) and the last equality follows from the definition of Φ in (21). By (24), we know that

$$\min \{c_\lambda \|\text{grad } \Phi_k(x_k)\|, c'\} \|\text{grad } \Phi_k(x_k)\| \leq \Phi_k(x_k) - \Phi_{k+1}(x_{k+1}) + \nu_k + 2\beta_k R^2.$$

Summing the above inequality over $k = K_1, K_1 + 1, \dots, K$ yields

$$\begin{aligned}
&\sum_{k=K_1}^K \min \{c_\lambda \|\text{grad } \Phi_k(x_k)\|, c'\} \|\text{grad } \Phi_k(x_k)\| \\
&\leq \Phi_1(x_1) - \Phi_{K+1}(x_{K+1}) + \sum_{k=1}^K (\nu_k + 2\beta_k R^2) \leq \Phi_1(x_1) - \Phi^* + \Upsilon,
\end{aligned}$$

where the last inequality is due to Assumption 1, (23), and (31). Hence, there exists some $\hat{k} \in \{K_1, K_1 + 1, \dots, K\}$ such that

$$\min \{c_\lambda \|\text{grad } \Phi_{\hat{k}}(x_{\hat{k}})\|, c'\} \|\text{grad } \Phi_{\hat{k}}(x_{\hat{k}})\| \leq \frac{\Phi_1(x_1) - \Phi^* + \Upsilon}{K - K_1}. \tag{32}$$

Substituting (26) into (32), we have

$$\|\text{grad } \Phi_{\hat{k}}(x_{\hat{k}})\| \leq \varepsilon. \quad (33)$$

The definition of $y_{k+\frac{1}{2}}$ in (20), together with (18), implies that $\nabla \Phi_k(x_k) = \nabla_x F(x_k, y_{k+\frac{1}{2}})$ and thus

$$\text{grad } \Phi_k(x_k) = \text{grad}_x F(x_k, y_{k+\frac{1}{2}}). \quad (34)$$

Therefore, we see from (33) and (34) that (30) holds. Together with (27), we conclude that $(x_{\hat{k}}, y_{\hat{k}+\frac{1}{2}})$ is an ε -RGS point of problem (1). \blacksquare

According to Theorem 2, since $\lambda = \varepsilon/(2R)$ and $c_\lambda = \mathcal{O}(\lambda)$ by our choice, we see that the iteration complexity of Algorithm 1 for obtaining an ε -RGS point of problem (1) is $\mathcal{O}(\varepsilon^{-3})$. It is worth mentioning that our iteration complexity result in Theorem 2 for the RADA algorithmic framework matches not only that for the less general ARPGDA in [74], but also matches the best-known complexity results for algorithms that tackle NC-L minimax problems with x being constrained on a nonempty compact convex set instead of a Riemannian manifold [20, 58, 65]. Furthermore, our iteration complexity analysis is based on the sufficient decrease condition (13) of the proposed value function Φ_k . By contrast, the iteration complexity analyses in [58, 74, 76] rely on the descent properties of the function $F(\cdot, y_k)$ and a different potential function, which do not extend to RADA.

Next, we establish the iteration complexity of Algorithm 1 for returning an ε -ROS point of problem (1).

THEOREM 3. *Consider the setting of Theorem 2. Then, $x_{\hat{k}}$ is an ε -ROS point of problem (1), where \hat{k} is the index given in Theorem 2.*

Proof The proof is similar to that of Theorem 2. From (33), we know there exists some $\hat{k} \in \{K_1, K_1 + 1, \dots, K\}$ such that

$$\|\text{grad } \Phi_{\hat{k}}(x_{\hat{k}})\| \leq \varepsilon. \quad (35)$$

Next, define $p_k = \text{prox}_{(\lambda+\beta_k)h^*}(\mathcal{A}(x_k) + \beta_k y_k)$. From (10), (18), and (20), we have

$$\text{grad } \Phi_k(x_k) = \text{grad } f(x_k) + \text{proj}_{T_{x_k} \mathcal{M}} \left(\nabla \mathcal{A}(x_k)^\top y_{k+\frac{1}{2}} \right) \quad \text{and} \quad y_{k+\frac{1}{2}} \in \partial h^*(p_k). \quad (36)$$

Then, it follows from (35) and (36) that

$$\begin{aligned} & \text{dist} \left(0, \text{grad } f(x_{\hat{k}}) + \text{proj}_{T_{x_{\hat{k}}} \mathcal{M}} \left(\nabla \mathcal{A}(x_{\hat{k}})^\top \partial h^*(p_{\hat{k}}) \right) \right) \\ & \leq \left\| \text{grad } f(x_{\hat{k}}) + \text{proj}_{T_{x_{\hat{k}}} \mathcal{M}} \left(\nabla \mathcal{A}(x_{\hat{k}})^\top y_{\hat{k}+\frac{1}{2}} \right) \right\| = \|\text{grad } \Phi_{\hat{k}}(x_{\hat{k}})\| \leq \varepsilon. \end{aligned} \quad (37)$$

On the other hand, by the Moreau decomposition (8), we have

$$\begin{aligned} \|p_{\hat{k}} - \mathcal{A}(x_{\hat{k}})\| &= \left\| \text{prox}_{(\lambda+\beta_{\hat{k}})h^*}(\mathcal{A}(x_{\hat{k}}) + \beta_{\hat{k}} y_{\hat{k}}) - \mathcal{A}(x_{\hat{k}}) \right\| \\ &= \left\| \beta_{\hat{k}} y_{\hat{k}} - (\lambda + \beta_{\hat{k}}) \text{prox}_{h/(\lambda+\beta_{\hat{k}})} \left(\frac{\mathcal{A}(x_{\hat{k}}) + \beta_{\hat{k}} y_{\hat{k}}}{\lambda + \beta_{\hat{k}}} \right) \right\| \\ &\stackrel{(a)}{\leq} (\lambda + 2\beta_{\hat{k}}) R \stackrel{(b)}{\leq} \varepsilon, \end{aligned} \quad (38)$$

where (a) uses $\text{prox}_{h/(\lambda+\beta_{\hat{k}})}(\cdot) \in \text{dom } h$ and (22), and (b) uses $\lambda = \varepsilon/(2R)$, $0 \leq \beta_{\hat{k}} \leq \beta_1/\hat{k}^\rho$, and $\hat{k} \geq K_1 = \lceil (4\beta_1 R/\varepsilon)^{\frac{1}{\rho}} \rceil$. Combining (37) and (38), we conclude that $x_{\hat{k}}$ is an ε -ROS point of problem (1). \blacksquare

Algorithm 2: RADA-PGD for problem (1)

-
- 1 Input $x_1 \in \mathcal{M}, y_1 \in \text{dom } h, \lambda > 0, \beta_1 \geq 0, \rho > 1, \tau_1, \tau_2 \in (0, 1)$, and a sequence $\{T_k\} \subset \mathbb{Z}_+$ with $1 \leq T_k \leq \bar{T}$, where \bar{T} is a preset positive integer.
 - 2 **for** $k = 1, 2, \dots$ **do**
 - 3 Set $x_{k,1} = x_k$.
 - 4 **for** $t = 1, 2, \dots, T_k$ **do**
 - 5 Compute $\zeta_{k,t} = \ell_k^{-1}$ and update $x_{k,t+1} \in \text{proj}_{\mathcal{M}}(x_{k,t} - \zeta_{k,t} \nabla \Phi_k(x_{k,t}))$.
 - 6 Update $x_{k+1} = x_{k,T_k+1}, y_{k+1}$ via (14), and β_{k+1} via (15).
-

Theorem 3 shows that the iteration complexity of Algorithm 1 for obtaining an ε -ROS point of problem (1) is $\mathcal{O}(\varepsilon^{-3})$ since $\lambda = \varepsilon/(2R)$ and $c_\lambda = \mathcal{O}(\lambda)$. This matches the best-known complexity results for the DSGM in [5] and the ManIAL in [13]. However, the results in [5, 13] apply only to the case where \mathcal{A} is a linear mapping, whereas our results work for a general nonlinear mapping \mathcal{A} .

Before we leave this section, let us point out that by Theorems 2 and 3, Algorithm 1 can find both an ε -RGS point and an ε -ROS point of problem (1) within $\mathcal{O}(\varepsilon^{-3})$ iterations. This is noteworthy, as the results in [45, 77] for the Euclidean setting suggest that in general extra computation is needed to convert between an ε -RGS point and an ε -ROS point. Moreover, existing works on NC-L minimax problems [20, 58, 74] only establish the iteration complexity for obtaining an ε -game-stationary point. The iteration complexity for obtaining an ε -optimization-stationary point is not addressed in these works.

4. Two Efficient Algorithms for Problem (1) In this section, we propose two efficient algorithms within the proposed RADA algorithmic framework, called RADA-PGD and RADA-RGD in Algorithms 2 and 3, respectively, for solving the Riemannian NC-L minimax problem (1). These two algorithms differ in the first-order method used for approximately solving the subproblem (12) to find a point $x_{k+1} \in \mathcal{M}$ that satisfies the sufficient decrease condition (13).

When the projection operator onto a closed manifold \mathcal{M} is tractable, which is the case in the motivating applications introduced in Section 1.1, we can employ the nonconvex projected gradient method to compute x_{k+1} that satisfies the sufficient decrease condition (13). Noting that Φ_k is differentiable as shown in (18), it is straightforward to perform an ordinary projected gradient step with a suitable constant stepsize depending on the Lipschitz constant of $\nabla \Phi_k$. It is also possible to perform multiple projected gradient steps to achieve a more accurate solution to (12) while satisfying (13). Within the RADA algorithmic framework, the first customized algorithm is RADA with projected gradient descent (RADA-PGD), which is formally presented in Algorithm 2. Here, \mathbb{Z}_+ is the set of all positive integers, and the integer $T_k \in [1, \bar{T}]$ denotes the number of the projected gradient descent steps to be performed at the k -th iteration with \bar{T} being a predetermined positive integer. The stepsize $\zeta_{k,t}$ is set as ℓ_k^{-1} , where ℓ_k is to be specified later in (46).

An alternative approach is to perform a Riemannian gradient descent step to obtain x_{k+1} satisfying the sufficient decrease condition (13). For this method, determining a suitable constant stepsize is not as easy as in RADA-PGD. Hence, for practical efficiency, we utilize the Barzilai-Borwein (BB) stepsize [3, 16, 24, 29–31, 71] as the initial stepsize and perform a backtracking line search to find a stepsize satisfying (13). Similar to Algorithm 2, we can also perform multiple Riemannian gradient steps to achieve a more accurate solution to (12). The corresponding algorithm is named RADA with Riemannian gradient descent (RADA-RGD) and is presented in Algorithm 3, where

Algorithm 3: RADA-RGD for problem (1)

- 1 Input $x_1 \in \mathcal{M}$, $y_1 \in \text{dom } h$, $\lambda > 0$, $\beta_1 \geq 0$, $0 < \zeta_{\min} < \zeta_{1,1} < \zeta_{\max}$, $\rho > 1$, $\eta \in (0, 1)$,
 $c_1 \in (0, 1)$, $\{\nu_k\} \in \mathbb{S}$, $\tau_1, \tau_2 \in (0, 1)$, and a sequence $\{T_k\} \subset \mathbb{Z}_+$ with $1 \leq T_k \leq \bar{T}$, where \bar{T}
 is a preset positive integer.
 - 2 **for** $k = 1, 2, \dots$ **do**
 - 3 **Set** $x_{k,1} = x_k$.
 - 4 **for** $t = 1, 2, \dots, T_k$ **do**
 - 5 Find the smallest nonnegative integer j such that

$$\Phi_k(\text{R}_{x_{k,t}}(-\zeta_{k,t}\eta^j \text{grad } \Phi_k(x_{k,t}))) - \Phi_k(x_{k,t}) \leq -c_1\zeta_{k,t}\eta^j \|\text{grad } \Phi_k(x_{k,t})\|^2 + \frac{\nu_k}{T_k}, \quad (40)$$
 and update $x_{k,t+1} = \text{R}_{x_{k,t}}(-\zeta_{k,t}\eta^j \text{grad } \Phi_k(x_{k,t}))$.
 - 6 Compute $\zeta_{k,t+1}^{\text{BB}}$ according to (39) and set

$$\zeta_{k,t+1} = \min \left\{ \max \left\{ \zeta_{k,t+1}^{\text{BB}}, \zeta_{\min} \right\}, \zeta_{\max} / \|\text{grad } \Phi_k(x_{k,t+1})\| \right\}. \quad (41)$$
 - 7 Update $x_{k+1} = x_{k,T_k+1}$, y_{k+1} via (14), and β_{k+1} via (15). Set $\zeta_{k+1,1} = \zeta_{k,T_k+1}$.
-

the integer $T_k \in [1, \bar{T}]$ denotes the number of the Riemannian gradient descent steps to be performed at the k -th iteration and \bar{T} is a predetermined positive integer. The Riemannian BB stepsize $\zeta_{k,t+1}^{\text{BB}}$ in Algorithm 3 is set as

$$\zeta_{k,t+1}^{\text{BB}} = \begin{cases} \frac{\|x_{k,t+1} - x_{k,t}\|^2}{|\langle x_{k,t+1} - x_{k,t}, v_{k,t} \rangle|}, & \text{for odd } t; \\ \frac{|\langle x_{k,t+1} - x_{k,t}, v_{k,t} \rangle|}{\|v_{k,t}\|^2}, & \text{for even } t, \end{cases} \quad (39)$$

where $v_{k,t} = \text{grad } \Phi_k(x_{k,t+1}) - \text{grad } \Phi_k(x_{k,t})$.

