Manifold Proximal Point Algorithms for Dual Principal Component Pursuit and Orthogonal Dictionary Learning[*]

Shixiang Chen[†] Zengde Deng[‡] Shiqian Ma[§] Anthony Man-Cho So[¶]

July 21, 2021

Abstract

We consider the problem of minimizing the ℓ_1 norm of a linear map over the sphere, which arises in various machine learning applications such as orthogonal dictionary learning (ODL) and robust subspace recovery (RSR). The problem is numerically challenging due to its nonsmooth objective and nonconvex constraint, and its algorithmic aspects have not been well explored. In this paper, we show how the manifold structure of the sphere can be exploited to design fast algorithms with provable guarantees for tackling this problem. Specifically, our contribution is fourfold. First, we present a manifold proximal point algorithm (ManPPA) for the problem and show that it converges at a global sublinear rate. Furthermore, we show that ManPPA can achieve a local quadratic convergence rate when applied to sharp instances of the problem. Second, we develop a semismooth Newton-based inexact augmented Lagrangian method for computing the search direction in each iteration of ManPPA and show that it has an asymptotic superlinear convergence rate. Third, we propose a stochastic variant of ManPPA called StManPPA, which is well suited for large-scale computation, and establish its sublinear convergence rate. Both ManPPA and StManPPA have provably faster convergence rates than existing subgradient-type methods. Fourth, using ManPPA as a building block, we propose a new heuristic method for solving a matrix analog of the problem, in which the sphere is replaced by the Stiefel manifold. The results from our extensive numerical experiments on the ODL and RSR problems demonstrate the efficiency and efficacy of our proposed methods.

1 Introduction

The problem of finding a subspace that captures the features of a given dataset and possesses certain properties is at the heart of many machine learning applications. One commonly encountered formulation of the problem, which is motivated largely by sparsity or robustness considerations, is given by

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) := \|\boldsymbol{Y}^\top \boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 = 1, \tag{1.1}$$

[*] A short version of this paper appeared in the 53rd Annual Asilomar Conference on Signals, Systems, and Computers, Nov. 3–6, 2019.

[†] Department of Industrial & Systems Engineering, Texas A&M University, College Station, TX, USA. Email: sxchen@tamu.edu

[‡] Cainiao Network, Hangzhou, China. Email: dengzengde@gmail.com

[§] Department of Mathematics, University of California, Davis, CA, USA. Research supported in part by NSF grants DMS-1953210 and CCF-2007797, and UC Davis CeDAR Innovative Data Science Seed Funding Program. Email: sqma@ucdavis.edu

[¶] Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China. Research supported in part by the Hong Kong RGC GRF Project CUHK 14203218 and in part by the CUHK Research Sustainability of Major RGC Funding Schemes Project 3133236. Email: manchoso@se.cuhk.edu.hk where $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ is a given matrix and $\|\cdot\|_r$ denotes the ℓ_r norm of a vector. To better understand how problem (1.1) arises in applications, let us consider two representative examples.

- **Orthogonal Dictionary Learning (ODL).** The goal of ODL is to find an orthonormal basis that can compactly represent a given set of p ($p \gg n$) data points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p \in \mathbb{R}^n$. Such a problem arises in many signal and image processing applications; see, e.g., [4, 24] and the references therein. By letting $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p] \in \mathbb{R}^{n\times p}$, the problem can be understood as finding an orthogonal matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ and a sparse matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ such that $\boldsymbol{Y} \approx \boldsymbol{X}\boldsymbol{A}$. Noting that this means $\boldsymbol{X}^\top \boldsymbol{Y} \approx \boldsymbol{A}$ should be sparse, one approach is to find a collection of sparse vectors from the row space of $\boldsymbol{Y}$ and apply some post-processing procedure to the collection to form the orthogonal matrix $\boldsymbol{X}$. This has been pursued in various works; see, e.g., [25, 19, 26, 2]. In particular, the work [2] considers the formulation (1.1) and shows that under a standard generative model of the data, one can recover $\boldsymbol{X}$ from certain local minimizers of problem (1.1).

- **Robust Subspace Recovery (RSR).** RSR is a fundamental problem in machine learning and data mining [12]. It is concerned with fitting a linear subspace to a dataset corrupted by outliers. Specifically, given a dataset $\boldsymbol{Y} = [\boldsymbol{X}, \boldsymbol{O}]\boldsymbol{\Gamma} \in \mathbb{R}^{n\times(p_1+p_2)}$, where the columns of $\boldsymbol{X} \in \mathbb{R}^{n\times p_1}$ are the inlier points spanning a d-dimensional subspace $\mathcal{S}$ of $\mathbb{R}^n$ ($d < p_1$), the columns of $\boldsymbol{O} \in \mathbb{R}^{n\times p_2}$ are outlier points without a linear structure, and $\boldsymbol{\Gamma} \in \mathbb{R}^{(p_1+p_2)\times(p_1+p_2)}$ is an unknown permutation, the goal is to recover the inlier subspace $\mathcal{S}$, or equivalently, to cluster the points into inliers and outliers. One recently proposed approach for solving this problem is the so-called dual principal component pursuit (DPCP) [28, 33]. A key task in DPCP is to find a hyperplane that contains all the inliers. Such a task can be tackled by solving problem (1.1). In fact, it has been shown in [28, 33] that under certain conditions on the inliers and outliers, any global minimizer of problem (1.1) yields a vector that is orthogonal to the inlier subspace $\mathcal{S}$.

Despite its attractive theoretical properties in various applications, problem (1.1) is numerically challenging to solve due to its nonsmooth objective and nonconvex constraint. Nevertheless, the manifold structure of the constraint set $\mathcal{M} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_2 = 1\}$ suggests that problem (1.1) could be amenable to manifold optimization techniques [1]. One approach is to apply smoothing to the nonsmooth objective in (1.1) and use existing algorithms for Riemannian smooth optimization to solve the resulting problem. For instance, when tackling the ODL problem, Sun et al. [26, 27] and Gilboa et al. [11] proposed to replace the absolute value function $t \mapsto |t|$ by the smooth surrogate $t \mapsto h_\mu(t) = \mu \log(\cosh(t/\mu))$ with $\mu > 0$ being a smoothing parameter, while Qu et al. [20] proposed to replace the ℓ_1 norm with the ℓ_4 norm. They then solve the resulting smoothed problems by either the Riemannian trust-region method [26, 27] or the Riemannian gradient descent method [11, 20]. Although it can be shown that these methods will yield the desired orthonormal basis under a standard generative model of the data, the smoothing approach can introduce significant analytic and computational difficulties [2]. Another approach, which avoids smoothing the objective, is to solve (1.1) directly using Riemannian nonsmooth optimization techniques. For instance, in the recent work [2], Bai et al. proposed to solve (1.1) using the Riemannian subgradient method (RSGM), which generates the iterates via

$$\boldsymbol{x}^{k+1} = \frac{\boldsymbol{x}^k - \eta_k \boldsymbol{v}^k}{\|\boldsymbol{x}^k - \eta_k \boldsymbol{v}^k\|_2}, \quad \boldsymbol{v}^k \in \partial_R f(\boldsymbol{x}^k). \tag{1.2}$$

Here, $\eta_k > 0$ is the step size; $\partial_R f(\cdot)$ denotes the Riemannian subdifferential of f and is given by

$$\partial_R f(\boldsymbol{x}) = (\boldsymbol{I}_n - \boldsymbol{x}\boldsymbol{x}^\top)\partial f(\boldsymbol{x}), \quad \forall\, \boldsymbol{x} \in \mathcal{M},$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\partial f(\cdot)$ is the usual subdifferential of the convex function f [31, Section 5]. Bai et al. [2] showed that for the ODL problem, RSGM with a suitable initialization will converge at a sublinear rate to a basis vector with high probability under a standard generative model of the data. Moreover, by running RSGM $O(n \log n)$ times, each time with an independent random initialization, one can recover the entire orthonormal basis with high probability. Around the same time, Zhu et al. [33] proposed a projected subgradient method (PSGM) for solving (1.1). The method generates the iterates via

$$\boldsymbol{x}^{k+1} = \frac{\boldsymbol{x}^k - \eta_k \boldsymbol{v}^k}{\|\boldsymbol{x}^k - \eta_k \boldsymbol{v}^k\|_2}, \quad \boldsymbol{v}^k \in \partial f(\boldsymbol{x}^k). \tag{1.3}$$

The updates (1.2) and (1.3) differ in the choice of the direction $\boldsymbol{v}^k$—the former uses a *Riemannian* subgradient of f at $\boldsymbol{x}^k$, while the latter uses a usual *Euclidean* subgradient. For the DPCP formulation of the RSR problem, Zhu et al. [33] showed that under certain assumptions on the data, PSGM with suitable initialization and piecewise geometrically diminishing step sizes will converge at a linear rate to a vector that is orthogonal to the inlier subspace $\mathcal{S}$. The step sizes take the form $\eta_k = \eta^{\lfloor (k-K_0)/K \rfloor + 1}$, where $\eta \in (0,1)$ and $K_0, K \geq 1$ satisfy certain conditions. In practice, however, the parameters η, K_0, K are difficult to determine. Therefore, Zhu et al. [33] also proposed a PSGM with modified backtracking line search (PSGM-MBLS), which works well in practice but has no convergence guarantee.

1.1 Motivations for this Work

Although the results in [2, 33] demonstrate, both theoretically and computationally, the efficacy of RSGM and PSGM for solving instances of (1.1) that arise from the ODL and RSR problems, respectively, two fundamental questions remain. First, while PSGM can be shown to achieve a *linear* convergence rate on the DPCP formulation of the RSR problem [33], only a *sublinear* convergence rate has been established for RSGM on the ODL problem [2]. Given the similarity of the updates (1.2) and (1.3), it is natural to ask whether the slower convergence rate of RSGM is an artifact of the analysis or due to the inherent structure of the ODL problem. Second, the convergence analyses in [2, 33] focus only on the ODL and RSR problems. In particular, they do not shed light on the performance of RSGM or PSGM when tackling general instances of problem (1.1). It would be of interest to fill this gap by identifying or developing practically fast methods that have more general convergence guarantees, especially since different applications may give rise to instances of problem (1.1) with different structures. In a recent attempt to address these questions, Li et al. [14] showed, among other things, that RSGM will converge at the sublinear rate of $\mathcal{O}(k^{-1/4})$ (here, k is the iteration counter) when applied to a general instance of problem (1.1) and at a linear rate when applied to a so-called *sharp* instance of problem (1.1). Informally, an optimization problem is said to possess the sharpness property if the objective function grows linearly with the distance to a set of local minima [5]. Such a property plays a crucial role in establishing fast convergence guarantees for a host of iterative methods; see, e.g., [5, 14, 16] and also [18, 32, 17] for related results. Since the ODL problem and the DPCP formulation of the RSR problem are known to possess the sharpness property under certain assumptions on the data [2, 33], the results in [14] imply that RSGM will converge linearly on these problems.

1.2 Our Contributions

In this paper, we depart from the subgradient-type approaches (such as RSGM (1.2) and PSGM (1.3)) and present another method called the manifold proximal point algorithm (ManPPA) to tackle problem (1.1). At each iterate $\boldsymbol{x}^k$, ManPPA computes a search directon by minimizing the sum of f and a proximal term defined in terms of the *Euclidean* distance over the tangent space to $\mathcal{M}$ at $\boldsymbol{x}^k$. This should be contrasted with other existing PPAs on manifolds (see, e.g., [9, 3]), in which the proximal term is defined in terms of the *Riemannian* distance. Such a difference is important. Indeed, although the search direction defined in ManPPA does not admit a closed-form formula, it can be computed in a highly efficient manner by exploiting the structure of problem (1.1); see Section 2.2. However, the search direction defined in the existing PPAs on manifolds can be as difficult to compute as a solution to the original problem. Consequently, the applicability of those methods is rather limited.

We now summarize our contributions as follows:

1. We show that ManPPA has a global sublinear convergence rate of $\mathcal{O}(k^{-1/2})$ when applied to a general instance of problem (1.1). Moreover, we show that if the instance has the sharpness property, then the local convergence rate of ManPPA is at least quadratic. Although the sublinear rate result follows from the results in [7], the quadratic rate result is new. Moreover, both rates are superior to those of RSGM established in [14]. Key to the proof of the quadratic rate result is a new *Riemannian subgradient inequality* (see Appendix C, Proposition C.1), which extends the classic *subgradient inequality* in the Euclidean space to the sphere $\mathcal{M}$. Such an inequality allows us to analyze ManPPA in a similar way as its Euclidean counterpart. It can also be of independent interest.

2. To compute the search direction in each iteration of ManPPA, we develop a semismooth Newton (SSN)-based inexact augmented Lagrangian method (ALM). Numerically, the proposed method can accurately compute the search direction in a highly efficient manner, which is crucial to the fast convergence of ManPPA. Theoretically, we show, for the first time, that the proposed SSN-based inexact ALM has an asymptotic superlinear convergence rate when finding the search direction.

3. We propose a stochastic version of ManPPA called StManPPA to tackle problem (1.1). StManPPA is well suited for the setting where the number of the data points p is extremely large, as each iteration involves only a simple closed-form update. We also analyze the convergence behavior of StManPPA. In particular, we show that it converges at the sublinear rate of $\mathcal{O}(k^{-1/4})$ when applied to a general instance of problem (1.1), which matches the convergence rate of RSGM established in [14]. Again, the aforementioned Riemannian subgradient inequality plays an important role in establishing this result, as it connects the analysis of StManPPA to those of various Euclidean stochastic methods.

4. Using ManPPA as a building block, we develop a new method for solving the following matrix analog of problem (1.1):

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times q}} \|\boldsymbol{Y}^\top \boldsymbol{X}\|_1 \quad \text{s.t.} \quad \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_q. \tag{1.4}$$

   Our interest in problem (1.4) stems from the observation that it provides alternative formulations of the ODL and RSR problems. Indeed, for the ODL problem, one can recover the entire orthonormal basis all at once by solving problem (1.4) with $q = n$. For the RSR problem, if one knows the dimension d of the inlier subspace $\mathcal{S}$, then one can recover it by solving problem (1.4) with $q = n - d$. We show that a good feasible solution to problem (1.4) can be found in a column-by-column manner by suitably modifying ManPPA. Although the proposed method is only a heuristic, our extensive numerical experiments show that it yields solutions of comparable quality to but is significantly faster than existing methods on the ODL and RSR problems.

1.3 Organization and Notation

The rest of the paper is organized as follows. In Section 2, we present ManPPA for solving problem (1.1) and describe a highly efficient method for solving the subproblem that arises in each iteration of ManPPA. We also analyze the convergence behavior of ManPPA. In Section 3, we propose StManPPA, a stochastic version of ManPPA that is well suited for large-scale computation, and analyze its convergence behavior. In Section 4 we discuss an extension of ManPPA for solving the matrix analog (1.4) of problem (1.1). In Section 5, we apply ManPPA to solve the ODL problem and the DPCP formulation of the RSR problem and compare its performance with some existing methods. We draw our conclusions in Section 6.

Besides the notation introduced earlier, we use L to denote the Lipschitz constant of f; i.e., $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ (note that $L \leq \sqrt{n} \max_{\boldsymbol{u} \in \mathbb{R}^n} \|\boldsymbol{Y}^\top \boldsymbol{u}\|_1 / \|\boldsymbol{u}\|_1$). Given a closed set $\mathcal{C} \subseteq \mathbb{R}^n$, we use $\mathrm{Proj}_{\mathcal{C}}(\boldsymbol{x})$ to denote the projection of $\boldsymbol{x}$ onto $\mathcal{C}$ and $\mathrm{dist}(\boldsymbol{x}, \mathcal{C}) := \inf_{\boldsymbol{y} \in \mathcal{C}} \|\boldsymbol{y} - \boldsymbol{x}\|_2$ to denote the distance between $\boldsymbol{x}$ and $\mathcal{C}$. Given a proper lower semicontinuous function $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, its proximal mapping is given by $\mathrm{prox}_h(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{w} \in \mathbb{R}^n}\, h(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{x}\|_2^2$. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ or $\boldsymbol{x}^\top \boldsymbol{y}$ to denote their usual inner product. Other notation is standard.

