REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
07-11-2012	Technical Report	-

4. TITLE AND SUBTITLE
Matrix Recipes for Hard Thresholding Methods

5a. CONTRACT NUMBER
W911NF-09-1-0383

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611103

6. AUTHORS
Anastasios Kyrillidis, Volkan Cevher

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
William Marsh Rice University
Office of Sponsored Research
William Marsh Rice University
Houston, TX 77005 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
56177-CS-MUR.144

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Given a set of possibly corrupted and incomplete linear measurements, we leverage low-dimensional models to best explain the data for provable solution quality in inversion. A non-exhaustive list of examples includes sparse vector and low-rank matrix approximation. Most of the well-known low dimensional models are inherently non-convex. However, recent approaches prefer convex surrogates that “relax” the problem in order to establish solution uniqueness and stability. In this paper, we tackle the linear inverse problems revolving around low-rank matrices by

15. SUBJECT TERMS
Affine rank minimization, compressed sensing, sparse approximation algorithms, hard thresholding, greedy methods, ?-approximation schemes, randomized algorithms.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Richard Baraniuk
UU	UU	UU	UU		19b. TELEPHONE NUMBER: 713-348-5132

Report Title

Matrix Recipes for Hard Thresholding Methods

ABSTRACT

Given a set of possibly corrupted and incomplete linear measurements, we leverage low-dimensional models to best explain the data for provable solution quality in inversion. A non-exhaustive list of examples includes sparse vector and low-rank matrix approximation. Most of the well-known low dimensional models are inherently non-convex. However, recent approaches prefer convex surrogates that “relax” the problem in order to establish solution uniqueness and stability. In this paper, we tackle the linear inverse problems revolving around low-rank matrices by preserving their non-convex structure. To this end, we present and analyze a new set of sparse and low-rank recovery algorithms within the class of hard thresholding methods. We provide strategies on how to set up these algorithms via basic “ingredients” for different configurations to achieve complexity vs. accuracy tradeoffs. Moreover, we propose acceleration schemes by utilizing memory-based techniques and randomized, ?-approximate, low-rank projections to speed-up the convergence as well as decrease the computational costs in the recovery process. For all these cases, we present theoretical analysis that guarantees convergence under mild problem conditions. Simulation results demonstrate notable performance improvements compared to state-of-the-art algorithms both in terms of data reconstruction and computational complexity.

Matrix Recipes for Hard Thresholding Methods

Anastasios Kyrillidis and Volkan Cevher

Abstract

Given a set of possibly corrupted and incomplete linear measurements, we leverage low-dimensional models to best explain the data for provable solution quality in inversion. A non-exhaustive list of examples includes sparse vector and low-rank matrix approximation. Most of the well-known low dimensional models are inherently non-convex. However, recent approaches prefer convex surrogates that "relax" the problem in order to establish solution uniqueness and stability. In this paper, we tackle the linear inverse problems revolving around low-rank matrices by preserving their non-convex structure. To this end, we present and analyze a new set of sparse and low-rank recovery algorithms within the class of hard thresholding methods. We provide strategies on how to set up these algorithms via basic "ingredients" for different configurations to achieve complexity vs. accuracy tradeoffs. Moreover, we propose acceleration schemes by utilizing memory-based techniques and randomized, ϵ-approximate, low-rank projections to speed-up the convergence as well as decrease the computational costs in the recovery process. For all these cases, we present theoretical analysis that guarantees convergence under mild problem conditions. Simulation results demonstrate notable performance improvements compared to state-of-the-art algorithms both in terms of data reconstruction and computational complexity.

Index Terms

Affine rank minimization, compressed sensing, sparse approximation algorithms, hard thresholding, greedy methods, ϵ-approximation schemes, randomized algorithms.

I. Introduction

A. Background

Compressed Sensing (CS) theory [1] lies at the heart of exciting developments in the areas of signal processing and convex/discrete optimization. CS roughly states that a *sparse vector* signal can be perfectly reconstructed from far fewer samples, compared to its ambient dimension, as dictated by the well-known Nyquist-Shannon theorem. To accomplish this, efficient sparse approximation algorithms have been proposed, accompanied with strong theoretical convergence and approximation guarantees [2].

CS theory is based on the sparse transform coding technique. To describe the main idea, let us assume that $\boldsymbol{x}^* \in \mathcal{R}^n$ is a dense n-dimensional vector of interest. Instead of processing $\boldsymbol{x}^*$ in its dense representation, the literature today offers signal basis transforms that promote data compression and sparsity. Thus, using the appropriate orthonormal basis matrix $\boldsymbol{\Psi} \in \mathcal{R}^{n\times n}$, $\boldsymbol{x}^*$ can be described as a s-sparse ($s \ll n$) linear combination of atoms $\{\boldsymbol{\psi}_i\}_{i=1}^n$ corresponding to the columns of $\boldsymbol{\Psi}$. This representation can be either exact, $\boldsymbol{x}^* = \boldsymbol{\Psi}\boldsymbol{\alpha}$, or approximate, $\boldsymbol{x}^* \approx \boldsymbol{\Psi}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathcal{R}^n$ denotes the set of coefficients with only s out of n entries being nonzero. Without loss of generality, we assume $\boldsymbol{\Psi}$ is the *identity matrix*, unless otherwise stated.

Based on this key observation, CS proposes an alternative sampling theorem where the samples are acquired through linear projections of the signal of interest using a (random) measurement matrix and the number of measurements needed for signal recovery is much less compared to traditional sampling techniques.

B. CS problem statement

In the standard CS framework, we seek to reconstruct a high-dimensional signal $\boldsymbol{x}^* \in \mathcal{R}^n$ through a low-dimensional observation vector $\boldsymbol{y} \in \mathcal{R}^m$ ($m < n$) such that:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}^* + \boldsymbol{\varepsilon}. \tag{1}$$

In this setting, $\boldsymbol{\Phi} \in \mathcal{R}^{m\times n}$ represents the measurement/sensing matrix and $\boldsymbol{\varepsilon} \in \mathcal{R}^m$ is an additive noise term.

To recover $\boldsymbol{x}^*$ given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$, *unconstrained* least-squares method is the classic approach to the solution of linear systems by minimizing the data error function $g(\boldsymbol{x}) := \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_2^2$ where $\|\cdot\|_q$ denotes the ℓ_q-norm. Nevertheless, the reconstruction of $\boldsymbol{x}^*$ from $\boldsymbol{y}$ is an ill-posed problem and there is no hope in finding the *true vector* without ambiguity; there is an infinite number of possible solutions that satisfy the linear system of equations. Therefore, additional signal prior knowledge should be exploited in the optimization solver. Using the fact that $\boldsymbol{x}^*$ is s-sparse, we concentrate on the following constrained minimization problem:

$$\underset{\boldsymbol{x}\in\mathcal{R}^n}{\text{minimize}} \quad g(\boldsymbol{x}) \quad \text{subject to} \quad \|\boldsymbol{x}\|_0 \leq s, \tag{2}$$

where $\|\boldsymbol{x}\|_0$ is the "norm" that counts the non-zeros in $\boldsymbol{x}$.

CS theory plays an important role in solving (2): assuming signal sparsity, the true solution $\boldsymbol{x}^*$ can be found using $m \ll n$ measurements, as long as the geometry of sparse signals is preserved after projection on the subspace defined by $\boldsymbol{\Phi}$. To achieve this, CS community concentrates on developing polynomial-time algorithms for sparse signal recovery from a limited number of non-adaptive samples. Although the collection of works in this direction grows fast, the problem of constructing efficient methods both in execution time and signal recovery performance remains widely open [2], [3].

C. From compressed sensing to rank minimization

In the general affine rank minimization (ARM) problem, the set of observations $\boldsymbol{y} \in \mathcal{R}^p$ is acquired as $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\boldsymbol{X}^* + \boldsymbol{\varepsilon}$ where $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ and $\boldsymbol{X}^* \in \mathcal{R}^{m\times n}$ is the rank-k matrix ($k \ll \min\{m, n\}$) that we desire to recover. As in CS, the challenge is to reconstruct the true low-rank matrix given that $p \ll m \times n$. For this purpose, we are interested in finding the simplest solution $\boldsymbol{X}$ of minimum rank that minimizes the data error $f(\boldsymbol{X}) := \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}\|_2^2$:

$$\underset{\boldsymbol{X}\in\mathcal{R}^{m\times n}}{\text{minimize}} \quad f(\boldsymbol{X}) \quad \text{subject to} \quad \text{rank}(\boldsymbol{X}) \leq k. \tag{3}$$

The ARM problem appears in many applications; low-dimensional embedding [3], matrix completion [4], image compression [5], function learning [6], [7] just to name a few.

In (2), the minimization problem can be equivalently rewritten as:

$$\underset{\boldsymbol{X}\in\mathcal{L}^{n}}{\text{minimize}} \quad f(\boldsymbol{X}) \quad \text{subject to} \quad \text{rank}(\boldsymbol{X}) \leq k, \tag{4}$$

where $\mathcal{L}^n$ denotes the set of square $n \times n$ diagonal matrices and $\boldsymbol{\mathcal{A}} : \mathcal{R}^{n\times n} \to \mathcal{R}^m$ is a generic linear operator such that, given $\boldsymbol{\Phi} \in \mathcal{R}^{m\times n}$ in (1), the solutions $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{X}}$ of (2) and (4), respectively, satisfy $\boldsymbol{\mathcal{A}}\widehat{\boldsymbol{X}} = \boldsymbol{\Phi}\widehat{\boldsymbol{x}}$ for $\widehat{\boldsymbol{X}} \in \mathcal{L}^n$ with $\widehat{\boldsymbol{x}} \in \mathcal{R}^n$ on the main diagonal. In other words, the problem in (2) is a special case of (3).

We present below some characteristic examples for the linear operator $\boldsymbol{\mathcal{A}}$:

Matrix Completion (MC): As a motivating example, consider the famous Netflix problem [8], a recommender system problem where user movie preferences are inferred by a limited subset of entries in a database. Matrix completion is an ill-posed problem since the number of variables exceeds the number of observations and, furthermore, there are entries for which *no knowledge* is available.

To set-up the optimization problem, let $\Omega = \{(i, j) : [\boldsymbol{X}^*]_{ij} \text{ is known}\} \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ be the set of ordered pairs that represent the coordinates of the observable entries—here, $[\boldsymbol{X}^*]_{ij}$ represents the element at the i-th row and j-th column of the matrix $\boldsymbol{X}^*$. The matrix completion problem can be solved as:

$$\underset{\boldsymbol{X}\in\mathcal{R}^{m\times n}}{\text{minimize}} \quad \left\|\boldsymbol{\mathcal{A}}_\Omega\boldsymbol{X} - \boldsymbol{\mathcal{A}}_\Omega\boldsymbol{X}^*\right\|_F \quad \text{subject to} \quad \text{rank}(\boldsymbol{X}) \leq k. \tag{5}$$

where $\boldsymbol{\mathcal{A}}_\Omega$ defines a linear mask over the observable entries Ω. We observe that (5) is a special case of the low-rank ARM problem.

Principal Component Analysis (PCA) and Robust PCA: In the PCA problem, we are interested in reconstructing $\boldsymbol{X}^*$ from the *noisy* set of observations $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\boldsymbol{X}^* + \boldsymbol{\varepsilon}$ where $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ is an identity linear map with $p = m \times n$. Furthermore, combining ideas from sparse and low-rank signal recovery methods, Robust PCA (RPCA) problem deals with the challenge of recovering a low rank $\mathbf{L}^*$ and a sparse component $\mathbf{M}^*$ from a *complete data matrix* $\mathbf{Y}$ such that $\mathbf{Y} = \mathbf{L}^* + \mathbf{M}^* + \boldsymbol{\varepsilon}$. We extend the proposed framework for this case in [9].

General linear maps: We can consider many other cases where $\boldsymbol{\mathcal{A}}$ is any *arbitrary* linear map. As an example, in the experiments we consider the case where $\boldsymbol{\mathcal{A}}$ is constituted by permuted noiselets [10]. Recent developments further indicate connections of ridge function learning with the ARM problem with general linear operators $\boldsymbol{\mathcal{A}}$ [6].

D. Two camps of recovery algorithms

Due to the non-convexity of $\|\cdot\|_0$ and rank$(\cdot)$ constraints, there are no known polynomial-time algorithms that solve (2), (3) with guaranteed optimality, without further assumptions on the measurement operator. As a consequence, various convex relaxations have been proposed to approximate the solution. In [11], Donoho et al. demonstrate that, in the sparse approximation problem, under basic incoherence properties of the sensing matrix $\boldsymbol{\Phi}$ and given $\boldsymbol{x}^*$ is sufficiently sparse, the combinatorial "norm" $\|\cdot\|_0$ in (2) can be substituted by its sparsity-inducing convex surrogate $\|\cdot\|_1$ with provable guarantees for unique signal recovery. In the ARM problem, Fazel et al. [12] identified the nuclear norm $\|\boldsymbol{X}\|_* := \sum_{i=1}^{\text{rank}(\boldsymbol{X})} \sigma_i$ as a convex surrogate of rank$(\boldsymbol{X})$ operator where we can leverage second-order optimization approaches, such as interior-point methods—here, σ_i denotes the i-th singular value of $\boldsymbol{X}$. Under basic incoherence properties of the sensing matrix $\boldsymbol{\mathcal{A}}$, [12] provides provable guarantees for unique signal recovery.

Here, we restrict our attention to low-rank minimization problem formulations, alternative to (3). Given the above, the equality-constrained nuclear-norm minimization problem:

$$\underset{\boldsymbol{X}\in\mathcal{R}^{m\times n}}{\text{minimize}} \quad \|\boldsymbol{X}\|_* \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\mathcal{A}}\boldsymbol{X}, \tag{6}$$

for the noiseless case and the regularized optimization problem in the presence of noise:

$$\underset{\boldsymbol{X}\in\mathcal{R}^{m\times n}}{\text{minimize}} \quad \frac{1}{2}f(\boldsymbol{X}) + \tau\|\boldsymbol{X}\|_*, \tag{7}$$

emerge as natural estimators of $\boldsymbol{X}^*$—in (7), $\tau > 0$ balances the error norm and the rank of the solution. Based on the LASSO operator for the vector case [13], (3) can also be relaxed to:

$$\underset{\boldsymbol{X}\in\mathcal{R}^{m\times n}}{\text{minimize}} \quad f(\boldsymbol{X}) \quad \text{subject to} \quad \|\boldsymbol{X}\|_* \leq \lambda, \tag{8}$$

where $\lambda > 0$ is a regularization parameter that governs the rank of the solution.

Once (2), (3) are relaxed to a convex problem, decades of knowledge on convex analysis and optimization can be leveraged. Interior point methods find a solution with fixed precision in polynomial time but their complexity might be prohibitive even for moderate-sized problems [14], [15]. More suitable for large-scale data analysis, first-order methods constitute low-complexity alternatives [16]–[19].

In contrast to the conventional convex relaxation approaches, iterative greedy algorithms maintain the combinatorial nature of (2), (3). Unfortunately, solving (2), (3) optimally is in general NP-hard [20]. Due to this computational intractability, the algorithms in this class greedily refine a s-sparse/rank-k solution using only "local" information available at the current iteration [21], [22].

E. Contributions.

In this work, we mainly focus on the ARM problem, unless otherwise stated, and exploit a special class of iterative algorithms known as the hard thresholding methods. Similar results can be derived for the vector case in a straightforward way [23]. Note that the transition from sparse vector approximation to ARM is non-trivial; while s-sparse signals "live" in the union of finite number of subspaces, the set of rank-k matrices expand to infinitely many subspaces. Thus, the selection rules do not generalize in a straightforward way.

To this end, we propose and analyze acceleration schemes for this class of algorithms with applications to low rank matrix approximation in linear inverse systems. Our contributions are the following:

"Ingredients" of hard thresholding methods: We analyze the behaviour and performance of hard thresholding methods from a global perspective. Three basic building blocks ("ingredients") are studied: i) step size selection μ_i, ii) memory exploitation, and iii) gradient or least-squares updates over restricted low-rank subspaces (e.g., adaptive block-coordinate descent). We highlight the impact of these key pieces on the convergence rate and signal reconstruction performance and provide optimal and/or efficient strategies on how to set up these "ingredients" under different problem conditions.

Low-rank matrix approximations in hard thresholding methods: In [24], the authors indicate the connection of combinatorial model projections in CS with the modular set function optimization problem. [24] shows that the solution efficiency can be significantly improved by ϵ-approximation algorithms. Based on this new algorithmic definition, we analyze the impact of ϵ-approximation low rank-revealing schemes in the proposed algorithms with well-characterized time and space complexities. Moreover, we provide extensive analysis to prove convergence using approximate low-rank projections.

Hard thresholding-based framework with improved convergence conditions: We study four hard thresholding variants that provide salient computational trade-offs for the class of greedy methods for low-rank matrix recovery. These methods, as they iterate, optimally exploit the non-convex scaffold of low rank matrices on which the approximation problem resides. Using simple analysis tools, we derive improved conditions that guarantee convergence, compared to state-of-the-art approaches.

The organization of the paper is as follows. In Section II, we set up the notation and provide some definitions and properties, essential for the rest of the paper. In Section III, we describe the basic algorithmic framworks in a nutshell, on which the rest of the work elaborates, while in Section IV we provide important "ingredients" for the class of hard-thresholding methods—detailed convergence analysis proofs are provided in Section V. The complexity analysis of the proposed algorithms is provided in Section VI. We study two acceleration schemes in Sections VII and VIII, based on memory utilization and ϵ-approximate low-rank projections, respectively. We further improve convergence speed by exploiting randomized SVD low rank projections based on power iteration-based subspace finder tools [25]. We provide empirical support for our claims for better data recovery performance and reduced complexity through experimental results on synthetic data in Section X. Finally, we conclude by providing our perspective on the future directions in Section XI.

II. Elementary Definitions and Properties

Notation. We reserve lower-case and bold lower-case letters for scalar and vector variable representation, respectively. Bold upper-case letters denote matrices while bold calligraphic upper-case letters represent linear operators. We reserve calligraphic upper-case letters for set representations. For a matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ and a set of indices $\mathcal{I} \in \{1, 2, \ldots, n\}$, the matrix $\boldsymbol{X}_{\mathcal{I}}$ denotes the submatrix of $\boldsymbol{X}$ with columns indexed by the set $\mathcal{I}$. We use $\boldsymbol{X}(i) \in \mathcal{R}^{m\times n}$ to represent the current matrix estimate at the i-th iteration.