Note that both RADA-PGD and RADA-RGD are single-loop algorithms, as the inner iteration number T_k is bounded by a predetermined positive integer \bar{T} . The choice of T_k is also flexible and can be adapted to the specific problem structure. Empirically, if the value function Φ_k is benign, for example, if the Lipschitz constant of its gradient can be well estimated, then setting $T_k \equiv 1$ is often sufficient. Otherwise, we suggest setting a slightly larger T_k to ensure a more substantial decrease in the value of Φ_k , thereby improving the practical performance of RADA-PGD and RADA-RGD. This is particularly important because the subproblem (12) is itself a minimax problem and can be viewed as a surrogate of problem (1). Thanks to the proposed value function Φ_k and the flexible setting of T_k , our approaches to updating x_{k+1} are expected to be more efficient than simply performing a gradient descent step on certain surrogate of $F(\cdot, y_k)$, as is done in [20, 53, 58, 74, 76]. This will be demonstrated in Section 6. Moreover, even when adapted to the Euclidean setting, our proposed RADA algorithmic framework and the associated RADA-PGD and RADA-RGD are new.

4.1. Verification of the Sufficient Decrease Condition (13) In this subsection, we show that the update x_{k+1} in RADA-PGD and RADA-RGD satisfies the sufficient decrease condition (13). Therefore, the iteration complexity of these two algorithms can be directly obtained from the results in Section 3.2.

First, we make the following (somewhat standard) assumptions for our analysis; see, e.g., [9, 11, 12, 15, 27, 28, 48].

ASSUMPTION 2. *The level set*

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

is compact, where Φ and Υ are defined in (21) and (23), respectively, and x_1 is the initial point of Algorithm 1.

ASSUMPTION 3. *The function f and the mapping \mathcal{A} in (1) 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} , i.e.,*

$$f(x') \leq f(x) + \langle \nabla f(x), x' - x \rangle + \frac{L_f}{2} \|x' - x\|^2, \quad \forall x, x' \in \mathcal{M}. \quad (42)$$

(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 over $\text{conv } \mathcal{M}$, respectively. In other words, for any $x, x' \in \text{conv } \mathcal{M}$, we have*

$$\|\mathcal{A}(x) - \mathcal{A}(x')\| \leq L_{\mathcal{A}}^0 \|x - x'\|, \quad (43a)$$

$$\|\nabla \mathcal{A}(x) - \nabla \mathcal{A}(x')\| \leq L_{\mathcal{A}}^1 \|x - x'\|. \quad (43b)$$

Moreover, the mapping $\nabla \mathcal{A}$ is bounded over $\text{conv } \mathcal{M}$, i.e.,

$$\rho_{\mathcal{A}} := \sup_{x \in \text{conv } \mathcal{M}} \|\nabla \mathcal{A}(x)\| < +\infty. \quad (44)$$

In our subsequent development, we allow for the possibility that the retraction at some $x \in \mathcal{M}$ is defined only locally; see [9, Definition 2.1]. As such, we make the following assumption, which stipulates that the retraction is well defined on a compact subset of the tangent bundle; cf. [9, Remark 2.2].

ASSUMPTION 4. *There exists a constant $\varrho > 0$ such that for any $(x, v) \in \mathcal{V} := \{(x, v) \mid x \in \Omega_{x_1}, v \in \mathbb{T}_x \mathcal{M}, \|v\| \leq \varrho\}$, the point $\mathbb{R}_x(v) \in \mathcal{M}$ is well defined.*

The following lemma, which is extracted from [9, Appendix B], shows that the retraction satisfies the first- and second-order boundedness conditions on \mathcal{V} .

LEMMA 2. *Suppose that Assumptions 2 and 4 hold. Then, there exist constants $\alpha_1, \alpha_2 > 0$ such that*

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

for any $(x, v) \in \mathcal{V}$.

Based on Lemma 2, we can establish the descent property of Φ_k as follows.

LEMMA 3. *Suppose that Assumptions 2, 3, and 4 hold. Then, the value function Φ_k in (11) satisfies the following properties:*

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

$$\Phi_k(x') \leq \Phi_k(x) + \langle \nabla \Phi_k(x), x' - x \rangle + \frac{\ell_k}{2} \|x' - x\|^2, \quad (45)$$

where

$$\ell_k = L_f + RL_A^1 + \frac{\rho_{\mathcal{A}} L_{\mathcal{A}}^0}{\lambda + \beta_k}. \quad (46)$$

(ii) **Riemannian Descent.** For any $(x, v) \in \mathcal{V}$, we have

$$\Phi_k(\mathbb{R}_x(v)) \leq \Phi_k(x) + \langle \text{grad } \Phi_k(x), v \rangle + \frac{L_k(x)}{2} \|v\|^2, \quad (47)$$

where

$$L_k(x) = \ell_k \alpha_1^2 + 2(\|\nabla f(x)\| + \rho_{\mathcal{A}} R) \alpha_2. \quad (48)$$

Proof We first prove (45). Let $P_k(x) = \Phi_k(x) - f(x)$. By (18), we have

$$\nabla P_k(x) = \nabla \mathcal{A}(x)^\top \mathcal{B}(x),$$

where

$$\mathcal{B}(x) = \text{prox}_{h/(\lambda+\beta_k)} \left(\frac{\mathcal{A}(x) + \beta_k y_k}{\lambda + \beta_k} \right).$$

We claim that ∇P_k is Lipschitz continuous. Indeed, for any $x, x' \in \text{conv } \mathcal{M}$, we have

$$\begin{aligned} \|\nabla P_k(x) - \nabla P_k(x')\| &= \|\nabla \mathcal{A}(x)^\top \mathcal{B}(x) - \nabla \mathcal{A}(x')^\top \mathcal{B}(x')\| \\ &\leq \|(\nabla \mathcal{A}(x) - \nabla \mathcal{A}(x'))^\top \mathcal{B}(x)\| + \|\nabla \mathcal{A}(x')^\top (\mathcal{B}(x) - \mathcal{B}(x'))\| \\ &\stackrel{(a)}{\leq} L_{\mathcal{A}}^1 \|\mathcal{B}(x)\| \cdot \|x - x'\| + \rho_{\mathcal{A}} \|\mathcal{B}(x) - \mathcal{B}(x')\| \\ &\stackrel{(b)}{\leq} L_{\mathcal{A}}^1 \|\mathcal{B}(x)\| \cdot \|x - x'\| + \frac{\rho_{\mathcal{A}}}{\lambda + \beta_k} \|\mathcal{A}(x) - \mathcal{A}(x')\| \\ &\stackrel{(c)}{\leq} \left(RL_A^1 + \frac{\rho_{\mathcal{A}} L_{\mathcal{A}}^0}{\lambda + \beta_k} \right) \|x - x'\|, \end{aligned}$$

where (a) is due to (43b) and (44), (b) is due to the nonexpansiveness of the proximal operator, and (c) is due to $\mathcal{B}(x) \in \text{dom } h$, (22), and (43a). Therefore, by [55, Lemma 1.2.3], we know that P_k satisfies the Euclidean descent property, namely, for any $x, x' \in \mathcal{M}$,

$$P_k(x') \leq P_k(x) + \langle \nabla P_k(x), x' - x \rangle + \frac{1}{2} \left(RL_A^1 + \frac{\rho_{\mathcal{A}} L_{\mathcal{A}}^0}{\lambda + \beta_k} \right) \|x' - x\|^2. \quad (49)$$

Combining (49) and (42) and noting that $\Phi_k = f + P_k$, we conclude that Φ_k satisfies the Euclidean descent property (45).

Moreover, using Lemma 2 and following a similar analysis as that in [9, Appendix B], we know that Φ_k satisfies the Riemannian descent property (47). This completes the proof. \blacksquare

To prove that x_{k+1} in Algorithm 2 satisfies the sufficient decrease condition (13), we need the following lemma.

LEMMA 4. *Let $\{x_{k,t}\}_{t=1}^{T_k+1}$ be the sequence generated by Algorithm 2 with $\zeta_{k,t} = \ell_k^{-1}$ at the k -th iteration. Suppose that Assumptions 2, 3, and 4 hold. If $x_{k,t} \in \Omega_{x_1}$, then*

$$\Phi_k(x_{k,t+1}) - \Phi_k(x_{k,t}) \leq - \min \left\{ \frac{\|\text{grad } \Phi_k(x_{k,t})\|}{2L_k(x_{k,t})}, \frac{\varrho}{2} \right\} \|\text{grad } \Phi_k(x_{k,t})\|.$$

Proof By the definition of $x_{k,t+1}$ in Algorithm 2, we have

$$\begin{aligned} x_{k,t+1} &\in \arg \min_{x \in \mathcal{M}} \left\| x - x_{k,t} + \frac{1}{\ell_k} \nabla \Phi_k(x_{k,t}) \right\|^2 \\ &= \arg \min_{x \in \mathcal{M}} \left\{ \langle \nabla \Phi_k(x_{k,t}), x - x_{k,t} \rangle + \frac{\ell_k}{2} \|x - x_{k,t}\|^2 \right\}. \end{aligned} \quad (50)$$

Consider $x'_{k,t+1} = R_{x_{k,t}}(d_{k,t})$ with $d_{k,t} = -c_{k,t} \text{grad } \Phi_k(x_{k,t})$, where

$$c_{k,t} := \min \left\{ \frac{1}{L_k(x_{k,t})}, \frac{\varrho}{\|\text{grad } \Phi_k(x_{k,t})\|} \right\}. \quad (51)$$