2 A Manifold Proximal Point Algorithm

Since problem (1.1) is nonconvex, our goal is to compute a *stationary point* of (1.1), which is a point $\bar{\boldsymbol{x}} \in \mathcal{M}$ that satisfies the first-order optimality condition

$$\boldsymbol{0} \in \partial_R f(\bar{\boldsymbol{x}}) = (\boldsymbol{I}_n - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top)\partial f(\bar{\boldsymbol{x}})$$

(see [31]). In the recent work [7], Chen et al. considered the more general problem of minimizing the sum of a smooth function and a nonsmooth convex function over the Stiefel manifold and developed a manifold proximal gradient method (ManPG) for finding a stationary point of it. When specialized to solve problem (1.1), the method generates the iterates via

$$\boldsymbol{x}^{k+1} = \mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k) = \frac{\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k}{\|\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k\|_2}, \tag{2.1}$$

where the search direction $\boldsymbol{d}^k$ is given by

$$\begin{array}{rcll} \boldsymbol{d}^k & = & \underset{\boldsymbol{d} \in \mathbb{R}^n}{\mathrm{argmin}} & \|\boldsymbol{Y}^\top(\boldsymbol{x}^k + \boldsymbol{d})\|_1 + \dfrac{1}{2t}\|\boldsymbol{d}\|_2^2 \\ & & \text{s.t.} & \boldsymbol{d}^\top \boldsymbol{x}^k = 0 \end{array} \tag{2.2}$$

and $\alpha_k > 0$, $t > 0$ are the step sizes. As the reader may readily recognize, without the constraint $\boldsymbol{d}^\top \boldsymbol{x}^k = 0$, the subproblem (2.2) is simply computing the proximal mapping of f at $\boldsymbol{x}^k$ and coincides with the update of the classic proximal point algorithm (PPA) [22]. The constraint $\boldsymbol{d}^\top \boldsymbol{x}^k = 0$ in (2.2), which states that the search direction $\boldsymbol{d}$ should lie on the tangent space to $\mathcal{M}$ at $\boldsymbol{x}^k$, is introduced to account for the manifold constraint in problem (1.1) and ensures that the next iterate $\boldsymbol{x}^{k+1}$ achieves sufficient decrease in objective value. Motivated by the above discussion, we call the method obtained by specializing ManPG to the setting of problem (1.1) *ManPPA* and present its details in Algorithm 1.

Naturally, ManPPA inherits the properties of ManPG established in [7]. However, due to the structure of problem (1.1), many of the developments in [7] have to be refined when designing ManPPA. In particular, the SSN method used by ManPG for finding the search direction in each

Algorithm 1 ManPPA for Solving Problem (1.1)

```
1: Input: $\mathbf{x}^0 \in \mathcal{M}$, $\beta \in (0,1)$, $t > 0$.
2: for $k = 0, 1, 2, \ldots$ do
3:     Solve the subproblem (2.2) to obtain $\boldsymbol{d}^k$.
4:     Let $j_k$ be the smallest nonnegative integer such that
           $f(\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \beta^{j_k}\boldsymbol{d}^k)) \leq f(\boldsymbol{x}^k) - \frac{\beta^{j_k}}{2t}\|\boldsymbol{d}^k\|_2^2.$
5:     Set $\boldsymbol{x}^{k+1}$ according to (2.1) with $\alpha_k = \beta^{j_k}$.
6: end for
```

iteration requires the computation of the proximal mapping of the nonsmooth part of the objective function. However, due to the presence of the matrix $\boldsymbol{Y}$, the objective function f of problem (1.1) does not have an easily computable proximal mapping. As such, the SSN method proposed in [7] cannot efficiently solve the subproblem (2.2). To circumvent this difficulty, we propose to use an inexact ALM, which can efficiently compute an accurate solution to (2.2); see Section 2.2.

Now, let us state the following result, which shows that the line search step in line 4 of Algorithm 1 is well defined. It simplifies [7, Lemma 5.2] and yields sharper constants. The proof can be found in Appendix B.

Proposition 2.1. *Let $\{(\boldsymbol{x}^k, \boldsymbol{d}^k)\}_k$ be the sequence generated by Algorithm 1. Define $\bar{\alpha} = \min\{1, 1/(tL)\}$. For any $\alpha \in (0, \bar{\alpha}]$, we have*

$$f(\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \alpha\boldsymbol{d}^k)) \leq f(\boldsymbol{x}^k) - \frac{\alpha}{2t}\|\boldsymbol{d}^k\|_2^2. \tag{2.3}$$

As a result, we have $\alpha_k = \beta^{j_k} > \beta\bar{\alpha}$ for any $k \geq 0$ in Algorithm 1, which implies that the line search step terminates after at most $\lceil \log_\beta \bar{\alpha} \rceil + 1$ iterations. In particular, if $t \leq 1/L$, then we have $\bar{\alpha} = 1$, which implies that we can take $j_k = 0$ in line 4 of Algorithm 1; i.e., no line search is needed.

2.1 Convergence Analysis of ManPPA

In this subsection, we study the convergence behavior of ManPPA. Recall from [7, Lemma 5.3] that if $\boldsymbol{d}^k = \mathbf{0}$ in (2.2), then $\boldsymbol{x}^k \in \mathcal{M}$ is a stationary point of problem (1.1). This motivates us to call $\boldsymbol{x}^k \in \mathcal{M}$ an *ϵ-stationary point* of problem (1.1) with $\epsilon \geq 0$ if the solution $\boldsymbol{d}^k$ to (2.2) satisfies $\|\boldsymbol{d}^k\|_2 \leq \epsilon$. By specializing the convergence results in [7, Theorem 5.5] for ManPG to ManPPA, we obtain the following theorem:

Theorem 2.2. *Any limit point of the sequence $\{\boldsymbol{x}^k\}_k$ generated by Algorithm 1 is a stationary point of problem (1.1). Moreover, Algorithm 1 with $t = 1/L$ returns an ϵ-stationary point $\boldsymbol{x}^k$ in at most $\lceil 2(f(\boldsymbol{x}^0) - f^*)/(L\epsilon^2) \rceil$ iterations, where f^* is the optimal value of problem (1.1).*

Theorem 2.2 shows that ManPPA has an iteration complexity of $\mathcal{O}(\epsilon^{-2})$, which is superior to the $\mathcal{O}(\epsilon^{-4})$ bound established for RSGM in [14].

Now, let us analyze the convergence rate of ManPPA in the setting where problem (1.1) possesses the sharpness property. Such a setting is highly relevant in applications, as both the ODL problem and DPCP formulation of the RSR problem give rise to sharp instances of problem (1.1) under certain assumptions on the data; see [2, Proposition C.8] and [14, Proposition 4]. To proceed, we first introduce the notion of sharpness.

Definition 2.3 (Sharpness; see, e.g., [5]). *We say that $\mathcal{X} \subseteq \mathcal{M}$ is a set of* weak sharp minima *for the function f with parameters (α, δ) (where $\alpha, \delta > 0$) if for any $\boldsymbol{x} \in \mathcal{B}(\delta) := \{\boldsymbol{x} \in \mathcal{M} \mid \operatorname{dist}(\boldsymbol{x}, \mathcal{X}) \leq \delta\}$, we have*

$$f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}) \geq \alpha \cdot \operatorname{dist}(\boldsymbol{x}, \mathcal{X}), \quad \forall \bar{\boldsymbol{x}} \in \mathcal{X}. \tag{2.4}$$

From the definition, we see that if $\mathcal{X}$ is a set of weak sharp minima of f, then it is the set of minimizers of f over $\mathcal{B}(\delta)$. Moreover, the function value grows linearly with the distance to $\mathcal{X}$. In the presence of such a regularity property, ManPPA can be shown to converge at a much faster rate. The following result, which has not appeared in the literature before and is thus new, constitutes the first main contribution of this paper.

Theorem 2.4. *Suppose that $\mathcal{X} \subseteq \mathcal{M}$ is a set of weak sharp minima for the function f with parameters (α, δ). Let $\{\boldsymbol{x}^k\}_k$ be the sequence generated by Algorithm 1 with $\operatorname{dist}(\boldsymbol{x}^0, \mathcal{X}) < \overline{\delta} := \min\left\{\delta, \frac{\alpha}{L}\right\}$ and $t \leq \min\left\{\frac{\overline{\delta}}{2\alpha - L\overline{\delta}}, \frac{2\overline{\delta}\alpha - L\overline{\delta}^2}{L^2}\right\}$. Then, we have*

$$\operatorname{dist}(\boldsymbol{x}^k, \mathcal{X}) \leq \overline{\delta}, \quad \forall k \geq 0, \tag{2.5}$$

$$\operatorname{dist}(\boldsymbol{x}^{k+1}, \mathcal{X}) \leq \mathcal{O}(\operatorname{dist}^2(\boldsymbol{x}^k, \mathcal{X})), \quad \forall k \geq 0. \tag{2.6}$$

Theorem 2.4 establishes the quadratic convergence rate of ManPPA when applied to a sharp instance of problem (1.1). Again, this is superior to the linear convergence rate of RSGM established in [14] for this setting. The proof of Theorem 2.4 can be found in Appendix C. Note that since $\mathcal{M}$ is nonconvex, one cannot directly apply standard convergence analysis techniques for PPA (see, e.g., [22]) to obtain Theorem 2.4. The key to overcoming this diffculty is the new Riemannian subgradient inequality we establish in Proposition C.1 (see Appendix C), which provides a path for extending the convergence analysis of PPA to that of ManPPA.

It should be pointed out that the results in Theorems 2.2 and 2.4 do not assume any generative model of the data matrix $\boldsymbol{Y}$. By contrast, the results developed in, e.g., [2] for the ODL problem and [33] for the DPCP formulation of the RSR problem do assume certain generative models of the data. Although the latter results may yield qualitatively sharper convergence guarantees for instances of (1.1) that arise from the ODL problem or the DPCP formulation of the RSR problem, the former apply to arbitrary instances of (1.1).

2.2 Solving the Subproblem (2.2)

Observe that each iteration of ManPPA requires solving the subproblem (2.2) to obtain the search direction. Thus, the efficiency of ManPPA depends not only on its convergence rate (which has already been studied in Theorems 2.2 and 2.4) but also on how fast the subproblem (2.2) can be solved. To address the latter, we note that (2.2) is a linearly constrained strongly convex quadratic minimization problem. This motivates us to adopt the SSN-based ALM originally developed in [15] for LASSO-type problems to solve it. As we shall see, such an approach yields a highly efficient method for solving the subproblem (2.2).

To set the stage for our development, let us drop the index k from (2.2) for simplicity and set $\boldsymbol{c} = \boldsymbol{Y}^\top \boldsymbol{x}$. Then, the subproblem (2.2) can be equivalently written as

$$\min_{\substack{\boldsymbol{d} \in \mathbb{R}^n, \\ \boldsymbol{u} \in \mathbb{R}^p}} \ \frac{1}{2}\|\boldsymbol{d}\|_2^2 + t\|\boldsymbol{u}\|_1 \ \text{ s.t. } \ \boldsymbol{Y}^\top \boldsymbol{d} + \boldsymbol{c} = \boldsymbol{u}, \ \boldsymbol{d}^\top \boldsymbol{x} = 0. \tag{2.7}$$

At this point, one may be tempted to use ADMM to solve problem (2.7). However, from a practical point of view, ADMM is often unable to return a high-accuracy solution in an efficient manner. Since

(2.7) is a subproblem in ManPPA, a low-accuracy solution will adversely affect the convergence rate of ManPPA. In fact, this has been observed in our numerical experiments. Therefore, we propose to use an inexact ALM, which can solve problem (2.7) efficiently and accurately. This makes it possible for ManPPA to achieve fast convergence. To describe the algorithm, let us first write down the augmented Lagrangian function corresponding to (2.7):

$$\mathcal{L}_\sigma(\boldsymbol{d},\boldsymbol{u};y,\boldsymbol{z}) := \frac{1}{2}\|\boldsymbol{d}\|_2^2 + t\|\boldsymbol{u}\|_1 + y\cdot\boldsymbol{d}^\top\boldsymbol{x} + \langle\boldsymbol{z},\boldsymbol{Y}^\top\boldsymbol{d}+\boldsymbol{c}-\boldsymbol{u}\rangle + \frac{\sigma}{2}(\boldsymbol{d}^\top\boldsymbol{x})^2 + \frac{\sigma}{2}\|\boldsymbol{Y}^\top\boldsymbol{d}+\boldsymbol{c}-\boldsymbol{u}\|_2^2. \tag{2.8}$$

Here, $y \in \mathbb{R}$ and $\boldsymbol{z} \in \mathbb{R}^p$ are Lagrange multipliers (dual variables) associated with the constraints in (2.7) and $\sigma > 0$ is a penalty parameter. Then, the inexact ALM for solving (2.7) can be described as follows [21, 15]:

Algorithm 2 Inexact ALM for Solving Problem (2.7)

```
Input: $\boldsymbol{d}^0 \in \mathbb{R}^n$, $\boldsymbol{u}^0 \in \mathbb{R}^p$, $y^0 \in \mathbb{R}$, $\boldsymbol{z}^0 \in \mathbb{R}^p$, $\sigma_0 > 0$.
for $j = 0, 1, \ldots$ do
    Compute
        $(\boldsymbol{d}^{j+1},\boldsymbol{u}^{j+1}) \approx \operatorname*{argmin}_{\boldsymbol{d}\in\mathbb{R}^n,\ \boldsymbol{u}\in\mathbb{R}^p} \Psi_j(\boldsymbol{d},\boldsymbol{u}) := \mathcal{L}_{\sigma_j}(\boldsymbol{d},\boldsymbol{u};y^j,\boldsymbol{z}^j).$    (2.9)
    Update dual variables:
        $y^{j+1} = y^j + \sigma_j(\boldsymbol{d}^{j+1})^\top\boldsymbol{x},$
        $\boldsymbol{z}^{j+1} = \boldsymbol{z}^j + \sigma_j(\boldsymbol{Y}^\top\boldsymbol{d}^{j+1} + \boldsymbol{c} - \boldsymbol{u}^{j+1}).$
    Update $\sigma_{j+1} \nearrow \sigma_\infty \le +\infty$.
end for
```

Since the subproblem (2.9) can only be solved inexactly in general, we adopt the following stopping criteria, which are standard in the literature (see [21, 15]):

$$\Psi_j(\boldsymbol{d}^{j+1},\boldsymbol{u}^{j+1}) - \Psi_j^* \le \frac{\varepsilon_j^2}{2\sigma_j}, \quad \sum_{j=0}^{\infty}\varepsilon_j < \infty, \tag{2.10a}$$

$$\Psi_j(\boldsymbol{d}^{j+1},\boldsymbol{u}^{j+1}) - \Psi_j^* \le \frac{\delta_j^2}{2\sigma_j}\|(y^{j+1},\boldsymbol{z}^{j+1}) - (y^j,\boldsymbol{z}^j)\|_2^2, \quad \sum_{j=0}^{\infty}\delta_j < \infty, \tag{2.10b}$$

$$\operatorname{dist}(\boldsymbol{0},\partial\Psi_j(\boldsymbol{d}^{j+1},\boldsymbol{u}^{j+1})) \le \frac{\delta_j'}{\sigma_j}\|(y^{j+1},\boldsymbol{z}^{j+1}) - (y^j,\boldsymbol{z}^j)\|_2, \quad \delta_j' \searrow 0. \tag{2.10c}$$

Here, Ψ_j^* is the optimal value of (2.9). Conditions (2.10a)–(2.10c) ensure that starting from any initial point $(\boldsymbol{d}^0,\boldsymbol{u}^0;y^0,\boldsymbol{z}^0)$, the inexact ALM (Algorithm 2) will converge at a superlinear rate to an optimal solution to problem (2.7). This result, which constitutes the second main contribution of this paper, is obtained from a new perturbation analysis of the solution set of the subproblem (2.2) and its dual. The proof can be found in Appendix D.