The rank of $\boldsymbol{X}$ is denoted as $\text{rank}(\boldsymbol{X}) \leq \min\{m, n\}$. The empirical data error is denoted as $f(\boldsymbol{X}) := \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}\|_2^2$ with gradient $\nabla f(\boldsymbol{X}) := -2\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X})$, where * is the adjoint operation over the linear operator $\boldsymbol{\mathcal{A}}$. The inner product between matrices $\boldsymbol{A},\ \boldsymbol{B} \in \mathcal{R}^{m\times n}$ is denoted as $\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \text{trace}(\boldsymbol{B}^T\boldsymbol{A})$, where ^{T} represents the transpose operation. I represents an identity matrix with dimensions apparent from the context.

Let $\mathcal{T}$ denote the set of *non-collinear*, non-zero rank-1 matrices in $\mathcal{R}^{m\times n}$—this set defines an uncountably infinite number of matrices in a finite dimensional space. Furthermore, let $\mathcal{U} \subseteq \mathcal{T}$ denote the superset that includes all the sets of *orthonormal*, rank-1 matrices that span *a subspace* included in the subspace induced by $\mathcal{T}$—i.e., for all sets of orthonormal, rank-1 matrices $\mathcal{S} \in \mathcal{U}$, the following hold true: i) $\text{span}(\mathcal{S}) \subseteq \text{span}(\mathcal{T})$ where $\text{span}(\cdot)$ denotes the subspace spanned by the input set of rank-1 matrices, ii) $\langle \boldsymbol{S}_i, \boldsymbol{S}_j\rangle = 0,\ \forall \boldsymbol{S}_i, \boldsymbol{S}_j \in \mathcal{S},\ i \neq j$, and iii) $\text{rank}(\boldsymbol{S}_i) = \text{rank}(\boldsymbol{S}_j) = 1$. With slight abuse of notation, given a set of rank-1 matrices $\mathcal{S}$, $\text{rank}(\text{span}(\mathcal{S}))$ calculates the *maximum* rank of matrices that lie on the subspace spanned by the set $\mathcal{S}$. Given a finite set $\mathcal{S} \in \mathcal{U}$, $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. For any matrix $\boldsymbol{X}$, we use $R(\boldsymbol{X})$ and $N(\boldsymbol{X})$ to denote its range and nullspace, respectively.

A. Matrix decompositions and subspace projections

Given a finite set $\mathcal{S} \in \mathcal{U}$, we declare that $\mathcal{S}$ spans a subspace of $\mathcal{R}^{m\times n}$ if and only if any matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ that lies in this subspace can be decomposed as a linear combination of rank-1 matrices from $\mathcal{S}$ [26]—i.e.,

$$\exists \alpha_i \in \mathcal{R}_+, \quad i = 1, \ldots, \text{rank}(\boldsymbol{X}) \;\; \text{such that} \;\; \boldsymbol{X} = \sum_{i=1}^{\text{rank}(\boldsymbol{X})} \alpha_i \boldsymbol{S}_i, \quad \boldsymbol{S}_i \in \mathcal{S}. \tag{9}$$

Moreover, we denote a set of orthonormal, rank-1 matrices that span the subspace induced by $\boldsymbol{X}$ as ortho($\boldsymbol{X}$). ortho($\boldsymbol{X}$) is obtained as a result of the following optimization problem:

$$\text{ortho}(\boldsymbol{X}) := \arg\min_{\mathcal{Z}} \{|\mathcal{Z}| : \mathcal{Z} \subseteq \mathcal{U}, \boldsymbol{X} \in \text{span}(\mathcal{Z}), \langle \boldsymbol{Z}_i, \boldsymbol{Z}_j \rangle = 0, \; i \neq j, \; \forall \boldsymbol{Z}_i, \boldsymbol{Z}_j \in \mathcal{Z}\}. \tag{10}$$

Remark 1. *For a matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$, ortho($\boldsymbol{X}$) is not unique—there is an infinite uncountable number of sets of orthonormal, rank-1 matrices that span a given subspace.*

Given a set $\mathcal{S} \subseteq \mathcal{U}$, we denote the orthogonal projection operator onto the subspace induced by $\mathcal{S}$ as $\mathcal{P}_{\mathcal{S}}$; furthermore, we denote the orthogonal projection operator onto the orthogonal subspace of $\mathcal{S}$ as $\mathcal{P}_{\mathcal{S}}^{\perp}$. Thus, for arbitrary $\boldsymbol{Y} \in \mathcal{R}^{m\times n}$, $\mathcal{P}_{\mathcal{S}}\boldsymbol{Y} \in \mathcal{R}^{m\times n}$ denotes the orthogonal projection of $\boldsymbol{Y}$ onto the subspace spanned by $\mathcal{S}$ and $\mathcal{P}_{\mathcal{S}}^{\perp}\boldsymbol{Y} \in \mathcal{R}^{m\times n}$ denotes the orthogonal projection of $\boldsymbol{Y}$ onto the orthogonal subspace defined by $\mathcal{S}$.

Remark 2. *Given $\mathcal{S} \subseteq \mathcal{U}$, $\mathcal{P}_{\mathcal{S}}$ is an idempotent linear transformation, that is, $\forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$, $\mathcal{P}_{\mathcal{S}}\mathcal{P}_{\mathcal{S}}\boldsymbol{X} = \mathcal{P}_{\mathcal{S}}\boldsymbol{X}$.*

For an arbitrary set $\mathcal{S} \subseteq \mathcal{U}$, we can always decompose a matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ into two matrix components, as follows:

$$\boldsymbol{X} := \mathcal{P}_{\mathcal{S}}\boldsymbol{X} + \mathcal{P}_{\mathcal{S}}^{\perp}\boldsymbol{X}, \quad \text{such that } \langle \mathcal{P}_{\mathcal{S}}\boldsymbol{X}, \mathcal{P}_{\mathcal{S}}^{\perp}\boldsymbol{X} \rangle = 0.$$

Remark 3. *Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ and $\mathcal{S}$ be a set of rank-1 matrices such that $\boldsymbol{X} \in$ span($\mathcal{S}$) and rank(span($\mathcal{S}$)) $\geq$ rank($\boldsymbol{X}$). Then, $\mathcal{P}_{\mathcal{S}}\boldsymbol{X} = \boldsymbol{X}$—i.e., since $\boldsymbol{X} \in$ span($\mathcal{S}$), the best projection of $\boldsymbol{X}$ onto the subspace induced by $\mathcal{S}$ is the matrix $\boldsymbol{X}$ itself.*

Given two sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{U}$, the following holds true for any matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$:

$$\mathcal{P}_{\mathcal{S}_1}\mathcal{P}_{\mathcal{S}_2}\boldsymbol{X} = \mathcal{P}_{\mathcal{S}_2}\mathcal{P}_{\mathcal{S}_1}\boldsymbol{X}. \tag{11}$$

With slight abuse of notation, we replace the series of projection operations $\mathcal{P}_{\mathcal{S}_1}\mathcal{P}_{\mathcal{S}_2}^{\perp}\boldsymbol{X}$ by one projection as follows: $\mathcal{P}_{\mathcal{S}_1\setminus\mathcal{S}_2}\boldsymbol{X}$—similarly, $\mathcal{P}_{\mathcal{S}_1\setminus\mathcal{S}_2}^{\perp}$ denotes the orthogonal projection onto the orthogonal subspace, spanned by $\mathcal{S}_1 \setminus \mathcal{S}_2$.

B. Singular Value Decomposition (SVD) and its properties

Definition 1. *[SVD] Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ be a rank-k ($k < \min\{m, n\}$) matrix. Then, the SVD of $\boldsymbol{X}$ is given by:*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = [\boldsymbol{U}_\alpha \quad \boldsymbol{U}_\beta] \begin{bmatrix} \widetilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_\alpha^T \\ \boldsymbol{V}_\beta^T \end{bmatrix}, \tag{12}$$

where $\boldsymbol{U}_\alpha \in \mathcal{R}^{m\times k}, \boldsymbol{U}_\beta \in \mathcal{R}^{m\times(m-k)}, \boldsymbol{V}_\alpha \in \mathcal{R}^{n\times k}, \boldsymbol{V}_\beta \in \mathcal{R}^{n\times(n-k)}$ and $\widetilde{\boldsymbol{\Sigma}} = diag(\sigma_1, \ldots, \sigma_k) \in \mathcal{R}^{k\times k}$ for $\sigma_1, \ldots, \sigma_k \in \mathcal{R}_+$. Here, the columns of $\boldsymbol{U}, \boldsymbol{V}$ represent the set of left and right singular vectors, respectively, and $\sigma_1, \ldots, \sigma_k$ denote the singular values.

Remark 4. *The SVD of a matrix $\boldsymbol{X}$ is an orthonormal decomposition of $\boldsymbol{X}$.*

For any matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ with arbitrary rank($\boldsymbol{X}$) $\leq \min\{m, n\}$, the best orthogonal projection $\mathcal{P}_k(\boldsymbol{X})$ onto the set of rank-k ($k \leq$ rank($\boldsymbol{X}$)) matrices $\mathcal{C}_k := \{\boldsymbol{A} \in \mathcal{R}^{m\times n} : \text{rank}(\boldsymbol{A}) \leq k\}$ defines the optimization problem:

$$\mathcal{P}_k(\boldsymbol{X}) = \arg\min_{\boldsymbol{Y}\in\mathcal{C}_k} \left\|\boldsymbol{Y} - \boldsymbol{X}\right\|_F. \tag{13}$$

The distinction between $\mathcal{P}_{\mathcal{S}}$ for $\mathcal{S} \subseteq \mathcal{U}$ and $\mathcal{P}_k$ for $k \in \mathcal{R}_+$ is apparent from context. According to the Eckart-Young theorem [27], the best rank-k approximation of a matrix $\boldsymbol{X}$ corresponds to its truncated SVD. In more detail, if $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ according to (12), then $\mathcal{P}_k(\boldsymbol{X}) := \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T$ and contains the k strongest principal compoments of $\boldsymbol{X}$ with the k largest in magnitude singular values and vectors—$\boldsymbol{\Sigma}_k \in \mathcal{R}^{k\times k}$ is a diagonal matrix that contains the first k diagonal entries of $\boldsymbol{\Sigma}$ and $\boldsymbol{U}_k$, $\boldsymbol{V}_k$ contain the corresponding left and right singular vectors, respectively.

Given a matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$, SVD naturally provides a set of orthonormal, rank-1 matrices, $\mathcal{S} \in$ ortho($\boldsymbol{X}$), as outer products of the left singular vectors associated with the non-zero singular values, according to the following definition.

Definition 2. *[Orthogonal projections using SVD] Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ be a matrix with arbitrary rank and SVD decomposition given by (12). Then, $\mathcal{S} := \{\boldsymbol{u}_i\boldsymbol{u}_i^T : i = 1, \ldots, rank(\boldsymbol{X})\}$ constitutes a set of orthonormal, rank-1 matrices, where $\boldsymbol{u}_i$ denotes the i-th left singular vector, such that $\boldsymbol{X} \in$ span($\mathcal{S}$). Moreover, the orthogonal projection onto $R(\boldsymbol{X})$ is given by $\mathcal{P}_{\mathcal{S}} = \boldsymbol{U}_\alpha\boldsymbol{U}_\alpha^T$, as defined in (12).*

Remark 5. *For $k \leq rank(\boldsymbol{X})$, the rank-k subspace that includes most of the energy of $\boldsymbol{X}$ (in Frobenius norm) is spanned by $\mathcal{S} := \{\boldsymbol{u}_i \boldsymbol{u}_i^T : i = 1, \ldots, k\}$, where $\boldsymbol{u}_i$ is the singular vector associated with the i-th largest singular value of $\boldsymbol{X}$. The corresponding low rank matrix is given by $\mathcal{P}_{\mathcal{S}}\boldsymbol{X}$.*

Remark 6. *Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ be a matrix with arbitrary rank and SVD decomposition given by (12). Then, for any $\mathcal{S} \subseteq \mathcal{U}$, the following holds true: $\left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F \leq \left\|\boldsymbol{X}\right\|_F$, that is, the projection of a matrix does not increase the energy in terms of the Frobenius norm.*

C. Restricted Isometry Property

Unfortunately, low rank assumption does not guarantee successful recovery of the true matrix for *any linear operator* $\boldsymbol{\mathcal{A}}$. In the analogous vector case, many conditions have been proposed in the literature to establish solution uniqueness and reconstruction stability such as null space property [28], exact recovery condition [29], etc. For the matrix case, [12] proved the so-called *restricted isometry property* (RIP) for the affine rank minimization problem.

Definition 3. *[Rank Restricted Isometry Property (R-RIP) for matrix linear operators [12]] A linear operator $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ satisfies the R-RIP with constant $\delta_k(\boldsymbol{\mathcal{A}}) \in (0, 1)$ if and only if:*

$$(1 - \delta_k(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}\right\|_F^2 \leq \left\|\boldsymbol{\mathcal{A}}\boldsymbol{X}\right\|_2^2 \leq (1 + \delta_k(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}\right\|_F^2, \quad \forall \boldsymbol{X} \in \mathcal{R}^{m\times n} \text{ such that } rank(\boldsymbol{X}) \leq k. \tag{14}$$

D. Some useful bounds using R-RIP

In this section, we present some lemmas that are useful in our subsequent developments—these lemmas are consequences of the R-RIP of $\boldsymbol{\mathcal{A}}$.

Lemma 1. *[21] Let $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ be a linear operator that satisfies the R-RIP with constant $\delta_k(\boldsymbol{\mathcal{A}})$. Then, $\forall \boldsymbol{v} \in \mathcal{R}^p$, the following holds true:*

$$\left\|\mathcal{P}_{\mathcal{S}}(\boldsymbol{\mathcal{A}}^*\boldsymbol{v})\right\|_F \leq \sqrt{1 + \delta_k(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{v}\right\|_2, \tag{15}$$

where $\mathcal{S} \subseteq \mathcal{U}$ is a set of orthonormal, rank-1 matrices such that $rank(\mathcal{P}_{\mathcal{S}}\boldsymbol{X}) \leq k,\ \forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$.

Lemma 2. *[21] Let $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ be a linear operator that satisfies the R-RIP with constant $\delta_k(\boldsymbol{\mathcal{A}})$. Then, $\forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$, the following holds true:*

$$(1 - \delta_k(\boldsymbol{\mathcal{A}}))\left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F \leq \left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F \leq (1 + \delta_k(\boldsymbol{\mathcal{A}}))\left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F, \tag{16}$$

where $\mathcal{S} \subseteq \mathcal{U}$ is a set of orthonormal, rank-1 matrices such that $rank(\mathcal{P}_{\mathcal{S}}\boldsymbol{X}) \leq k,\ \forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$.

Lemma 3. *[22] Let $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ be a linear operator that satisfies the R-RIP with constant $\delta_k(\boldsymbol{\mathcal{A}})$ and $\mathcal{S} \subseteq \mathcal{U}$ be a set of orthonormal, rank-1 matrices such that $rank(\mathcal{P}_{\mathcal{S}}\boldsymbol{X}) \leq k,\ \forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$. Then, for $\mu > 0$, $\boldsymbol{\mathcal{A}}$ satisfies:*

$$\lambda(\mu\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}}) \in [\mu(1 - \delta_k(\boldsymbol{\mathcal{A}})), \mu(1 + \delta_k(\boldsymbol{\mathcal{A}}))]. \tag{17}$$

where $\lambda(\boldsymbol{\mathcal{B}})$ represents the range of eigenvalues of the linear operator $\boldsymbol{\mathcal{B}} : \mathcal{R}^p \to \mathcal{R}^{m\times n}$. Moreover, $\forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$, it follows:

$$\left\|(I - \mu\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}})\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F \leq \max\left\{\mu(1 + \delta_k(\boldsymbol{\mathcal{A}})) - 1, 1 - \mu(1 - \delta_k(\boldsymbol{\mathcal{A}}))\right\}\left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{X}\right\|_F. \tag{18}$$

Lemma 4. *[22] Let $\boldsymbol{\mathcal{A}} : \mathcal{R}^{m\times n} \to \mathcal{R}^p$ be a linear operator that satisfies the R-RIP with constant $\delta_k(\boldsymbol{\mathcal{A}})$ and $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{U}$ be two sets of orthonormal, rank-1 matrices such that $rank(\mathcal{P}_{\mathcal{S}_1\cup\mathcal{S}_2}\boldsymbol{X}) \leq k,\ \forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$. Then the following inequality holds:*

$$\left\|\mathcal{P}_{\mathcal{S}_1}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_1}^{\perp}\boldsymbol{X}\right\|_F \leq \delta_k(\boldsymbol{\mathcal{A}})\left\|\mathcal{P}_{\mathcal{S}_1}^{\perp}\boldsymbol{X}\right\|_F,\ \forall \boldsymbol{X} \in span(\mathcal{S}_2). \tag{19}$$

A well-known lemma used in the convergence rate proofs of this class of greedy hard thresholding algorithms is defined next.

Lemma 5. *[Optimality condition [30]]Let $\Theta \subseteq \mathcal{R}^{m\times n}$ be a convex subspace and $f : \Theta \to \mathcal{R}$ be a smooth objective function defined over Θ. Let $\boldsymbol{X}^* \in \Theta$ be a local minimum of the objective function f over the set Θ. Then*

$$\langle \nabla f(\boldsymbol{X}^*), \boldsymbol{X} - \boldsymbol{X}^* \rangle \geq 0, \quad \forall \boldsymbol{X} \in \Theta, \tag{20}$$

for all convex sets Θ.

III. Algrebraic Pursuits in a nutshell

In this section, we provide an overview of the proposed framework. The main theorems are presented in section V where detailed proofs are provided in the appendix.

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, k, Tolerance η, MaxIterations
Initialize: $\boldsymbol{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat

1: $\mathcal{D}_i \leftarrow \text{ortho}\Big(\mathcal{P}_k\big(\mathcal{P}_{\mathcal{X}_i}^{\perp} \nabla f(\boldsymbol{X}(i))\big)\Big)$ *(Best rank-k subspace orthogonal to $\mathcal{X}_i$)*

2: $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{X}_i$ *(Active subspace expansion)*

3: $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\boldsymbol{X}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\|_2^2}$ *(Step size selection)*

4: $\boldsymbol{V}(i) \leftarrow \boldsymbol{X}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))$ *(Error norm reduction via gradient descent)*

5: $\boldsymbol{W}(i) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$ with $\mathcal{W}_i \leftarrow \text{ortho}(\boldsymbol{W}(i))$ *(Best rank-k subspace selection)*

6: $\xi_i \leftarrow \arg\min_{\xi} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\boldsymbol{W}(i) - \frac{\xi}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))\|_2^2}$ *(Step size selection)*

7: $\boldsymbol{X}(i+1) \leftarrow \boldsymbol{W}(i) - \frac{\xi_i}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))$ with $\mathcal{X}_{i+1} \leftarrow \text{ortho}(\boldsymbol{X}(i+1))$ *(De-bias using gradient descent)*

$i \leftarrow i+1$

until $\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\|_2 \leq \eta\|\boldsymbol{X}(i)\|_2$ or MaxIterations.