We know that $(x_{k,t}, d_{k,t}) \in \mathcal{V}$ from (51) where \mathcal{V} is defined in Assumption 4. By the optimality of $x_{k,t+1}$ in (50), we get

$$\begin{aligned} &\langle \nabla \Phi_k(x_{k,t}), x_{k,t+1} - x_{k,t} \rangle + \frac{\ell_k}{2} \|x_{k,t+1} - x_{k,t}\|^2 \\ &\leq \langle \nabla \Phi_k(x_{k,t}), x'_{k,t+1} - x_{k,t} \rangle + \frac{\ell_k}{2} \|x'_{k,t+1} - x_{k,t}\|^2 \\ &= \langle \nabla \Phi_k(x_{k,t}), d_{k,t} \rangle + \langle \nabla \Phi_k(x_{k,t}), x'_{k,t+1} - x_{k,t} - d_{k,t} \rangle + \frac{\ell_k}{2} \|x'_{k,t+1} - x_{k,t}\|^2 \\ &\leq \langle \nabla \Phi_k(x_{k,t}), d_{k,t} \rangle + \|\nabla \Phi_k(x_{k,t})\| \cdot \|x'_{k,t+1} - x_{k,t} - d_{k,t}\| + \frac{\ell_k}{2} \|x'_{k,t+1} - x_{k,t}\|^2 \\ &\stackrel{(a)}{\leq} -c_{k,t} \|\text{grad } \Phi_k(x_{k,t})\|^2 + \alpha_2 (\|\nabla f(x)\| + \rho_A R) \|d_{k,t}\|^2 + \frac{\alpha_1^2 \ell_k}{2} \|d_{k,t}\|^2 \\ &\stackrel{(b)}{=} -c_{k,t} \left(1 - \frac{L_k(x_{k,t})}{2} c_{k,t} \right) \|\text{grad } \Phi_k(x_{k,t})\|^2, \end{aligned}$$

where (a) follows from Lemma 2, $\text{prox}_{h/(\lambda+\beta_k)}((\mathcal{A}(x) + \beta_k y_k)/(\lambda + \beta_k)) \in \text{dom } h$, (18), (22), and (44), and (b) is due to (48). In addition, applying (45) with $x' = x_{k,t+1}$ and $x = x_{k,t}$ yields

$$\begin{aligned} \Phi_k(x_{k,t+1}) - \Phi_k(x_{k,t}) &\leq \langle \nabla \Phi_k(x_{k,t}), x_{k,t+1} - x_{k,t} \rangle + \frac{\ell_k}{2} \|x_{k,t+1} - x_{k,t}\|^2 \\ &\leq -c_{k,t} \left(1 - \frac{L_k(x_{k,t})}{2} c_{k,t} \right) \|\text{grad } \Phi_k(x_{k,t})\|^2 \stackrel{(a)}{\leq} -\frac{c_{k,t}}{2} \|\text{grad } \Phi_k(x_{k,t})\|^2, \end{aligned}$$

where (a) is due to (51). The desired result follows directly from substituting (51) into the above inequalities. ■

Based on Lemma 4, we now present Proposition 1, which shows that Algorithm 2 is within the RADA algorithmic framework.

PROPOSITION 1. *Let $\{x_k\}$ be the sequence generated by Algorithm 2 with $\zeta_{k,t} = \ell_k^{-1}$. Suppose that Assumptions 2, 3, and 4 hold. Then, the sequence $\{x_k\}$ lies in the compact level set Ω_{x_1} . Moreover, for $k = 1, 2, \dots$, we have*

$$\Phi_k(x_{k+1}) - \Phi_k(x_k) \leq - \min \left\{ \frac{\|\text{grad } \Phi_k(x_k)\|}{2L_k(x_k)}, \frac{\rho}{2} \right\} \|\text{grad } \Phi_k(x_k)\|. \quad (52)$$

Proof We apply mathematical induction. First, we show that (52) holds when $k = 1$. Since $x_{1,1} = x_1 \in \Omega_{x_1}$, using Lemma 4 with $t = 1$ and the definition of $c_{k,t}$ in (51), we have

$$\Phi_1(x_{1,2}) - \Phi_1(x_{1,1}) \leq - \min \left\{ \frac{\|\text{grad } \Phi_1(x_{1,1})\|}{2L_1(x_{1,1})}, \frac{\rho}{2} \right\} \|\text{grad } \Phi_1(x_{1,1})\| \leq 0.$$

This gives

$$\Phi(x_{1,2}) \stackrel{(a)}{\leq} \Phi_1(x_{1,2}) + \left(\frac{\lambda}{2} + 2\beta_1 \right) R^2 \stackrel{(b)}{\leq} \Phi_1(x_{1,1}) + \Upsilon \stackrel{(c)}{\leq} \Phi(x_1) + \Upsilon, \quad (53)$$

where (a) follows from (31), (b) uses the definition of Υ in (23), and (c) holds by the definition of Φ in (21) and the fact that

$$\begin{aligned} \Phi_1(x) &= \max_{y \in \mathcal{E}_2} \left\{ f(x) + \langle \mathcal{A}(x), y \rangle - h(y) - \frac{\lambda}{2} \|y\|^2 - \frac{\beta_1}{2} \|y - y_1\|^2 \right\} \\ &\leq \max_{y \in \mathcal{E}_2} \{ f(x) + \langle \mathcal{A}(x), y \rangle - h(y) \} = \Phi(x). \end{aligned} \quad (54)$$

Hence, we have $x_{1,2} \in \Omega_{x_1}$ from (53). Repeating the above procedure, we further know that

$$\Phi_1(x_{1,t+1}) - \Phi_1(x_{1,t}) \leq - \min \left\{ \frac{\|\text{grad } \Phi_1(x_{1,t})\|}{2L_1(x_{1,t})}, \frac{\rho}{2} \right\} \|\text{grad } \Phi_1(x_{1,t})\| \leq 0 \quad (55)$$

and $x_{1,t+1} \in \Omega_{x_1}$ for $t = 1, 2, \dots, T_1$. Summing the above inequality over $t = 1, 2, \dots, T_1$ yields

$$\begin{aligned} \Phi_1(x_{1,T_1+1}) - \Phi_1(x_{1,1}) &= \sum_{t=1}^{T_1} (\Phi_1(x_{1,t+1}) - \Phi_1(x_{1,t})) \\ &\leq - \min \left\{ \frac{\|\text{grad } \Phi_1(x_1)\|}{2L_1(x_1)}, \frac{\rho}{2} \right\} \|\text{grad } \Phi_1(x_1)\|. \end{aligned} \quad (56)$$

Recalling that $x_{1,1} = x_1$ and $x_{1,T_1+1} = x_2$, it follows from (56) that (52) holds for $k = 1$.

Next, we assume that $x_j \in \Omega_{x_1}$ for some $j \geq 2$ and (52) holds for $k = 1, 2, \dots, j-1$. It remains to prove that $x_{j+1} \in \Omega_{x_1}$ and (52) holds for $k = j$. Combining (25) and (52) with $k = 1, 2, \dots, j-1$ yields

$$\Phi_{k+1}(x_{k+1}) - \Phi_k(x_k) \leq 2\beta_k R^2, \quad \forall k = 1, 2, \dots, j-1. \quad (57)$$

Since $x_{j,1} = x_j \in \Omega_{x_1}$, Lemma 4 implies that

$$\Phi_j(x_{j,2}) \leq \Phi_j(x_{j,1}) = \Phi_j(x_j). \quad (58)$$

Similar to the proof for $j = 1$, we have

$$\begin{aligned}
\Phi(x_{j,2}) &\leq \Phi_j(x_{j,2}) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&\stackrel{(a)}{\leq} \Phi_j(x_j) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&= \Phi_1(x_1) + \sum_{k=2}^j (\Phi_k(x_k) - \Phi_{k-1}(x_{k-1})) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&\stackrel{(b)}{\leq} \Phi_1(x_1) + \frac{\lambda}{2} R^2 + \sum_{k=1}^{+\infty} (\nu_k + 2\beta_k R^2) \stackrel{(c)}{\leq} \Phi(x_1) + \Upsilon,
\end{aligned} \tag{59}$$

where (a) is from (58), (b) is due to (57) and $\beta_j \leq \beta_1$, and (c) is due to (54) and the definition of Υ in (23). Hence, we get $x_{j,2} \in \Omega_{x_1}$. Similar to (55) and (56), we deduce from Lemma 4 that

$$\Phi_j(x_{j,t+1}) - \Phi_j(x_{j,t}) \leq -\min \left\{ \frac{\|\text{grad } \Phi_j(x_{j,t})\|}{2L_j(x_{j,t})}, \frac{\varrho}{2} \right\} \|\text{grad } \Phi_j(x_{j,t})\| \leq 0 \tag{60}$$

for $t = 1, 2, \dots, T_j$. It follows from $x_{j,1} = x_j$ and $x_{j,T_j+1} = x_{j+1}$ that

$$\Phi_j(x_{j+1}) - \Phi_j(x_j) = \sum_{t=1}^{T_j} (\Phi_j(x_{j,t+1}) - \Phi_j(x_{j,t})) \leq -\min \left\{ \frac{\|\text{grad } \Phi_j(x_j)\|}{2L_j(x_j)}, \frac{\varrho}{2} \right\} \|\text{grad } \Phi_j(x_j)\|.$$

Moreover, similar to (59), we have

$$\begin{aligned}
\Phi(x_{j,T_j+1}) &\leq \Phi_j(x_{j,T_j+1}) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&\stackrel{(a)}{\leq} \Phi_j(x_j) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&= \Phi_1(x_1) + \sum_{k=2}^j (\Phi_k(x_k) - \Phi_{k-1}(x_{k-1})) + \left(\frac{\lambda}{2} + 2\beta_j\right) R^2 \\
&\stackrel{(b)}{\leq} \Phi_1(x_1) + \frac{\lambda}{2} R^2 + \sum_{k=1}^{+\infty} (\nu_k + 2\beta_k R^2) \stackrel{(c)}{\leq} \Phi(x_1) + \Upsilon,
\end{aligned}$$

where (a) is from (60) and $x_{j,1} = x_j$, (b) is due to (57) and $\beta_j \leq \beta_1$, and (c) is due to (54) and the definition of Υ in (23). Therefore, noting that $x_{j+1} = x_{j,T_j+1}$, we have $x_{j+1} \in \Omega_{x_1}$. Moreover, similar to the derivation of (56), we know from (60) that (52) holds for $k = j$. This completes the inductive step and hence also the proof. \blacksquare

Denote

$$\rho_\Phi := \max_{x \in \Omega_{x_1}} \|\nabla f(x)\| + \rho_{\mathcal{A}} R \quad \text{and} \quad \underline{L} := \ell_1 \alpha_1^2 + 2\rho_{\mathcal{A}} R \alpha_2, \tag{61}$$

where $\rho_{\mathcal{A}}$ and R are defined in (44) and (22), respectively. Then, for any $x \in \Omega_{x_1}$ and $k = 1, 2, \dots$, we have

$$\|\text{grad } \Phi_k(x)\| \leq \|\nabla \Phi_k(x)\| \leq \rho_\Phi \tag{62}$$

from (18), (22), and (44). Moreover, we have

$$0 < \underline{L} \leq L_k(x) \quad (63)$$

by substituting $0 \leq \beta_k \leq \beta_1$ into (48). To prove that x_{k+1} in Algorithm 3 satisfies the sufficient decrease condition (13), we need the following lemma.