Now, it remains to discuss how to solve the subproblem (2.9) in an efficient manner. Again, let us drop the index j in (2.9) for simplicity. By simple manipulation, we have

$$\Psi(\boldsymbol{d},\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{d}\|_2^2 + \frac{\sigma}{2}\left(\boldsymbol{d}^\top\boldsymbol{x} + \frac{y}{\sigma}\right)^2 - \frac{y^2}{2\sigma} - \frac{\|\boldsymbol{z}\|_2^2}{2\sigma} + t\|\boldsymbol{u}\|_1 + \frac{\sigma}{2}\left\|\boldsymbol{Y}^\top\boldsymbol{d} + \boldsymbol{c} + \frac{\boldsymbol{z}}{\sigma} - \boldsymbol{u}\right\|_2^2.$$

Consider the function $\boldsymbol{d} \mapsto \psi(\boldsymbol{d}) := \inf_{\boldsymbol{u}\in\mathbb{R}^p} \Psi(\boldsymbol{d},\boldsymbol{u})$. Upon letting $\boldsymbol{w} = \boldsymbol{Y}^\top \boldsymbol{d} + \boldsymbol{c} + \frac{\boldsymbol{z}}{\sigma} \in \mathbb{R}^p$ and using the definition of the proximal mapping of $\boldsymbol{u} \mapsto h(\boldsymbol{u}) := t\|\boldsymbol{u}\|_1$, we have

$$\psi(\boldsymbol{d}) = \frac{1}{2}\|\boldsymbol{d}\|_2^2 + \frac{\sigma}{2}\left(\boldsymbol{d}^\top \boldsymbol{x} + \frac{y}{\sigma}\right)^2 - \frac{y^2}{2\sigma} - \frac{\|\boldsymbol{z}\|_2^2}{2\sigma} + h(\mathrm{prox}_{h/\sigma}(\boldsymbol{w})) + \frac{\sigma}{2}\|\boldsymbol{w} - \mathrm{prox}_{h/\sigma}(\boldsymbol{w})\|_2^2.$$

It follows that $(\bar{\boldsymbol{d}}, \bar{\boldsymbol{u}}) = \mathrm{argmin}_{\boldsymbol{d}\in\mathbb{R}^n, \boldsymbol{u}\in\mathbb{R}^p}\ \Psi(\boldsymbol{d},\boldsymbol{u})$ if and only if

$$\bar{\boldsymbol{d}} = \underset{\boldsymbol{d}\in\mathbb{R}^n}{\mathrm{argmin}}\ \psi(\boldsymbol{d}), \quad \bar{\boldsymbol{u}} = \mathrm{prox}_{h/\sigma}\left(\boldsymbol{Y}^\top \bar{\boldsymbol{d}} + \boldsymbol{c} + \frac{\boldsymbol{z}}{\sigma}\right).$$

Using [23, Theorem 2.26] and the Moreau decomposition $\boldsymbol{w} = \mathrm{prox}_{h/\sigma}(\boldsymbol{w}) + (1/\sigma)\mathrm{prox}_{\sigma h^*}(\sigma \boldsymbol{w})$, where h^* is the conjugate function of h, it can be deduced that ψ is strongly convex and continuously differentiable with

$$\nabla\psi(\boldsymbol{d}) = \boldsymbol{d} + \sigma\left(\boldsymbol{d}^\top \boldsymbol{x} + \frac{y}{\sigma}\right)\boldsymbol{x} + \boldsymbol{Y}\mathrm{prox}_{\sigma h^*}(\sigma \boldsymbol{w}).$$

Thus, we can find $\bar{\boldsymbol{d}}$ by solving the nonsmooth equation

$$\nabla\psi(\boldsymbol{d}) = \boldsymbol{0}. \tag{2.11}$$

Towards that end, we apply an SSN method, which finds the solution by successive linearization of the map $\nabla\psi$. To implement the method, we first need to compute the generalized Jacobian of $\nabla\psi$ [8, Definition 2.6.1], denoted by $\partial(\nabla\psi)$. By the chain rule [8, Corollary of Theorem 2.6.6] and the Moreau decomposition, each element $\boldsymbol{V} \in \partial(\nabla\psi)$ takes the form

$$\boldsymbol{V} = \boldsymbol{I}_n + \sigma\boldsymbol{Y}(\boldsymbol{I}_p - \boldsymbol{Q})\boldsymbol{Y}^\top + \sigma\boldsymbol{x}\boldsymbol{x}^\top, \tag{2.12}$$

where $\boldsymbol{Q} \in \partial\mathrm{prox}_{h/\sigma}(\boldsymbol{w})$. Using the definition of h, it can be shown that the diagonal matrix $\boldsymbol{Q} = \mathrm{Diag}(\boldsymbol{q})$ with

$$q_i = \begin{cases} 0 & \text{if } |w_i| \le t/\sigma, \\ 1 & \text{otherwise,} \end{cases} \qquad i = 1, \dots, p$$

is an element of $\partial\mathrm{prox}_{h/\sigma}(\boldsymbol{w})$ [15, Section 3.3] and hence can be used to define an element $\boldsymbol{V} \in \partial(\nabla\psi)$ via (2.12). Note that the matrix $\boldsymbol{V}$ so defined is positive definite. As such, the following generic iteration of the SSN method for solving (2.11) is well defined:

$$\boldsymbol{v} = -\boldsymbol{V}^{-1}\nabla\psi(\boldsymbol{d}^j), \tag{2.13a}$$

$$\boldsymbol{d}^{j+1} = \boldsymbol{d}^j + \rho_j \boldsymbol{v} \tag{2.13b}$$

Here, $\rho_j > 0$ is the step size. Moreover, since $\boldsymbol{I}_p - \boldsymbol{Q}$ is a diagonal matrix whose entries are either 0 or 1, the matrix $\boldsymbol{V}$ can be assembled in a very efficient manner; again, see [15]. The detailed implementation of the SSN method for solving (2.11) is given in Algorithm 3.

The SSN method (Algorithm 3) will converge at a superlinear rate to the unique solution $\bar{\boldsymbol{d}}$ to (2.11). The proof can be found in Appendix D.

3 Stochastic Manifold Proximal Point Algorithm

In this section, we propose, for the first time, a stochastic ManPPA (StManPPA) for solving problem (1.1), which is well suited for the setting where p (typically representing the number of data points)

Algorithm 3 SSN method for Solving the Nonsmooth Equation (2.11)

```
Input: $\mu \in (0, 1/2)$, $\bar{\eta} \in [0, 1)$, $\tau \in (0, 1]$, $\delta \in (0, 1)$.
for $j = 0, 1, \ldots$ do
    Choose $\boldsymbol{Q}^j \in \partial \mathrm{prox}_{h/\sigma}\left(\boldsymbol{Y}^\top \boldsymbol{d} + \boldsymbol{c} + \frac{\boldsymbol{z}}{\sigma}\right)$. Let $\boldsymbol{V}^j = \boldsymbol{I}_n + \sigma \boldsymbol{Y}(\boldsymbol{I}_p - \boldsymbol{Q}^j)\boldsymbol{Y}^\top + \sigma \boldsymbol{x}\boldsymbol{x}^\top$. Find an
    approximate solution $\boldsymbol{v}^j$ to the linear system
        $\boldsymbol{V}^j \boldsymbol{v} = -\nabla\psi(\boldsymbol{d}^j)$
    that satisfies
        $\|\boldsymbol{V}^j \boldsymbol{v}^j + \nabla\psi(\boldsymbol{d}^j)\|_2 \leq \min\left\{\bar{\eta}, \|\nabla\psi(\boldsymbol{d}^j)\|_2^{1+\tau}\right\}.$
    Let $m_j$ be the smallest nonnegative integer such that
        $\psi(\boldsymbol{d}^j + \delta^{m_j}\boldsymbol{v}^j) \leq \psi(\boldsymbol{d}^j) + \mu\delta^{m_j}\langle \nabla\psi(\boldsymbol{d}^j), \boldsymbol{v}^j\rangle.$
    Set $\boldsymbol{d}^{j+1} = \boldsymbol{d}^j + \rho_j \boldsymbol{v}^j$ with $\rho_j = \delta^{m_j}$.
end for
```

is much larger than n (typically representing the ambient dimension of the data points). To begin, observe that problem (1.1) has the finite-sum structure

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{j=1}^{p} \left|\boldsymbol{y}_j^\top \boldsymbol{x}\right| \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 = 1,$$

where $\boldsymbol{y}_j \in \mathbb{R}^n$ is the j-th column of Y. When p is extremely large, computing the matrix-vector product $\boldsymbol{Y}^\top \boldsymbol{x}$ can be expensive. To circumvent this difficulty, in each iteration of StManPPA, a column of $\boldsymbol{Y}$, say $\boldsymbol{y}_j$, is randomly chosen and the search direction $\boldsymbol{d}^k$ is given by

$$\begin{aligned} \boldsymbol{d}^k \;=\; & \underset{\boldsymbol{d}\in\mathbb{R}^n}{\operatorname{argmin}} && \left|\boldsymbol{y}_j^\top(\boldsymbol{x}^k + \boldsymbol{d})\right| + \frac{1}{2t}\|\boldsymbol{d}\|_2^2 \\ & \quad\text{s.t.} && \boldsymbol{d}^\top \boldsymbol{x}^k = 0. \end{aligned} \tag{3.1}$$

The key advantage of StManPPA is that the subproblem (3.1) admits a closed-form solution that is very easy to compute.

Proposition 3.1. *Let $\mu = t(\boldsymbol{y}_j^\top \boldsymbol{x}^k)$. Then, the solution to (3.1) is given by*

$$\boldsymbol{d}^k = \begin{cases} \mu\boldsymbol{x}^k - t\boldsymbol{y}_j & \text{if } (1+\mu)\mu/t - t\|\boldsymbol{y}_j\|_2^2 > 0, \\ -\mu\boldsymbol{x}^k + t\boldsymbol{y}_j & \text{if } (1-\mu)\mu/t + t\|\boldsymbol{y}_j\|_2^2 < 0, \\ \dfrac{\mu^2\boldsymbol{x}^k - t\mu\boldsymbol{y}_j}{t^2\|\boldsymbol{y}_j\|_2^2 - \mu^2} & \text{otherwise.} \end{cases} \tag{3.2}$$

Proof. The first-order optimality conditions of (3.1) are

$$\boldsymbol{0} \in \frac{1}{t}\boldsymbol{d} + \partial\left|\boldsymbol{y}_j^\top(\boldsymbol{x}^k + \boldsymbol{d})\right| \boldsymbol{y}_j - \lambda\boldsymbol{x}^k, \tag{3.3a}$$

$$0 = \boldsymbol{d}^\top \boldsymbol{x}^k. \tag{3.3b}$$

Suppose that $\boldsymbol{d}$ is a solution to (3.3). If $\boldsymbol{y}_j^\top(\boldsymbol{x}^k+\boldsymbol{d})>0$, then $\partial\left|\boldsymbol{y}_j^\top(\boldsymbol{x}^k+\boldsymbol{d})\right|=1$, and (3.3a) implies that $\boldsymbol{0}=\boldsymbol{d}/t+\boldsymbol{y}_j-\lambda\boldsymbol{x}^k$. This, together with (3.3b) and the fact that $\|\boldsymbol{x}^k\|_2=1$, gives $\lambda=\boldsymbol{y}_j^\top\boldsymbol{x}^k$. It follows that $\boldsymbol{d}=t(\boldsymbol{y}_j^\top\boldsymbol{x}^k)\boldsymbol{x}^k-t\boldsymbol{y}_j=\mu\boldsymbol{x}^k-t\boldsymbol{y}_j$ and hence $\boldsymbol{y}_j^\top(\boldsymbol{x}^k+\boldsymbol{d})>0$ is equivalent to $(1+\mu)\mu/t-t\|\boldsymbol{y}_j\|_2^2>0$. This establishes the first case in (3.2). The other two cases in (3.2), which correspond to $\boldsymbol{y}_j^\top(\boldsymbol{x}^k+\boldsymbol{d})<0$ and $\boldsymbol{y}_j^\top(\boldsymbol{x}^k+\boldsymbol{d})=0$, can be derived using a similar argument. □

We now present the details of StManPPA in Algorithm 4. It is worth noting that our proposed StManPPA is different from the ones developed in the recent work [29]. Indeed, in each iteration, the former only needs to solve a subproblem that involves a single component of the objective function, while the latter need to compute the proximal mapping of the entire objective function.

Algorithm 4 StManPPA for Solving Problem (1.1)

```
Input: $\boldsymbol{x}^0 \in \mathcal{M}$, $t_0, t_1, \ldots, t_T > 0$.
for $k = 0, 1, \ldots, T$ do
    Select $j_k \in \{1, \ldots, p\}$ uniformly at random and solve the subproblem (3.1) with $j = j_k$, $t = t_k$ to obtain $\boldsymbol{d}^k$.
    Set $\boldsymbol{x}^{k+1} = \mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \boldsymbol{d}^k)$.
end for
Output: $\bar{\boldsymbol{x}} = \boldsymbol{x}^k$ with probability $t_k / \sum_{k=0}^T t_k$.
```

3.1 Convergence Analysis of StManPPA

In this section, we present our convergence results for StManPPA. Let us begin with some preparations. Define $f_j:\mathbb{R}^n\to\mathbb{R}$ to be the function $f_j(\boldsymbol{x})=\left|\boldsymbol{y}_j^\top\boldsymbol{x}\right|$ and let $L_j>0$ denote the Lipschitz constant of f_j, where $j=1,\ldots,p$. Set $\bar{L}:=\max_{j\in\{1,\ldots,p\}}L_j$. Furthermore, define the *Moreau envelope* and *proximal mapping* on $\mathcal{M}$ by

$$e_f(\boldsymbol{z})=\min_{\boldsymbol{x}\in\mathcal{M}}\ f(\boldsymbol{x})+\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2, \tag{3.4a}$$

$$\mathrm{mprox}_f(\boldsymbol{z})\in\operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{M}}\ f(\boldsymbol{x})+\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2, \tag{3.4b}$$

respectively. The proximal mapping mprox is well defined since the constraint set $\mathcal{M}$ is compact. As it turns out, the proximal mapping mprox can be used to define an alternative notion of stationarity for problem (1.1). Indeed, for any $\lambda>0$ and $\boldsymbol{x}\in\mathcal{M}$, if we denote $\hat{\boldsymbol{x}}=\mathrm{mprox}_{\lambda f}(\boldsymbol{x})$, then the optimality condition of (3.4) yields

$$\boldsymbol{0}\in\partial_R f(\hat{\boldsymbol{x}})+\frac{1}{\lambda}(\boldsymbol{I}_n-\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top)(\hat{\boldsymbol{x}}-\boldsymbol{x}).$$

Since $\boldsymbol{I}_n-\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top$ is a projection operator and hence nonexpansive, we obtain

$$\mathrm{dist}(\boldsymbol{0},\partial_R f(\hat{\boldsymbol{x}}))\le\frac{1}{\lambda}\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2.$$

In particular, if $\frac{1}{\lambda}\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2\le\epsilon$, then (i) $\hat{\boldsymbol{x}}$ is ϵ-stationary in the sense that $\mathrm{dist}(\boldsymbol{0},\partial_R f(\hat{\boldsymbol{x}}))\le\epsilon$ and (ii) $\boldsymbol{x}$ is close to the ϵ-stationary point $\hat{\boldsymbol{x}}$. This motivates us to use

$$\mathcal{M}\ni\boldsymbol{x}\mapsto\Theta_\lambda(\boldsymbol{x}):=\frac{1}{\lambda}\|\boldsymbol{x}-\mathrm{mprox}_{\lambda f}(\boldsymbol{x})\|_2$$

as a stationarity measure for problem (1.1). We call $\boldsymbol{x} \in \mathcal{M}$ an *ϵ-nearly stationary point* of problem (1.1) if $\Theta_\lambda(\boldsymbol{x}) \leq \epsilon$. It is worth noting that such a notion has also been used in [14] to study the stochastic RSGM.

We are now ready to establish the convergence rate of StManPPA, which constitutes the third main contribution of this paper.

Theorem 3.2. *For any $\lambda \in (0, 1/(p\bar{L}))$, the point $\bar{\boldsymbol{x}}$ output by Algorithm 4 satisfies*

$$\mathbb{E}\left[\Theta_\lambda(\bar{\boldsymbol{x}})^2\right] \leq \frac{2\lambda e_\lambda(\boldsymbol{x}^0) + \bar{L}^2 \sum_{k=0}^{T} t_k^2}{\lambda((1/p) - \lambda\bar{L}) \sum_{k=0}^{T} t_k},$$

where the expectation is taken over all random choices made by the algorithm. In particular, if the step sizes $\{t_k\}_k$ satisfy $\sum_{k=0}^{\infty} t_k = \infty$ and $\sum_{k=0}^{\infty} t_k^2 < \infty$, then $\mathbb{E}\left[\Theta_\lambda(\bar{\boldsymbol{x}})^2\right] \to 0$. Moreover, if we take $t_k = \frac{1}{\sqrt{T+1}}$ for $k = 0, 1, \ldots, T$, then the number of iterations needed by StManPPA to obtain a point $\bar{\boldsymbol{x}} \in \mathcal{M}$ satisfying $\mathbb{E}[\Theta_\lambda(\bar{\boldsymbol{x}})] \leq \epsilon$ is $\mathcal{O}(\epsilon^{-4})$.

The proof of Theorem 3.2 can be found in Appendix E. Again, it makes crucial use of our newly established Riemannian subgradient inequality (see Appendix C, Proposition C.1), which allows StManPPA to be analyzed in a similar way as various Euclidean stochastic methods. We remark that the iteration complexity bound $\mathcal{O}(\epsilon^{-4})$ of StManPPA established in Theorem 3.2 is comparable to that of RSGM established in [14].