Algorithm 1: MATRIX ALPS I

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, k, Tolerance η, MaxIterations
Initialize: $\boldsymbol{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat

1: $\mathcal{D}_i \leftarrow \text{ortho}\Big(\mathcal{P}_k\big(\mathcal{P}_{\mathcal{X}_i}^{\perp} \nabla f(\boldsymbol{X}(i))\big)\Big)$ *(Best rank-k subspace orthogonal to $\mathcal{X}_i$)*

2: $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{X}_i$ *(Active subspace expansion)*

3: $\boldsymbol{V}(i) \leftarrow \arg\min_{\boldsymbol{V}:\boldsymbol{V}\in\text{span}(\mathcal{S}_i)} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{V}\big\|_2^2$ *(Error norm reduction via least-squares optimization)*

4: $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$ with $\mathcal{X}_{i+1} \leftarrow \text{ortho}(\boldsymbol{X}(i+1))$ *(Best rank-k subspace selection)*

$i \leftarrow i+1$

until $\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\|_2 \leq \eta\|\boldsymbol{X}(i)\|_2$ or MaxIterations.

Algorithm 2: ADMiRA Instance

A. Precedence

Explicit descriptions of the proposed algorithms are provided in Algorithms 1 and 2, in pseudocode form. Algorithm 1 follows from the ALgrebraic PursuitS (ALPS) scheme for the vector case [31]. MATRIX ALPS I provides efficient strategies for adaptive step size selection and additional signal estimate updates at each iteration (these motions are explained in detail in the next subsection). Algorithm 2 (ADMiRA) [21] further improves the performance of Algorithm 1 by introducing least squares optimization steps over restricted subspaces—this technique borrows from a series of vector reconstruction algorithms such as CoSaMP [32], Subspace Pursuit (SP) [33] and Hard Thresholding Pursuit (HTP) [34]. To start-off, we first derive conditions under which MATRIX ALPS I and ADMiRA algorithms recover the underlying low rank matrix, in terms of R-RIP constant bounds. The resulting conditions are competitive with the state-of-the-art approaches [5], [21].

In a nutshell, both algorithms simply seek to improve the subspace selection by iteratively collecting an extended subspace $\mathcal{S}_i$ with $|\mathcal{S}_i| \leq 2k$ and then finding the rank-k matrix that fits the measurements in this restricted subspace using least squares techniques or gradient descent motions.

Similarly to the measurement-optimal greedy algorithms for the sparse vector reconstruction problem [32], [33], our method is a first-order gradient descent algorithm; hence, it requires the computation of the gradient $\nabla f(\cdot)$, a *superlinear* runtime operation as a function of $O(mn)$. [24] provides an efficient randomized scheme with sublinear time complexity for the vector analogue problem.

B. Algebraic Pursuits in a nutshell

At each iteration, the Algorithms 1 and 2 perform motions from the following list:

1) *Best rank-k subspace orthogonal to $\mathcal{X}_i$ and active subspace expansion:* We identify the best rank-k subspace of the current gradient $\nabla f(\boldsymbol{X}(i))$, orthogonal to $\mathcal{X}_i$ and then merge this low-rank subspace with $\mathcal{X}_i$. This motion guarantees that, at each iteration, we expand the current rank-k subspace estimate with k new, rank-1 orthogonal subspaces to explore, avoiding premature termination of the algorithm.

2a) *Error norm reduction via greedy descent with adaptive step size selection (Algorithm 1):* We decrease the data error by performing a single gradient descent step over the objective function. This scheme is based on a one-shot step size selection procedure (Step size selection step)—detailed description of this approach is given in Section IV.

2b) *Error norm reduction via least squares optimization (Algorithm 2):* We decrease the data error $f(\boldsymbol{X})$ as much as possible on the active $O(k)$-low rank subspace. Assuming $\boldsymbol{\mathcal{A}}$ is well-conditioned over low-rank subspaces, the main complexity of this operation is dominated by the solution of a symmetric linear system of equations.

3) *Best rank-k subspace selection:* We project the constrained solution onto the set of rank-k matrices $\mathcal{C}_k := \{\boldsymbol{A} \in \mathcal{R}^{m\times n} : \text{rank}(\boldsymbol{A}) \leq k\}$ to arbitrate the active support set. This step is calculated in polynomial time complexity as a function of $m \times n$ using SVD or other matrix rank-revealing decomposition algorithms—further discussions about this step and its approximations can be found in Sections VIII and IX.

4) *De-bias using gradient descent (Algorithm 1):* We de-bias the current estimate $\boldsymbol{W}(i)$ by performing an additional gradient descent step, decreasing the data error. The step size selection procedure follows the same motions as in 2a).

IV. Ingredients for hard thresholding methods

A. Step size selection

To emphasize how step size selection μ_i affects the convergence rate of Algorithm 1, we provide ideas on selecting the step size μ_i adaptively.

For the vector case (1), recent works on the performance of IHT algorithm provide strong convergence rate guarantees in terms of R-RIP constants; c.f. [35]. However, as a prerequisite to achieve these strong isometry constant bounds, the step size is set $\mu_i = 1, \forall i$, given that $\|\boldsymbol{\Phi}\|_2^2 < 1$ [34]—similar analysis can be found in [5] for the matrix case. From a different perspective, [36] proposes a constant step size $\mu_i = 1/(1+\delta_{2K}),\ \forall i$, for the vector case, based on a simple convergence analysis of the gradient descent method.

Unfortunately, most of the above problem assumptions are not naturally met; the authors in [37] provide an intuitive example where IHT algorithm behaves differently under various scalings of the sensing matrix $\boldsymbol{\Phi}$—similar counterexamples can be devised for the matrix case. Violation of these configuration details usually lead to unpredictable signal recovery performance of hard thresholding methods. Therefore, more sophisticated step size selection procedures should be devised to tackle these computational issues during actual recovery. On the other hand, the computation of RIP constants has exponential time complexity for the vector case strategy of [36] and exhaustive combinatorial search is necessary.

Existing approaches broadly fall into two categories: constant and adaptive step size selection. In this work, we present efficient strategies to adaptively select the step size μ_i that implies fast convergence rate—without violating the R-RIP conditions on $\boldsymbol{\mathcal{A}}$. Constant step size strategies easily follow from [23] and are not listed in this work.

Adaptive step size selection. There is limited work on the adaptive step size selection for hard thresholding methods. To the best of our knowledge, apart from [23], [37]- [38] are the only studies that attempt this via line searching for the vector case—no references are available for the matrix case.

According to Algorithm 1, let $\boldsymbol{X}(i) \in \mathcal{R}^{m\times n}$ be the rank-k matrix estimate with subspace spanned by the set of orthonormal, rank-1 matrices in $\mathcal{X}_i$, at the i-th iteration. Using regular gradient descent motions, the new rank-k estimate $\boldsymbol{W}(i)$ can be calculated through:

$$\boldsymbol{V}_i = \boldsymbol{X}(i) - \frac{\mu}{2}\nabla f(\boldsymbol{X}(i)), \quad \boldsymbol{W}(i) = \mathcal{P}_k(\boldsymbol{V}(i)), \tag{21}$$

with $\mathcal{W}_i \leftarrow \text{ortho}(\boldsymbol{W}(i))$. It then holds that the subspace spanned by $\mathcal{W}_i$ originates: i) either from the subspace of $\mathcal{X}_i$, ii) or from the best subspace (in terms of the Frobenius norm metric) of the current gradient $\nabla f(\boldsymbol{X}(i))$, *orthogonal to* $\mathcal{X}_i$, iii) or from the combination of orthonormal, rank-1 matrices lying on the union of the above two subspaces. The statements above can be summarized in the following expression:

$$\text{span}(\mathcal{W}_i) \in \text{span}\left(\mathcal{D}_i \cup \mathcal{X}_i\right) \tag{22}$$

for any step size μ_i and $\mathcal{D}_i := \text{ortho}\Big(\mathcal{P}_k\big(\mathcal{P}_{\mathcal{X}_i}^{\perp}\nabla f(\boldsymbol{X}(i))\big)\Big)$. Since $\text{rank}(\text{span}(\mathcal{W}_i)) \leq k$, we easily deduce the following key observation:

Remark 7. *Let $\mathcal{S}_i$ be a set of rank-1 matrices where rank(span($\mathcal{S}_i$)) $\leq 2k$, defined as follows:*

$$\mathcal{S}_i = \mathcal{D}_i \cup \mathcal{X}_i. \tag{23}$$

Given $\mathcal{W}_i$ is unknown before the i-th iteration, $\mathcal{S}_i$ spans the smallest subspace that contains $\mathcal{W}_i$ such that the following equality

$$\mathcal{P}_k\left(\boldsymbol{X}(i) - \frac{\mu_i}{2}\nabla f(\boldsymbol{X}(i))\right) = \mathcal{P}_k\left(\boldsymbol{X}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\right) \tag{24}$$

necessarily holds.

To compute step-size μ_i, we use:

$$\mu_i = \arg\min_{\mu} \left\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\left(\boldsymbol{X}(i) - \frac{\mu}{2}\nabla_{\mathcal{S}_i} f(\boldsymbol{X}(i))\right)\right\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\|_2^2}, \tag{25}$$

i.e., μ_i is the minimizer of the objective function, given the current gradient $\nabla f(\boldsymbol{X}(i))$. Note that:

$$1 - \delta_{2k}(\boldsymbol{\mathcal{A}}) \leq \frac{1}{\mu_i} \leq 1 + \delta_{2k}(\boldsymbol{\mathcal{A}}), \tag{26}$$

due to R-RIP—i.e., we select $2k$ subspaces such that μ_i satisfies the R-RIP condition. We can derive similar arguments for the additional step size selection ξ_i in Step 6 of Algorithm 1.

We observe that adaptive μ_i scheme results in more restrictive "worst-case" isometry constants compared to [5], [34], [39], but faster convergence and better stability are empirically observed in general. Figures 1(a)-(b) illustrate some characteristic examples. The performance varies for different problem configurations. For $\mu > 1$, SVP diverges for various test cases. We note that, for large fixed matrix dimensions m, n, adaptive step size selection becomes computationally expensive compared to constant step size selection strategies, as the rank of $\boldsymbol{X}^*$ increases.

B. Updates over Restricted Subspaces

In Algorithm 1, at each iteration, the new estimate $\boldsymbol{W}(i) \leftarrow \mathcal{P}_k\left(\boldsymbol{V}(i)\right)$ can be further refined by applying a single or multiple gradient descent updates with line search restricted on $\mathcal{W}_i$ [34] (Step 7 in Algorithm 1):

$$\boldsymbol{X}(i+1) \leftarrow \boldsymbol{W}(i) - \frac{\xi_i}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i)), \quad \text{where } \xi_i = \frac{\|\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\nabla f(\boldsymbol{W}(i))\|_2^2}.$$

In spirit, the gradient step above is the same as block coordinate descent in convex optimization except that we find the subspaces adaptively (almost greedily). Figure 1(c) depicts the acceleration achieved by using additional gradient updates over restricted low-rank subspaces.

C. Memory-based acceleration

Memory-based techniques can be used to improve convergence speed and stability. We keep the discussion on memory utilization for Section VII where we presnt a new algorithmic framework for low-rank matrix recovery.

V. Convergence guarantees

In this section, we present the theoretical convergence guarantees of Algorithms 1 and 2 as functions of R-RIP constants. To characterize the performance of the proposed algorithms, both in terms of convergence rate and noise resilience, we use the following recursive expression:

$$\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\|_F \leq \rho\|\boldsymbol{X}(i) - \boldsymbol{X}^*\|_F + \gamma\|\boldsymbol{\varepsilon}\|_2. \tag{27}$$

In (27), γ denotes the approximation guarantee and provides insights into algorithm's reconstruction capabilities when additive noise is present; $\rho < 1$ expresses the convergence rate towards a region around $\boldsymbol{X}^*$, whose radius is determined by $\frac{\gamma}{1-\rho}\|\boldsymbol{\varepsilon}\|_2$. In short, (27) characterizes how the distance to the true signal $\boldsymbol{X}^*$ is decreased and how the noise level affects the accuracy of the solution, at each iteration.

A. Matrix ALPS I

An important lemma for our derivations below is given next:

Lemma 6. *[Active subspace expansion] Let $\boldsymbol{X}(i) \in \mathcal{R}^{m\times n}$ be the matrix estimate at the i-th iteration and let $\mathcal{X}_i$ be a set of orthonormal, rank-1 matrices such that $\mathcal{X}_i \leftarrow ortho(\boldsymbol{X}(i))$. Then, at each iteration, the Active Subspace Expansion step in Algorithms 1 and 2 identifies information in $\boldsymbol{X}^*$, such that:*

$$\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\right\|_F \leq (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{28}$$

where $\mathcal{S}_i = \mathcal{X}_i \cup \mathcal{D}_i$ and $\mathcal{X}^* \leftarrow ortho(\boldsymbol{X}^*)$.

Lemma 6 states that, at each iteration, the active subspace expansion step identifies a $2k$ rank subspace in $\mathcal{R}^{m\times n}$ such that the amount of unrecovered energy of $\boldsymbol{X}^*$—i.e., the projection of $\boldsymbol{X}^*$ onto the orthogonal subspace of span$(\mathcal{S}_i)$—is bounded by (28).

Then, Theorem 1 characterizes the iteration invariant of Algorithm 1 for the matrix case:

Theorem 1. *[Iteration invariant for* Matrix ALPS I*] The $(i+1)$-th matrix estimate $\boldsymbol{X}(i+1)$ of* Matrix ALPS I *satisfies the following recursion:*

$$\left\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\right\|_F \leq \rho\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{29}$$

where $\rho := \left(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right)\left(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right)$ *and*

$$\gamma := \left(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right)\left(\frac{2\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\right) + \frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}. \tag{30}$$

Moreover, when $\delta_{3k}(\boldsymbol{\mathcal{A}}) < 0.1235$, the iterations are contractive.

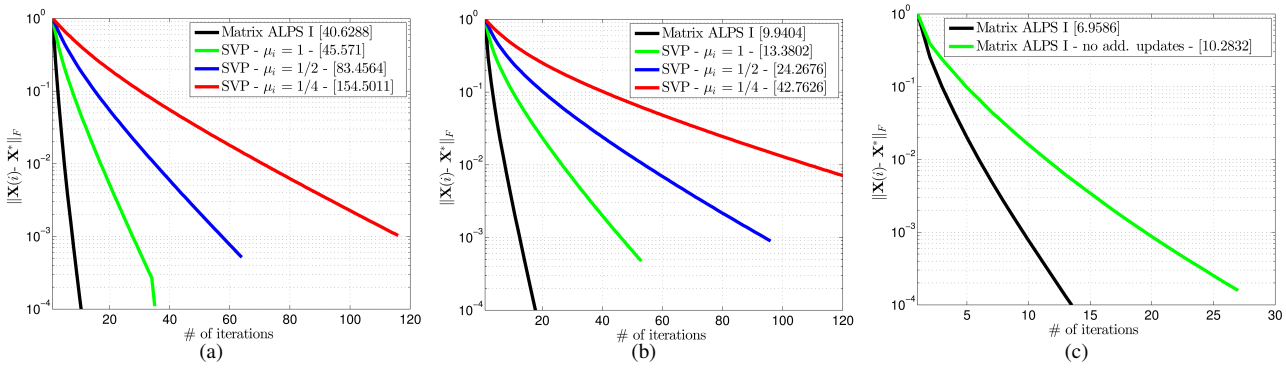

Fig. 1. *Median error per iteration for various step size policies and 20 Monte-Carlo repetitions. In brackets, we present the mean time consumed for convergene in seconds. (a) $m = n = 2048$, $p = 0.4n^2$, and rank $k = 70$—$\boldsymbol{\mathcal{A}}$ is formed by permuted and subsampled noiselets [40]. (b) $m = 2048$, $n = 512$, $p = 0.4n^2$, and rank $k = 50$—we use underdetermined linear map $\boldsymbol{\mathcal{A}}$ according to the MC problem (c) $m = 2048$, $n = 512$, $p = 0.4n^2$, and rank $k = 40$—we use underdetermined linear map $\boldsymbol{\mathcal{A}}$ according to the MC problem.*

To provide some intuition behind this result, assume that $\boldsymbol{X}^*$ is a rank-k matrix. Then, according to Theorem 1, for $\rho < 1$, the approximation parameter γ in (29) satisfies:

$$\gamma < 5.7624, \quad \text{for} \quad \delta_{3k}(\boldsymbol{\mathcal{A}}) < 0.1235. \tag{31}$$

Moreover, we derive the following:

$$\rho < \frac{1 + 2\delta_{3k}(\boldsymbol{\mathcal{A}})}{(1 - \delta_{3k}(\boldsymbol{\mathcal{A}}))^2}\left(4\delta_{3k}(\boldsymbol{\mathcal{A}}) + 8\delta_{3k}^2(\boldsymbol{\mathcal{A}})\right) < \frac{1}{2} \Rightarrow \delta_{3k}(\boldsymbol{\mathcal{A}}) < 0.079, \tag{32}$$

which is *a stronger* R-RIP condition assumption compared to state-of-the-art approaches [21]. In the next section, we further improve this guarantee using Algorithm 2.

Unfolding the recursive formula (28), we obtain the following upper bound for $\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F$ at the i-th iteration:

$$\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F \leq \rho^i \left\|\boldsymbol{X}(0) - \boldsymbol{X}^*\right\|_F + \frac{\gamma}{1-\rho}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{33}$$

Then, given $\boldsymbol{X}(0) = 0$, MATRIX ALPS I finds a rank-k solution $\widehat{\boldsymbol{X}} \in \mathcal{R}^{m\times n}$ such that $\left\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^*\right\|_F \leq \frac{\gamma+1-\rho}{1-\rho}\left\|\boldsymbol{\varepsilon}\right\|_2$ after $i := \left\lceil \frac{\log(\|\boldsymbol{X}^*\|_F / \|\boldsymbol{\varepsilon}\|_2)}{\log(1/\rho)} \right\rceil$ iterations.

If we ignore the steps 5 and 6 in Algorithm 1, we obtain another projected gradient descent variant for the affine rank minimization problem, for which we obtain the following performance guarantees—the proof follows from the proof of Theorem 1.

Corollary 1. *[MATRIX ALPS I Instance] In Algorithm 1, we ignore steps 5 and 6 and let $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k(\boldsymbol{Y}_i)$ with $\mathcal{X}_{i+1} \leftarrow ortho(\boldsymbol{X}(i+1))$ in step 4. Then, using same analysis, we observe that the following recursion is satisfied:*

$$\left\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\right\|_F \leq \rho\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{34}$$

for $\rho := \left(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right)$ and $\gamma := \left(\frac{2\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\right)$. Moreover, $\rho < 1$ when $\delta_{3k}(\boldsymbol{\mathcal{A}}) < 0.1594$.