LEMMA 5. *Let $\{x_{k,t}\}_{t=1}^{T_k+1}$ be the sequence generated by Algorithm 3 with $\zeta_{\max} \leq \varrho$ at the k -th iteration, where ϱ is given in Assumption 4. Suppose that Assumptions 2, 3, and 4 hold. If $x_{k,t} \in \Omega_{x_1}$, then*

$$\Phi_k(x_{k,t+1}) - \Phi_k(x_{k,t}) \leq -\frac{c_1 \min \{ \rho_{\Phi}^{-1} \zeta_{\max} \underline{L}, \zeta_{\min} \underline{L}, 2\eta(1-c_1) \}}{L_k(x_{k,t})} \|\text{grad } \Phi_k(x_{k,t})\|^2 + \frac{\nu_k}{T_k}. \quad (64)$$

Proof Let \hat{j} be the smallest nonnegative integer satisfying (40) and $\hat{\zeta}_{k,t} := \zeta_{k,t} \eta^{\hat{j}}$. First, if $\hat{j} \geq 1$, then by the backtracking line search procedure, we have

$$\Phi_k(x_{k,t}^+) - \Phi_k(x_{k,t}) > -c_1 \frac{\hat{\zeta}_{k,t}}{\eta} \|\text{grad } \Phi_k(x_{k,t})\|^2 + \frac{\nu_k}{T_k} \geq -c_1 \frac{\hat{\zeta}_{k,t}}{\eta} \|\text{grad } \Phi_k(x_{k,t})\|^2, \quad (65)$$

where $x_{k,t}^+ := R_{x_{k,t}}(-\hat{\zeta}_{k,t} \eta^{-1} \text{grad } \Phi_k(x_{k,t}))$. Since $\hat{\zeta}_{k,t} \eta^{-1} = \zeta_{k,t} \eta^{\hat{j}-1} \leq \zeta_{k,t}$ and (41) implies that

$$\zeta_{k,t} \|\text{grad } \Phi_k(x_{k,t})\| \leq \zeta_{\max} \leq \varrho,$$

we get $(x_{k,t}, -\hat{\zeta}_{k,t} \eta^{-1} \text{grad } \Phi_k(x_{k,t})) \in \mathcal{V}$, where \mathcal{V} is defined in Assumption 4. By Lemma 3 (ii), we have

$$\Phi_k(x_{k,t}^+) - \Phi_k(x_{k,t}) \leq -\left(1 - \frac{\hat{\zeta}_{k,t} L_k(x_{k,t})}{2\eta}\right) \frac{\hat{\zeta}_{k,t}}{\eta} \|\text{grad } \Phi_k(x_{k,t})\|^2. \quad (66)$$

Combining (65) and (66) yields

$$\hat{\zeta}_{k,t} \geq \frac{2\eta(1-c_1)}{L_k(x_{k,t})}.$$

Second, if $\hat{j} = 0$, then $\hat{\zeta}_{k,t} = \zeta_{k,t}$. We bound

$$\hat{\zeta}_{k,t} \geq \min \left\{ \zeta_{k,t}, \frac{2\eta(1-c_1)}{L_k(x_{k,t})} \right\} \geq \frac{\min \{ \rho_{\Phi}^{-1} \zeta_{\max} \underline{L}, \zeta_{\min} \underline{L}, 2\eta(1-c_1) \}}{L_k(x_{k,t})}, \quad (67)$$

where the second inequality holds by (63), (62), and $\zeta_{k,t} \geq \min \{ \zeta_{\min}, \zeta_{\max} / \|\text{grad } \Phi_k(x_{k,t+1})\| \}$ as defined in (41). By substituting (67) into (40), we get (64), as desired. \blacksquare

Based on Lemma 5, we have the following proposition, which shows that Algorithm 3 is within the RADA algorithmic framework.

PROPOSITION 2. *Let $\{x_k\}$ be the sequence generated by Algorithm 3. Suppose that Assumptions 2, 3, and 4 hold. Then, the sequence $\{x_k\}$ lies in the compact level set Ω_{x_1} . Moreover, for $k = 1, 2, \dots$, we have*

$$\Phi_k(x_{k+1}) - \Phi_k(x_k) \leq -\frac{c_1 \min \{ \rho_{\Phi}^{-1} \zeta_{\max} \underline{L}, \zeta_{\min} \underline{L}, 2\eta(1-c_1) \}}{L_k(x_{k,t})} \|\text{grad } \Phi_k(x_k)\|^2 + \nu_k.$$

Proof The result is obtained by using the same techniques as in the proof of Proposition 1. ■
Recalling that $\beta_k \geq 0$, we know from (46), (48), and (61) that

$$L_k(x_k) \leq \bar{L} := \alpha_1^2 \left(L_f + RL_{\mathcal{A}}^1 + \frac{\rho_{\mathcal{A}} L_{\mathcal{A}}^0}{\lambda} \right) + 2\rho_{\Phi} \alpha_2 = \mathcal{O}(\lambda^{-1}).$$

Therefore, from Propositions 1 and 2, we see that the sufficient decrease condition (13) holds with

$$c_{\lambda} = \frac{1}{2\bar{L}} \quad \text{and} \quad c' = \frac{\varrho}{2}$$

for Algorithm 2 and with

$$c_{\lambda} = \frac{\min \{ \rho_{\Phi}^{-1} \zeta_{\max} \bar{L}, \zeta_{\min} \bar{L}, 2\eta(1 - c_1) \}}{\bar{L}}, \quad \text{and} \quad c' = c_{\lambda} \rho_{\Phi}$$

for Algorithm 3. Here, we use $\|\text{grad } \Phi_k(x)\| = \min \{ \|\text{grad } \Phi_k(x)\|, \rho_{\Phi} \}$ as shown by (62). Consequently, by invoking Theorems 2 and 3, we obtain the following complexity result:

THEOREM 4. *Given a constant $\varepsilon > 0$, let $\{(x_k, y_k)\}$ be the sequence generated by Algorithms 2 or 3 with $\lambda = \varepsilon/(2R)$. Suppose that Assumptions 1, 2, 3, and 4 hold. Then, there exists an index $\hat{k} \leq K$ with $K = \mathcal{O}(\varepsilon^{-3})$ such that $(x_{\hat{k}}, y_{\hat{k}+\frac{1}{2}})$ is an ε -RGS point and $x_{\hat{k}}$ is an ε -ROS point of problem (1).*

5. Connections with Existing Algorithms Recall that the minimax problem (1) is equivalent to the nonsmooth composite problem (7). As it turns out, the proposed RADA algorithmic framework and the proposed RADA-RGD algorithm for solving problem (1) have interesting connections to the RALMs [13, 14, 83] and RADMM [40] for solving problem (7), respectively. In what follows, we elaborate on these connections and explain why RADA and RADA-RGD can be more advantageous than RALM and RADMM for tackling problem (1).

5.1. Connection between RADA and RALM In this subsection, we show how the updates of the proposed RADA algorithmic framework for solving the minimax problem (1) relates to those of the RALMs [13, 14, 83] for solving the equivalent nonsmooth composite problem (7).

On one hand, using (17), the subproblem (12) for obtaining the update x_{k+1} in RADA can be written as

$$\min_{x \in \mathcal{M}} \{ f(x) + M_{(\lambda+\beta_k)h^*}(\mathcal{A}(x) + \beta_k y_k) \}. \quad (68)$$

Moreover, by the Moreau decomposition (8), the update formula (14) for y_{k+1} in RADA can be written as

$$y_{k+1} = \frac{1}{\lambda + \beta_k} (\beta_k y_k + \mathcal{A}(x_{k+1}) - \text{prox}_{(\lambda+\beta_k)h^*}(\mathcal{A}(x_{k+1}) + \beta_k y_k)). \quad (69)$$

On the other hand, consider applying the RALM in [13, 14, 83] to the following equivalent reformulation of (7):

$$\min_{x \in \mathcal{M}, q \in \mathcal{E}_2} \{ f(x) + h^*(q) \} \quad \text{s.t.} \quad \mathcal{A}(x) - q = 0. \quad (70)$$

Here, q is an auxiliary variable. The augmented Lagrangian function associated with problem (70) is given by

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

where y is the Lagrange multiplier corresponding to the constraint $\mathcal{A}(x) - q = 0$ and $\sigma > 0$ is the penalty parameter. At the k -th iteration, instead of utilizing the subproblem

$$\min_{x \in \mathcal{M}, q \in \mathcal{E}_2} \mathcal{L}_{\sigma_k}(x, q; y_k), \quad (71)$$

RALM utilizes the equivalent subproblem

$$\min_{x \in \mathcal{M}} \left\{ \mathcal{L}_k(x) := f(x) + M_{h^*/\sigma_k} \left(\mathcal{A}(x) + \frac{y_k}{\sigma_k} \right) \right\} \quad (72)$$

to obtain the update x_{k+1} , where the equivalence follows by eliminating the variable q in (71). Specifically, RALM first approximately solves (72) to obtain a point $x_{k+1} \in \mathcal{M}$ that satisfies

$$\|\text{grad } \mathcal{L}_k(x_{k+1})\| \leq \varepsilon_k, \quad (73)$$

where the sequence of tolerances $\{\varepsilon_k\}$ converges to 0. Then, it updates

$$y_{k+1} = y_k + \sigma_k \left(\mathcal{A}(x_{k+1}) - \text{prox}_{h^*/\sigma_k} \left(\mathcal{A}(x_{k+1}) + \frac{y_k}{\sigma_k} \right) \right). \quad (74)$$

Now, let us compare the subproblems (68) and (72) for updating x and the formulas (69) and (74) for updating y . When $\lambda = 0$ and $\beta_k = \sigma_k^{-1}$, the subproblem (68) in RADA coincides with the subproblem (72) in RALM, and the update formula (69) in RADA reduces to the update formula (74) in RALM. However, when $\lambda > 0$, both (68) and (69) in RADA differ from their respective counterparts (72) and (74) in RALM. More importantly, RADA and RALM differ in their stopping criteria for the subproblems. Indeed, RADA only needs to find a solution x_{k+1} that satisfies the sufficient decrease condition (13), whereas RALM needs to find a solution x_{k+1} that satisfies the inexactness condition (73). The former offers significantly more flexibility when solving the subproblem. It should be mentioned that (68) is also related to the so-called dampened or perturbed augmented Lagrangian function (see, e.g., in [17, 37]), which is used in the design of primal-dual algorithms for linearly constrained composite optimization problems in the Euclidean space. Furthermore, when $\beta_k \equiv 0$ and $\lambda > 0$, the subproblem (68) becomes

$$\min_{x \in \mathcal{M}} \{f(x) + M_{\lambda h^*}(\mathcal{A}(x))\}, \quad (75)$$

which corresponds to the subproblem in the Riemannian smoothing methods in [5, 59, 80].

The above relationship between the proposed RADA algorithmic framework and RALM provides important insights into the update formula (15) for β_{k+1} based on the parameter δ_{k+1} defined in (16), as well as the numerical performance of the two approaches. Specifically, at the k -th iteration of RALM, we can express the update q_{k+1} as

$$q_{k+1} = \text{prox}_{h^*/\sigma_k} \left(\mathcal{A}(x_{k+1}) + \frac{y_k}{\sigma_k} \right).$$

Using (69), we see that our Algorithm 1 has an analogous update, namely,

$$p_{k+1} = \text{prox}_{(\lambda+\beta_k)h^*} (\mathcal{A}(x_{k+1}) + \beta_k y_k).$$

Now, using (16), we compute

$$\begin{aligned}
 \delta_{k+1} &= \|(\lambda + \beta_k)y_{k+1} - \beta_k y_k\|_\infty \\
 &\stackrel{(a)}{=} \left\| -\beta_k y_k + (\lambda + \beta_k) \text{prox}_{h/(\lambda + \beta_k)} \left(\frac{\mathcal{A}(x_{k+1}) + \beta_k y_k}{\lambda + \beta_k} \right) \right\|_\infty \\
 &\stackrel{(b)}{=} \|\mathcal{A}(x_{k+1}) - \text{prox}_{(\lambda + \beta_k)h^*}(\mathcal{A}(x_{k+1}) + \beta_k y_k)\|_\infty \\
 &= \|\mathcal{A}(x_{k+1}) - p_{k+1}\|_\infty,
 \end{aligned}$$

where (a) is due to (14) and (b) follows from the Moreau decomposition (8). This shows that the parameter δ_{k+1} in (16) can be viewed as a primal residual in RALM.

5.2. Connection between RADA-RGD and RADMM In this subsection, we first elaborate on the relationship between RADA-RGD with $T_k \equiv 1$ and the RADMM in [40]. We then show that RADA-RGD with $T_k \equiv 1$ is equivalent to a new Riemannian sGS-ADMM.