4 Extension to Stiefel Manifold Constraint

In this section, we consider the matrix analog (1.4) of problem (1.1), which also arises in many applications such as certain "one-shot" formulations of the ODL and RSR problems (see Section 1.2). Currently, there are two existing approaches for solving problem (1.4), namely a sequential linear programming (SLP) approach and an iteratively reweighted least squares (IRLS) approach [28]. In the SLP approach, the columns of $\boldsymbol{X}$ are extracted one at a time. Suppose that we have already obtained the first ℓ columns of $\boldsymbol{X}$ ($\ell = 0, 1, \ldots, q-1$) and arrange them in the matrix $\boldsymbol{X}_\ell \in \mathbb{R}^{n \times \ell}$ (with $\boldsymbol{X}_0 = \boldsymbol{0}$). Then, the $(\ell+1)$-st column of $\boldsymbol{X}$ is obtained by solving

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \ \|\boldsymbol{Y}^\top \boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 = 1, \ \boldsymbol{X}_\ell^\top \boldsymbol{x} = \boldsymbol{0}.$$

This is achieved by the alternating linearization and projection (ALP) method, which generates the iterates via

$$\begin{aligned} \boldsymbol{z}^k &= \operatorname*{argmin}_{\boldsymbol{z} \in \mathbb{R}^n} \ \|\boldsymbol{Y}^\top \boldsymbol{z}\|_1 \ \text{ s.t. } \ \boldsymbol{z}^\top \boldsymbol{x}^{k-1} = 1, \ \boldsymbol{X}_\ell^\top \boldsymbol{z} = \boldsymbol{0}, \\ \boldsymbol{x}^k &= \boldsymbol{z}^k / \|\boldsymbol{z}^k\|_2. \end{aligned}$$

Note that the $\boldsymbol{z}$-subproblem is a linear program, which can be efficiently solved by off-the-shelf solvers.

In the IRLS approach, the following variant of problem (1.4), which favors row-wise sparsity of $\boldsymbol{Y}^\top \boldsymbol{X}$ and has also been studied by Lerman et al. in [13], is considered:

$$\min_{\boldsymbol{X} \in \mathbb{R}^{n \times q}} \ \|\boldsymbol{Y}^\top \boldsymbol{X}\|_{1,2} \quad \text{s.t.} \quad \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_q. \tag{4.1}$$

Here, $\|\boldsymbol{Y}^\top\boldsymbol{X}\|_{1,2}$ denotes the sum of the Euclidean norms of the rows of $\boldsymbol{Y}^\top\boldsymbol{X}$. The IRLS method for solving (4.1) generates the iterates via

$$\boldsymbol{X}^k = \operatorname*{argmin}_{\boldsymbol{X}\in\mathbb{R}^{n\times q}} \sum_{j=1}^{p} w_{j,k}\|\boldsymbol{X}^\top\boldsymbol{y}_j\|_2^2 \quad \text{s.t.} \quad \boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{I}_q,$$

where $w_{j,k} = 1/\max\{\delta, \|\boldsymbol{X}^{k-1\top}\boldsymbol{y}_j\|_2\}$ and $\delta > 0$ is a perturbation parameter to prevent the denominator from being 0. The solution $\boldsymbol{X}^k$ can be obtained via an SVD and is thus easy to implement. However, there has been no convergence guarantee for the IRLS method so far.

Recently, Wang et al. [30] proposed a proximal alternating maximization method for solving a maximization version of (1.4), which arises in the so-called ℓ_1-*PCA* problem (see [12]). However, the method cannot be easily adapted to solve problem (1.4).

The similarity between problems (1.1) and (1.4) suggests that the latter can also be solved by ManPPA, which is indeed the case. However, the SSN method for solving the resulting nonsmooth equation (i.e., the matrix analog of (2.11)) can be slow, as the dimension of the linear system (2.13) is high. Here, we propose an alternative method called *sequential ManPPA*, which solves (1.4) in a column-by-column manner and constitutes the fourth main contribution of this paper. The method is based on the observation that the objective function in (1.4) is separable in the columns of $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$. To find $\boldsymbol{x}_1$, we simply solve

$$\min_{\boldsymbol{x}_1\in\mathbb{R}^n} \|\boldsymbol{Y}^\top\boldsymbol{x}_1\|_1 \quad \text{s.t.} \quad \|\boldsymbol{x}_1\|_2 = 1$$

using ManPPA as it is an instance of problem (1.1). Now, suppose that we have found the first ℓ columns of $\boldsymbol{X}$ ($\ell = 0, 1, \ldots, q-1$) and arrange them in the matrix $\boldsymbol{Q}_\ell \in \mathbb{R}^{n\times\ell}$ (with $\boldsymbol{Q}_0 = \boldsymbol{0}$). Then, we find the $(\ell+1)$-st column $\boldsymbol{x}_{\ell+1}$ by solving

$$\min_{\boldsymbol{x}_{\ell+1}\in\mathbb{R}^n} \|\boldsymbol{Y}^\top\boldsymbol{x}_{\ell+1}\|_1 \ \text{s.t.} \ \|\boldsymbol{x}_{\ell+1}\|_2 = 1, \ \boldsymbol{Q}_\ell^\top\boldsymbol{x}_{\ell+1} = \boldsymbol{0}. \tag{4.2}$$

As it turns out, problem (4.2) is equivalent to the deflation process discussed in [27] for sequentially recovering the columns of an orthogonal dictionary. Specifically, let $\boldsymbol{V}_\ell$ be an orthonormal basis of the orthogonal complement of $\boldsymbol{Q}_\ell$. We can then find $\boldsymbol{x}_{\ell+1}$ by solving

$$\min_{\boldsymbol{q}\in\mathbb{R}^{n-\ell}} \|\boldsymbol{Y}^\top\boldsymbol{V}_\ell\boldsymbol{q}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1. \tag{4.3}$$

Note that (4.3) has the same form as (1.1) and thus can be solved by RSGM or PSGM. By contrast, problem (4.2) has an extra linear constraint and cannot be solved by RSGM or PSGM directly. Nevertheless, our ManPPA can solve both (4.2) and (4.3). Let us now briefly discuss how to use ManPPA to solve the former. To simplify notation, let us drop the index ℓ and denote $\boldsymbol{x} = \boldsymbol{x}_{\ell+1}$, $\boldsymbol{Q} = \boldsymbol{Q}_\ell$. In the k-th iteration of ManPPA, we update the iterate by (2.1), where the search direction $\boldsymbol{d}^k$ is computed by

$$\begin{array}{ll} \min\limits_{\substack{\boldsymbol{d}\in\mathbb{R}^n,\\ \boldsymbol{u}\in\mathbb{R}^p}} & \dfrac{1}{2}\|\boldsymbol{d}\|_2^2 + t\|\boldsymbol{u}\|_1 \\ \text{s.t.} & \boldsymbol{Y}^\top(\boldsymbol{x}^k + \boldsymbol{d}) = \boldsymbol{u}, \ \boldsymbol{d}^\top[\boldsymbol{Q}, \boldsymbol{x}^k] = \boldsymbol{0}. \end{array}$$

This subproblem can be solved using the SSN-based inexact ALM framework in Section 2. We omit the details for succinctness. The sequential ManPPA is guaranteed to find a feasible solution to problem (1.4). Moreover, as we shall see in Section 5, it is computationally much more efficient than the ALP and IRLS methods on the ODL problem and the DPCP formulation of the RSR problem. However, it remains open whether the solution found by sequential ManPPA is a stationary point of problem (1.4). We leave this question for future research.

5 Numerical Experiments

In this section, we compare our proposed ManPPA, StManPPA, and sequential ManPPA with the existing methods PSGM-MBLS, ALP, and IRLS. We do not include RSGM with diminishing step sizes in our comparison, as the numerical results in [33] show that they are slower than PSGM-MBLS and cannot achieve high accuracy. In all the tests, we used the step size $t = 0.1$ for ManPPA and set the maximum number of iterations of ManPPA, inexact ALM, and SSN to 100, 30, and 20, respectively. We stopped ManPPA if the relative change in the objective values satisfies $|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k-1})|/f(\boldsymbol{x}^{k-1}) \leq 10^{-9}$. In the k-th iteration of ManPPA, we stopped the inexact ALM if the primal and dual feasibility of problem (2.7) satisfy

$$\max\left\{\sqrt{\|\boldsymbol{Y}^\top \boldsymbol{d}^{j+1} + \boldsymbol{c} - \boldsymbol{u}^{j+1}\|_2^2 + ((\boldsymbol{d}^{j+1})^\top \boldsymbol{x})^2}, \|\nabla \psi_j(\boldsymbol{d}^{j+1})\|_2\right\} \leq \epsilon_k = 0.1^k.$$

For SSN, we used the same termination criteria as the ones given in [15] and solved the linear equation (2.13a) by Cholesky decomposition. We refer the reader to the companion technical report [6] for the setting of the parameters in the inexact ALM and SSN.

For StManPPA, we used the piecewise geometrically diminishing step sizes $t_k = \beta^{\lfloor k/p \rfloor} t_0$ for $k = 0, 1, \ldots, pT$ with $t_0 = 0.6$ and $T = 500$. Such step sizes are motivated by those used in PSGM [33]. We use StManPPA-β to specify the parameter β used in the algorithm. We stopped the algorithm if the relative change in the objective values satisfies $|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k-1})|/f(\boldsymbol{x}^{k-1}) \leq 10^{-12}$. For PSGM-MBLS, ALP, and IRLS, we used their default settings of the parameters. We stopped ALP if the change in the objective values satisfies $|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k-1})| \leq 10^{-6}$, while we stopped IRLS if the change in the objective values satisfies $|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k-1})| \leq 10^{-11}$. With these stopping criteria, the solutions returned by these algorithms usually achieve the same accuracy.

5.1 DPCP Formulations of the RSR Problem

In this section, we consider the DPCP formulations of the RSR problem. For the recovery of a vector that is orthogonal to the inlier subspace (the *vector* case), we compared the performance of ManPPA, StManPPA, PSGM-MBLS, ALP, and IRLS on problem (1.1). For the recovery of the entire inlier subspace of known dimension d (the *matrix* case), we compared the performance of sequential ManPPA, PSGM-MBLS, ALP, and IRLS on problem (1.4) with $q = n - d$.

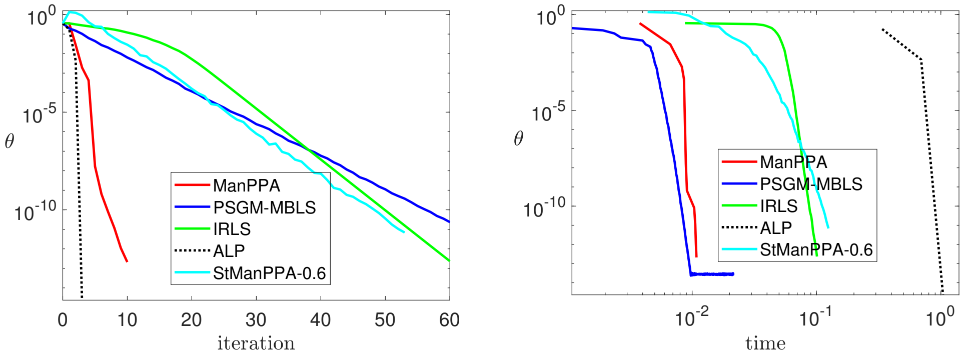
5.1.1 Synthetic Data

We first test the algorithms on synthetic data. The data matrix takes the form $\boldsymbol{Y} = [\boldsymbol{X}, \boldsymbol{O}] \in \mathbb{R}^{n \times (p_1 + p_2)}$. We generated the inlier data $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{C}$, where $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$ is an orthonormal matrix and $\boldsymbol{C} \in \mathbb{R}^{d \times p_1}$ is a coefficient matrix. The matrix $\boldsymbol{Q}$ was generated by orthonormalizing an $n \times d$ standard Gaussian random matrix via QR decomposition, while the matrix $\boldsymbol{C}$ was generated as a $d \times p_1$ standard Gaussian random matrix. The outlier data $\boldsymbol{O} \in \mathbb{R}^{n \times p_2}$ was generated as a standard Gaussian random matrix. Finally, the columns of the matrix $\boldsymbol{Y}$ were normalized. The d-dimensional subspace spanned by $\boldsymbol{X}$ is denoted by $\mathcal{S}$ and its orthogonal complement is denoted by $\mathcal{S}^\perp$.

Vector case. We set the initial point $\boldsymbol{x}^0$ of all algorithms to be the eigenvector of $\boldsymbol{Y}\boldsymbol{Y}^\top$ corresponding to the eigenvalue with minimum magnitude. We compared the performance of the algorithms on problem (1.1) with different dimension n, number of inliers p_1, and number of outliers p_2 in Figure 1. The first row of Figure 1 reports the principal angle[1] θ between $\boldsymbol{x}^k$ and $\mathcal{S}^\perp$ versus

[1] The principal angle is the distance between $\boldsymbol{x}^k$ and $\mathcal{S}^\perp$. Any optimal solution $\boldsymbol{x}^*$ to problem (1.1) is orthogonal to the inlier subspace $\mathcal{S}$ [33, Theorem 1]. Using the Lipschitz continuity of f, we know that θ also measures the function value gap $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$.

the iteration number. The second row reports θ versus CPU time. Note that $\boldsymbol{x}^k = \sin(\theta)\boldsymbol{n} + \cos(\theta)\boldsymbol{s}$, where $\boldsymbol{n} = \mathrm{Proj}_{\mathcal{S}}(\boldsymbol{x}^k)/\|\mathrm{Proj}_{\mathcal{S}}(\boldsymbol{x}^k)\|_2$ and $\boldsymbol{s} = \mathrm{Proj}_{\mathcal{S}^\perp}(\boldsymbol{x}^k)/\|\mathrm{Proj}_{\mathcal{S}^\perp}(\boldsymbol{x}^k)\|_2$. From Figure 1, we see that PSGM-MBLS is the fastest, while ManPPA is slightly slower. However, they are both much faster than other compared methods. Moreover, the principal angle θ of ManPPA decreases much faster than PSGM-MBLS in terms of iteration number. This can be attributed to the quadratic convergence rate of ManPPA (Theorem 2.4).



(a) $n = 30, p_1 = 500, p_2 = 1167$

(b) $n = 30, p_1 = 500, p_2 = 1167$

Figure 1: Numerical results for the DPCP formulation (1.1). (a): Principal angle versus iteration number. (b): Principal angle versus CPU time.

In Figure 2 we report the quadratic fitting curves of the different algorithms. As shown in [33], the DPCP formulation can tolerate $O((\#\text{inliers})^2)$ outliers; i.e., $p_2 = \mathcal{O}(p_1^2)$. For different $p_2 \in \{40, 80, 120, \ldots, 600\}$, we find the smallest $p_1 \in \{60, 70, 80, \ldots, 260\}$ such that $\theta < 10^{-1}$. Here, the principal angle θ is the mean value of 10 trials; i.e., we find pairs (p_1, p_2) such that for a fixed p_1, p_2 is the largest number of outliers that can be tolerated. We then use a quadratic function to fit these pairs (p_1, p_2). A higher curve indicates that more outliers can be tolerated and hence the algorithm is more robust. From Figure 2, we see that the curve corresponding to PSGM-MBLS is the lowest one and thus the least robust, while ManPPA and StManPPA are more robust. In Figure 3 we report the CPU time versus p_1 and p_2. For each algorithm, the shadow area represents the standard deviation (std) of 10 random trials, while the line within the shadow is the mean of those trials. From the left two subfigures of Figure 3, we see that the stds of IRLS and ALP are quite significant. In particular, they are usually more than ten times larger than the stds of other compared algorithms. To better illustrate the stds of the other four algorithms, we plot their CPU times in the right two subfigures of Figure 3. From these two figures, we see that ManPPA has a larger std than those of the other three algorithms. Overall, we see that PSGM-MBLS is the fastest and ManPPA is second, and they are both much faster than the other compared algorithms. Figures 2 and 3 suggest that ManPPA is slightly slower than PSGM-MBLS but is more robust. Moreover, the choice of the parameter β for StManPPA is crucial and challenging, as StManPPA-0.9 is faster but less robust than StManPPA-0.8. We leave the determination of the best parameter β for StManPPA as a future work.

Matrix case. We solved problem (1.4) using sequential ManPPA and compared its performance with PSGM-MBLS (applied to (4.3)), ALP, and IRLS. We report the results for $q = 2$ and $q = 4$ in Figures 4 and 5, respectively. Since ALP has a very high std, for better illustration of the CPU time comparison, we exclude it from the right two subfigures of Figures 4 and 5. The results suggest that

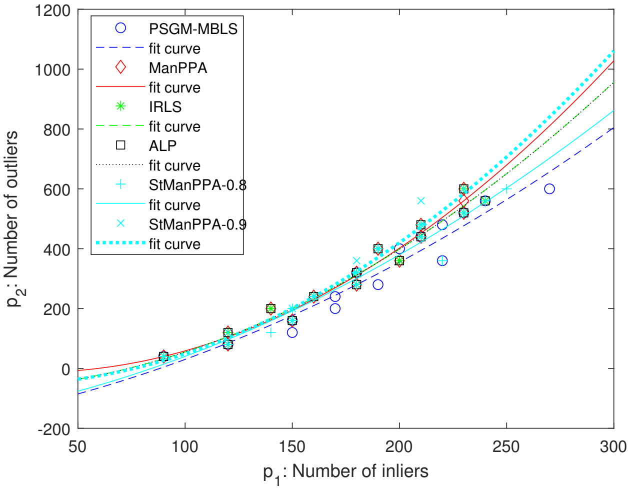


Figure 2: Quadratic fitting curves ($n = 30$).

sequential ManPPA is not as robust as IRLS but is the second most efficient one among the four compared algorithms. Moreover, we find that although PSGM-MBLS is very efficient, its fitting curve is not very good. This is due to the fact that PSGM-MBLS is very sensitive to the choice of step size.