We observe that the additional estimate update over restricted support sets results in more restrictive isometry constants compared to Theorem 1. In practice, additional updates result in faster convergence and more stable signal reconstruction, as shown in Figure 1(c).

B. ADMiRA Instance

In MATRIX ALPS I, the gradient descent steps constitute a first-order approximation to least-squares minimization problems. Replacing Step 4 in Algorithm 1 with the following optimization problem:

$$\boldsymbol{V}(i) \leftarrow \underset{\boldsymbol{V}:\boldsymbol{V}\in \text{span}(\mathcal{S}_i)}{\arg\min} \left\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{V}\right\|_2^2, \tag{35}$$

we obtain ADMiRA (furthermore, we remove the de-bias step in Algorithm 1). Assuming that the linear operator $\boldsymbol{\mathcal{A}}$, restricted on sufficiently low-rank subspaces, is well-conditioned in terms of the R-RIP assumption, the optimization problem (35) has a

unique optimal minimizer. By exploiting the optimality condition in Lemma 5, ADMiRA instance in Algorithm 2 features the following guarantee:

Theorem 2. *[Iteration invariant] The $(i+1)$-th matrix estimate $\boldsymbol{X}(i+1)$ of ADMiRA answers the following recursive expression:*

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq \rho\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F+\gamma\big\|\boldsymbol{\varepsilon}\big\|_F,$$

where

$$\rho := \big(2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}})\big)\sqrt{\frac{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}, \tag{36}$$

and

$$\gamma := \sqrt{\frac{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\sqrt{2(1+\delta_{3k}(\boldsymbol{\mathcal{A}}))}+\Big(\frac{\sqrt{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}+\sqrt{3}\Big)\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}. \tag{37}$$

Moreover, when $\delta_{3k}(\boldsymbol{\mathcal{A}})<0.2267$, the iterations are contractive.

Similarly to MATRIX ALPS I analysis, the parameter γ in Theorem 2 satisfies:

$$\gamma < 5.1848, \text{ for } \delta_{3k}(\boldsymbol{\mathcal{A}})<0.2267. \tag{38}$$

Furthermore, to compare the approximation guarantees of Theorem 2 with [21], we further observe:

$$\delta_{3k}(\boldsymbol{\mathcal{A}})<0.1214, \text{ for } \rho<1/2. \tag{39}$$

We remind that [21] provides convergence guarantees for ADMiRA with $\delta_{4k}(\boldsymbol{\mathcal{A}})<0.04$ for $\rho=1/2$.

VI. COMPLEXITY ANALYSIS

In each iteration, computational requirements of the proposed hard thresholding methods mainly depend on the total number of linear mapping operations $\boldsymbol{\mathcal{A}}$, gradient descent steps, least-squares optimizations and matrix decompositions for low rank approximation. Different algorithmic configurations (e.g. removing steps 6 and 7 in Algorithm 1) lead to hard thresholding variants with less computational complexity per iteration and better R-RIP conditions for convergence but a degraded performance in terms of stability and convergence speed is observed in practice. On the other hand, these additional processing steps increase the required time-complexity per iteration; hence, low iteration counts are desired to trade-off these operations.

A non-exhaustive list of linear map examples includes the identity operator (Principal component analysis (PCA) problem), Fourier/Wavelets/Noiselets tranformations and the famous Matrix Completion problem where $\boldsymbol{\mathcal{A}}$ is a mask operator such that only a fraction of elements in $\boldsymbol{X}$ is observed. Assuming the most demanding case where $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^*$ are dense linear maps with no structure, the computation of the gradient $\nabla f(\boldsymbol{X}(i))$ at each iteration requires $O(pkmn)$ arithmetic operations.

Given a set $\mathcal{S}$ of orthonormal, rank-1 matrices, the projection $\mathcal{P}_{\mathcal{S}}\boldsymbol{X}$ for any matrix $\boldsymbol{X}\in\mathcal{R}^{m\times n}$ needs $O(m^2n)$ time complexity as a matrix-matrix multiplication operation. In MATRIX ALPS I, the adaptive step size selection steps require the calculation of μ_i and ξ_i quantities in $O(\min\{pkmn,m^2n\})$ time complexity. In ADMiRA solving a least-squares system restricted on rank-$2k$ and rank-k subspaces requires $O(pk^2)$ complexity—according to [32], [21], the complexity of this step can be further reduced using approximation techniques such as the Richardson method or conjugate gradients algorithm.

Contrariwise to the vector case where the projection onto the union of sparse vector subspaces can be easily computed via sorting the signal coefficients, the best projection of an arbitrary matrix onto the set of low rank matrices requires more sophisticated linear algebra matrix decompositions such as SVD. Using the Lanczos approach, we require $O(kmn)$ arithmetic operations to compute a rank-k matrix approximation for a given constant accuracy—a prohibitive time-complexity that does not scale well for many practical applications. Sections VIII and IX describe approximate low rank matrix projections and how they affect the convergence guarantees of the proposed algorithms.

Overall, in MATRIX ALPS I, the operation that dominates, with respect to the total number of operations at each iteration, requires $O(\min\{pkmn,m^2n\})$ time complexity while ADMiRA requires $O(pkmn)$ time complexity per iteration.

VII. MEMORY UTILIZATION

Iterative algorithms can use memory to gain momentum in convergence. The success of the memory-based approaches depends on the iteration dependent momentum term by leveraging previous estimates. Based on Nesterov's optimal gradient methods, we propose a hard thresholding variant, described in Algorithm 3—the vector case variant was previously proposed in [41].

Similarly to μ_i strategies, τ_i can be preset as constant or adaptively computed at each iteration. Constant momentum step size selection has no additional computational cost but convergence rate acceleration is not guaranteed for some problem formulations. On the other hand, empirical evidence has shown that adaptive τ_i selection strategies result to faster convergence compared to zero-memory methods with *similar complexity*.

For the case of strongly convex objective functions, Nesterov [42] proposed the following constant momentum step size selection scheme: $\tau_i=\frac{\alpha_i(1-\alpha_i)}{\alpha_i^2+\alpha_{i+1}}$, where $\alpha_0\in(0,1)$ and α_{i+1} is computed as the root $\in(0,1)$ of

$$\alpha_{i+1}^2=(1-\alpha_{i+1})\alpha_i^2+q\alpha_{i+1}, \text{ for } q\triangleq\frac{1}{\kappa^2(\boldsymbol{\mathcal{A}})}=\frac{\sigma_{\min}^2(\boldsymbol{\mathcal{A}})}{\sigma_{\max}^2(\boldsymbol{\mathcal{A}})},$$

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, k, Tolerance η, MaxIterations
Initialize: $\boldsymbol{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $\boldsymbol{Q}(0) \leftarrow 0$, $\mathcal{Q}_0 \leftarrow \{\emptyset\}$, $\tau_i\ \forall i$, $i \leftarrow 0$
repeat

1: $\mathcal{D}_i \leftarrow \text{ortho}\Big(\mathcal{P}_k\big(\mathcal{P}^{\perp}_{\mathcal{Q}_i} \nabla f(\boldsymbol{Q}(i))\big)\Big)$ *(Best rank-k subspace orthogonal to $\mathcal{Q}_i$)*
2: $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{Q}_i$ *(Active subspace expansion)*
3: $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\boldsymbol{Q}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\|_2^2}$ *(Step size selection)*
4: $\boldsymbol{V}(i) \leftarrow \boldsymbol{Q}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))$ *(Error norm reduction via gradient descent)*
5: $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$ *(Best rank-k subspace selection)*
6: $\boldsymbol{Q}(i+1) \leftarrow \boldsymbol{X}(i+1) + \tau_i(\boldsymbol{X}(i+1) - \boldsymbol{X}(i))$ *(Momentum update)*
7: $\mathcal{Q}_{i+1} \leftarrow \text{ortho}(\mathcal{X}_i \cup \mathcal{X}_{i+1})$
$i \leftarrow i+1$
until $\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\|_2 \leq \eta\|\boldsymbol{X}(i)\|_2$ or MaxIterations.

Algorithm 3: MATRIX ALPS II

where $\kappa(\boldsymbol{\mathcal{A}})$ denotes the condition number of $\boldsymbol{\mathcal{A}}$ and $\sigma_{\min}(\boldsymbol{\mathcal{A}}), \sigma_{\max}(\boldsymbol{\mathcal{A}})$ denote the minimum and maximum singular values of $\boldsymbol{\mathcal{A}}$. In this scheme, exact calculation of q parameter is computationally expensive for large-scale data problems and approximation schemes are leveraged to compensate this complexity bottleneck.

Based upon the similar ideas as adaptive μ_i selection, we propose to select τ_i as the minimizer of the objective function:

$$\tau_i = \arg\min_{\tau} \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{Q}(i+1)\|_2^2 = \frac{\langle \boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i), \boldsymbol{\mathcal{A}}\boldsymbol{X}(i) - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i-1)\rangle}{\|\boldsymbol{\mathcal{A}}\boldsymbol{X}(i) - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i-1)\|_2^2}, \tag{40}$$

where $\boldsymbol{\mathcal{A}}\boldsymbol{X}(i), \boldsymbol{\mathcal{A}}\boldsymbol{X}(i-1)$ are *previously computed*. According to (40), τ_i is dominated by the calculation of a vector inner product, a computationally cheaper process than q calculation. Convergence rate performance of the above schemes is depicted in Fig. 2(a) for the vector case (1) [23].

Theorem 3 characterizes Algorithm 3 for *constant* momentum step size selection. To keep the main ideas simple, we ignore the additional gradient updates in Algorithm 3. In addition, we only consider the noiseless case for clarity. The convergence rate proof for these cases is left to the reader.

Theorem 3. *[Iteration invariant for* MATRIX ALPS II*] Let $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\boldsymbol{X}^*$ be a noiseless set of observations. To recover $\boldsymbol{X}^*$ from $\boldsymbol{y}$ and $\boldsymbol{\mathcal{A}}$, the $(i+1)$-th matrix estimate $\boldsymbol{X}(i+1)$ of* MATRIX ALPS II *satisfies the following recursion:*

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F \leq \alpha(1+\tau_i)\big\|\boldsymbol{X}(i) - \boldsymbol{X}^*\big\|_F + \alpha\tau_i\big\|\boldsymbol{X}(i-1) - \boldsymbol{X}^*\big\|_F, \tag{41}$$

where $\alpha := \frac{4\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + (2\delta_{3k}(\boldsymbol{\mathcal{A}}) + 2\delta_{4k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}$. Moreover, solving the above second-order recurrence, the following inequality holds true:

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F \leq \left(\frac{\alpha(1+\tau_i) + \sqrt{\alpha^2(1+\tau_i)^2 + 4\alpha\tau_i}}{2}\right)^{i+1} \big\|\boldsymbol{X}(0) - \boldsymbol{X}^*\big\|_F. \tag{42}$$

Theorem 3 provides convergence rate behaviour proof for the case where τ_i is constant $\forall i$. The more elaborate case where τ_i follows the policy described in (40) is left as an open question for future work. To provide some insight for (42), for $\tau_i = 1/4,\ \forall i$ and $\tau_i = 1/2,\ \forall i$, $\delta_{4k}(\boldsymbol{\mathcal{A}}) < 0.1187$ and $\delta_{4k}(\boldsymbol{\mathcal{A}}) < 0.095$ guarantee convergence in Algorithm 3, respectively. Moreover, Figure 2(b) shows acceleration in practice compared to the zero-memory case (MATRIX ALPS I).

VIII. ACCELERATING MATRIX ALPS: ϵ-APPROXIMATION OF SVD VIA COLUMN SUBSET SELECTION

A time-complexity bottleneck in the proposed schemes is the computation of the singular value decomposition to find subspaces that describe the, yet, unexplored information in matrix $\boldsymbol{X}^*$. Unfortunately, following the Eckart-Young theorem, the computational cost of SVD for best subspace tracking is prohibitive for many applications.

Based on [43], [44], we can obtain randomized SVD approximations of a matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ using *column subset selection* ideas: given $\boldsymbol{X}$, we compute leverage scores for each column that represent their "significance". In particular, we define a probability distribution that weights each column depending on the amount of information they contain—usually, the distribution is related to the ℓ_2-norm of the columns. The main idea of this approach is to compute a surrogate rank-k matrix $\mathcal{P}_k^{\epsilon}(\boldsymbol{X})$ by subsampling the columns according to this distribution. It turns out that the total number of sampled columns is a function of the parameter ϵ. Moreover, [45], [46] proved that, given a target rank k and an approximation parameter ϵ, we can compute an ϵ-approximate rank-k matrix $\mathcal{P}_k^{\epsilon}(\boldsymbol{X})$ according to the following defintion.

Definition 4. *[ϵ-approximate low-rank projection] Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ be an arbitrary matrix. Then, $\mathcal{P}_k^{\epsilon}(\boldsymbol{X})$ projection provides a rank-k matrix approximation to $\boldsymbol{X}$ such that:*

$$\big\|\mathcal{P}_k^{\epsilon}(\boldsymbol{X}) - \boldsymbol{X}\big\|_F^2 \leq (1+\epsilon)\big\|\mathcal{P}_k(\boldsymbol{X}) - \boldsymbol{X}\big\|_F^2, \tag{43}$$

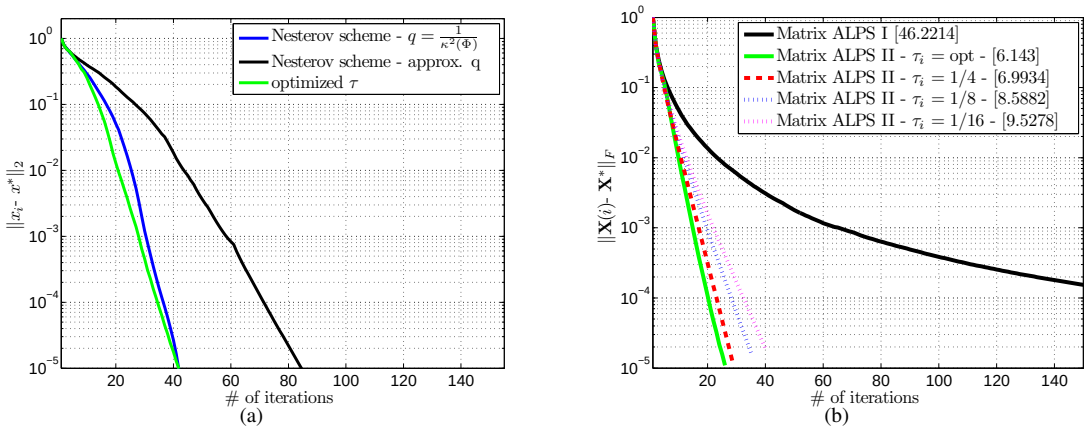

Fig. 2. *(a) Convergence rate example using memory for the vector case in (1) with $n = 2000, p = 600$ and sparsity level $s = 120$. Blue and black lines represent Nesterov's τ_i selection scheme with $q = \frac{\sigma^2_{min}(\mathbf{\Phi}^*\mathbf{\Phi})}{\sigma^2_{max}(\mathbf{\Phi}^*\mathbf{\Phi})}$ and $q \sim \frac{\mu_i^{min}}{\mu_i^{max}}$, respectively; green line represents the proposed momentum step size selection. (b). Median error per iteration for various momentum step size policies and 50 Monte-Carlo repetitions. Here, $m = n = 256$, $p = 0.4n^2$, and rank $k = 22$. We use permuted and subsampled noiselets for the linear map $\boldsymbol{\mathcal{A}}$. In brackets, we present the median time for convergence in seconds.*

where $\mathcal{P}_k(\boldsymbol{X}) = \arg\min_{\boldsymbol{Y}:rank(\boldsymbol{Y})\leq k} \|\boldsymbol{X} - \boldsymbol{Y}\|_F$.

Using ϵ-approximation schemes to perform the Active subspace selection Step, the following upper bound holds. The proof is provided in the Appendix:

Lemma 7. *[ϵ-approximate active subspace expansion] Let $\boldsymbol{X}(i) \in \mathcal{R}^{m\times n}$ be the matrix estimate at the i-th iteration and let $\mathcal{X}_i$ be a set of orthonormal, rank-1 matrices such that $\mathcal{X}_i \leftarrow ortho(\boldsymbol{X}(i))$. Furthermore, let $\mathcal{D}_i^\epsilon := ortho\Big(\mathcal{P}_k^\epsilon\big(\mathcal{P}_{\mathcal{X}_i}^\perp \nabla f(\boldsymbol{X}(i))\big)\Big)$ be a set of orthonormal, rank-1 matrices that span rank-k subspace such that (43) is satisfied for $\boldsymbol{X} := \mathcal{P}_{\mathcal{X}_i}^\perp \nabla f(\boldsymbol{X}(i))$. Then, at each iteration, the Active Subspace Expansion step in Algorithms 1 and 2 captures information contained in the true matrix $\boldsymbol{X}^*$, such that:*

$$\begin{aligned}\big\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\big\|_F \leq \big(2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}) + \sqrt{\epsilon}(1 + 2\delta_{2k}(\boldsymbol{\mathcal{A}}) + \delta_k(\boldsymbol{\mathcal{A}})\big)\big\|\boldsymbol{X}(i) - \boldsymbol{X}^*\big\|_F \\ + \sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\big\|\boldsymbol{\varepsilon}\big\|_2 + \sqrt{\epsilon}\big\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F,\end{aligned} \tag{44}$$

where $\mathcal{S}_i = \mathcal{X}_i \cup \mathcal{D}_i^\epsilon$ and $\mathcal{X}^* \leftarrow ortho(\boldsymbol{X}^*)$.

Furthermore, to prove the following theorems, we extend Lemma (11) as follows. The proof easily follows from the proof of Lemma (11), using Definition (4):

Lemma 8. *[ϵ-approximation rank-k subspace selection] Let $\boldsymbol{V}(i) \in \mathcal{R}^{m\times n}$ be a rank-$2k$ proxy matrix in the subspace spanned by $\mathcal{S}_i$ and let $\widehat{\boldsymbol{W}}(i) \leftarrow \mathcal{P}_k^\epsilon(\boldsymbol{V}(i))$ denote the rank-k ϵ-approximation to $\boldsymbol{V}(i)$, according to (13). Then:*

$$\big\|\widehat{\boldsymbol{W}}(i) - \boldsymbol{V}(i)\big\|_F^2 \leq (1+\epsilon)\big\|\boldsymbol{W}(i) - \boldsymbol{V}(i)\big\|_F \leq (1+\epsilon)\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F \leq (1+\epsilon)\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F. \tag{45}$$

where $\boldsymbol{W}(i) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$.