Consider applying the RADMM in [40] to the following equivalent formulation of the smoothed problem (75):

$$\min_{x \in \mathcal{M}} \{f(x) + M_{\lambda h^*}(p)\} \quad \text{s.t.} \quad \mathcal{A}(x) - p = 0. \quad (76)$$

The augmented Lagrangian function associated with problem (76) is given by

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

where y is the Lagrange multiplier corresponding to the constraint $\mathcal{A}(x) - p = 0$ and $\sigma > 0$ is the penalty parameter. At the k -th iteration, RADMM computes the updates

$$x_{k+1} = R_{x_k}(-\zeta_k \text{grad}_x \tilde{\mathcal{L}}_{\sigma_k}(x_k, p_k; y_k)), \quad (77a)$$

$$p_{k+1} = \arg \min_{p \in \mathcal{E}_2} \tilde{\mathcal{L}}_{\sigma_k}(x_{k+1}, p; y_k), \quad (77b)$$

$$y_{k+1} = y_k + \sigma_k(\mathcal{A}(x_{k+1}) - p_{k+1}), \quad (77c)$$

where the stepsize $\zeta_k > 0$ in (77a) and the penalty parameter $\sigma_k > 0$ are both taken to be constants in [40].

Next, we modify RADA-RGD in Algorithm 3 by introducing an additional variable p , so that the resulting algorithm, which is shown in Algorithm 4, can be used to solve problem (7). Here, for simplicity of presentation, we omit the line search procedure in RADA-RGD.

The key difference between (77) and Algorithm 4 with $T_k \equiv 1$ is that the latter updates x and p in a symmetric Gauss-Seidel fashion, i.e., $p \rightarrow x \rightarrow p$. Therefore, Algorithm 4 with $T_k \equiv 1$ can be viewed as a Riemannian sGS-ADMM (cf. [10, 43, 44] for the sGS-ADMM in Euclidean convex settings) for solving the Riemannian nonsmooth composite problem (7). To the best of our knowledge, this is the first time a Riemannian sGS-type ADMM with a convergence guarantee is proposed.

Now, let us establish the equivalence between the original RADA-RGD (i.e., Algorithm 3) and its modified version (i.e., Algorithm 4).

PROPOSITION 3. *The updates of x and y in Algorithm 3 are equivalent to (78) and (79) in Algorithm 4, respectively.*

Algorithm 4: RADA-RGD reformulation for problem (7)

- 1 Input $x_1 \in \mathcal{M}$, $y_1 \in \mathcal{E}_2$, $\lambda > 0$, $\tau_1, \tau_2 \in (0, 1)$, and a sequence $\{T_k\} \subset \mathbb{Z}_+$ with $1 \leq T_k \leq \bar{T}$, where \bar{T} is a preset positive integer.
 - 2 **for** $k = 1, 2, \dots$ **do**
 - 3 Set $x_{k,1} = x_k$.
 - 4 **for** $t = 1, 2, \dots, T_k$ **do**
 - 5 Compute $\zeta_{k,t}$ as in Algorithm 3 and update

$$p_{k,t} = \arg \min_{p \in \mathcal{E}_2} \tilde{\mathcal{L}}_{1/\beta_k}(x_{k,t}, p; y_k),$$

$$x_{k,t+1} = \mathbb{R}_{x_{k,t}}(-\zeta_{k,t} \text{grad}_x \tilde{\mathcal{L}}_{1/\beta_k}(x_{k,t}, p_{k,t}; y_k)). \quad (78)$$
 - 6 Set $x_{k+1} = x_{k,T_k+1}$.
 - 7 Update $p_{k+1} = \arg \min_{p \in \mathcal{E}_2} \tilde{\mathcal{L}}_{1/\beta_k}(x_{k+1}, p; y_k)$ and

$$y_{k+1} = y_k + \frac{1}{\beta_k} (\mathcal{A}(x_{k+1}) - p_{k+1}). \quad (79)$$
 - 8 Update β_{k+1} as (15).
-

Proof Denote $p(x) := \arg \min_{p \in \mathcal{E}_2} \tilde{\mathcal{L}}_{1/\beta_k}(x, p; y_k)$. By [40, Lemma 1], we have

$$p(x) = \frac{\beta_k}{\lambda + \beta_k} \text{prox}_{(\lambda + \beta_k)h^*}(\mathcal{A}(x) + \beta_k y_k) + \frac{\lambda}{\lambda + \beta_k} (\mathcal{A}(x) + \beta_k y_k),$$

which implies that

$$\mathcal{A}(x) - p(x) = \frac{\beta_k}{\lambda + \beta_k} (\mathcal{A}(x) - \lambda y_k - \text{prox}_{(\lambda + \beta_k)h^*}(\mathcal{A}(x) + \beta_k y_k)). \quad (80)$$

We first show the equivalence of the updates of y in the two algorithms. Note that $p(x_{k+1}) = p_{k+1}$. Substituting (80) with $x = x_{k+1}$ into (79) yields

$$y_{k+1} = \frac{\beta_k}{\lambda + \beta_k} y_k + \frac{1}{\lambda + \beta_k} (\mathcal{A}(x_{k+1}) - \text{prox}_{(\lambda + \beta_k)h^*}(\mathcal{A}(x_{k+1}) + \beta_k y_k)),$$

which is exactly the update (69), or equivalently, the update (14) in Algorithm 3.

We now show the equivalence of the updates of x in the two algorithms. Substituting (80) into $\nabla_x \tilde{\mathcal{L}}_{1/\beta_k}(x, p(x); y_k)$ yields

$$\begin{aligned} \nabla_x \tilde{\mathcal{L}}_{1/\beta_k}(x, p(x); y_k) &= \nabla f(x) + \nabla \mathcal{A}(x)^\top \left(y_k + \frac{1}{\beta_k} (\mathcal{A}(x) - p(x)) \right) \\ &= \nabla f(x) + \frac{1}{\lambda + \beta_k} \nabla \mathcal{A}(x)^\top (\mathcal{A}(x) + \beta_k y_k - \text{prox}_{(\lambda + \beta_k)h^*}(\mathcal{A}(x) + \beta_k y_k)), \end{aligned}$$

which, together with (18) and the Moreau decomposition (8), implies that

$$\nabla_x \tilde{\mathcal{L}}_{1/\beta_k}(x, p(x); y_k) = \nabla \Phi_k(x).$$

TABLE 2. Average performance comparison on SPCA.

	RADA-RGD					RALM-II					ManPG-Ada				
	$-\Phi$	var	iter	cpu	spar	$-\Phi$	var	iter	cpu	spar	$-\Phi$	var	iter	cpu	spar
μ	Synthetic, $d = 1000, N = 50, r = 10$														
0.5	859.3	0.991	636	1.9	26.6	859.0	0.991	26	5.4	26.7	859.1	0.991	18366	11.1	26.5
0.75	810.2	0.979	2288	1.3	35.5	809.9	0.978	28	5.3	35.7	810.1	0.978	19581	12.0	35.5
1	765.8	0.965	494	1.5	42.1	765.5	0.964	30	5.2	42.3	765.4	0.964	24640	14.5	42.3
1.25	724.2	0.952	569	1.7	47.4	723.8	0.951	33	5.3	47.4	724.0	0.951	18271	12.0	47.4
1.5	685.8	0.936	527	1.6	51.8	685.7	0.936	36	5.4	51.9	685.3	0.936	13713	9.1	51.9
r	Synthetic, $d = 2000, N = 50, \mu = 3$														
8	1163.1	0.919	532	2.5	46.1	1161.6	0.923	61	16.2	47.6	1161.0	0.915	15004	12.4	49.2
10	1195.2	0.904	624	3.7	59.3	1193.6	0.896	62	21.1	59.8	1190.0	0.885	16399	19.2	61.0
12	1189.7	0.890	816	5.5	65.1	1188.7	0.891	70	29.6	64.3	1188.6	0.890	12614	22.6	65.2
14	1145.9	0.867	698	5.7	69.8	1144.5	0.869	64	31.5	69.0	1144.6	0.865	15179	46.8	69.9
16	1247.8	0.895	739	7.0	75.6	1245.0	0.892	50	28.5	76.1	1240.1	0.894	23799	98.3	76.0
μ	coil-100, $d = 1024, N = 100, r = 10$														
1	531.0	0.960	268	1.1	47.2	530.8	0.959	34	5.6	47.4	530.4	0.959	17205	11.9	47.3
1.25	488.5	0.940	234	1.0	55.3	487.8	0.940	35	4.9	55.2	487.6	0.940	11865	8.9	55.2
1.5	453.5	0.923	266	1.2	61.3	451.1	0.917	39	5.1	61.7	448.5	0.911	11828	8.7	61.7
1.75	419.7	0.906	319	1.4	66.4	417.2	0.898	41	5.7	66.5	411.5	0.885	10475	7.7	66.5
2	388.2	0.890	344	1.5	70.6	380.5	0.866	43	5.6	70.2	373.7	0.848	10950	7.9	70.4
r	coil-100, $d = 1024, N = 100, \mu = 2$														
8	384.8	0.913	254	0.9	63.2	380.3	0.898	40	3.9	63.5	375.2	0.883	16313	7.1	64.2
10	388.2	0.890	344	1.5	70.6	380.5	0.866	43	5.7	70.2	373.7	0.848	10950	7.9	70.4
12	403.3	0.875	654	3.1	76.0	395.7	0.848	40	5.8	75.6	391.5	0.837	10463	10.7	75.7
14	399.9	0.860	636	3.5	80.5	391.2	0.829	41	8.1	79.3	384.5	0.814	14758	22.6	79.4
16	398.7	0.842	640	3.8	83.3	388.7	0.808	38	8.2	82.9	385.4	0.800	12004	25.3	82.8

Therefore, noting that $p(x_{k,t}) = p_{k,t}$, we can rewrite (78) as

$$x_{k,t+1} = R_{x_{k,t}}(-\zeta_{k,t} \text{grad } \Phi_k(x_{k,t})),$$

which is exactly the update of x in RADA-RGD. \blacksquare

6. Numerical Results In this section, we report the numerical results of our proposed RADA algorithmic framework for solving SPCA, FPCA, and SSC problems. All the tests are implemented in MATLAB 2023b and evaluated on Apple M2 Pro CPU. Our code is available at <https://github.com/XuMeng00124/RADAopt>.

Before reporting the results, we specify the default choices of some parameters in our tests. We choose $\zeta_{\min} = 10^{-20}$, $\zeta_{\max} = 10^{20}$, $\tau_1 = 0.999$, $\tau_2 = 0.9$, $\rho = 1.5$, and $\lambda = \varepsilon/(2R)$ for RADA-PGD and RADA-RGD. We also set $\lambda = \varepsilon/(2R)$ for ARPGDA. For RADA-RGD, we set $\nu_k = 2T_k R^2 \beta_k$, $c_1 = 10^{-4}$, and $\eta = 0.1$. The value of R can be easily computed for each specific problem and is given by $\mu\sqrt{dr}$, 1, and μN for problems (3), (4), and (6), respectively. The initial setting of the parameter β_1 varies across different problem settings. For DSGM, ARPGDA, and RADMM, we apply the same line search strategy initialized with the BB stepsize as in RADA-RGD for updating x . Unless otherwise stated, we terminate RADA-PGD and RADA-RGD if they return an ε -RGS point. The stopping rules for the other algorithms will be specified later. We utilize the QR

factorization and the standard projection as the retraction on the Stiefel manifold and the Grassmann manifold, respectively [1, 2, 6, 63]. The standard projection onto the Grassmann manifold is obtained by taking the m leading eigenvectors of the matrix to be projected. The Riemannian gradient on the Grassmann manifold is computed as in [63]. Finally, both the QR-based retraction on the Stiefel manifold and the projection-based retraction on the Grassmann manifold are globally defined for all tangent vectors. Therefore, Assumption 4 holds with $\varrho = +\infty$ for both manifolds.