5.1.2 Real 3D Point Cloud Road Data

Next, we compared ManPPA with PSGM-MBLS on the road detection challenge of the KITTI dataset [10]. This dataset contains image data together with the corresponding 3D points collected by a rotating 3D laser scanner. Similar to [33], we only used the 360° 3D point clouds to determine which points lie on the road plane (inliers) and which do not (outliers). By using homogeneous coordinates, this can be cast as a robust hyperplane learning problem (1.1) in $\mathbb{R}^4$. As reported in [33], PSGM-MBLS is the fastest algorithm when compared with other state-of-the-art methods. Thus, we only compared the performance of ManPPA and PSGM-MBLS on problem (1.1). Table 1 reports the area under the Receiver Operator Curve (ROC) and the CPU time. We see that all ROC values of ManPPA are better than those of PSGM-MBLS, with some sacrifice on the CPU time.

Table 1: Area under ROC and CPU time for annotated 3D point clouds with index 0, 21 in KITTY-CITY-48 and 1, 45, 120, 137, 153 in KITTYCITY-5. The number in parenthesis is the percentage of outliers.

	KITTY-CITY-48		KITTY-CITY-5				
	0 (56%)	21 (57%)	1 (37%)	45 (38%)	120 (53%)	137 (48%)	153(67%)
Area under ROC							
ManPPA	0.99437	0.99077	0.99810	0.99898	0.87629	0.99969	0.75481
PSGM-MBLS	0.99420	0.99062	0.99802	0.99891	0.86782	0.99968	0.74933
CPU time							
ManPPA	0.129	0.174	0.091	0.066	0.106	0.108	0.066
PSGM-MBLS	0.028	0.015	0.034	0.029	0.029	0.014	0.017

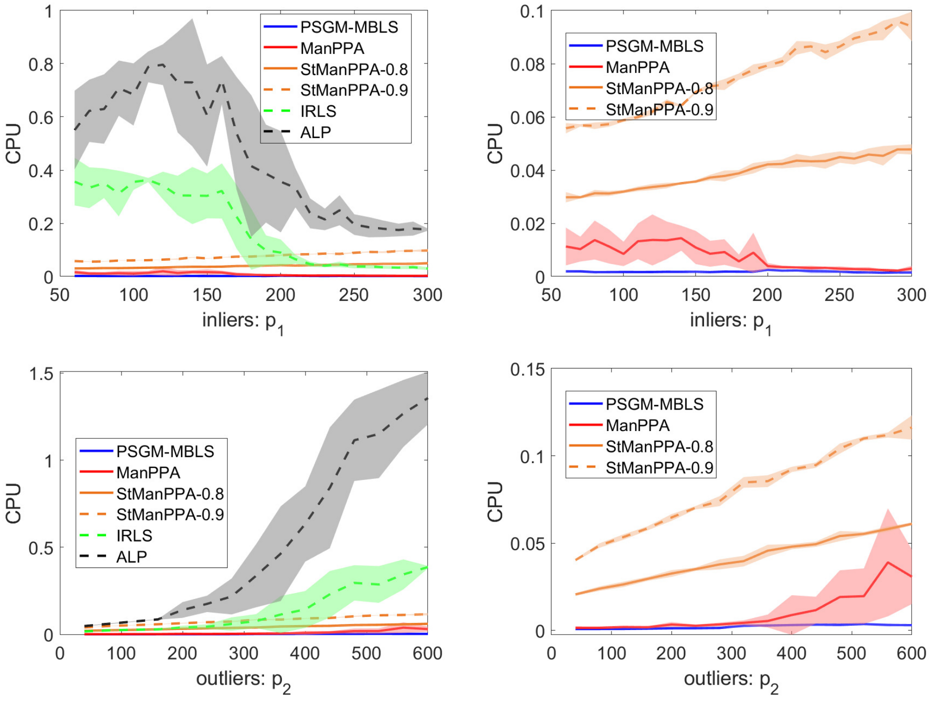


Figure 3: Upper: CPU time versus the number of inliers p_1 ($n = 30$, $p_2 = 320$). Lower: CPU time versus the number of outliers p_2 ($n = 30$, $p_1 = 200$). The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

5.2 ODL Problem

We generated instances of the ODL problem by first randomly generating an orthogonal matrix $\hat{\boldsymbol{X}} \in \mathbb{R}^{n\times n}$ and a Bernoulli-Gaussian matrix $\hat{\boldsymbol{A}} \in \mathbb{R}^{n\times p}$ with parameter γ (see, e.g., [25]), then setting $\boldsymbol{Y} = \hat{\boldsymbol{X}}\hat{\boldsymbol{A}}$.

Vector case. We first compared the performance of ManPPA and StManPPA with ALP, IRLS, and PSGM-MBLS on problem (1.1) with $n = 30$, $p = \lceil 10n^{1.5}\rceil$. Figures 6 and 7 report the iteration numbers and CPU times of the compared algorithms. The quantity θ is the angle between $\boldsymbol{x}^k$ returned by the algorithm and its nearest column in $\hat{\boldsymbol{X}}$. From Figures 6 and 7, we see that PSGM-MBLS is the fastest algorithm in terms of CPU time, while ManPPA is slightly slower. However, they are both much faster than the other compared algorithms. Moreover, we see that ManPPA is much faster than PSGM-MBLS in terms of iteration number. This again can be attributed to the quadratic convergence rate of ManPPA (Theorem 2.4).

In Figures 8 and 9 we report the linear fitting curves for $\log(n)$ and $\log(p)$ and CPU times of ManPPA, StManPPA-0.8, StManPPA-0.9, IRLS, ALP, and PSGM-MBLS. Note that for the ODL problem, it has been found empirically in [2] that the sample size p and dimension n should satisfy $p = O(n^2)$ to guarantee recovery. The linear fitting curves were found in the following manner. For a given dimension $n \in \{5, 10, 15 \ldots, 50\}$, we find the smallest sample number $p \in 2n + \{10, 20, 30, \ldots, 800\}$ such that $\theta < 10^{-1}$. Here the principal angle θ is the mean value of 10 trials. We then use a linear function to fit the points $\{(\log(n), \log(p))\}_{n,p}$. From Figures 8 and 9, we find that PSGM-MBLS is the fastest but its fitting curve is high, which suggests that it is not robust. This is because PSGM-MBLS is very sensitive to the choice of step size. ManPPA appears

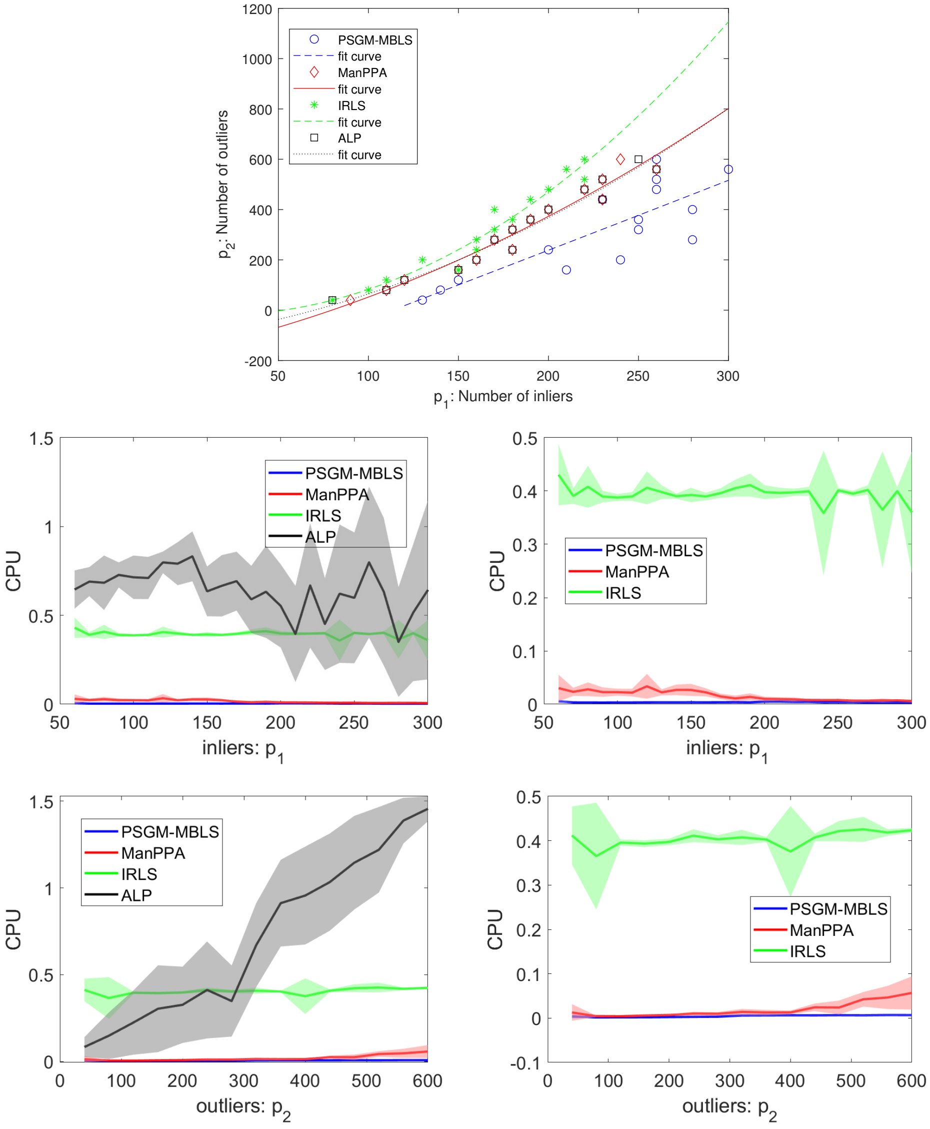


Figure 4: Comparison on the DPCP formulation (1.4) with $n = 30$, $q = 2$. First row: Quadratic fitting curves. Second row: CPU time versus the number of inliers p_1 ($p_2 = 320$). Third row: CPU time versus the number of outliers p_2 ($p_1 = 200$). The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

to be the second fastest but is very robust based on the fitting curve.

Matrix case. To find the entire orthogonal basis, we use sequential ManPPA to solve problem (1.4) with $q = n$. We compared sequential ManPPA with PSGM-MBLS (applied to (4.3)) and SLP based on ALP, and report the results in Figures 10 and 11. We see that sequential ManPPA is not as robust as ALP, but it is much faster than ALP. Note that there is nothing for IRLS to do, as it tackles the objective function $\|\boldsymbol{Y}^\top \boldsymbol{X}\|_{1,2}$, which is a constant when $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_n$. We also find that the fitting curve of PSGM-MBLS is very high, which indicates that it fails to recover the dictionary in many cases. Again, this is due to the high sensitivity of PSGM-MBLS to the choice of step size.

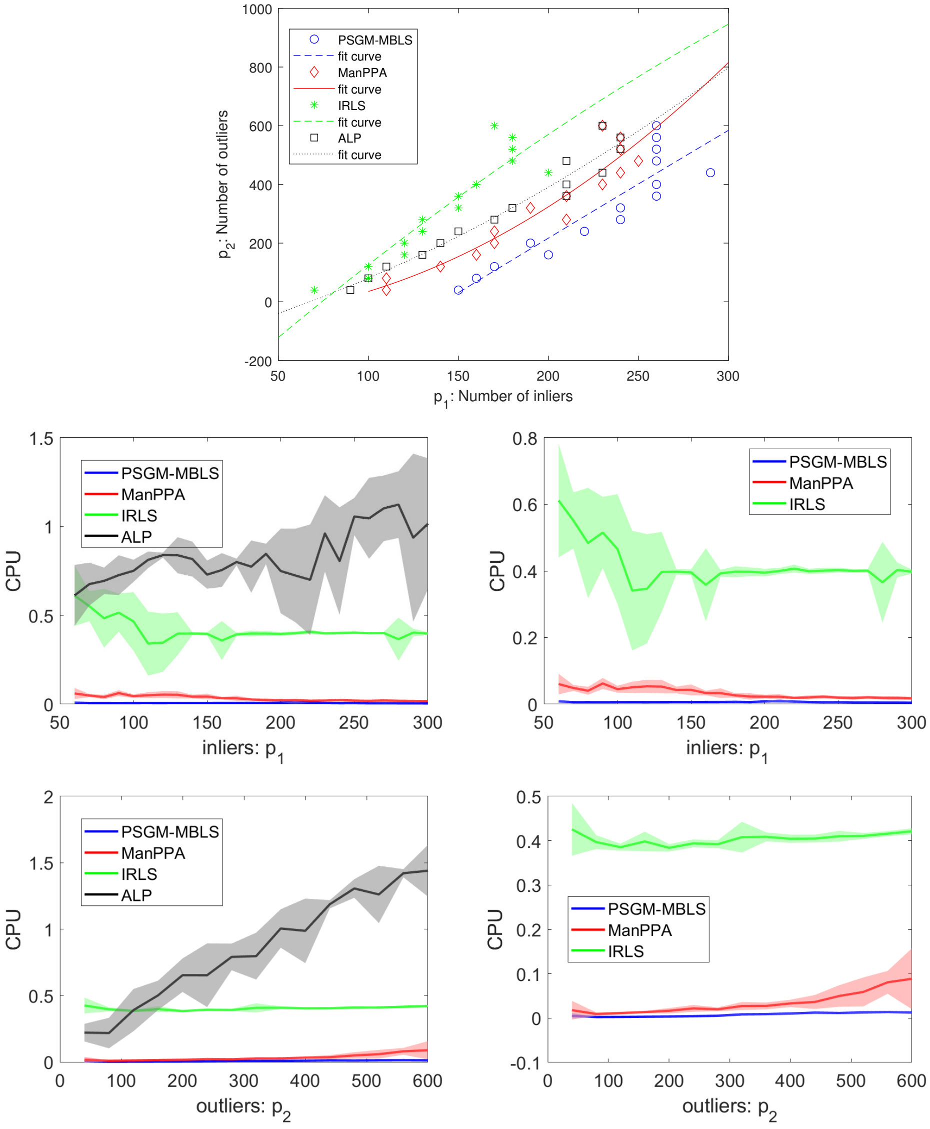


Figure 5: Comparison on the DPCP formulation (1.4) with $n = 30$, $q = 4$. First row: Quadratic fitting curves. Second row: CPU time versus the number of inliers p_1 ($p_2 = 320$). Third row: CPU time versus the number of outliers p_2 ($p_1 = 200$). The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

6 Conclusions

In this paper, we presented ManPPA and its stochastic variant StManPPA for solving problem (1.1). By exploiting the manifold structure of the constraint set $\mathcal{M}$, these methods not only are practically efficient but also possess convergence guarantees that are provably superior to those of existing subgradient-type methods. Using ManPPA as a building block, we also proposed a new sequential approach to solving the matrix analog (1.4) of problem (1.1). We conducted extensive numerical experiments to compare the performance of our proposed algorithms with existing ones on the ODL problem and DPCP formulation of the RSR problem. The results demonstrated the efficiency and efficacy of our proposed methods.

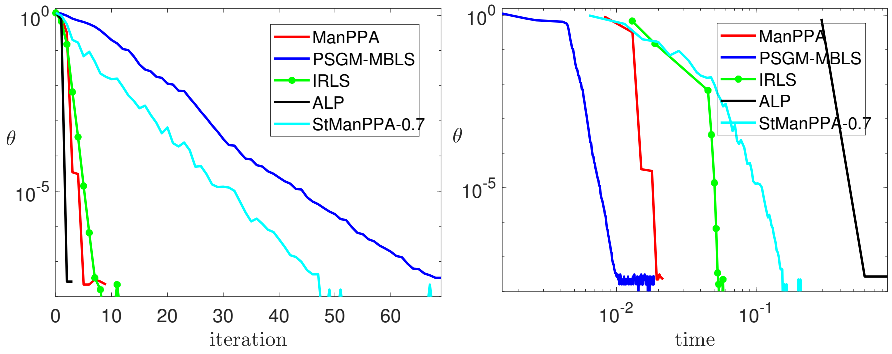


Figure 6: Numerical results for the ODL problem (1.1): $n = 30$, $p = \lceil 10n^{1.5} \rceil$, $\gamma = 0.1$.