A. MATRIX ALPS I *using ϵ-approximate low-rank projection via column subset selection*

Using ϵ-approximate SVD in MATRIX ALPS I, the following iteration invariant theorem holds:

Theorem 4. *[Iteration invariant with ϵ-approximate projections for* MATRIX ALPS I*] Assume $\big\|\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F \leq \lambda$ for some constant $\lambda > 0$. The $(i+1)$-th matrix estimate $\boldsymbol{X}(i+1)$ of* MATRIX ALPS I *with ϵ-approximate projections $\mathcal{D}_i^\epsilon \leftarrow ortho\Big(\mathcal{P}_k^\epsilon\big(\mathcal{P}_{\mathcal{X}_i}^\perp \nabla f(\boldsymbol{X}(i))\big)\Big)$ and $\widehat{\boldsymbol{W}}(i) \leftarrow \mathcal{P}_k^\epsilon(\boldsymbol{V}(i))$ in Algorithm 1 satisfies the following recursion:*

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F \leq \rho\big\|\boldsymbol{X}(i) - \boldsymbol{X}^*\big\|_F + \gamma\|\boldsymbol{\varepsilon}\|_2 + \beta\lambda, \tag{46}$$

where

$$\rho := \big(1 + \frac{3\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big)(2+\epsilon)\left[\Big(1 + \frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(4\delta_{3k}(\boldsymbol{\mathcal{A}}) + \sqrt{\epsilon}(1+3\delta_{2k}(\boldsymbol{\mathcal{A}}))\Big) + \frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right], \tag{47}$$

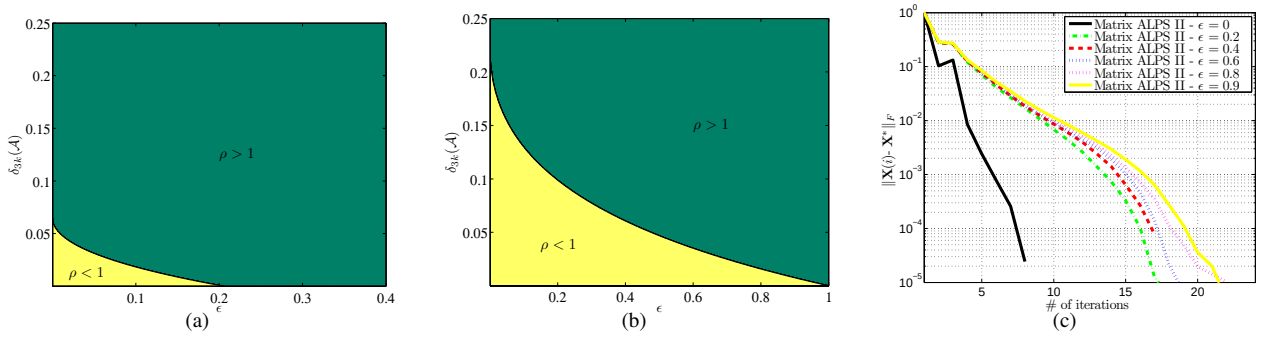

Fig. 3. *(a)-(b)Trade-off curve between ϵ-approximation parameter and convergence conditions in terms of $\delta_{3k}(\mathcal{A})$ RIP constant. (c) Performance comparison using ϵ-approximation SVD [46] in* MATRIX ALPS II. *$m = n = 256$, $p = 0.4n^2$, rank of $\boldsymbol{X}^*$ equals 2 and $\mathcal{A}$ constituted by permuted noiselets. The non-smoothness in the error curves is due to the extreme low rank-ness of $\boldsymbol{X}^*$ for this problem setting.*

$$\beta := \left(1 + \frac{3\delta_k(\mathcal{A})}{1-\delta_k(\mathcal{A})}\right)(2+\epsilon)\left(1 + \frac{\delta_{3k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\right)\sqrt{\epsilon}, \tag{48}$$

and

$$\gamma := \left(1 + \frac{3\delta_k(\mathcal{A})}{1-\delta_k(\mathcal{A})}\right)(2+\epsilon)\left[\left(1 + \frac{\delta_{3k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\right)\sqrt{2(1+\delta_{2k}(\mathcal{A}))} + 2\frac{\sqrt{1+\delta_{2k}(\mathcal{A})}}{1-\delta_{2k}(\mathcal{A})}\right]. \tag{49}$$

Figure 3(a) shows the trade-off between approximation parameter ϵ and necessary R-RIP conditions on $\delta_{3k}(\mathcal{A})$ such that $\rho < 1$. We observe that the space $(\delta_{3k}(\mathcal{A}), \epsilon) \in (0,1) \times (0,1)$ is partitioned into two regions, defining a phase transition curve on the boundary of the two parts. We note that, due to different convergence analysis compared to Section V, MATRIX ALPS I in Theorem 4 satisfies $\rho < 1$ for $\epsilon = 0$ *if and only if* $\delta_{3k}(\mathcal{A}) < 0.0637$—this complies with Figure 3(a) on the vertical axis for $\epsilon = 0$.

B. ADMiRA using ϵ-approximate low-rank projection via column subset selection

Similarly, the following theorem holds for ADMiRA using approximate SVDs:

Theorem 5. *[Iteration invariant with ϵ-approximate projections for ADMiRA] Assume $\left\|\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F \leq \lambda$ for some constant $\lambda > 0$. The $(i+1)$-th matrix estimate $\boldsymbol{X}(i+1)$ of ADMiRA in Algorithm 2 with ϵ-approximate projections $\mathcal{D}_i \leftarrow \text{ortho}\left(\mathcal{P}_k^{\epsilon}\left(\mathcal{P}_{\mathcal{X}_i}^{\perp}\nabla f(\boldsymbol{X}(i))\right)\right)$ and $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k^{\epsilon}(\boldsymbol{V}(i))$ answers the following recursive expression:*

$$\left\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\right\|_F \leq \rho\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_F + \beta\lambda,$$

where

$$\rho := \sqrt{\frac{1+(1+\epsilon+2\sqrt{1+\epsilon})\delta_{3k}^2(\mathcal{A})}{1-\delta_{3k}^2(\mathcal{A})}}\left(2\delta_{2k}(\mathcal{A}) + 2\delta_{3k}(\mathcal{A}) + \sqrt{\epsilon}(1 + 2\delta_{2k}(\mathcal{A}) + \delta_k(\mathcal{A})\right), \tag{50}$$

$$\beta := \sqrt{\epsilon}\cdot\sqrt{\frac{1+(1+\epsilon+2\sqrt{1+\epsilon})\delta_{3k}^2(\mathcal{A})}{1-\delta_{3k}^2(\mathcal{A})}}, \tag{51}$$

and

$$\gamma := \sqrt{1+(1+\epsilon+2\sqrt{1+\epsilon})\delta_{3k}^2(\mathcal{A})}\left(1 + \frac{\sqrt{1+\delta_{2k}(\mathcal{A})}}{1-\delta_{3k}(\mathcal{A})} + \sqrt{\frac{2(1+\delta_{2k}(\mathcal{A}))}{1-\delta_{3k}^2(\mathcal{A})}}\right). \tag{52}$$

We need the following lemma to prove the result above—a detailed proof can be found in the Appendix.

Lemma 9. *Let $\boldsymbol{V}(i)$ be the least squares solution in Step 3 of the ADMiRA algorithm and let $\boldsymbol{X}(i+1)$ be a proxy, rank-k matrix to $\boldsymbol{V}(i)$ according to: $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k^{\epsilon}(\boldsymbol{V}(i))$. Then, $\left\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\right\|_F$ can be expressed in terms of the distance from $\boldsymbol{V}(i)$ to $\boldsymbol{X}^*$ as follows:*

$$\left\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\right\|_F \leq \sqrt{1+\left((1+\epsilon)+2\sqrt{1+\epsilon}\right)\delta_{3k}^2(\mathcal{A})}\left(\left\|\boldsymbol{V}(i) - \boldsymbol{X}^*\right\|_F + \sqrt{\frac{\left((1+\epsilon)+2\sqrt{1+\epsilon}\right)\left(1+\delta_{2k}(\mathcal{A})\right)}{1+\left((1+\epsilon)+2\sqrt{1+\epsilon}\right)\delta_{3k}^2(\mathcal{A})}}\left\|\boldsymbol{\varepsilon}\right\|_2\right). \tag{53}$$

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, k, q, Tolerance η, MaxIterations
Initialize: $\boldsymbol{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $\boldsymbol{Q}(0) \leftarrow 0$, $\mathcal{Q}_0 \leftarrow \{\emptyset\}$, $\tau_i \ \forall i$, $i \leftarrow 0$
repeat
1: $\mathcal{D}_i \leftarrow \text{RANDOMIZEDPOWERITERATION}\big(\mathcal{P}_{\mathcal{Q}_i}^{\perp} \nabla f(\mathbf{Q}(i)),\ k,\ q\big)$ *(Rank-k subspace via Randomized Power Iteration)*
2: $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{Q}_i$ *(Active subspace expansion)*
3: $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\boldsymbol{Q}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))\|_2^2}$ *(Step size selection)*
4: $\boldsymbol{V}(i) \leftarrow \boldsymbol{Q}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{Q}(i))$ *(Error norm reduction via gradient descent)*
5: $\mathcal{U} \leftarrow \text{RANDOMIZEDPOWERITERATION}\big(\mathbf{V}(i),\ k,\ q\big)$ *(Rank-k subspace via Randomized Power Iteration)*
6: $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_{\mathcal{U}}\mathbf{V}(i)$ *(Best rank-k subspace selection)*
7: $\boldsymbol{Q}(i+1) \leftarrow \boldsymbol{X}(i+1) + \tau_i(\boldsymbol{X}(i+1) - \boldsymbol{X}(i))$ *(Momentum update)*
8: $\mathcal{Q}_{i+1} \leftarrow \text{ortho}(\mathcal{X}_i \cup \mathcal{X}_{i+1})$
$i \leftarrow i+1$
until $\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\|_2 \leq \eta\|\boldsymbol{X}(i)\|_2$ or MaxIterations.

Algorithm 4: Randomized MATRIX ALPS II with QR Factorization

Then, the proof of Theorem 5 easily follows by combining Lemma 7—10. Following the same process as in MATRIX ALPS I algorithm, we easily derive the trade-off curve as depicted in Figure 3(b). The same remarks apply here as in the MATRIX ALPS I case.

To illustrate the impact of SVD ϵ-approximation on the signal reconstruction performance of the proposed algorithms, we replace the *best* rank-k projections in steps 1 and 5 of Algorithm 1 by the ϵ-approximation SVD algorithm, presented in [46]. In this paper, the column subset selection algorithm satisfies the following theorem:

Theorem 6. *Let $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ be a signal of interest with arbitrary rank $< \min\{m,n\}$ and let $\boldsymbol{X}_k$ represent the best rank-k approximation of $\boldsymbol{X}$. After $2(k+1)(\log(k+1)+1)$ passes over the data, the Linear Time Low-Rank Matrix Approximation algorithm in [46] computes a rank-k approximation $\mathcal{P}_k^{\epsilon}(\boldsymbol{X}) \in \mathcal{R}^{m\times n}$ such that Definition 4 is satisfied with probability at least 3/4.*

The proof of Theorem 6 is provided in [46]. In total, Linear Time Low-Rank Matrix Approximation algorithm [46] requires $O(mn(k/\epsilon + k^2\log k) + (m+n)(k^2/\epsilon^2 + k^3\log k/\epsilon + k^4\log^2 k))$ time-complexity and $O(\min\{m,n\}(k/\epsilon + k^2\log k))$ space complexity. However, while column subset selection methods such as [46] reduce the overall complexity of low-rank projections in theory, in practice this applies only in very high-dimensional settings. To strengthen this argument, in Figure 3(c) we compare SVD-based MATRIX ALPS II with MATRIX ALPS II using the ϵ-approximate column subset selection method in [46]. We observe that the total number of iterations for convergence increases due to ϵ-approximate low-rank projections, as expected. Nevertheless, we observe that, on average, the column subset selection process [46] is computationally prohibitive compared to regular SVD calculation due to the time overhead in the column selection procedure—fewer passes over the data are desirable in practice to trade-off the increased number of iterations for convergence. In the next Section, we present alternatives based on recent trends in randomized matrix decompositions and how we can use them in low-rank recovery.

IX. ACCELERATING MATRIX ALPS: SVD APPROXIMATION USING RANDOMIZED MATRIX DECOMPOSITIONS

Finding low-cost SVD approximations to tackle the above complexity issues is a challenging task. Recent works on probabilistic methods for matrix approximation [25] dictate a family of efficient approximate projections on the set of rank-deficient matrices with clear computational advantages over regular SVD computation in practice and attractive theoretical guarantees. In this work, we elaborate over the low-cost, power-iteration *subspace tracking* scheme, described in Algorithms 4.3 and 4.4 in [25]. Our proposed algorithm is described in Algorithm 4.

The convergence guarantees of Algorithm 4 follow the same motions described in Section VIII, where ϵ is a function of $m,\ n,\ k$ and q. An extensive theoretical study on randomized low-rank approximations and their impact in the ARM problem is left for future work.

X. EXPERIMENTS

A. List of algorithms

In the following experiments, we compare algorithms drawn from the following list: (i) the Singular Value Projection (SVP) algorithm [5], a non-convex first-order projected gradient descent algorithm with *constant* step size selection (here, we study the case where $\mu = 1$), (ii) the inexact ALM algorithm [18] based on augmented Langrance multipliers, (iii) the OptSpace algorithm [47], a gradient descent algorithm on the Grassmann manifold, (iv) the Grassmannian Rank-One Update Subspace Estimation (GROUSE) and the Grassmannian Robust Adaptive Subspace Tracking (GRASTA) algorithms [48], [49], two stochastic gradient descent algorithms that operate on the Grassmannian—moreover, to allay the impact of outliers in the subspace selection step, GRASTA incorporates the augmented Lagrangian of ℓ_1-norm loss function into the Grassmannian optimization framework, (v) the Riemannian Trust Region Matrix Completion (RTRMC) algorithm [50], a matrix completion method using first- and second-order Riemannian trust-region approaches, (vi) the Low rank Matrix Fitting algorithm (LMatFit) [51], a nonlinear successive over-relaxation algorithm and (vii) the algorithms MATRIX ALPS I, ADMiRA [21], MATRIX ALPS II and Randomized MATRIX ALPS II with QR Factorization (referred shortly as MATRIX ALPS II with QR) presented in this paper.

B. Implementation details

To properly compare the algorithms in the above list, we preset a set of parameters that are common. We denote the ratio between the number of observed samples and the number of variables in $\boldsymbol{X}^*$ as SR $:= p/(m \cdot n)$ (sampling ratio). Furthemore, we reserve FR to represent the degree of freedom in a rank-k matrix to the number of observations—this corresponds to the following definition FR $:= (k(m+n-k))/p$. In most of the experiments, we fix the number of observable data $p = 0.3mn$ and vary the dimensions and the rank k of the matrix $\boldsymbol{X}^*$. This way, we create a wide range of different problem configurations with variable FR.

In all algorithms, we fix the maximum number of iterations to 700, unless otherwise stated. To solve a least squares problem over a restricted low-rank subspace, we use conjugate gradients with maximum number of iterations given by $\mathrm{cg\_maxiter} := 500$ and tolerance parameter $\mathrm{cg\_tol} := 10^{-10}$. We use the same stopping criteria for the majority of algorithms under consideration:

$$\frac{\left\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\right\|_F}{\left\|\boldsymbol{X}(i)\right\|_F} \leq \mathrm{tol}, \tag{54}$$

where $\boldsymbol{X}(i),\ \boldsymbol{X}(i-1)$ denote the current and the previous estimate of $\boldsymbol{X}^*$ and $\mathrm{tol} := 5 \cdot 10^{-5}$. If this is not the case, we tweak the algorithms to minimize the total execution time and achieve similar reconstruction performance as the rest of the algorithms. For SVD calculations, we use the lansvd implementation in PROPACK package [52]—moreover, all the algorithms in comparison use the same linear operators $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^*$ for gradient and SVD calculations and conjugate-gradient least-squares minimizations. For fairness, we modified all the algorithms so that they *exploit the true rank*.

C. Limitations of $\left\|\cdot\right\|_*$-based algorithms: a toy example

While nucluear norm heuristic is widely used in solving the low-rank minimization problem with impressive reconstruction performance in polynomial time cost, [53] presents simple problem cases where convex, nuclear norm-based, algorithms *fail* in practice. Using the $\left\|\cdot\right\|_*$-norm in the objective function as the convex surrogate of the rank$(\cdot)$ metric might lead to a candidate set with multiple solutions, introducing ambiguity in the selection process. Borrowing the example in [53], we test the list of algorithms above on a toy problem setting. To this end, we design the following problem: let $\boldsymbol{X}^* \in \mathcal{R}^{5\times 4}$ be the matrix of interest with rank$(\boldsymbol{X}^*) = 2$, as shown in Figure 4(a). We consider the case where we have access to $\boldsymbol{X}^*$ only through a subset of its entries, as shown in Figure 4(b).

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ ? & ? & ? & 1 \\ 2 & ? & ? & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(a) (b)

Fig. 4. *Matrix Completion toy example for $\boldsymbol{X}^* \in \mathcal{R}^{5\times 4}$. We reserve '?' to denote the unobserved entried.*

In Figure 5, we present the reconstruction performance of various matrix completion solvers after 300 iterations. Although there are multiple solutions that induce the recovered matrix and have the same rank as $\boldsymbol{X}^*$, most of the algorithms in comparison reconstruct $\boldsymbol{X}^*$ successfully. We note that, in some cases, the inadequancy of an algorithm to reconstruct $\boldsymbol{X}^*$ is not because of the (relaxed) problem formulation but due to its fast—but inaccurate—implementation (fast convergence versus reconstruction accuracy trade-off).

D. Synthetic data

General affine rank minimization using noiselets: In this experiment, the set of observations $\boldsymbol{y} \in \mathcal{R}^p$ satisfy:

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}\boldsymbol{X}^* + \boldsymbol{\varepsilon} \tag{55}$$

Here, we use permuted and subsampled noiselets for the linear operator $\boldsymbol{\mathcal{A}}$ [10]. The signal $\boldsymbol{X}^*$ is generated as the multiplication of two low-rank matrices, $\mathbf{L} \in \mathcal{R}^{m\times k}$ and $\mathbf{R} \in \mathcal{R}^{n\times k}$, such that $\boldsymbol{X}^* = \mathbf{L}\mathbf{R}^T$ and $\left\|\boldsymbol{X}^*\right\|_F = 1$. Both $\mathbf{L}$ and $\mathbf{R}$ have random independent and identically distributed (iid) Gaussian entries with zero mean and unit variance. In the noisy case, the additive noise term $\boldsymbol{\varepsilon} \in \mathcal{R}^p$ contains entries drawn from a zero mean Gaussian distribution with $\left\|\boldsymbol{\varepsilon}\right\|_2 \in \{10^{-3}, 10^{-4}\}$.