6.1. Results on SPCA In this subsection, we compare our proposed RADA-RGD with RALM-II (i.e., LS-II in [83]) and ManPG-Ada [11], a more efficient version of ManPG, on synthetic datasets and the real dataset coil-100 [54], which contains 7200 image samples of 100 objects taken from different angles with $d = 1024$. For RADA-RGD, we set $\beta_1 = 0.1d\sqrt{r}$ and $T_k \equiv 10$. The codes for RALM-II and ManPG-Ada are downloaded from <https://github.com/miskcoo/almssn> and <https://github.com/chenshixiang/ManPG>, respectively. For ManPG-Ada, we relax the line search parameter from 0.5 to 10^{-4} and adopt a nonmonotone line search strategy to improve stepsize selection and empirical performance. We terminate all three methods once an ε -ROS point is found. More specifically, we terminate RADA-RGD when

$$\max \left\{ \text{dist} \left(0, \text{grad } f(x_k) + \text{proj}_{T_{x_k} \mathcal{M}} \left(\nabla \mathcal{A}(x_k)^\top \partial h^*(p_k) \right) \right), \|p_k - \mathcal{A}(x_k)\| \right\} \leq \varepsilon, \quad (81)$$

where $p_k = y_{k+\frac{1}{2}}$ is defined in (20). For RALM-II, we apply the same criterion (81) but with $p_k = y_k$. Recall that ManPG solves the subproblem

$$V_k := \arg \min_{V \in T_{x_k} \mathcal{M}} \left\{ \langle \text{grad } f(x_k), V \rangle + \frac{\ell_f}{2} \|V\|^2 + h^*(x_k + V) \right\}$$

at each iteration. Here, ℓ_f is the Lipschitz constant of ∇f . We terminate ManPG-Ada if $\max\{\ell_f \|V_k\|, \|V_k\|\} \leq \varepsilon$. The maximum number of iterations is set as 50000. In each test, we generate the synthetic data matrix A following the method in [83] or randomly select 100 samples from the coil-100 dataset.

First, we set $\varepsilon = 10^{-4}$. The average results over 20 runs, each using either a different randomly generated synthetic matrix A or a different set of selected samples from the coil-100 dataset, are reported in Table 2. Here, Φ is defined in (21), ‘‘cpu’’ represents the cpu time in seconds, and ‘‘iter’’ denotes the outer iteration number. We also compare the normalized variance (denoted by ‘‘var’’) and the sparsity of X (denoted by ‘‘spar’’). The former is defined as $\langle AA^\top, \hat{X} \hat{X}^\top \rangle / \max_{X \in \mathcal{S}(d,r)} \langle AA^\top, XX^\top \rangle$, where \hat{X} denotes the returned solution by the algorithm. The latter is defined as the percentage of entries with the absolute value less than 10^{-5} .

From Table 2, we see that the solutions obtained by these algorithms are comparable in terms of the normalized variance and sparsity. Our proposed RADA-RGD always returns the best solution in terms of the value of Φ . Moreover, the proposed single-loop RADA-RGD algorithm is significantly more efficient than the compared nested-loop algorithms. The performance difference between RADA-RGD and RALM-II is consistent with the analysis in Section 5. Specifically, although RALM-II requires significantly fewer iterations compared to RADA-RGD, it requires a much higher level of accuracy than RADA-RGD when solving the subproblem. Consequently, RALM-II is less efficient than RADA-RGD for solving the SPCA problems considered.

Next, we set $\varepsilon = 10^{-8}$ and tighten the subproblem stopping criterion of ManPG-Ada to 10^{-10} (otherwise, the stationarity gap stagnates and fails to reach the same accuracy level as RADA-RGD and RALM-II) to observe the stationarity gap versus the runtime of two representative instances in Figure 1. To ensure a fair and readable comparison, we record the runtime of RALM-II at

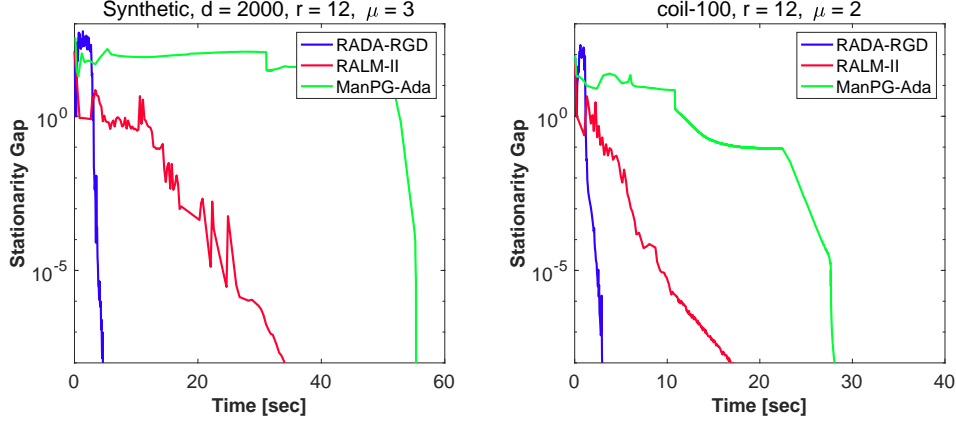


FIGURE 1. Stationarity gap versus runtime for two representative instances of SPCA.

each outer iteration and that of RADA-RGD and ManPG-Ada every 10 iterations. This is because RALM-II involves the fewest but most expensive outer iterations. As shown in Figure 1, RADA-RGD reduces the stationarity gap significantly faster than both RALM-II and ManPG-Ada, which is consistent with the results reported in Table 2.

6.2. Results on FPCA For the FPCA problem (4), we generate the data points $\{a_i\}_{i=1}^N$ independently according to the standard Gaussian distribution as in [74] and divide them into $m = N$ groups. We compare RADA-RGD with the single-loop algorithms ARPGDA [74], RADMM [40], and DSGM [5]. Note that the existing convergence guarantees for RADMM and DSGM do not apply to FPCA since the mapping \mathcal{A} is nonlinear. For RADA-RGD, we set $\beta_1 = 10^4 m^2 \sqrt{r}$, $T_k \equiv 5$, and $\varepsilon = 10^{-8}$. For ARPGDA, we set $\beta_k = 10^3 m^2 \sqrt{r} / k^{1.5}$. For DSGM and RADMM, we set the smoothing parameter $\lambda_k = 10/k^{1/3}$. For RADMM, a significant effort is devoted to fine-tuning the penalty parameter in the augmented Lagrangian function, and the choice $\sigma_k = 10^{-7} k^{1.5}$ is found to work well in our tests. It should be noted that although this choice of σ_k violates the convergence condition in [40], the resulting RADMM demonstrates superior numerical performance. For the compared algorithms, a more relaxed criterion is employed: The algorithm is terminated if the improvement in Φ is less than 10^{-8} over 1000 consecutive iterations. The maximum number of iterations for all algorithms is set as 20000.

Table 3 presents the results on FPCA averaged over 20 runs with different randomly generated initial points. From this table, we observe that our proposed RADA-RGD consistently returns the best solutions in terms of the value of Φ while requiring the least computational time. Specifically, on average, our proposed RADA-RGD is more than 10 times faster than the other compared algorithms when d is large. We further plot the stationarity gap versus the number of gradient evaluations for two representative instances in Figure 2. As illustrated in the figure, RADA-RGD achieves a faster reduction in the stationarity gap than the other first-order methods. Moreover, the similarities and differences between RADA-RGD and RADMM, as discussed in Section 5.2, provide important insights into the superior performance of RADA-RGD. In particular, as shown in Algorithm 4, an additional update of p is performed before updating x , which means that the Riemannian gradient used to update x in RADA-RGD involves the most up-to-date information of p . The superior performance of RADA-RGD over ARPGDA further demonstrates that performing multiple Riemannian gradient steps to ensure sufficient descent of the proposed value function Φ_k is much more efficient than simply performing a Riemannian gradient descent step on certain surrogate of $F(\cdot, y)$ for some fixed y , as is done in ARPGDA and its Euclidean counterparts [20, 53, 58, 76].

TABLE 3. Average performance comparison on FPCA.

	RADA-RGD			ARPGDA			RADMM			DSGM		
	$-\Phi$	cpu	iter	$-\Phi$	cpu	iter	$-\Phi$	cpu	iter	$-\Phi$	cpu	iter
N	$d = 1000, r = 2$											
20	120.54	0.01	64	118.08	0.24	7715	120.18	0.20	6144	113.38	0.16	3123
25	98.88	0.01	77	97.00	0.24	6708	97.94	0.24	6885	97.01	0.19	3599
30	84.16	0.02	85	82.29	0.22	5764	83.11	0.30	7858	82.32	0.24	4276
35	73.69	0.02	102	72.16	0.33	7833	72.80	0.27	6289	69.74	0.24	3836
40	65.40	0.02	98	64.17	0.32	7257	64.77	0.25	5814	60.24	0.25	3824
d	$N = 20, r = 4$											
200	56.86	0.01	87	56.51	0.02	1579	56.59	0.02	1178	53.73	0.06	1890
400	104.24	0.01	73	103.29	0.07	3174	103.08	0.11	4771	103.39	0.11	2420
600	146.57	0.01	62	145.55	0.32	9975	146.20	0.22	6717	140.12	0.20	3821
800	193.26	0.01	53	191.28	0.38	9094	192.64	0.31	7248	192.25	0.32	5244
1000	237.18	0.02	75	235.59	0.51	10124	236.65	0.38	7423	224.79	0.38	5270

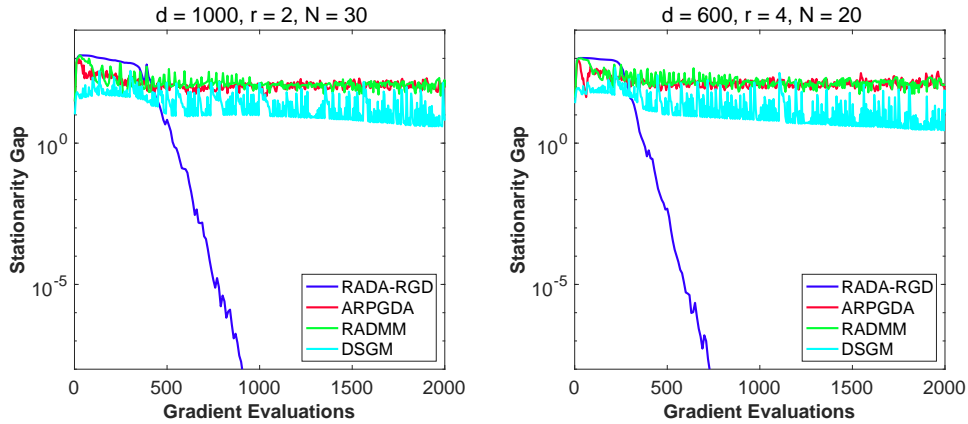


FIGURE 2. Stationarity gap versus number of gradient evaluations for two representative instances of FPCA.

6.3. Results on SSC In this subsection, we compare our proposed RADA-PGD and RADA-RGD with ARPGDA [74], DSGM [5], and NADMM [51] for solving the SSC problem (5). Note that NADMM is specifically designed for solving (5). The main difference between NADMM and RADMM is that NADMM performs a Euclidean projected gradient descent step to update x .

First, we perform tests on synthetic datasets, wherein the data points $\{a_i\}_{i=1}^N$ are independently generated according to the standard Gaussian distribution and $W_{ij} = |\langle a_i, a_j \rangle|$ as in [52]. We take $Q_1 = X_1 X_1^\top$ as the initial point, where X_1 consists of the m eigenvectors associated to the m smallest eigenvalues of L . For RADA, we set $\varepsilon = 10^{-4}$. For RADA-PGD, we set $\beta_1 = N^2 \sqrt{m}$, $T_k \equiv 1$, and $\zeta_{k,t} = \ell_k^{-1}$ with $\ell_k = (\lambda + \beta_k)^{-1}$. For RADA-RGD, we set $\beta_1 = N^2 \sqrt{m}$ and $T_k \equiv 3$. For DSGM and NADMM, we set $\lambda_k = 10^{-2}/k^{1/3}$. Additionally, we carefully tune the parameter β_k in ARPGDA for each test to achieve its best possible performance. For NADMM, we set the penalty parameter σ_k in the augmented Lagrangian function as $0.1k^{1/3}$, which enhances the efficiency of NADMM in our tests but does not satisfy the conditions of the theoretical guarantee in [51]. For the compared algorithms, a more relaxed criterion is employed: The algorithm is terminated if the improvement of the objective function Φ is less than 10^{-8} over 50 consecutive iterations.