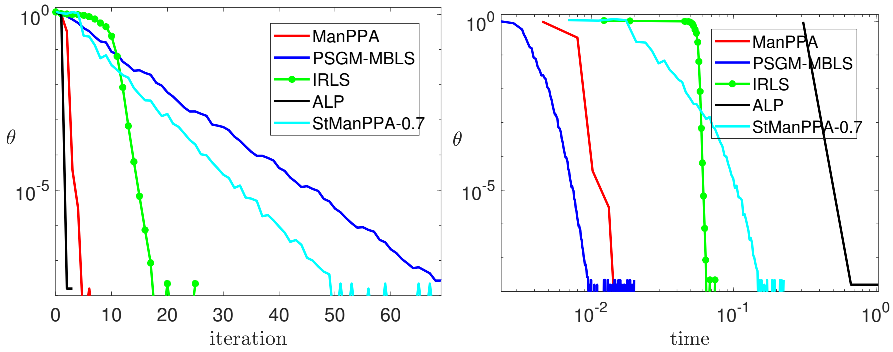


Figure 7: Numerical results for the ODL problem (1.1): $n = 30$, $p = \lceil 10n^{1.5} \rceil$, $\gamma = 0.3$.

Appendix

A Useful Properties of $\mathrm{Proj}_{\mathcal{M}}$

In this section, we collect some useful properties of the projector $\mathrm{Proj}_{\mathcal{M}}$.

Proposition A.1. *For any $\boldsymbol{x} \in \mathcal{M}$ and $\boldsymbol{d} \in \mathbb{R}^n$ satisfying $\boldsymbol{d}^\top \boldsymbol{x} = 0$ (i.e., $\boldsymbol{d}$ is a tangent vector at $\boldsymbol{x}$), we have*

$$\|\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x} + \boldsymbol{d}) - (\boldsymbol{x} + \boldsymbol{d})\|_2 \le \frac{1}{2}\|\boldsymbol{d}\|_2^2. \tag{A.1}$$

Moreover, if $\|\boldsymbol{d}\|_2 \le D$ for some $D \in (0, +\infty)$, then

$$\|\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x} + \boldsymbol{d}) - \boldsymbol{x}\|_2 \ge \frac{1}{(1 + D^2)^{3/4}}\|\boldsymbol{d}\|_2. \tag{A.2}$$

Proof. It is straightforward to verify that

$$\left\| \frac{\boldsymbol{x} + \boldsymbol{d}}{\|\boldsymbol{x} + \boldsymbol{d}\|_2} - (\boldsymbol{x} + \boldsymbol{d}) \right\|_2 = \sqrt{1 + \|\boldsymbol{d}\|_2^2} - 1.$$

We then have (A.1) by using the fact that $\sqrt{1 + x^2} - 1 \le \frac{1}{2}x^2$ for all $x \in \mathbb{R}$. Similarly, since

$$\left\| \frac{\boldsymbol{x} + \boldsymbol{d}}{\|\boldsymbol{x} + \boldsymbol{d}\|_2} - \boldsymbol{x} \right\|_2^2 = 2\left(1 - \frac{1}{\sqrt{1 + \|\boldsymbol{d}\|_2^2}}\right)$$

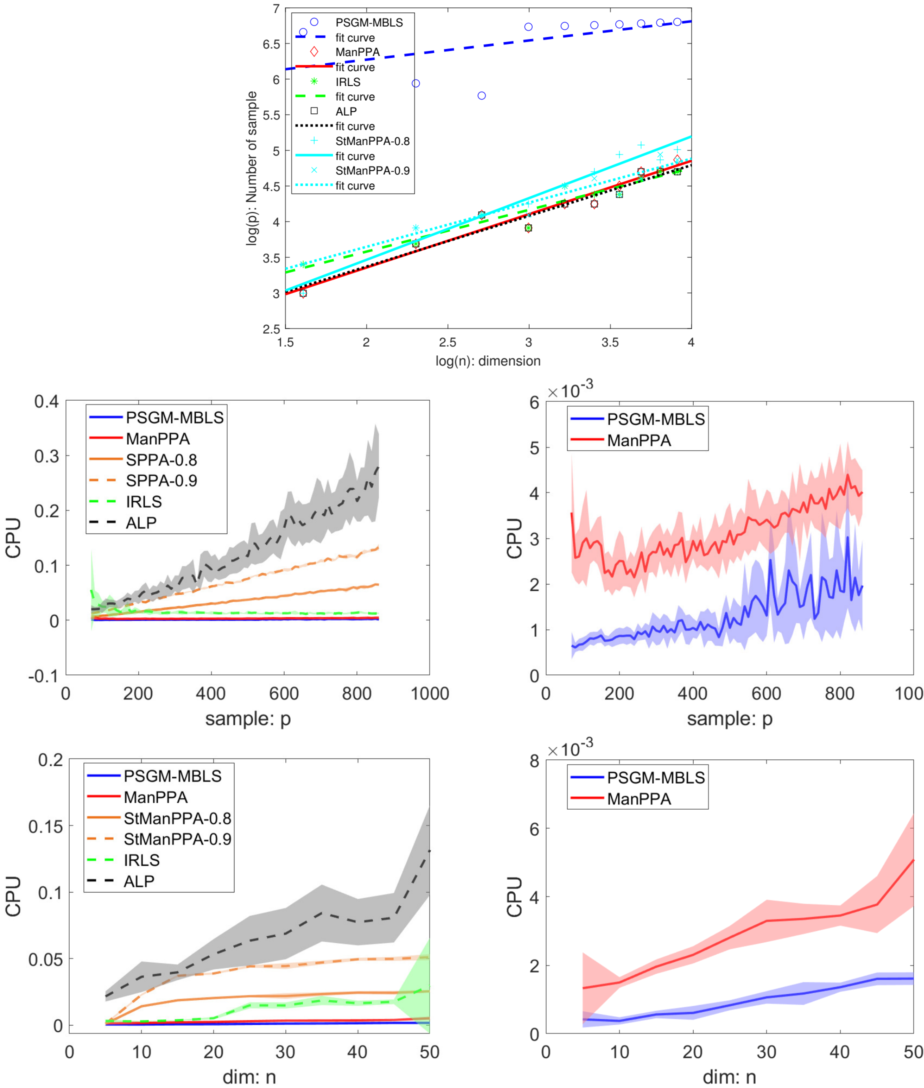


Figure 8: Comparison on the ODL problem (1.1) with $\gamma = 0.1$. First row: Fitting curves. Second row: CPU time versus the number of samples p, $(n = 30)$. Third row: CPU time versus the number of dimension n, $(p = 300)$. The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

and $\frac{1}{\sqrt{1+x^2}} \leq 1 - \frac{1}{2(1+D^2)^{3/2}}x^2$ for all $x \in [0, D]$, we get (A.2). $\square$

Proposition A.2. *For any $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{M}$ and $\boldsymbol{d} \in \mathbb{R}^n$ satisfying $\boldsymbol{d}^\top \boldsymbol{x} = 0$, we have*

$$\|\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x} + \boldsymbol{d}) - \boldsymbol{z}\|_2 \leq \|\boldsymbol{x} + \boldsymbol{d} - \boldsymbol{z}\|_2.$$

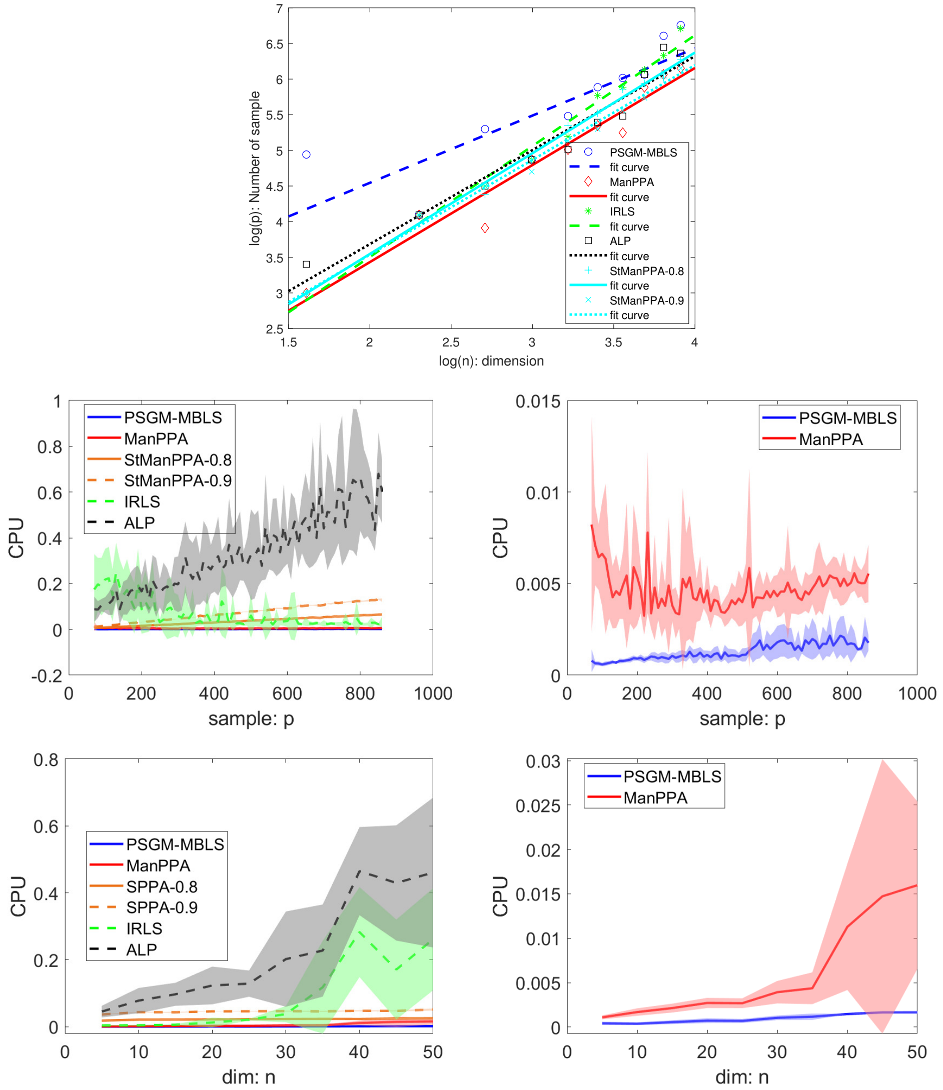


Figure 9: Comparison on the ODL problem (1.1) with $\gamma = 0.3$. First row: Fitting curves. Second row: CPU time versus the number of samples p, $(n = 30)$. Third row: CPU time versus the number of dimension n, $(p = 300)$. The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

Proof. We compute

$$
\begin{aligned}
&\left\| \frac{\boldsymbol{x}+\boldsymbol{d}}{\|\boldsymbol{x}+\boldsymbol{d}\|_2} - \boldsymbol{z} \right\|_2 = 2 - 2\frac{(\boldsymbol{x}+\boldsymbol{d})^\top \boldsymbol{z}}{\|\boldsymbol{x}+\boldsymbol{d}\|_2} \\
=&2 + 2(\boldsymbol{x}+\boldsymbol{d})^\top \boldsymbol{z}\left(1 - \frac{1}{\|\boldsymbol{x}+\boldsymbol{d}\|_2}\right) - 2(\boldsymbol{x}+\boldsymbol{d})^\top \boldsymbol{z} \\
\leq&2 + 2(\|\boldsymbol{x}+\boldsymbol{d}\|_2 - 1) - 2(\boldsymbol{x}+\boldsymbol{d})^\top \boldsymbol{z} \\
\leq&\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{z}\|_2^2 + \|\boldsymbol{d}\|_2^2 - 2(\boldsymbol{x}+\boldsymbol{d})^\top \boldsymbol{z} \qquad (A.3) \\
=&\|\boldsymbol{x}+\boldsymbol{d}-\boldsymbol{z}\|_2,
\end{aligned}
$$

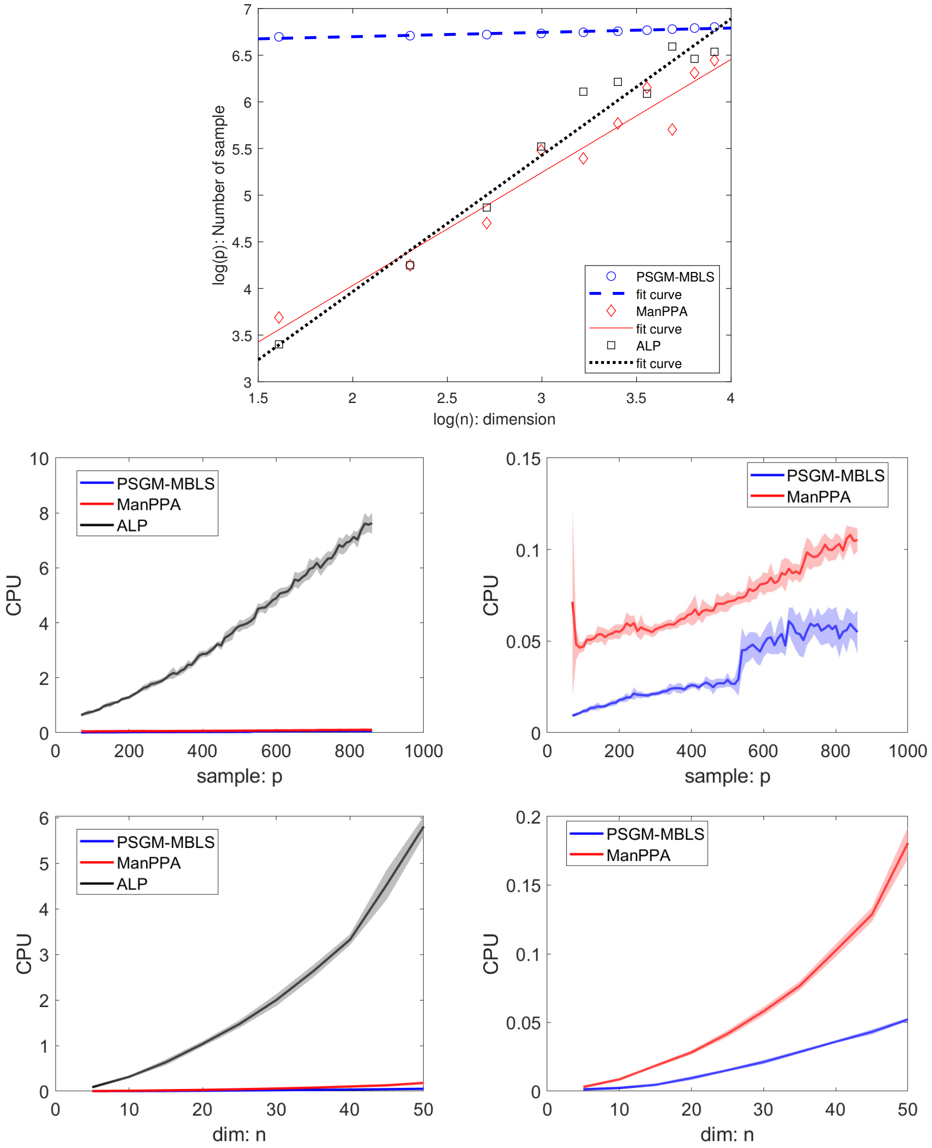


Figure 10: Comparison on the ODL problem (1.4) with $\gamma = 0.1$. First row: Linear fitting curves. Second row: CPU time versus the number of samples p, $(n = 30)$. Third row: CPU time versus the number of dimension n, $(p = 300)$. The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

where (A.3) follows from the fact that $\|\boldsymbol{x}+\boldsymbol{d}\|_2 - 1 = \sqrt{1+\|\boldsymbol{d}\|_2^2} - 1 \leq \frac{1}{2}\|\boldsymbol{d}\|_2^2$. □

B Proof of Proposition 2.1

Since f is Lipschitz with constant L and $\boldsymbol{d}^{k\top}\boldsymbol{x}^k = 0$, we have

$$\left| f(\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \alpha\boldsymbol{d}^k)) - f(\boldsymbol{x}^k + \alpha\boldsymbol{d}^k) \right| \leq L \left\| \frac{\boldsymbol{x}^k + \alpha\boldsymbol{d}^k}{\|\boldsymbol{x}^k + \alpha\boldsymbol{d}^k\|_2} - (\boldsymbol{x}^k + \alpha\boldsymbol{d}^k) \right\|_2 \leq \frac{\alpha^2 L}{2}\|\boldsymbol{d}^k\|_2^2$$