We compare the following algorithms: SVP, ADMiRA, MATRIX ALPS I, MATRIX ALPS II and MATRIX ALPS II with QR for various problem configurations, as depicted in Table I (there is no available code with arbitrary sensing operators for the rest algorithms). In Table I, we show the median values of reconstruction error, number of iterations and execution time over 50 Monte Carlo iterations. For all cases, we assume SR $= 0.3$ and we set the maximum number of iterations to 700. Bold font denotes the fastest execution time. Furthermore, Figure 6 illustrates the effectiveness of the algorithms for some representative problem configurations.

In Table 6, MATRIX ALPS II and MATRIX ALPS II with QR obtain accurate low-rank solutions much faster than the rest of the algorithms in comparison. In high dimensional settings, MATRIX ALPS II with QR scales better as the problem dimensions

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(a) SVT

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(b) FPC

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(c) SVP ($\mu = 1$)

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(d) ALM

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 0 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(e) OptSpace

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(f) SET

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(g) ADMiRA

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(h) GRASTA

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(i) LMatFit

$$\begin{pmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix}$$

(j) MATRIX ALPS II

Fig. 5. *Toy example reconstruction performance for various algorithms. We observe that $\boldsymbol{X}^*$ is an integer matrix—since the algorithms under consideration return real matrices as solutions, we round the solution elementwise.*

TABLE I
GENERAL ARM USING NOISELETS.

Configuration				FR	SVP			ADMiRA			MATRIX ALPS I		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
256	512	5	0	0.097	38	$2.2 \cdot 10^{-4}$	0.78	27	$4.4 \cdot 10^{-5}$	2.26	13.5	$1 \cdot 10^{-5}$	0.7
256	512	5	10^{-3}	0.097	38	$6 \cdot 10^{-4}$	0.91	700	$2 \cdot 10^{-3}$	65.94	16	$7 \cdot 10^{-4}$	0.92
256	512	5	10^{-4}	0.097	38	$2.1 \cdot 10^{-4}$	0.94	700	$4.1 \cdot 10^{-4}$	69.03	11.5	$7.9 \cdot 10^{-5}$	0.72
256	512	10	0	0.193	50	$3.4 \cdot 10^{-4}$	1.44	38	$5 \cdot 10^{-5}$	4.42	13	$3.9 \cdot 10^{-5}$	0.92
256	512	10	10^{-3}	0.193	50	$9 \cdot 10^{-4}$	1.39	700	$1.7 \cdot 10^{-3}$	56.94	29	$1.2 \cdot 10^{-3}$	1.78
256	512	10	10^{-4}	0.193	50	$3.5 \cdot 10^{-4}$	1.38	700	$9.3 \cdot 10^{-5}$	64.69	14	$1.4 \cdot 10^{-4}$	0.93
256	512	20	0	0.38	86	$7 \cdot 10^{-4}$	3.32	700	$4.1 \cdot 10^{-5}$	81.93	45	$2 \cdot 10^{-4}$	4.09
256	512	20	10^{-3}	0.38	86	$1.5 \cdot 10^{-3}$	3.45	700	$4.2 \cdot 10^{-2}$	77.35	69	$2.3 \cdot 10^{-3}$	5.05
256	512	20	10^{-4}	0.38	86	$7 \cdot 10^{-4}$	3.26	700	$4 \cdot 10^{-2}$	79.47	46	$4 \cdot 10^{-4}$	4.1
512	1024	30	0	0.287	66	$4.9 \cdot 10^{-4}$	8.79	295	$5.4 \cdot 10^{-5}$	143.53	24	$1 \cdot 10^{-4}$	8.01
512	1024	40	0	0.38	86	$7 \cdot 10^{-4}$	10.09	700	$4.3 \cdot 10^{-2}$	251.27	45	$2 \cdot 10^{-4}$	11.08
1024	2048	50	0	0.24	57	$4.3 \cdot 10^{-4}$	42.88	103	$5.2 \cdot 10^{-5}$	312.62	18	$5.7 \cdot 10^{-5}$	35.86

					MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time
256	512	5	0	0.097	8	$7.1 \cdot 10^{-6}$	0.42	10	$9.1 \cdot 10^{-6}$	**0.39**
256	512	5	10^{-3}	0.097	9	$7 \cdot 10^{-4}$	**0.56**	20	$7 \cdot 10^{-4}$	0.93
256	512	5	10^{-4}	0.097	8	$7 \cdot 10^{-5}$	0.5	10	$7.8 \cdot 10^{-5}$	**0.46**
256	512	10	0	0.193	10	$2.3 \cdot 10^{-5}$	0.68	13	$2.4 \cdot 10^{-5}$	**0.64**
256	512	10	10^{-3}	0.193	19	$1 \cdot 10^{-3}$	**1.29**	27	$1 \cdot 10^{-3}$	1.35
256	512	10	10^{-4}	0.193	10	$1.1 \cdot 10^{-4}$	0.68	13	$1.1 \cdot 10^{-4}$	**0.62**
256	512	20	0	0.38	21	$1 \cdot 10^{-4}$	1.92	24	$1 \cdot 10^{-4}$	**1.26**
256	512	20	10^{-3}	0.38	36	$1.5 \cdot 10^{-3}$	2.67	39	$1.5 \cdot 10^{-3}$	**1.69**
256	512	20	10^{-4}	0.38	21	$2 \cdot 10^{-4}$	1.87	24	$2 \cdot 10^{-4}$	**1.22**
512	1024	30	0	0.287	14	$4.5 \cdot 10^{-5}$	4.7	18	$3.3 \cdot 10^{-5}$	**4.15**
512	1024	40	0	0.38	21	$1 \cdot 10^{-4}$	6.01	24	$1 \cdot 10^{-4}$	**4.53**
1024	2048	50	0	0.24	12	$2.5 \cdot 10^{-5}$	22.76	15	$3.3 \cdot 10^{-5}$	**17.94**

increase, leading to faster convergence. Moreover, its execution time is at least a few orders of magnitude smaller compared to SVP, ADMiRA and MATRIX ALPS I implementations.

Robust matrix completion: We design matrix completion problems in the following way. The signal of interest $\boldsymbol{X}^* \in \mathcal{R}^{m \times n}$ is synthesized as a rank-k matrix, factorized as $\boldsymbol{X}^* := \mathbf{L}\mathbf{R}^T$ with $\left\|\boldsymbol{X}^*\right\|_F = 1$ where $\mathbf{L} \in \mathcal{R}^{m \times k}$ and $\mathbf{R} \in \mathcal{R}^{n \times k}$ as defined above. In sequence, we subsample $\boldsymbol{X}^*$ by observing $p = 0.3mn$ entries, drawn uniformly at random. We denote the set of ordered pairs that represent the coordinates of the observable entries as $\Omega = \{(i,j) : [\boldsymbol{X}^*]_{ij} \text{ is known}\} \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$

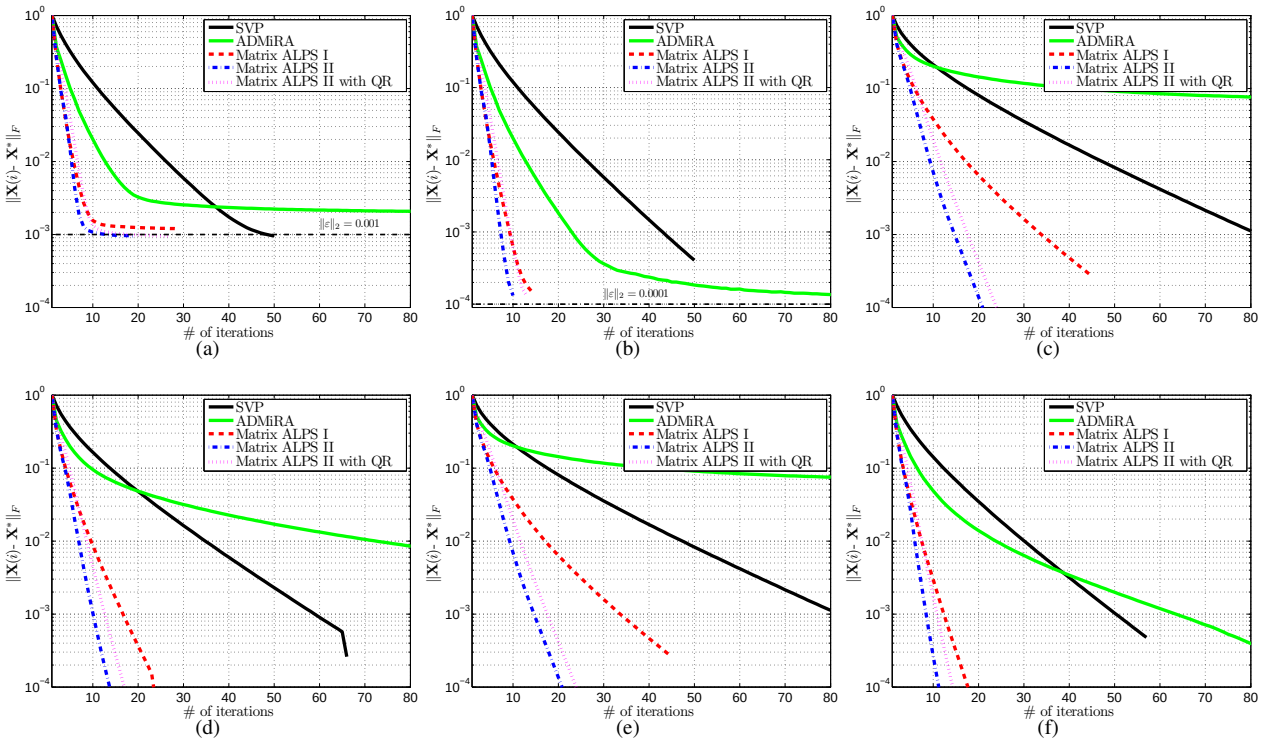

Fig. 6. *Low rank signal reconstruction using noiselet linear operator. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $p = 0.3mn$. *(a)* $m = 256$, $n = 512$, $k = 10$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-3}$. *(b)* $m = 256$, $n = 512$, $k = 10$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-4}$. *(c)* $m = 256$, $n = 512$, $k = 20$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(d)* $m = 512$, $n = 1024$, $k = 30$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(e)* $m = 512$, $n = 1024$, $k = 40$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(f)* $m = 1024$, $n = 2048$, $k = 50$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$.

and let $\boldsymbol{\mathcal{A}}_\Omega$ denote the linear operator (mask) that samples a matrix according to Ω. Then, the set of observations satisfies:

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}_\Omega \boldsymbol{X}^* + \boldsymbol{\varepsilon}, \tag{56}$$

i.e., the known entries of $\boldsymbol{X}^*$ are structured as a vector $\boldsymbol{y} \in \mathcal{R}^p$, disturbed by a dense noise vector $\boldsymbol{\varepsilon} \in \mathcal{R}^p$ with fixed-energy, which is populated by iid zero-mean Gaussians.

To demonstrate the reconstruction accuracy and the convergence speeds, we generate various problem configurations (both noisy and noiseless settings), according to (56). The energy of the additive noise takes values $\|\boldsymbol{\varepsilon}\|_2 \in \{10^{-3}, 10^{-4}\}$. All the algorithms are tested for the same signal-matrix-noise realizations. A summary of the results can be found in Tables II, III and, IV where we present the median values of reconstruction error, number of iterations and execution time over 50 Monte Carlo iterations. For all cases, we assume SR $= 0.3$ and set the maximum number of iterations to 700. Bold font denotes the fastest execution time. Some convergence error curves for specific cases are illustrated in Figures 7 and 8.

In Table 7, LMaFit [51] implementation presents the fastest convergence for small scale problem configuration where $m = 300$ and $n = 600$. We note that part of LMaFit implementation uses C code for acceleration. GROUSE [48] is a competitive low-rank recovery method with attractive execution times for the *extreme low rank* problem settings due to stochastic gradient descent techniques. Nevertheless, its execution time performance degrades significantly as we increase the rank of $\boldsymbol{X}^*$. Moreover, we observe how randomized low rank projections accelerate the convergence speed where MATRIX ALPS II with QR converges faster than MATRIX ALPS II. In Tables III and IV, we increase the problem dimensions. Here, MATRIX ALPS II with QR has faster convergence for most of the cases and scales well as the problem size increases. We note that we do not exploit stochastic gradient descent techniques in the recovery process to accelerate convergence which is left for future work.

E. Real data

We use real data images to highlight the reconstruction performance of the proposed schemes. To this end, we perform grayscale image denoising from an incomplete set of observed pixels—similar experiments can be found in [51]. Based on the matrix completion setting, we observe a limited number of pixels from the original image and perform a low rank approximation based only on the set of measurements. While the true underlying image might not be low-rank, we apply our solvers to obtain low-rank approximations.

TABLE II
MATRIX COMPLETION PROBLEM FOR $m = 300$ AND $n = 600$. "$-$" DEPICTS NO INFORMATION OR NOT APPLICABLE DUE TO TIME OVERHEAD.

Configuration				FR	SVP			GROUSE			TFOCS		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	43	$2.9 \cdot 10^{-4}$	0.59	–	$1.52 \cdot 10^{-4}$	0.08	–	$8.69 \cdot 10^{-5}$	3.36
300	600	5	10^{-3}	0.083	42	$6 \cdot 10^{-4}$	0.65	–	$2 \cdot 10^{-4}$	0.082	–	$5 \cdot 10^{-4}$	3.85
300	600	5	10^{-4}	0.083	43	$3 \cdot 10^{-4}$	0.64	–	$2 \cdot 10^{-4}$	0.079	–	$1 \cdot 10^{-4}$	3.5
300	600	10	0	0.165	54	$4 \cdot 10^{-4}$	0.9	–	$4.5 \cdot 10^{-6}$	0.22	–	$2 \cdot 10^{-4}$	6.43
300	600	10	10^{-3}	0.165	54	$9 \cdot 10^{-4}$	0.89	–	$2 \cdot 10^{-4}$	0.16	–	$8 \cdot 10^{-4}$	7.83
300	600	10	10^{-4}	0.165	54	$4 \cdot 10^{-4}$	0.91	–	$2 \cdot 10^{-4}$	0.16	–	$1 \cdot 10^{-4}$	6.75
300	600	20	0	0.326	85	$8 \cdot 10^{-4}$	2.04	–	$1 \cdot 10^{-4}$	0.81	–	$2 \cdot 10^{-4}$	30.04
300	600	40	0	0.637	241	$3.4 \cdot 10^{-3}$	11.1	–	$3.1 \cdot 10^{-3}$	13.94	–	–	–
					Inexact ALM			OptSpace			GRASTA		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	24	$6.7 \cdot 10^{-5}$	0.47	31	$2.8 \cdot 10^{-6}$	2.41	–	$2.2 \cdot 10^{-4}$	2.07
300	600	5	10^{-3}	0.083	24	$6 \cdot 10^{-4}$	0.49	297	$5 \cdot 10^{-4}$	22.82	–	$1 \cdot 10^{-4}$	2.07
300	600	5	10^{-4}	0.083	24	$1 \cdot 10^{-4}$	0.49	267	$1 \cdot 10^{-4}$	21.56	–	$8 \cdot 10^{-5}$	2.1
300	600	10	0	0.165	26	$1 \cdot 10^{-4}$	0.6	37	$2.3 \cdot 10^{-6}$	8.42	–	$8.6 \cdot 10^{-6}$	4.5
300	600	10	10^{-3}	0.165	26	$8 \cdot 10^{-4}$	0.59	304	$8 \cdot 10^{-4}$	66.02	–	$5.5 \cdot 10^{-3}$	3.43
300	600	10	10^{-4}	0.165	26	$1 \cdot 10^{-4}$	0.61	304	$1 \cdot 10^{-4}$	65.56	–	$5.3 \cdot 10^{-3}$	3.44
300	600	20	0	0.326	44	$3 \cdot 10^{-4}$	1.37	–	–	–	–	$5 \cdot 10^{-4}$	10.51
300	600	40	0	0.637	134	$1.6 \cdot 10^{-3}$	7.08	–	–	–	–	$5.2 \cdot 10^{-3}$	251.34
					RTRMC			LMaFit			MATRIX ALPS I		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	13	$1.2 \cdot 10^{-4}$	0.59	20	$2.2 \cdot 10^{-4}$	**0.054**	22	$1.8 \cdot 10^{-5}$	0.76
300	600	5	10^{-3}	0.083	13	$1 \cdot 10^{-4}$	0.59	19	$5 \cdot 10^{-4}$	**0.049**	37	$7 \cdot 10^{-4}$	1.34
300	600	5	10^{-4}	0.083	13	$2 \cdot 10^{-4}$	0.59	21	$1 \cdot 10^{-4}$	**0.052**	18	$1 \cdot 10^{-4}$	0.61
300	600	10	0	0.165	16	$1.1 \cdot 10^{-3}$	1.03	23	$1 \cdot 10^{-4}$	**0.064**	16	$1 \cdot 10^{-4}$	0.65
300	600	10	10^{-3}	0.165	17	$1 \cdot 10^{-4}$	1.09	26	$8 \cdot 10^{-4}$	**0.077**	30	$1.1 \cdot 10^{-3}$	1.16
300	600	10	10^{-4}	0.165	17	$2 \cdot 10^{-4}$	1.09	32	$1 \cdot 10^{-4}$	**0.097**	16	$1 \cdot 10^{-4}$	0.63
300	600	20	0	0.326	22	$4 \cdot 10^{-4}$	2.99	37	$2 \cdot 10^{-4}$	**0.12**	37	$2 \cdot 10^{-4}$	2.05
300	600	40	0	0.637	35	$3 \cdot 10^{-5}$	11.83	233	$4.9 \cdot 10^{-4}$	**2.52**	500	$6.5 \cdot 10^{-2}$	45.67
					ADMiRA			MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	k	$\|\boldsymbol{\varepsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	59	$5.2 \cdot 10^{-5}$	2.86	10	$1.7 \cdot 10^{-5}$	0.34	14	$3.2 \cdot 10^{-5}$	0.45
300	600	5	10^{-3}	0.083	700	$4 \cdot 10^{-3}$	30.96	12	$6 \cdot 10^{-4}$	0.44	24	$6 \cdot 10^{-4}$	0.81
300	600	5	10^{-4}	0.083	700	$4.5 \cdot 10^{-3}$	31.45	10	$1 \cdot 10^{-4}$	0.36	14	$1 \cdot 10^{-4}$	0.47
300	600	10	0	0.165	47	$1 \cdot 10^{-3}$	2.56	12	$3 \cdot 10^{-5}$	0.48	16	$3.4 \cdot 10^{-5}$	0.49
300	600	10	10^{-3}	0.165	700	$1.5 \cdot 10^{-3}$	28.49	19	$9 \cdot 10^{-4}$	0.74	29	$9 \cdot 10^{-4}$	0.95
300	600	10	10^{-4}	0.165	700	$1 \cdot 10^{-4}$	31.99	12	$1 \cdot 10^{-4}$	0.49	16	$1 \cdot 10^{-4}$	0.54
300	600	20	0	0.326	700	$1.2 \cdot 10^{-3}$	41.86	20	$1 \cdot 10^{-4}$	1.16	23	$1 \cdot 10^{-4}$	0.79
300	600	20	0	0.326	–	–	–	72	$2 \cdot 10^{-4}$	7.21	68	$2 \cdot 10^{-4}$	2.6

Figures 9 and 10 depict the reconstruction results. In the first test case, we use a 512×512 grayscale image as shown in the top left corner of Figure 9. For this case, we observe only the 35% of the total number of pixels, randomly selected—a realization is depicted in the top middle plot in Figure 9. In sequel, we fix the desired rank to $k = 40$. The best rank-40 approximation using SVD is shown in the top right corner of Figure 9 where the full set of pixels is observed. Given a fixed common tolerance and the same stopping criteria, Figure 9 shows the recovery performance achieved by a range of algorithms under consideration for 10 Monte-Carlo realizations. We repeat the same experiment for the second image in Figure 10. Here, the size of the image is 256×256, the desired rank is set to $k = 30$ and we observe the 33% of the image pixels. In constrast to the image denoising procedure above, we measure the reconstruction error of the computed solutions with respect to the *best rank*-30 *approximation* of the true image. In both cases, we note that MATRIX ALPS II has a better phase transition performance as compared to the rest of the algorithms.