TABLE 4. Average performance comparison on SSC (synthetic datasets).

m	RADA-PGD			RADA-RGD			ARPGDA			NADMM			DSGM		
	Φ	cpu	iter	Φ	cpu	iter	Φ	cpu	iter	Φ	cpu	iter	Φ	cpu	iter
$N = 200, \mu = 0.005$															
2	1.976	0.1	72	1.976	0.2	67	1.980	0.6	580	1.978	0.6	838	1.983	8.0	4455
3	2.973	0.1	92	2.973	0.3	78	2.987	6.0	6494	2.977	0.8	1136	2.988	2.9	2168
4	3.970	0.2	104	3.970	0.2	77	3.988	5.6	6238	3.975	0.9	1324	3.989	4.5	3528
5	4.967	0.2	123	4.967	0.2	76	4.987	6.0	6446	4.973	1.2	1759	4.992	6.8	5375
6	5.965	0.2	141	5.965	0.3	86	5.987	5.8	6185	5.971	1.7	2421	5.994	9.8	7601
$N = 500, m = 3$															
0.001	2.496	1.0	116	2.496	3.0	158	2.500	22.2	3757	2.498	3.8	1244	2.504	73.5	9198
0.002	2.985	1.1	110	2.985	2.0	93	2.990	46.5	7294	2.986	1.9	531	2.996	25.9	3084
0.005	3.007	1.7	152	3.007	3.2	143	3.010	3.3	503	3.007	3.0	798	3.009	29.7	3282
0.01	3.022	1.9	165	3.022	3.5	152	3.031	12.2	1796	3.022	13.9	3480	3.024	16.5	1681
0.02	3.052	2.6	227	3.052	4.1	164	3.057	3.2	436	3.052	17.1	3704	3.052	13.3	1134

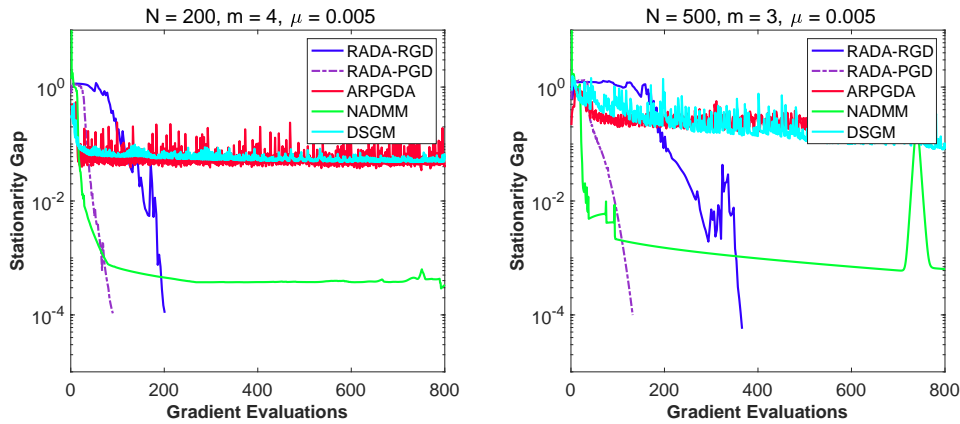


FIGURE 3. Stationarity gap versus number of gradient evaluations for two representative instances of SSC.

Table 4 reports the results on SSC averaged over 20 runs with randomly generated data points, and Figure 3 illustrates the stationarity gap versus the number of gradient evaluations for two representative instances. From both the table and the figure, we observe that the proposed RADA-PGD and RADA-RGD consistently achieve the best values of Φ and effectively reduce the stationarity gap. Notably, RADA-PGD is the most efficient among all compared algorithms. Similar to RADA-RGD, RADA-PGD with $T_k \equiv 1$ can also be interpreted as a Riemannian sGS-type ADMM. This interpretation sheds light on the superior performance of RADA-PGD over NADMM, given that sGS-type ADMM has demonstrated better performance than ADMM in many scenarios [10, 43, 44]. Furthermore, RADA-PGD demonstrates even better performance than RADA-RGD, especially when $N = 500$. This can be attributed to two reasons. First, RADA-PGD only needs to compute the Euclidean gradient instead of the Riemannian gradient, which requires an additional projection onto the tangent space. Second, the parameter ℓ_k in (45) becomes $(\lambda + \beta_k)^{-1}$ for this specific problem, making it straightforward to compute. Consequently, we can use $\ell_k^{-1} = \lambda + \beta_k$ directly as an appropriate stepsize at the k -th iteration without any additional cost.

Next, to further evaluate the different algorithms, we compare their clustering performance on four real datasets (i.e., Wine, Iris, Glass, and Letter) from the UCI machine learning repository

TABLE 5. Performance comparison on SSC (UCI datasets).

	RADA-PGD	RADA-RGD	ARPGDA	DSGM	NADMM
Wine: $N = 178, m = 3, \kappa = 1, \mu = 0.001$					
NMI	0.893	0.893	0.893	0.893	0.893
Φ	1.899	1.899	1.899	1.899	1.899
iter	148	115	932	1952	659
cpu	0.26	0.52	1.60	5.12	0.97
Iris: $N = 149, m = 3, \kappa = 0.2, \mu = 0.005$					
NMI	0.776	0.776	0.776	0.757	0.776
Φ	1.106	1.106	1.106	1.106	1.106
iter	161	115	320	1491	5995
cpu	0.21	0.44	0.59	3.17	4.82
Glass: $N = 213, m = 3, \kappa = 1, \mu = 0.0002$					
NMI	0.471	0.471	0.471	0.471	0.471
Φ	1.818	1.818	1.818	1.818	1.818
iter	120	63	268	8474	177
cpu	0.17	0.30	0.44	28.35	0.34
Letter: $N = 700, m = 12, \kappa = 1, \mu = 0.001$					
NMI	0.386	0.384	0.339	0.297	0.315
Φ	16.63	16.64	16.70	16.73	16.66
iter	271	222	4456	10000	10000
cpu	8.71	23.89	216.65	450.76	313.04

[35]. Due to the large size of the Letter dataset, we randomly select a subset of 1000 samples from the first 17 classes. The affinity matrix W is constructed using the Gaussian kernel as in [51], i.e., $W_{ij} = \exp(-\|a_i - a_j\|^2/\kappa)$. We fine-tune the parameters κ and μ for each dataset so that the solutions to the obtained optimization problems yield good clustering performance. We utilize the normalized mutual information (NMI) scores [66] to measure the clustering performance. Note that a higher NMI score indicates better clustering performance. The parameters and stopping criteria for all algorithms remain the same as those used in the tests on synthetic datasets, with the exception that $\varepsilon = 10^{-3}$ is used for both RADA-PGD and RADA-RGD. We also fine-tune the parameter β_k for ARPGDA in each test.

Table 5 reports the comparison of NMI scores, objective function values, and computational time for the four real datasets. From the table, we see that RADA-PGD and RADA-RGD consistently find solutions with the best quality in terms of NMI scores and objective function values for all datasets. For the Letter dataset, the proposed RADA-PGD enjoys a significantly lower computational time, while DSGM and NDAMM hit the preset maximum number of iterations of 10000.

7. Concluding Remarks In this paper, we introduced a flexible RADA algorithmic framework based on a novel value function for solving a class of Riemannian NC-L minimax problems. Within this framework, we proposed two customized efficient algorithms called RADA-PGD and RADA-RGD, which perform one or multiple projected/Riemannian gradient descent steps and then a proximal gradient ascent step at each iteration. Theoretically, we proved that the proposed algorithmic framework can find both ε -RGS and ε -ROS points of the Riemannian NC-L minimax problem within $\mathcal{O}(\varepsilon^{-3})$ iterations, achieving the best-known iteration complexity. We also showed how the proposed algorithmic framework can be adapted to solve an equivalent Riemannian non-smooth reformulation of the minimax problem. As a by-product, we made the intriguing observation that RADA and RADA-RGD are closely related to, but essentially different from, RALM and RADMM, respectively. Computationally, we demonstrated the superiority of our proposed algorithms compared to existing state-of-the-art algorithms through extensive numerical results on SPCA, FPCA, and SSC. It is worth noting that our proposed RADA algorithmic framework and

the associated algorithms are not derived from existing Euclidean methods; rather, the ideas behind their design even lead to new algorithms for NC-L minimax problems in the Euclidean setting.

Our work suggests several directions for future research. One direction is to design efficient algorithms with strong theoretical guarantees for solving more general Riemannian minimax problems. Another direction is to develop stochastic versions of our algorithms. It is also worthwhile to develop efficient algorithms for broader classes of Riemannian nonsmooth composite problems by suitably exploiting their intrinsic minimax structure.

Acknowledgments

The authors are grateful to the Area Editor, the Associate Editor, and the anonymous reviewers for their valuable comments and suggestions, which have helped improve the quality of the manuscript.

References

- [1] Absil PA, Mahony R, Sepulchre R (2008) *Optimization Algorithms on Matrix Manifolds* (Princeton Univ. Press).
- [2] Absil PA, Malick J (2012) Projection-like retractions on matrix manifolds. *SIAM J. Optim.* 22(1):135–158.
- [3] Barzilai J, Borwein JM (1988) Two-point step size gradient methods. *IMA J. Numer. Anal.* 8(1):141–148.
- [4] Beck A (2017) *First-Order Methods in Optimization* (SIAM).
- [5] Beck A, Rosset I (2023) A dynamic smoothing technique for a class of nonsmooth optimization problems on manifolds. *SIAM J. Optim.* 33(3):1473–1493.
- [6] Bendokat T, Zimmermann R, Absil PA (2024) A Grassmann manifold handbook: Basic geometry and computational aspects. *Adv. Comput. Math.* 50(1):1–51.
- [7] Borckmans PB, Selvan SE, Boumal N, Absil PA (2014) A Riemannian subgradient algorithm for economic dispatch with valve-point effect. *J. Comput. Appl. Math.* 255:848–866.
- [8] Boumal N (2023) *An Introduction to Optimization on Smooth Manifolds* (Cambridge Univ. Press).
- [9] Boumal N, Absil PA, Cartis C (2019) Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.* 39(1):1–33.
- [10] Chen L, Sun D, Toh KC (2017) An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Math. Program.* 161:237–270.
- [11] Chen S, Ma S, So AMC, Zhang T (2020) Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM J. Optim.* 30(1):210–239.
- [12] Chen S, Ma S, So AMC, Zhang T (2024) Nonsmooth optimization over the Stiefel manifold and beyond: Proximal gradient method and recent variants. *SIAM Rev.* 66(2):319–352.
- [13] Deng K, Hu J, Wu J, Wen Z (2025) Oracle complexities of augmented Lagrangian methods for nonsmooth manifold optimization. *Math. Oper. Res.* Doi: [10.1287/moor.2024.0498](https://doi.org/10.1287/moor.2024.0498).
- [14] Deng K, Peng Z (2023) A manifold inexact augmented Lagrangian method for nonsmooth optimization on Riemannian submanifolds in Euclidean space. *IMA J. Numer. Anal.* 43(3):1653–1684.
- [15] Drusvyatskiy D, Paquette C (2019) Efficiency of minimizing compositions of convex functions and smooth maps. *Math. Program.* 178:503–558.
- [16] Gao B, Liu X, Chen X, Yuan Y (2018) A new first-order algorithmic framework for optimization problems with orthogonality constraints. *SIAM J. Optim.* 28(1):302–332.
- [17] Hajinezhad D, Hong M (2019) Perturbed proximal primal-dual algorithm for nonconvex nonsmooth optimization. *Math. Program.* 176(1):207–245.
- [18] Han A, Mishra B, Jawanpuria P, Gao J (2023) Nonconvex-nonconcave min-max optimization on Riemannian manifolds. *Trans. Mach. Learn. Res.* .
- [19] Han A, Mishra B, Jawanpuria P, Kumar P, Gao J (2023) Riemannian Hamiltonian methods for min-max optimization on manifolds. *SIAM J. Optim.* 33(3):1797–1827.