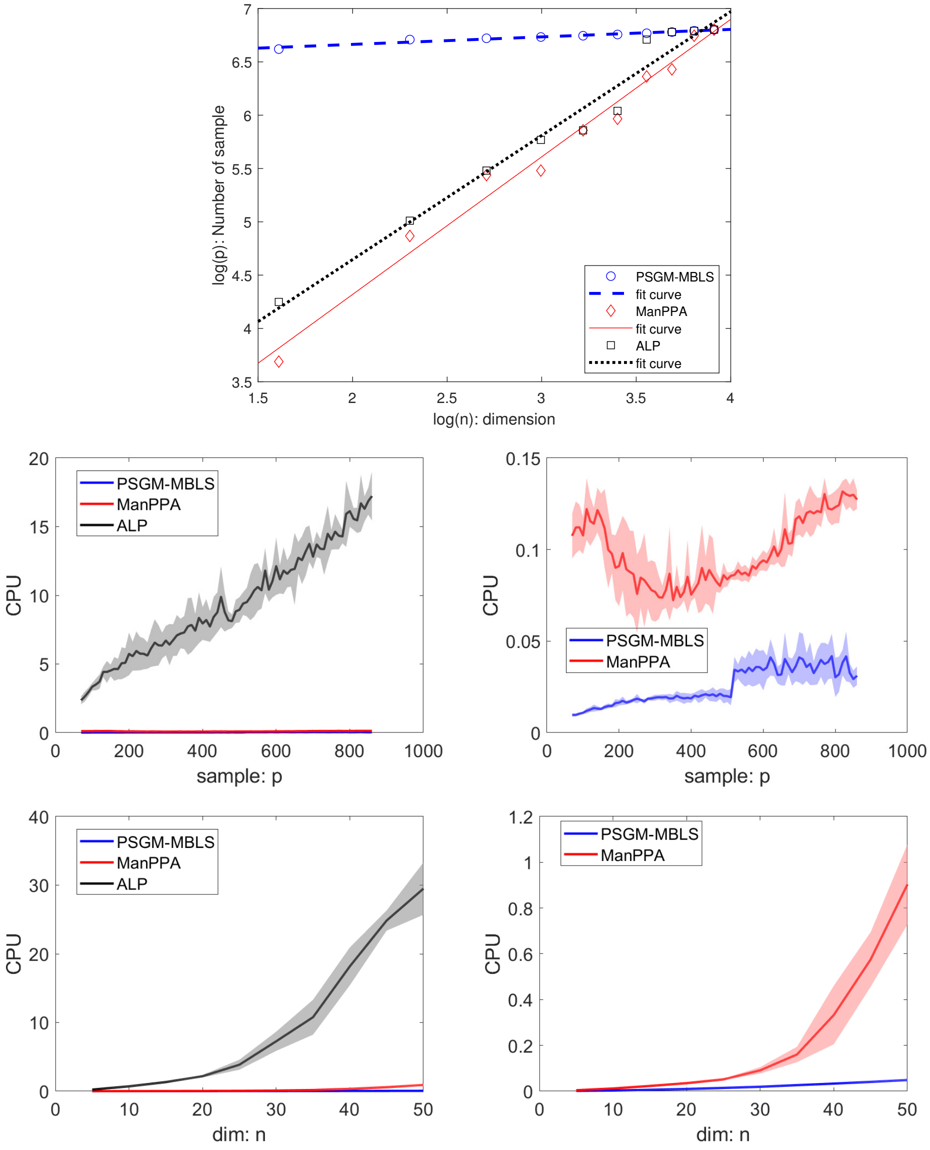


Figure 11: Comparison on the ODL problem (1.4) with $\gamma = 0.3$. First row: Linear fitting curves. Second row: CPU time versus the number of samples p, $(n = 30)$. Third row: CPU time versus the number of dimension n, $(p = 300)$. The shadow area corresponds to the std and the line within the shadow is the mean of 10 random trials.

by Proposition A.1. Hence, for any $\alpha \in (0, \bar{\alpha}]$, we have

$$
\begin{aligned}
f(\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)) &\le f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) + \frac{\alpha^2 L}{2} \|\boldsymbol{d}^k\|_2^2 \\
&\le (1-\alpha) f(\boldsymbol{x}^k) + \alpha f(\boldsymbol{x}^k + \boldsymbol{d}^k) + \frac{\alpha^2 L}{2} \|\boldsymbol{d}^k\|_2^2 \qquad \text{(B.1a)} \\
&\le f(\boldsymbol{x}^k) - \frac{\alpha}{t} \|\boldsymbol{d}^k\|_2^2 + \frac{\alpha^2 L}{2} \|\boldsymbol{d}^k\|_2^2 \qquad \text{(B.1b)} \\
&\le f(\boldsymbol{x}^k) - \frac{\alpha}{2t} \|\boldsymbol{d}^k\|_2^2, \qquad \text{(B.1c)}
\end{aligned}
$$

where (B.1a) follows from the convexity of f, (B.1b) holds because the strong convexity of the objective function in subproblem (2.2), together with the optimality of $\boldsymbol{d} = \boldsymbol{d}^k$ and feasibility of $\boldsymbol{d} = \boldsymbol{0}$ for (2.2), implies that $f(\boldsymbol{x}^k + \boldsymbol{d}^k) + \frac{1}{t}\|\boldsymbol{d}^k\|_2^2 \le f(\boldsymbol{x}^k)$, and (B.1c) is due to $\alpha \le 1/(tL)$. If $t \le 1/L$, then $\bar{\alpha} = 1$. This completes the proof.

C Proof of Theorem 2.4

We begin with two preparatory results. The first states that the restriction of the objective function f in (1.1) on the nonconvex constraint set $\mathcal{M}$ satisfies a Riemannian subgradient inequality, which means that f behaves almost like a convex function on $\mathcal{M}$.

Proposition C.1. *Let $\boldsymbol{x} \in \mathcal{M}$ and $\boldsymbol{d} \in \mathbb{R}^n$ be such that $\boldsymbol{d}^\top \boldsymbol{x} = 0$. Define $\boldsymbol{x}^+ = \boldsymbol{x} + \boldsymbol{d}$. Then, for any $\boldsymbol{z} \in \mathcal{M}$ and $\boldsymbol{s} \in \partial f(\boldsymbol{x}^+)$, we have*

$$f(\boldsymbol{z}) - f(\boldsymbol{x}^+) \ge \langle (\boldsymbol{I}_n - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{s}, \boldsymbol{z} - \boldsymbol{x}^+ \rangle - \frac{L}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2.$$

Proof. Since f is convex on $\mathbb{R}^n$, we have

$$f(\boldsymbol{z}) - f(\boldsymbol{x}^+) \ge \langle \boldsymbol{s}, \boldsymbol{z} - \boldsymbol{x}^+ \rangle = \langle (\boldsymbol{I}_n - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{s}, \boldsymbol{z} - \boldsymbol{x}^+ \rangle + \langle \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{s}, \boldsymbol{z} - \boldsymbol{x}^+ \rangle$$

Now, observe that

$$\begin{aligned} \langle \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{s}, \boldsymbol{z} - \boldsymbol{x}^+ \rangle &= \langle \boldsymbol{s}, \boldsymbol{x}\boldsymbol{x}^\top (\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{d})) \rangle \\ &= \langle \boldsymbol{s}, \boldsymbol{x}(\boldsymbol{x}^\top \boldsymbol{z} - 1) \rangle \qquad \text{(C.1a)} \\ &\ge -\frac{1}{2}\|\boldsymbol{s}\|_2 \|\boldsymbol{z} - \boldsymbol{x}\|_2^2, \qquad \text{(C.1b)} \end{aligned}$$

where (C.1a) is due to $\boldsymbol{x}^\top \boldsymbol{x} = 1$ and $\boldsymbol{d}^\top \boldsymbol{x} = 0$, while (C.1b) follows from the fact that $|\boldsymbol{x}^\top \boldsymbol{z} - 1| = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2$. Since f is Lipschitz with constant L, we have $\|\boldsymbol{s}\|_2 \le L$. □

The second establishes a key recursion for the iterates generated by ManPPA.

Proposition C.2. *Let $\{\boldsymbol{x}^k\}_k$ be the sequence generated by Algorithm 1 with $t \le 1/L$. Then, for any $\bar{\boldsymbol{x}} \in \mathcal{M}$, we have*

$$\|\boldsymbol{x}^{k+1} - \bar{\boldsymbol{x}}\|_2^2 \le (1 + tL)\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 - 2t\left(f(\boldsymbol{x}^k) - f(\bar{\boldsymbol{x}})\right) + t^2L^2.$$

Proof. Since $t \le 1/L$, we have $\boldsymbol{x}^{k+1} = \mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \boldsymbol{d}^k)$ by Proposition 2.1. From the optimality condition of the subproblem (2.2), there exists an $\boldsymbol{s}^k \in \partial f(\boldsymbol{x}^k + \boldsymbol{d}^k)$ such that

$$\boldsymbol{d}^k = -t(\boldsymbol{I}_n - \boldsymbol{x}^k \boldsymbol{x}^{k\top})\boldsymbol{s}^k. \qquad \text{(C.2)}$$

Denoting $\boldsymbol{x}^{k+} = \boldsymbol{x}^k + \boldsymbol{d}^k$, we have

$$\begin{aligned} &\|\boldsymbol{x}^{k+1} - \bar{\boldsymbol{x}}\|_2^2 = \left\|\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \boldsymbol{d}^k) - \bar{\boldsymbol{x}}\right\|_2^2 \\ \le& \left\|\boldsymbol{x}^k + \boldsymbol{d}^k - \bar{\boldsymbol{x}}\right\|_2^2 \qquad \text{(C.3a)} \\ =& \|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 + 2\langle \boldsymbol{d}^k, \boldsymbol{x}^{k+} - \bar{\boldsymbol{x}} \rangle - \|\boldsymbol{d}^k\|_2^2 \\ =& \|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 - 2t\langle (\boldsymbol{I}_n - \boldsymbol{x}^k \boldsymbol{x}^{k\top})\boldsymbol{s}^k, \boldsymbol{x}^{k+} - \bar{\boldsymbol{x}} \rangle - \|\boldsymbol{d}^k\|_2^2 \qquad \text{(C.3b)} \\ \le& (1 + tL)\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 + 2t(f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^{k+})) - \|\boldsymbol{d}^k\|_2^2 \qquad \text{(C.3c)} \\ \le& (1 + tL)\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 + 2t(f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^k)) + 2tL\|\boldsymbol{d}^k\|_2 - \|\boldsymbol{d}^k\|_2^2 \qquad \text{(C.3d)} \\ \le& (1 + tL)\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2 + 2t(f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^k)) + t^2L^2, \qquad \text{(C.3e)} \end{aligned}$$

where (C.3a) follows from Proposition A.2, (C.3b) follows from (C.2), (C.3c) follows from Proposition C.1, (C.3d) follows from the Lipschitz continuity of f, and (C.3e) follows from the fact that $2tL\|\boldsymbol{d}^k\|_2 - \|\boldsymbol{d}^k\|_2^2 = -(\|\boldsymbol{d}^k\|_2 - tL)^2 + t^2L^2 \le t^2L^2$. □

We are now ready to prove Theorem 2.4. We first prove (2.5) by induction. Let $\boldsymbol{x}^* \in \mathcal{X}$ be such that $\operatorname{dist}(\boldsymbol{x}^k, \mathcal{X}) = \|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2$. By invoking Proposition C.2 with $\bar{\boldsymbol{x}} = \boldsymbol{x}^*$, we have

$$\begin{aligned}
&\operatorname{dist}^2(\boldsymbol{x}^{k+1}, \mathcal{X}) \le \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2^2 \\
\le&(1+tL)\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2^2 - 2t\left(f(\boldsymbol{x}^k) - f(\boldsymbol{x}^*)\right) + t^2L^2 \\
\le&(1+tL)\operatorname{dist}^2(\boldsymbol{x}^k, \mathcal{X}) - 2\alpha t \cdot \operatorname{dist}(\boldsymbol{x}^k, \mathcal{X}) + t^2L^2,
\end{aligned}$$

where the last inequality follows from (2.4). Consider the function $[0, \overline{\delta}] \ni s \mapsto \phi(s) = (1 + tL)s^2 - 2t\alpha s + t^2L^2$. Observe that ϕ attains its maximum at $s = \overline{\delta}$ if $\overline{\delta} \ge \frac{2t\alpha}{1+tL}$. Given that $t \le \min\left\{\frac{\overline{\delta}}{2\alpha - L\overline{\delta}}, \frac{2\overline{\delta}\alpha - L\overline{\delta}^2}{L^2}\right\}$, we indeed have $\overline{\delta} \ge \frac{2t\alpha}{1+tL}$ and hence $\phi(s) \le \phi(\overline{\delta}) \le \overline{\delta}^2$ for all $s \in [0, \overline{\delta}]$. In particular, we have $\operatorname{dist}(\boldsymbol{x}^{k+1}, \mathcal{X}) \le \overline{\delta}$ whenever $\operatorname{dist}(\boldsymbol{x}^k, \mathcal{X}) \le \overline{\delta}$. This establishes (2.5).

Next, we prove (2.6). Again, let $\boldsymbol{x}^* \in \mathcal{X}$ be such that $\operatorname{dist}(\boldsymbol{x}^k, \mathcal{X}) = \|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2$. Since $\alpha \le L$ by (2.4), we have $\overline{\delta} \le 1$. This implies that $\boldsymbol{x}^{k\top}\boldsymbol{x}^* = \frac{2 - \|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2^2}{2} \ge \frac{1}{2}$. Hence, the vector $\bar{\boldsymbol{d}}^k = \frac{\boldsymbol{x}^*}{\boldsymbol{x}^{k\top}\boldsymbol{x}^*} - \boldsymbol{x}^k$ is well defined and satisfies $\bar{\boldsymbol{d}}^{k\top}\boldsymbol{x}^k = 0$ (i.e., $\bar{\boldsymbol{d}}^k$ is a tangent vector at $\boldsymbol{x}^k$), $\operatorname{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \bar{\boldsymbol{d}}^k) = \boldsymbol{x}^*$, and $\|\bar{\boldsymbol{d}}^k\|_2 \le \sqrt{3}$. By the strong convexity of the objective function in subproblem (2.2) and noting the optimality of $\boldsymbol{d}^k$ and feasibility of $\bar{\boldsymbol{d}}^k$ for (2.2), we have

$$f(\boldsymbol{x}^k + \boldsymbol{d}^k) + \frac{1}{2t}\|\boldsymbol{d}^k\|_2^2 + \frac{1}{2t}\|\boldsymbol{d}^k - \bar{\boldsymbol{d}}^k\|_2^2 \le f(\boldsymbol{x}^k + \bar{\boldsymbol{d}}^k) + \frac{1}{2t}\|\bar{\boldsymbol{d}}^k\|_2^2. \tag{C.4}$$

Furthermore, by the Lipschitz continuity of f and Proposition A.1, we get

$$f(\boldsymbol{x}^k + \bar{\boldsymbol{d}}^k) \le f(\operatorname{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \bar{\boldsymbol{d}}^k)) + \frac{L}{2}\|\bar{\boldsymbol{d}}^k\|_2^2 = f(\boldsymbol{x}^*) + \frac{L}{2}\|\bar{\boldsymbol{d}}^k\|_2^2, \tag{C.5a}$$

$$f(\boldsymbol{x}^{k+1}) = f(\operatorname{Proj}_{\mathcal{M}}(\boldsymbol{x}^k + \boldsymbol{d}^k)) \le f(\boldsymbol{x}^k + \boldsymbol{d}^k) + \frac{L}{2}\|\boldsymbol{d}^k\|_2^2. \tag{C.5b}$$

Combining (C.4), (C.5a), and (C.5b), we have

$$f(\boldsymbol{x}^{k+1}) + \left(\frac{1}{2t} - \frac{L}{2}\right)\|\boldsymbol{d}^k\|_2^2 + \frac{1}{2t}\|\boldsymbol{d}^k - \bar{\boldsymbol{d}}^k\|_2^2 \le f(\boldsymbol{x}^*) + \left(\frac{L}{2} + \frac{1}{2t}\right)\|\bar{\boldsymbol{d}}^k\|_2^2. \tag{C.6}$$

Since $t \le \frac{\overline{\delta}}{2\alpha - L\overline{\delta}} \le 1/L$, we have $\frac{1}{2t} - \frac{L}{2} \ge 0$. Moreover, since $\operatorname{dist}(\boldsymbol{x}^{k+1}, \mathcal{X}) \le \overline{\delta} \le \delta$, we have $f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^*) \ge \alpha \cdot \operatorname{dist}(\boldsymbol{x}^{k+1}, \mathcal{X})$ by (2.4). It then follows from (C.6) and Proposition A.1 that

$$\alpha \cdot \operatorname{dist}(\boldsymbol{x}^{k+1}, \mathcal{X}) \le \left(\frac{L}{2} + \frac{1}{2t}\right)\|\bar{\boldsymbol{d}}^k\|_2^2 \le 8\left(\frac{L}{2} + \frac{1}{2t}\right)\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2^2 = 4\left(L + \frac{1}{t}\right)\operatorname{dist}^2(\boldsymbol{x}^k, \mathcal{X}).$$

This completes the proof.