XI. CONCLUSIONS

In this paper, we present some new strategies and also review some existing ones for hard thresholding methods for recovering low-rank matrices from dimensionality reducing, linear projections. These methods exploit further problem structure in optimization to reduce computational complexity without sacrificing stability.

TABLE III
MATRIX COMPLETION PROBLEM FOR $m = 700$ AND $n = 1000$. "$-$" DEPICTS NO INFORMATION OR NOT APPLICABLE DUE TO TIME OVERHEAD.

Configuration				FR	SVP			Inexact ALM			GROUSE		
m	n	k	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
700	1000	5	0	0.04	34	$1.9 \cdot 10^{-4}$	1.77	23	$6.5 \cdot 10^{-5}$	1.69	–	$3.5 \cdot 10^{-5}$	**0.23**
700	1000	5	10−3	0.04	34	$4.2 \cdot 10^{-4}$	1.92	23	$3.7 \cdot 10^{-4}$	1.87	–	$3.1 \cdot 10^{-4}$	**0.24**
700	1000	30	0	0.239	61	$4.6 \cdot 10^{-4}$	6.39	29	$1.2 \cdot 10^{-4}$	3.91	–	$3.2 \cdot 10^{-5}$	3.15
700	1000	30	10^{-3}	0.239	61	$1.1 \cdot 10^{-3}$	6.33	29	$1 \cdot 10^{-3}$	3.87	–	$8 \cdot 10^{-4}$	3.14
700	1000	50	0	0.393	95	$8.5 \cdot 10^{-4}$	14.47	49	$3.2 \cdot 10^{-4}$	9.02	–	$1.3 \cdot 10^{-5}$	10.31
700	1000	50	10^{-3}	0.393	95	$1.6 \cdot 10^{-3}$	15.15	49	$1.4 \cdot 10^{-3}$	9.11	–	$8 \cdot 10^{-4}$	10.34
700	1000	110	0	0.833	683	$1.2 \cdot 10^{-2}$	253.1	374	$5.8 \cdot 10^{-3}$	152.61	–	$1.2 \cdot 10^{-1}$	110.93
700	1000	110	10^{-3}	0.833	682	$1.3 \cdot 10^{-2}$	256.21	374	$6.8 \cdot 10^{-3}$	154.34	–	$1.05 \cdot 10^{-1}$	111.05
					LMaFit			MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	k	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
700	1000	5	0	0.04	24	$7.2 \cdot 10^{-6}$	0.67	8	$1.5 \cdot 10^{-5}$	1.15	15	$8.3 \cdot 10^{-5}$	1.05
700	1000	5	10^{-3}	0.04	17	$3.7 \cdot 10^{-4}$	0.5	10	$4.5 \cdot 10^{-4}$	1.38	15	$3.8 \cdot 10^{-4}$	1.1
700	1000	30	0	0.239	34	$9.2 \cdot 10^{-6}$	**1.95**	14	$4.5 \cdot 10^{-5}$	3.69	35	$1.1 \cdot 10^{-4}$	2.6
700	1000	30	10^{-3}	0.239	30	$1 \cdot 10^{-3}$	**1.71**	25	$1.1 \cdot 10^{-3}$	6.1	35	$1 \cdot 10^{-3}$	2.61
700	1000	50	0	0.393	53	$2.7 \cdot 10^{-5}$	4.59	25	$8.6 \cdot 10^{-5}$	8.87	57	$1.6 \cdot 10^{-5}$	**4.47**
700	1000	50	10^{-3}	0.393	52	$1.4 \cdot 10^{-3}$	4.53	40	$1.6 \cdot 10^{-3}$	14.38	57	$1.4 \cdot 10^{-3}$	**4.49**
700	1000	110	0	0.833	584	$9 \cdot 10^{-4}$	101.95	280	$8 \cdot 10^{-4}$	214.93	553	$7 \cdot 10^{-4}$	**51.72**
700	1000	110	10^{-3}	0.833	584	$3.7 \cdot 10^{-3}$	102.15	336	$4.7 \cdot 10^{-3}$	261.98	551	$3.7 \cdot 10^{-3}$	**51.62**

TABLE IV
MATRIX COMPLETION PROBLEM FOR $m = 500$ AND $n = 2000$. "$-$" DEPICTS NO INFORMATION OR NOT APPLICABLE DUE TO TIME OVERHEAD.

Configuration				FR	SVP			Inexact ALM			GROUSE		
m	n	k	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
500	2000	30	0	0.083	64	$5.3 \cdot 10^{-4}$	10.18	32	$1.9 \cdot 10^{-4}$	6.47	–	$1.6 \cdot 10^{-4}$	**2.46**
500	2000	30	10^{-3}	0.083	64	$1.1 \cdot 10^{-3}$	6.69	32	$1 \cdot 10^{-3}$	4.51	–	$6 \cdot 10^{-4}$	**1.94**
500	2000	30	10^{-4}	0.083	64	$5.4 \cdot 10^{-4}$	10.14	32	$2.2 \cdot 10^{-4}$	6.51	–	$1.6 \cdot 10^{-4}$	**2.46**
500	2000	50	0	0.408	103	$1.1 \cdot 10^{-4}$	15.74	54	$5 \cdot 10^{-4}$	10.8	–	$8 \cdot 10^{-5}$	7.32
500	2000	50	10^{-3}	0.408	103	$1.8 \cdot 10^{-3}$	24.97	54	$1.55 \cdot 10^{-3}$	16.14	–	$9 \cdot 10^{-4}$	8.6
500	2000	50	10^{-4}	0.408	102	$1.1 \cdot 10^{-3}$	24.85	54	$5 \cdot 10^{-4}$	16.17	–	$7 \cdot 10^{-5}$	8.59
500	2000	80	0	0.645	239	$3.5 \cdot 10^{-3}$	92.91	134	$1.7 \cdot 10^{-3}$	59.33	–	$1 \cdot 10^{-4}$	79.64
500	2000	80	10^{-3}	0.645	239	$4.2 \cdot 10^{-3}$	94.86	134	$2.8 \cdot 10^{-3}$	60.68	–	$1 \cdot 10^{-4}$	79.98
500	2000	80	10^{-4}	0.645	239	$3.6 \cdot 10^{-3}$	93.95	134	$1.8 \cdot 10^{-3}$	60.76	–	$1 \cdot 10^{-4}$	79.48
500	2000	100	0	0.8	523	$1.1 \cdot 10^{-2}$	259.13	307	$6 \cdot 10^{-3}$	173.14	–	$4.5 \cdot 10^{-2}$	143.41
500	2000	100	10^{-3}	0.8	525	$1.2 \cdot 10^{-2}$	262.19	308	$7 \cdot 10^{-3}$	176.04	–	$5.2 \cdot 10^{-2}$	142.85
500	2000	100	10^{-4}	0.8	523	$1.1 \cdot 10^{-2}$	262.11	307	$6 \cdot 10^{-3}$	170.47	–	$5.1 \cdot 10^{-2}$	144.78
					LMaFit			MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	k	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
500	2000	30	0	0.083	37	$1.3 \cdot 10^{-5}$	3.05	13	$3.1 \cdot 10^{-5}$	4.84	37	$1.2 \cdot 10^{-5}$	4.04
500	2000	30	10^{-3}	0.083	37	$1 \cdot 10^{-3}$	2.52	22	$1.1 \cdot 10^{-3}$	5.35	37	$1 \cdot 10^{-3}$	3.32
500	2000	30	10^{-4}	0.083	35	$1 \cdot 10^{-4}$	2.86	13	$1.3 \cdot 10^{-4}$	4.85	37	$1.6 \cdot 10^{-4}$	4.05
500	2000	50	0	0.408	60	$6 \cdot 10^{-5}$	6.06	22	$1 \cdot 10^{-4}$	7.6	60	$2 \cdot 10^{-4}$	**5.67**
500	2000	50	10^{-3}	0.408	60	$1.4 \cdot 10^{-3}$	7.26	36	$1.6 \cdot 10^{-3}$	19.64	59	$1.6 \cdot 10^{-3}$	**6.91**
500	2000	50	10^{-4}	0.408	60	$2 \cdot 10^{-4}$	7.29	22	$2 \cdot 10^{-4}$	11.87	59	$2 \cdot 10^{-4}$	**6.75**
500	2000	80	0	0.645	183	$3 \cdot 10^{-4}$	33.65	61	$2 \cdot 10^{-4}$	49.53	151	$3 \cdot 10^{-4}$	**18.66**
500	2000	80	10^{-3}	0.645	183	$2.3 \cdot 10^{-3}$	33.48	92	$2.4 \cdot 10^{-3}$	75.51	151	$2.3 \cdot 10^{-3}$	**18.87**
500	2000	80	10^{-4}	0.645	183	$3 \cdot 10^{-4}$	33.47	61	$4 \cdot 10^{-4}$	49.52	151	$3 \cdot 10^{-4}$	**18.92**
500	2000	100	0	0.8	519	$1.5 \cdot 10^{-3}$	115.11	148	$4 \cdot 10^{-4}$	153.74	429	$7 \cdot 10^{-4}$	**55.1**
500	2000	100	10^{-3}	0.8	529	$3.6 \cdot 10^{-3}$	117.7	228	$3.7 \cdot 10^{-3}$	239.92	427	$3.4 \cdot 10^{-3}$	**55.7**
500	2000	100	10^{-3}	0.8	520	$1.6 \cdot 10^{-3}$	116.66	148	$6 \cdot 10^{-4}$	154.46	428	$8 \cdot 10^{-4}$	**55.07**

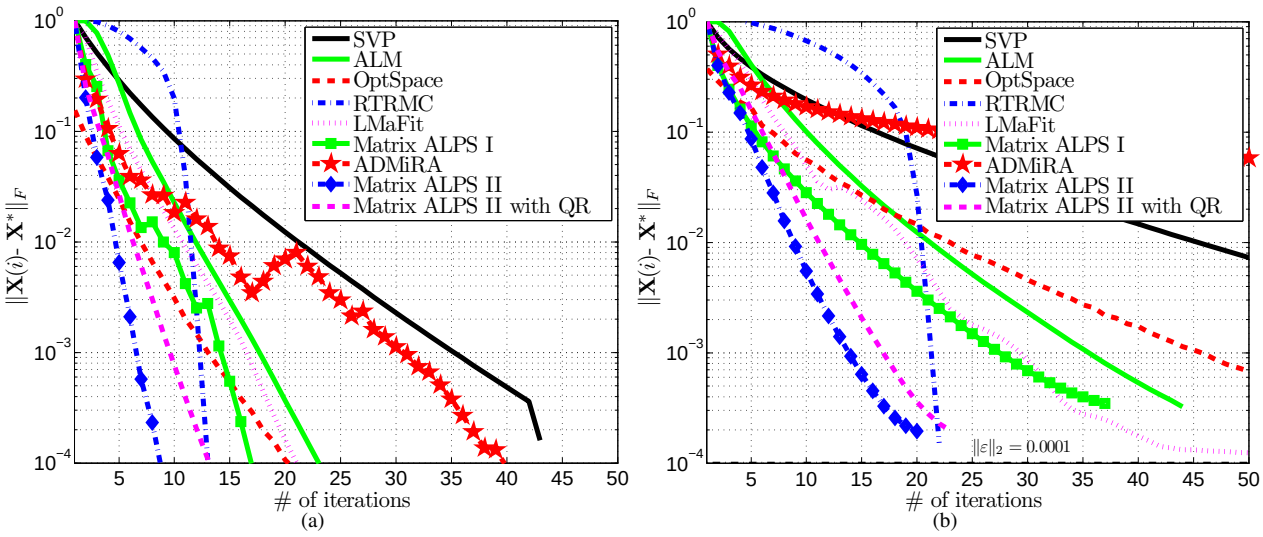

Fig. 7. *Low rank matrix recovery for the matrix completion problem. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $p = 0.3mn$. *(a)* $m = 300$, $n = 600$, $k = 5$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(b)* $m = 300$, $n = 600$, $k = 20$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-4}$.

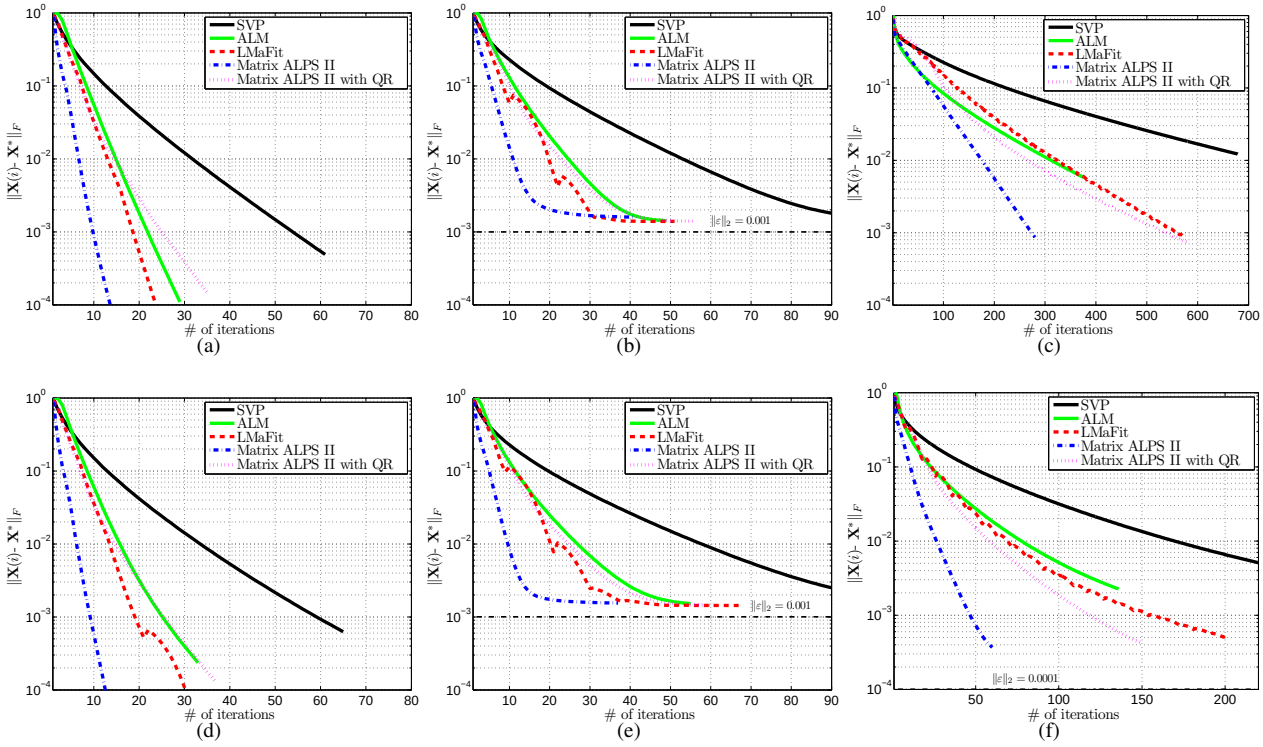

Fig. 8. *Low rank matrix recovery for the matrix completion problem. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $p = 0.3mn$. *(a)* $m = 700$, $n = 1000$, $k = 30$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(b)* $m = 700$, $n = 1000$, $k = 50$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-3}$. *(c)* $m = 700$, $n = 1000$, $k = 110$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(d)* $m = 500$, $n = 2000$, $k = 10$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(e)* $m = 500$, $n = 2000$, $k = 50$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-3}$. *(f)* $m = 500$, $n = 2000$, $k = 80$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-4}$.