- [20] He J, Zhang H, Xu Z (2024) An approximation proximal gradient algorithm for nonconvex-linear minimax problems with nonconvex nonsmooth terms. *J. Glob. Optim.* 90:73–92.
- [21] Hosseini S, Huang W, Yousefpour R (2018) Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM J. Optim.* 28(1):596–619.
- [22] Hosseini S, Uschmajew A (2017) A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM J. Optim.* 27(1):173–189.
- [23] Hu J, Liu X, Wen Z, Yuan Y (2020) A brief introduction to manifold optimization. *J. Oper. Res. Soc. China* 8:199–248.
- [24] Hu J, Milzarek A, Wen Z, Yuan Y (2018) Adaptive quadratically regularized Newton method for Riemannian optimization. *SIAM Journal on Matrix Analysis and Applications* 39(3):1181–1207.
- [25] Hu X, Xiao N, Liu X, Toh KC (2024) A constraint dissolving approach for nonsmooth optimization over the Stiefel manifold. *IMA J. Numer. Anal.* 44(6):3717–3748.
- [26] Huang F, Gao S (2023) Gradient descent ascent for minimax problems on Riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.* 45(7):8466–8476.
- [27] Huang W, Wei K (2022) Riemannian proximal gradient methods. *Math. Program.* 194(1):371–413.
- [28] Huang W, Wei K (2023) An inexact Riemannian proximal gradient method. *Comput. Optim. Appl.* 85(1):1–32.
- [29] Iannazzo B, Porcelli M (2018) The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA J. Numer. Anal.* 38(1):495–517.
- [30] Jiang B, Dai YH (2015) A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Math. Program.* 153(2):535–575.
- [31] Jiang B, Liu YF (2023) A Riemannian exponential augmented Lagrangian method for computing the projection robust Wasserstein distance. *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, 79999–80023.
- [32] Jin C, Netrapalli P, Jordan M (2020) What is local optimality in nonconvex-nonconcave minimax optimization? *Proc. Int. Conf. Mach. Learn.*, 4880–4889 (PMLR).
- [33] Jolliffe IT, Trendafilov NT, Uddin M (2003) A modified principal component technique based on the LASSO. *J. Comput. Graph. Stat.* 12(3):531–547.
- [34] Jordan M, Lin T, Vlatakis-Gkaragkounis EV (2022) First-order algorithms for min-max optimization in geodesic metric spaces. *Proc. Adv. Neural Inf. Process. Syst.*, volume 35, 6557–6574.
- [35] Kelly M, Longjohn R, Nottingham K (2007) The UCI Machine Learning Repository. <https://archive.ics.uci.edu>.
- [36] Kong W, Monteiro RDC (2021) An accelerated inexact proximal point method for solving nonconvex-concave min-max problems. *SIAM J. Optim.* 31(4):2558–2585.
- [37] Kong W, Monteiro RDC (2024) Global complexity bound of a proximal ADMM for linearly constrained non-separable nonconvex composite programming. *SIAM J. Optim.* 34(1):201–224.
- [38] Kovnatsky A, Glashoff K, Bronstein MM (2016) MADMM: A generic algorithm for non-smooth optimization on manifolds. *Proc. Comput. Vis. ECCV*, 680–696 (Springer).
- [39] Lai R, Osher S (2014) A splitting method for orthogonality constrained problems. *J. Sci. Comput.* 58:431–449.
- [40] Li J, Ma S, Srivastava T (2024) A Riemannian alternating direction method of multipliers. *Math. Oper. Res.* 50(4):3222–3242.
- [41] Li J, Zhu L, So AMC (2025) Nonsmooth nonconvex-nonconcave minimax optimization: Primal-dual balancing and iteration complexity analysis. *Math. Program.* 214:591–641.
- [42] Li X, Chen S, Deng Z, Qu Q, Zhu Z, So AMC (2021) Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM J. Optim.* 31(3):1605–1634.
- [43] Li X, Sun D, Toh KC (2016) A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions. *Math. Program.* 155(1):333–373.

- [44] Li X, Sun D, Toh KC (2019) A block symmetric Gauss-Seidel decomposition theorem for convex composite quadratic programming and its applications. *Math. Program.* 175:395–418.
- [45] Lin T, Jin C, Jordan M (2020) On gradient descent ascent for nonconvex-concave minimax problems. *Proc. Int. Conf. Mach. Learn.*, 6083–6093 (PMLR).
- [46] Lin T, Jin C, Jordan MI (2020) Near-optimal algorithms for minimax optimization. *Proc. Conf. Learn. Theory*, 2738–2779 (PMLR).
- [47] Liu C, Boumal N (2020) Simple algorithms for optimization on Riemannian manifolds with constraints. *Appl. Math. Optim.* 82(3):949–981.
- [48] Liu H, So AMC, Wu W (2019) Quadratic optimization with orthogonality constraint: Explicit Łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. *Math. Program.* 178(1):215–262.
- [49] Liu X, Xiao N, Yuan Y (2024) A penalty-free infeasible approach for a class of nonsmooth optimization problems over the Stiefel manifold. *J. Sci. Comput.* 99(2):30.
- [50] Liu YF, Chang TH, Hong M, Wu Z, So AMC, Jorswieck EA, Yu W (2024) A survey of recent advances in optimization methods for wireless communications. *IEEE J. Sel. Areas Commun.* 42(11):2992–3031.
- [51] Lu C, Feng J, Lin Z, Yan S (2018) Nonconvex sparse spectral clustering by alternating direction method of multipliers and its convergence analysis. *Proc. AAAI Conf. Artif. Intell.*, volume 32.
- [52] Lu C, Yan S, Lin Z (2016) Convex sparse spectral clustering: Single-view to multi-view. *IEEE Trans. Image Process.* 25(6):2833–2843.
- [53] Lu S, Tsaknakis I, Hong M, Chen Y (2020) Hybrid block successive approximation for one-sided non-convex min-max problems: Algorithms and applications. *IEEE Trans. Signal Process.* 68:3676–3691.
- [54] Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-100). *Technical Report CUCS-006-96*.
- [55] Nesterov Y (2018) *Lectures on Convex Optimization*, volume 137 (Springer).
- [56] Nouiehed M, Sanjabi M, Huang T, Lee JD, Razaviyayn M (2019) Solving a class of non-convex min-max games using iterative first order methods. *Proc. Adv. Neural Inf. Process. Syst.*, volume 32, 14934–14942.
- [57] Ostrovskii DM, Lowy A, Razaviyayn M (2021) Efficient search of first-order Nash equilibria in nonconvex-concave smooth min-max problems. *SIAM J. Optim.* 31(4):2508–2538.
- [58] Pan W, Shen J, Xu Z (2021) An efficient algorithm for nonconvex-linear minimax optimization problem and its application in solving weighted maximin dispersion problem. *Comput. Optim. Appl.* 78:287–306.
- [59] Peng Z, Wu W, Hu J, Deng K (2023) Riemannian smoothing gradient type algorithms for nonsmooth optimization problem on compact Riemannian submanifold embedded in Euclidean space. *Appl. Math. Optim.* 88(3):85.
- [60] Rafique H, Liu M, Lin Q, Yang T (2022) Weakly-convex-concave min-max optimization: Provable algorithms and applications in machine learning. *Optim. Methods Softw.* 37(3):1087–1121.
- [61] Rockafellar RT, Wets RJB (2009) *Variational Analysis*, volume 317 (Springer Sci. Business Media).
- [62] Samadi S, Tantipongpipat U, Morgenstern JH, Singh M, Vempala S (2018) The price of fair PCA: One extra dimension. *Proc. Adv. Neural Inf. Process. Syst.*, volume 31, 10999–11010.
- [63] Sato H, Iwai T (2014) Optimization algorithms on the Grassmann manifold with application to matrix eigenvalue problems. *Jpn. J. Ind. Appl. Math.* 31:355–400.
- [64] Shen J, Davis AJ, Lu D, Bai Z (2025) Hidden convexity of fair PCA and fast solver via eigenvalue optimization. *arXiv prepr. arXiv:2503.00299*.
- [65] Shen J, Wang Z, Xu Z (2023) Zeroth-order single-loop algorithms for nonconvex-linear minimax problems. *J. Glob. Optim.* 87(2):551–580.
- [66] Strehl A, Ghosh J (2002) Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3:583–617.
- [67] Sun K, Sun XA (2024) Dual descent augmented Lagrangian method and alternating direction method of multipliers. *SIAM J. Optim.* 34(2):1679–1707.

- [68] Thekumparampil KK, Jain P, Netrapalli P, Oh S (2019) Efficient algorithms for smooth minimax optimization. *Proc. Adv. Neural Inf. Process. Syst.*, volume 32, 12659–12670.
- [69] Tian L, So AMC (2024) No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians. *Math. Program.* 208(1):51–74.
- [70] Wang Z, Liu B, Chen S, Ma S, Xue L, Zhao H (2022) 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.
- [71] Wen Z, Yin W (2013) A feasible method for optimization with orthogonality constraints. *Math. Program.* 142(1):397–434.
- [72] Wu Z, Jiang B, Liu YF, Shao M, Dai YH (2024) Efficient CI-based one-bit precoding for multiuser downlink massive MIMO systems with PSK modulation. *IEEE Trans. Wireless Commun.* 23(5):4861–4875.
- [73] Xiao N, Liu X, Yuan Y (2021) Exact penalty function for $\ell_{2,1}$ norm minimization over the Stiefel manifold. *SIAM J. Optim.* 31(4):3097–3126.
- [74] Xu M, Jiang B, Pu W, Liu YF, So AMC (2024) An efficient alternating Riemannian/projected gradient descent ascent algorithm for fair principal component analysis. *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 7195–7199.
- [75] Xu Z, Wang Z, Shen J, Dai Y (2024) Derivative-free alternating projection algorithms for general nonconvex-concave minimax problems. *SIAM J. Optim.* 34(2):1879–1908.
- [76] Xu Z, Zhang H, Xu Y, Lan G (2023) A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. *Math. Program.* 201(1):635–706.
- [77] Yang J, Orvieto A, Lucchi A, He N (2022) Faster single-loop algorithms for minimax optimization without strong concavity. *Proc. Int. Conf. Artif. Intell. Stat.*, 5485–5517 (PMLR).
- [78] Yang J, Zhang S, Kiyavash N, He N (2020) A Catalyst framework for minimax optimization. *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 5667–5678.
- [79] Zalcberg G, Wiesel A (2021) Fair principal component analysis and filter design. *IEEE Trans. Signal Process.* 69:4835–4842.
- [80] Zhang C, Chen X, Ma S (2024) A Riemannian smoothing steepest descent method for non-Lipschitz optimization on embedded submanifolds of \mathbb{R}^n . *Math. Oper. Res.* 49(3):1710–1733.
- [81] Zhang J, Xiao P, Sun R, Luo Z (2020) A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems. *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 7377–7389.
- [82] Zhang P, Zhang J, Sra S (2023) Sion’s minimax theorem in geodesic metric spaces and a Riemannian extragradient algorithm. *SIAM J. Optim.* 33(4):2885–2908.
- [83] Zhou Y, Bao C, Ding C, Zhu J (2023) A semismooth Newton based augmented Lagrangian method for nonsmooth optimization on matrix manifolds. *Math. Program.* 201(1):1–61.
- [84] Zou H, Xue L (2018) A selective overview of sparse principal component analysis. *Proc. IEEE* 106(8):1311–1320.