D Convergence Results for Inexact ALM and SSN

The convergence behavior of the inexact ALM (Algorithm 2) solving problem (2.7) and the SSN method (Algorithm 3) for solving the nonsmooth equation (2.11) can be deduced from existing results in the literature. We begin with the convergence result for the inexact ALM.

Proposition D.1. *Let $\{(\boldsymbol{d}^j, \boldsymbol{u}^j, y^j, \boldsymbol{z}^j)\}_j$ be the sequence generated by Algorithm 2 with stopping criterion (2.10a). Then, the sequence $\{(\boldsymbol{d}^j, \boldsymbol{u}^j)\}_j$ is bounded and converges to the unique optimal solution to problem (2.7). Moreover, if $0 < \sigma_j \nearrow \sigma_\infty = \infty$ and the stopping criteria (2.10b) and (2.10c) are also used, then for all sufficiently large j, the sequence $\{(\boldsymbol{d}^j, \boldsymbol{u}^j, y^j, \boldsymbol{z}^j)\}_j$ converges asymptotically superlinearly to the set of KKT points of (2.7).*

Proof. Observe that the function $\boldsymbol{d} \mapsto \ell(\boldsymbol{d}) := \frac{1}{2}\|\boldsymbol{d}\|_2^2$ is strongly convex, self-conjugate (i.e., $\ell^*(\boldsymbol{d}) = \ell(\boldsymbol{d})$), and has a Lipschitz continuous gradient. Moreover, the conjugate of the indicator function

$$\boldsymbol{u} \mapsto \mathbb{I}_{\{\|\cdot\|_\infty \le t\}}(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \|\boldsymbol{u}\|_\infty \le t, \\ +\infty & \text{otherwise} \end{cases}$$

is $\boldsymbol{u} \mapsto h(\boldsymbol{u}) = t\|\boldsymbol{u}\|_1$. Hence, we may write the dual of problem (2.7) as

$$\max_{y\in\mathbb{R},\ \boldsymbol{z}\in\mathbb{R}^p} \ -\left(\frac{1}{2}\|\boldsymbol{Y}\boldsymbol{z} + y\boldsymbol{x}\|_2^2 + \boldsymbol{c}^\top \boldsymbol{z} + \mathbb{I}_{\{\|\cdot\|_\infty \le t\}}(-\boldsymbol{z})\right). \tag{D.1}$$

It is easy to verify that problem (2.7) has a unique optimal solution, and that problem (D.1) satisfies the Slater condition and its optimal solution set is also nonempty. It follows from [21, Theorem 4] that the first part of Proposition D.1 holds.

Next, let $g : \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}$ denote the objective function in problem (D.1); i.e.,

$$g(y, \boldsymbol{z}) := \frac{1}{2}\|\boldsymbol{Y}\boldsymbol{z} + y\boldsymbol{x}\|_2^2 + \boldsymbol{c}^\top \boldsymbol{z} + \mathbb{I}_{\{\|\cdot\|_\infty \le t\}}(-\boldsymbol{z}).$$

Furthermore, define the mapping $\Gamma : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \rightrightarrows \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}$ associated with the primal-dual pair (2.7) and (D.1) by

$$\Gamma(\boldsymbol{d}, \boldsymbol{u}; y, \boldsymbol{z}) := \left\{ (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \pi_4) \ \middle|\ \begin{array}{rcl} \boldsymbol{\pi}_1 & = & \boldsymbol{d} - y\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{z}, \\ \boldsymbol{\pi}_2 & \in & t\partial\|\boldsymbol{u}\|_1 + \boldsymbol{z}, \\ -\boldsymbol{\pi}_3 & = & \boldsymbol{Y}^\top \boldsymbol{d} - \boldsymbol{u} + \boldsymbol{c}, \\ -\pi_4 & = & \boldsymbol{d}^\top \boldsymbol{x}. \end{array} \right\}$$

It is easy to verify that if $(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, 0) \in \Gamma(\bar{\boldsymbol{d}}, \bar{\boldsymbol{u}}; \bar{y}, \bar{\boldsymbol{z}})$, then $(\bar{\boldsymbol{d}}, \bar{\boldsymbol{u}})$ is optimal for problem (2.7) and $(\bar{y}, \bar{\boldsymbol{z}})$ is optimal for problem (D.1).

Now, it is well known (see, e.g., [32, Section 4.2] and the references therein) that $(y, \boldsymbol{z}) \mapsto \partial g(y, \boldsymbol{z})$ is a polyhedral multifunction; i.e., its graph

$$\text{gph}(\partial g) := \{(y, \boldsymbol{z}; s, \boldsymbol{t}) \in \mathbb{R} \times \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \mid (s, \boldsymbol{t}) \in \partial g(y, \boldsymbol{z})\}$$

is the union of a finite collection of polyhedral convex sets. Moreover, using the fact that $\boldsymbol{s} \in \partial\|\boldsymbol{u}\|_1$ if and only if

$$s_i \in \begin{cases} \{1\} & \text{if } u_i > 0, \\ [-1, 1] & \text{if } u_i = 0, \\ \{-1\} & \text{if } u_i < 0, \end{cases}$$

it can be verified that the KKT mapping Γ is also a polyhedral multifunction. Hence, by invoking [15, Proposition 2], [32, Fact 2] and following the arguments in the proof of [15, Theorem 3.3], we conclude that the second part of Proposition D.1 holds. □

Next, we have the following convergence result for the SSN method.

Proposition D.2. *The sequence $\{\boldsymbol{d}^j\}_j$ generated by Algorithm 3 converges superlinearly to the unique optimal solution $\bar{\boldsymbol{d}}$ to the nonsmooth equation (2.11).*

Proof. This follows from the fact that $\text{prox}_{h/\sigma}$ is strongly semismooth (see, e.g., [15, Definition 3.5] for the definition) and the arguments in the proof of [15, Theorem 3.6]. □

E Proof of Theorem 3.2

Let $\hat{\boldsymbol{x}}^k = \operatorname{mprox}_{\lambda f}(\boldsymbol{x}^k) \in \mathcal{M}$. By definition of e_λ in (3.4a) and Proposition A.2, we have

$$e_\lambda(\boldsymbol{x}^{k+1}) \le f(\hat{\boldsymbol{x}}^k) + \frac{1}{2\lambda}\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^{k+1}\|_2^2 \le f(\hat{\boldsymbol{x}}^k) + \frac{1}{2\lambda}\|\hat{\boldsymbol{x}}^k - (\boldsymbol{x}^k + \boldsymbol{d}^k)\|_2^2. \tag{E.1}$$

From the optimality condition of (3.1), we get

$$\boldsymbol{d}^k \in -t_k(\boldsymbol{I}_n - \boldsymbol{x}^k\boldsymbol{x}^{k\top})\partial f_{j_k}(\boldsymbol{x}^k + \boldsymbol{d}^k).$$

Hence, we compute

$$\begin{aligned}
&\|\hat{\boldsymbol{x}}^k - (\boldsymbol{x}^k + \boldsymbol{d}^k)\|_2^2 \\
=&\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 - \|\boldsymbol{d}^k\|_2^2 - 2\langle \hat{\boldsymbol{x}}^k - \boldsymbol{x}^k - \boldsymbol{d}^k, \boldsymbol{d}^k\rangle \\
\le&\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 - \|\boldsymbol{d}^k\|_2^2 + 2t_k\left(f_{j_k}(\hat{\boldsymbol{x}}^k) - f_{j_k}(\boldsymbol{x}^k + \boldsymbol{d}^k) + \frac{L_{j_k}}{2}\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2\right) &\text{(E.2a)}\\
\le&\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 - \|\boldsymbol{d}^k\|_2^2 + 2t_k\left(f_{j_k}(\hat{\boldsymbol{x}}^k) - f_{j_k}(\boldsymbol{x}^k)\right) + 2t_kL_{j_k}\left(\frac{1}{2}\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \|\boldsymbol{d}^k\|_2\right), &\text{(E.2b)}
\end{aligned}$$

where (E.2a) follows from Proposition C.1 and (E.2b) is due to the Lipschitz continuity of f_{j_k}. Upon taking the expectation on both sides of (E.2) with respect to j_k conditioned on $\boldsymbol{x}^k$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\|\hat{\boldsymbol{x}}^k - (\boldsymbol{x}^k + \boldsymbol{d}^k)\|_2^2 \mid \boldsymbol{x}^k\right] \\
\le&(1 + t_k\bar{L})\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \frac{2t_k}{p}\left(f(\hat{\boldsymbol{x}}^k) - f(\boldsymbol{x}^k)\right) + 2t_k\bar{L}\cdot\mathbb{E}\left[\|\boldsymbol{d}^k\|_2 \mid \boldsymbol{x}^k\right] - \mathbb{E}\left[\|\boldsymbol{d}^k\|_2^2 \mid \boldsymbol{x}^k\right] \\
\le&(1 + t_k\bar{L})\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \frac{2t_k}{p}\left(f(\hat{\boldsymbol{x}}^k) - f(\boldsymbol{x}^k)\right) + t_k^2\bar{L}^2 &\text{(E.3a)}\\
=&\left(1 + t_k\bar{L} - \frac{t_k}{p\lambda}\right)\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \frac{2t_k}{p}(e_\lambda(\boldsymbol{x}^k) - f(\boldsymbol{x}^k)) + t_k^2\bar{L}^2 \\
\le&\left(1 + t_k\bar{L} - \frac{t_k}{p\lambda}\right)\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + t_k^2\bar{L}^2, &\text{(E.3b)}
\end{aligned}$$

where (E.3a) follows from the fact that $\mathbb{E}\left[\|\boldsymbol{d}^k\|_2 \mid \boldsymbol{x}^k\right] \le \sqrt{\mathbb{E}\left[\|\boldsymbol{d}^k\|_2^2 \mid \boldsymbol{x}^k\right]}$ and $a\sqrt{x} - x \le a^2/4$ for any $a, x \ge 0$; (E.3b) follows from the definition of e_λ. Putting (E.1) and (E.3b) together gives

$$\begin{aligned}
e_\lambda(\boldsymbol{x}^{k+1}) &\le f(\hat{\boldsymbol{x}}^k) + \frac{1}{2\lambda}\left(1 + t_k\bar{L} - \frac{t_k}{p\lambda}\right)\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \frac{t_k^2\bar{L}^2}{2\lambda} \\
&= e_\lambda(\boldsymbol{x}^k) + \frac{(\bar{L} - 1/(p\lambda))t_k}{2\lambda}\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2 + \frac{t_k^2\bar{L}^2}{2\lambda}.
\end{aligned}$$

Taking expectation on both sides with respect to $\boldsymbol{x}^k$ yields

$$\frac{(1/(p\lambda) - \bar{L})t_k}{2\lambda}\mathbb{E}\left[\|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2^2\right] \le \mathbb{E}\left[e_\lambda(\boldsymbol{x}^k)\right] - \mathbb{E}\left[e_\lambda(\boldsymbol{x}^{k+1})\right] + \frac{t_k^2\bar{L}^2}{2\lambda}.$$

Upon summing the above inequality over $k = 0, 1, \ldots T$ and noting that $\lambda < 1/(p\bar{L})$ and $e_\lambda(\boldsymbol{z}) \ge 0$ for any $\boldsymbol{z} \in \mathbb{R}^n$, we obtain

$$\sum_{k=0}^{T} t_k\mathbb{E}\left[\frac{1}{\lambda^2}\|\boldsymbol{x}^k - \operatorname{mprox}_{\lambda f}(\boldsymbol{x}^k)\|_2^2\right] \le \frac{2}{1/p - \lambda\bar{L}}e_\lambda(\boldsymbol{x}^0) + \frac{\bar{L}^2}{\lambda(1/p - \lambda\bar{L})}\sum_{k=0}^{T} t_k^2.$$

Upon dividing both sides of the above inequality by $\sum_{k=0}^{T} t_k$ and noting that the left-hand side becomes $\mathbb{E}\left[\Theta_\lambda(\bar{\boldsymbol{x}})^2\right]$, the proof is complete.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, 2009. 2

[2] Y. Bai, Q. Jiang, and J. Sun. Subgradient descent learns orthogonal dictionaries. In *ICLR*, 2019. 2, 3, 6, 7, 17

[3] G. C. Bento, O. P. Ferreira, and J. G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *J. Optim. Theory Appl.*, 173:548–562, 2017. 4

[4] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009. 2

[5] J. V. Burke and M. C. Ferris. Weak sharp minima in mathematical programming. *SIAM J. Control Optim.*, 31(5):1340–1359, 1993. 3, 7

[6] S. Chen, Z. Deng, S. Ma, and A. M.-C. So. Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning. *arXiv preprint https://arxiv.org/abs/2005.02356*, 2020. 14

[7] S. Chen, S. Ma, A. M.-C. So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM J. Optim.*, 30(1):210–239, 2020. 4, 5, 6

[8] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Society for Industrial and Applied Mathematics, 1990. 9

[9] O. P. Ferreira and P. R. Oliveira. Proximal point algorithm on Riemannian manifolds. *Optim.*, 51(2):257–270, 2002. 4

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, 2013. 16

[11] D. Gilboa, S. Buchanan, and J. Wright. Efficient dictionary learning with gradient descent. In *ICML*, pages 2252–2259, 2019. 2

[12] G. Lerman and T. Maunu. An overview of robust subspace recovery. *Proc. IEEE*, 106(8):1380–1410, 2018. 2, 13

[13] G. Lerman, M. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Found. Comput. Math.*, 15(2):363–410, 2015. 12

[14] X. Li, S. Chen, Z. Deng, Q. Qu, Z. Zhu, and A. M.-C. So. Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM J. Optim.*, 31(3):1605–1634, 2021. 3, 4, 6, 7, 12

[15] X. Li, D. Sun, and K.-C. Toh. A highly efficient semi-smooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM J. Optim.*, 28:433–458, 2018. 7, 8, 9, 14, 27

[16] X. Li, Z. Zhu, A. M.-C. So, and R. Vidal. Nonconvex robust low-rank matrix recovery. *SIAM J. Optim.*, 30(1):660–686, 2020. 3

[17] H. Liu, A. M.-C. So, and W. Wu. Quadratic optimization with orthogonality constraint: Explicit Łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. *Math. Programm., Ser. A*, 178(1–2):215–262, 2019. 3

[18] H. Liu, M.-C. Yue, and A. M.-C. So. On the estimation performance and convergence rate of the generalized power method for phase synchronization. *SIAM J. Optim.*, 27(4):2426–2446, 2017. 3

[19] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in subspace: linear sparsity using alternating directions. *IEEE Trans. Inf. Theory*, 62(10):5855–5880, 2016. 2

[20] Q. Qu, Y. Zhai, X. Li, Y. Zhang, and Z. Zhu. Geometric analysis of nonconvex optimization landscapes for overcomplete learning. In *ICLR*, 2020. 2

[21] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976. 8, 27

[22] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14:877–898, 1976. 5, 7

[23] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer–Verlag, 2004. 9

[24] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proc. IEEE*, 98(6):1045–1057, 2010. 2

[25] D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *COLT*, 2012. 2, 17

[26] J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Trans. Inf. Theory*, 63(2):853–884, 2017. 2

[27] J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere II: Recovery by Riemannian trust-region method. *IEEE Trans. Inf. Theory*, 63(2):885–914, 2017. 2, 13

[28] M. C. Tsakiris and R. Vidal. Dual principal component pursuit. *J. Mach. Learn. Res.*, 19(18):1–50, 2018. 2, 12

[29] B. Wang, S. Ma, and L. Xue. Riemannian stochastic proximal gradient methods for nonsmooth optimization over the Stiefel manifold. *arXiv preprint arXiv:2005.01209*, 2020. 11

[30] P. Wang, H. Liu, and A. M.-C. So. Globally convergent accelerated proximal alternating maximization method for L1–principal component analysis. In *ICASSP*, pages 8147–8151, 2019. 13

[31] W. H. Yang, L.-H. Zhang, and R. Song. Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific J. Optim.*, 10(2):415–434, 2014. 3, 5

[32] Z. Zhou and A. M.-C. So. A unified approach to error bounds for structured convex optimization problems. *Math. Programm., Ser. A*, 165(2):689–728, 2017. 3, 27

[33] Z. Zhu, Y. Wang, D. Robinson, D. Naiman, R. Vidal, and M. Tsakiris. Dual principal component pursuit: Improved analysis and efficient algorithms. In *NeurIPS*, pages 2171–2181, 2018. 2, 3, 7, 14, 15, 16