In theory, constant μ_i selection schemes are accompanied with strong RIP constant conditions but empirical evidence reveal

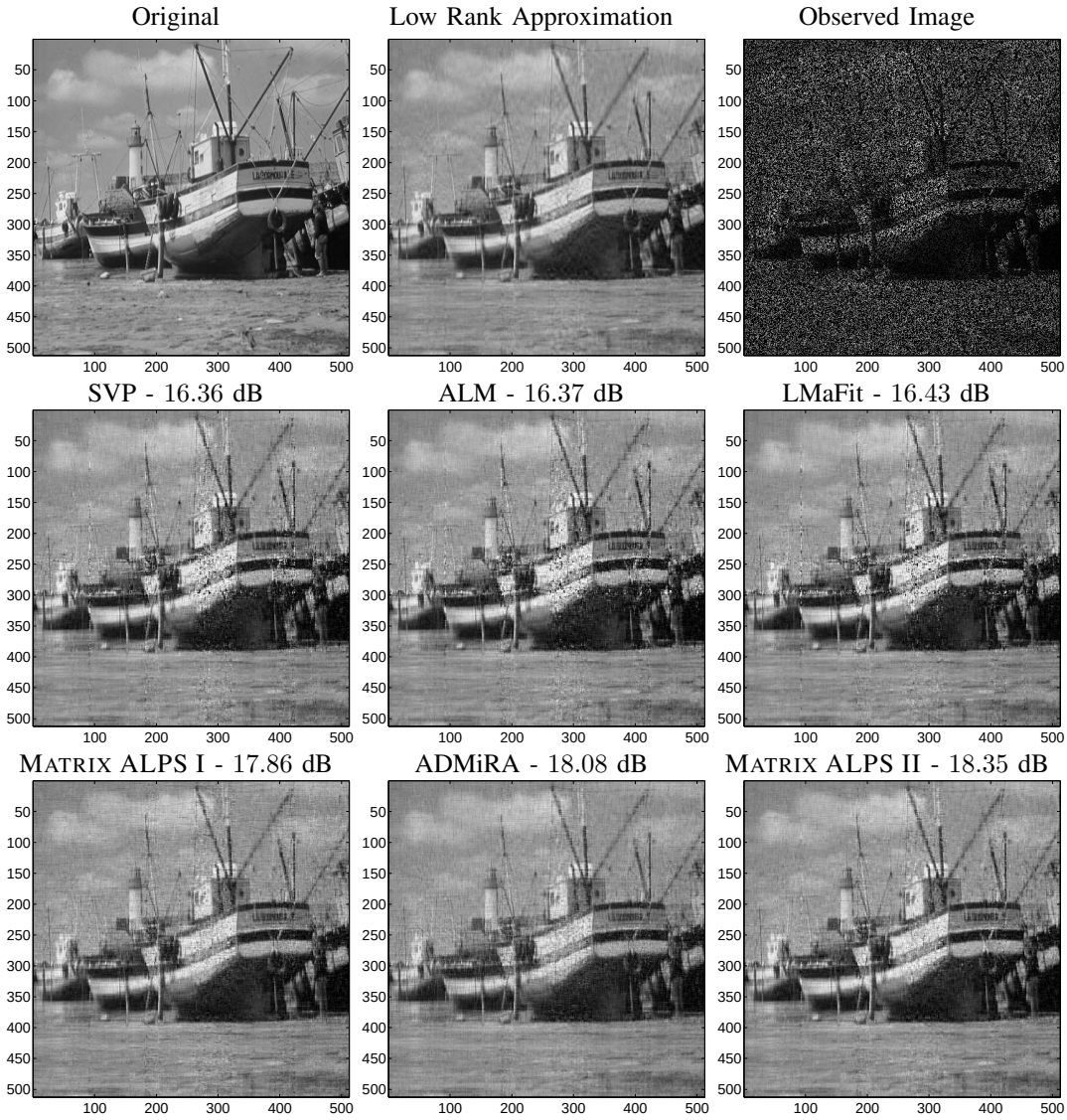

Fig. 9. Reconstruction performance in image denoising settings. The image size is 512×512 and the desired rank is preset to $k = 40$. We observe 35% of the pixels of the true image. We depict the median reconstruction error with respect to the true image in dB over 10 Monte Carlo realizations.

signal reconstruction vulnerabilities for deviations from the initial problem assumptions. While convergence derivations of adaptive schemes are characterized by weaker bounds, the performance gained by this choice in terms of convergence rate, is quite significant. Memory-based methods lead to convergence speed with (almost) no extra cost on the complexity of hard thresholding methods—theoretical eveidence prove the efficiency of memory utilization in signal recovery but more theoretical justification is neeed as future work. Lastly, further estimate refinement over sparse support sets using gradient update steps or pseudoinversion optimization techniques provides signal reconstruction efficacy, but more computational power is needed per iteration.

Affine rank minimization on real data deals with very large matrices which, in many cases, is impossible to load into the Random Access Memory (RAM) of a computer; therefore, even first-order gradient descent procedures are prohibitively expensive and require huge processing power and memory storage restricting the application of these algorithms only on small-sized matrices. Recent developments on geometric functional analysis have shown encouraging results dictating that sampling from large matrices can approximate efficiently large data sets with small error in terms of the Frobenius norm. In this work, we connect ϵ-approximation low-rank revealing schemes with first-order gradient descent algorithms to solve general affine rank minimization problems—to the best of our knowledge, this is the first attempt to theoretically characterize the performance of iterative greedy algorithms with ϵ-approximation schemes.

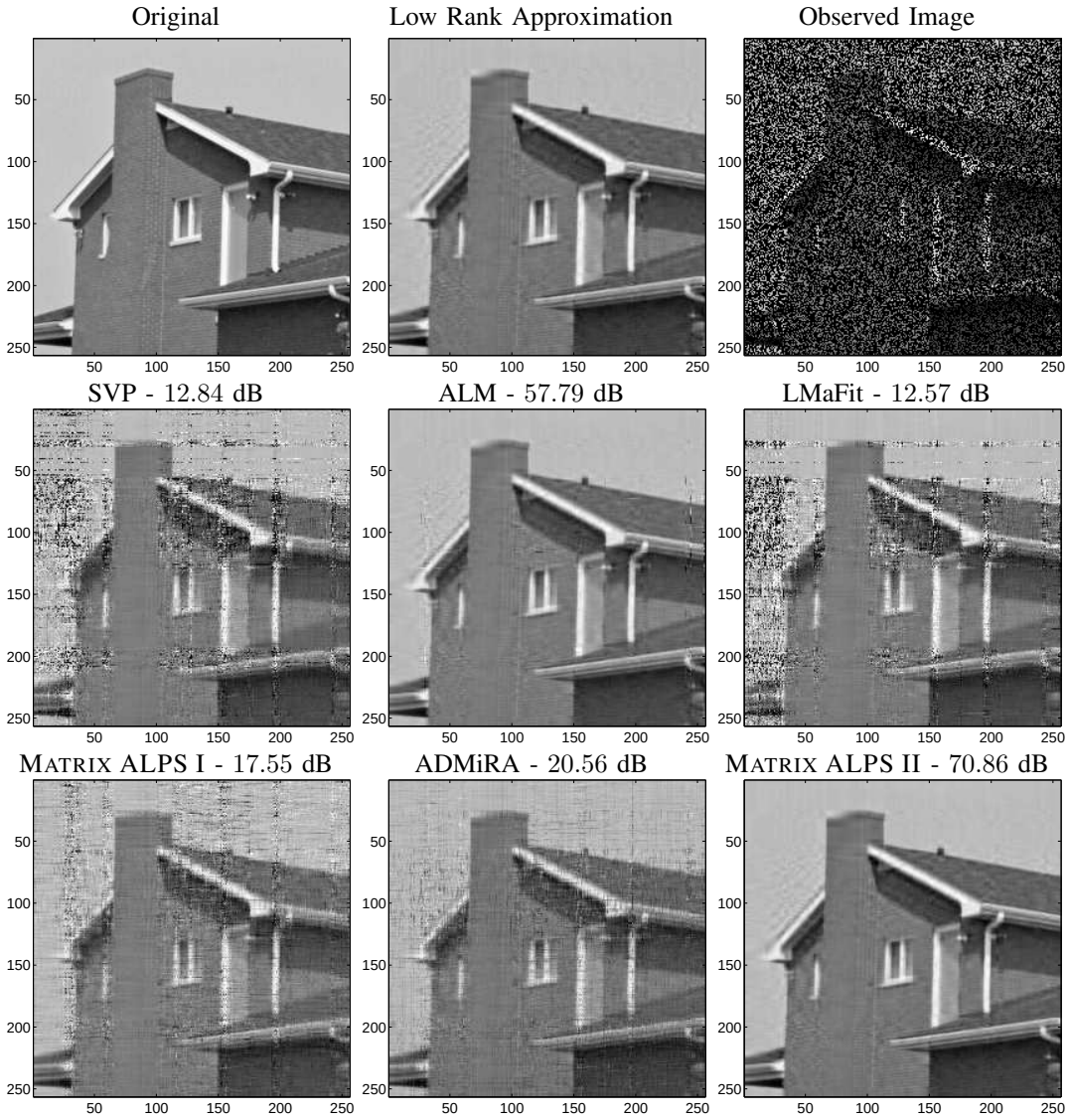

Fig. 10. Reconstruction performance in image denoising settings. The image size is 256×256 and the desired rank is preset to $k = 30$. We observe 33% of the pixels of the best rank-30 approximation of the image. We depict the median reconstruction with respect to the best rank-30 approximation in dB over 10 Monte Carlo realizations

In all cases, experimental results illustrate the effectiveness of the proposed schemes on different problem configurations.

XII. Acknowledgements

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, ARO MURI W911NF0910383, and DARPA KeCoM program #11-DARPA-1055. VC also would like to acknowledge Rice University for his Faculty Fellowship.

Appendix

A. Proof of Lemma 6

Given $\mathcal{X}^* := \text{ortho}(\boldsymbol{X}^*)$, we define the following quantities: $\mathcal{S}_i := \mathcal{X}_i \cup \mathcal{D}_i,\ \mathcal{S}_i^* := \mathcal{X}_i \cup \mathcal{X}^*$. Then:

$$\mathcal{P}_{\mathcal{S}_i \setminus \mathcal{S}_i^*} = \mathcal{P}_{\mathcal{D}_i \setminus (\mathcal{X}^* \cup \mathcal{X}_i)}, \quad \text{and} \quad \mathcal{P}_{\mathcal{S}_i^* \setminus \mathcal{S}_i} = \mathcal{P}_{\mathcal{X}^* \setminus (\mathcal{D}_i \cup \mathcal{X}_i)}. \tag{57}$$

Since the subspace defined in $\mathcal{D}_i$ is the best rank-k subspace, orthogonal to the subspace spanned by $\mathcal{X}_i$, the following holds true:

$$\left\|\mathcal{P}_{\mathcal{D}_i\setminus\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \Rightarrow \tag{58}$$

$$\left\|\mathcal{P}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 + \left\|\mathcal{P}_{\mathcal{D}_i\setminus\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 + \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \Rightarrow \tag{59}$$

$$\left\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{S}_i^*}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \tag{60}$$

Removing the common subspaces in $\mathcal{S}_i$ and $\mathcal{S}_i^*$, we get

$$\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\right\|_F^2 \Rightarrow \tag{61}$$

$$\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{62}$$

On the left hand side, we have:

$$\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{63}$$

$$\overset{(i)}{\leq} \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{64}$$

$$\overset{(ii)}{=} \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{65}$$

$$\overset{(iii)}{=} \left\|(I - \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*})(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{66}$$

$$\leq \left\|(I - \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*})(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{67}$$

$$\overset{(iv)}{\leq} \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{X}^* - \boldsymbol{X}(i)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{68}$$

$$\overset{(v)}{\leq} \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{X}^* - \boldsymbol{X}(i)\right\|_F + \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{69}$$

$$\overset{(vi)}{\leq} 2\delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{X}^* - \boldsymbol{X}(i)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{70}$$

where (i) due to triangle inequality over Frobenius metric norm, (ii) since $\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}(i) - \boldsymbol{X}^*) = 0$, (iii) by using the fact that $\boldsymbol{X}(i) - \boldsymbol{X}^* := \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}(i) - \boldsymbol{X}^*) + \mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}(i) - \boldsymbol{X}^*)$, (iv) due to Lemma 3, (v) due to Lemma 4 and (vi) since $\left\|\mathcal{P}^{\perp}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F \leq \left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F$.

For the right hand side of (62), we calculate:

$$\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{71}$$

$$= \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i)) - \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F$$

$$= \left\|(\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i} - I)(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right.$$

$$\left. + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{72}$$

$$\geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F - \left\|(\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i} - I)(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F$$

$$- \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F - \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{73}$$

$$\geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{X}(i))\right\|_F - 2\delta_{2k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F - \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \tag{74}$$

by using Lemmas 3 and 4. Combining (70) and (74) in (62), we get:

$$\left\|\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\boldsymbol{X}^*\right\|_F \leq (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \left\|\mathcal{P}_{(\mathcal{S}_i^*\setminus\mathcal{S}_i)\cup(\mathcal{S}_i\setminus\mathcal{S}_i^*)}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \Rightarrow \tag{75}$$

$$\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\right\|_F \leq (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{76}$$

B. Proof of Theorem 1

Let $\mathcal{X}^* \leftarrow \text{ortho}(\boldsymbol{X}^*)$ be a set of orthonormal, rank-1 matrices that span the range of $\boldsymbol{X}^*$. In Algorithm 1, $\boldsymbol{W}(i)$ is the best rank-k approximation of $\boldsymbol{V}(i)$. Thus:

$$\left\|\boldsymbol{W}(i) - \boldsymbol{V}(i)\right\|_F^2 \leq \left\|\boldsymbol{X}^* - \boldsymbol{V}(i)\right\|_F^2 \Rightarrow \tag{77}$$

$$\left\|\boldsymbol{W}(i) - \boldsymbol{X}^* + \boldsymbol{X}^* - \boldsymbol{V}(i)\right\|_F^2 \leq \left\|\boldsymbol{X}^* - \boldsymbol{V}(i)\right\|_F^2 \Rightarrow \tag{78}$$

$$\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F^2 + \left\|\boldsymbol{V}(i) - \boldsymbol{X}^*\right\|_F^2 + 2\langle\boldsymbol{W}(i) - \boldsymbol{X}^*, \boldsymbol{X}^* - \boldsymbol{V}(i)\rangle \leq \left\|\boldsymbol{X}^* - \boldsymbol{V}(i)\right\|_F^2 \Rightarrow \tag{79}$$

$$\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F^2 \leq 2\langle\boldsymbol{W}(i) - \boldsymbol{X}^*, \boldsymbol{V}(i) - \boldsymbol{X}^*\rangle \tag{80}$$

From Algorithm 1, it is obvious that $i)$ $\boldsymbol{V}(i) \in \text{span}(\mathcal{S}_i)$, $ii)$ $\boldsymbol{X}(i) \in \text{span}(\mathcal{S}_i)$ and $iii)$ $\boldsymbol{W}(i) \in \text{span}(\mathcal{S}_i)$. We define $\mathcal{E} := \mathcal{S}_i \cup \mathcal{X}^*$ where $\text{rank}(\text{span}(\mathcal{E})) \leq 3k$ and let $\mathcal{P}_{\mathcal{E}}$ be the orthogonal projection onto the subspace defined by $\mathcal{E}$.

Since $\boldsymbol{W}(i) - \boldsymbol{X}^* \in \text{span}(\mathcal{E})$ and $\boldsymbol{V}(i) - \boldsymbol{X}^* \in \text{span}(\mathcal{E})$, the following hold true:

$$\boldsymbol{W}(i) - \boldsymbol{X}^* = \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*) \;\text{ and }\; \boldsymbol{V}(i) - \boldsymbol{X}^* = \mathcal{P}_{\mathcal{E}}(\boldsymbol{V}(i) - \boldsymbol{X}^*) \tag{81}$$

due to Remark 3.

Then, (80) can be written as:

$$\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F^2 \leq 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\rangle \Rightarrow \tag{82}$$

$$\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F^2 \leq 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}\left(\boldsymbol{X}(i) + \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^* - \boldsymbol{X}(i)) + \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon} - \boldsymbol{X}^*\right)\rangle \Rightarrow \tag{83}$$

$$\begin{aligned}\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F^2 \leq\; & \underbrace{2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^* - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}(i) - \boldsymbol{X}^*))\rangle}_{\doteq A} \\ & + \underbrace{2\mu_i\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle}_{\doteq B}\end{aligned} \tag{84}$$

In B, we observe:

$$B := 2\mu_i\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle \overset{(i)}{=} 2\mu_i\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle \tag{85}$$

$$\overset{(ii)}{\leq} 2\mu_i \left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F \left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\right\|_F \tag{86}$$

$$\overset{(iii)}{\leq} 2\mu_i\sqrt{1 + \delta_{2k}(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F \left\|\boldsymbol{\varepsilon}\right\|_2 \tag{87}$$

where (i) holds due to Remark 3 and since $\mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{\mathcal{E}} = \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i} = \mathcal{P}_{\mathcal{S}_i}$ for $\mathcal{S}_i \subseteq \mathcal{E}$, (ii) is due to Cauchy-Schwarz inequality and, (iii) is easily derived using Lemma 1.

In A, we perform the following motions:

$$A := 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{W}(i) - \boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^* - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}(i) - \boldsymbol{X}^*))\rangle \tag{88}$$

$$= 2\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) - \mu_i \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\rangle \tag{89}$$

$$= 2\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\rangle \tag{90}$$

$$\overset{(i)}{=} 2\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\left[\mathcal{P}_{\mathcal{S}_i} + \mathcal{P}_{\mathcal{S}_i}^{\perp}\right]\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\rangle \tag{91}$$

$$= 2\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, (I - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\rangle - 2\mu_i\langle \boldsymbol{W}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\rangle \tag{92}$$

$$\overset{(ii)}{\leq} 2\left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F \left\|(I - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F + 2\mu_i \left\|\boldsymbol{W}(i) - \boldsymbol{X}^*\right\|_F \left\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F \tag{93}$$

where (i) is due to $\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) := \mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) + \mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)$ and (ii) follows from Cauchy-Schwarz inequality. Since $\frac{1}{1+\delta_{2k}(\boldsymbol{\mathcal{A}})} \leq \mu_i \leq \frac{1}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}$, Lemma 3 implies:

$$\lambda(I - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}) \in \left[1 - \frac{1 - \delta_{2k}(\boldsymbol{\mathcal{A}})}{1 + \delta_{2k}(\boldsymbol{\mathcal{A}})}, \frac{1 + \delta_{2k}(\boldsymbol{\mathcal{A}})}{1 - \delta_{2k}(\boldsymbol{\mathcal{A}})} - 1\right] \leq \frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1 - \delta_{2k}(\boldsymbol{\mathcal{A}})}. \tag{94}$$

and thus:

$$\left\|(I - \mu_i \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F \leq \frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1 - \delta_{2k}(\boldsymbol{\mathcal{A}})}\left\|\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F. \tag{95}$$

Furthermore, according to Lemma 4:

$$\left\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F \leq \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F \tag{96}$$

since $\text{rank}(\mathcal{P}_{\mathcal{E}\cup\mathcal{S}_i}\boldsymbol{X}) \leq 3k,\ \forall \boldsymbol{X} \in \mathcal{R}^{m\times n}$. Since $\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*) = \mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\boldsymbol{X}^*$ where $\mathcal{D}_i := \mathcal{P}_k\left(\mathcal{P}_{\mathcal{X}_i}^{\perp}\nabla f(\boldsymbol{X}(i))\right)$, then:

$$\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i) - \boldsymbol{X}^*)\right\|_F = \left\|\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\boldsymbol{X}^*\right\|_F \leq (2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}}))\left\|\boldsymbol{X}(i) - \boldsymbol{X}^*\right\|_F + \sqrt{2(1 + \delta_{2k}(\boldsymbol{\mathcal{A}}))}\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{97}$$

using Lemma 6. Combining the above in (93), we compute:

$$A \leq 2\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F\big\|(I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F + 2\mu_i\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F\big\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F \tag{98}$$

$$\leq \frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F + \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F \tag{99}$$

$$\leq \Big(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F$$
$$+ \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\big\|\boldsymbol{\varepsilon}\big\|_2 \tag{100}$$

Combining (87) and (100) in (84), we get:

$$\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F \leq \Big(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F$$
$$+ \Big(\frac{2\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\Big)\big\|\boldsymbol{\varepsilon}\big\|_2 \tag{101}$$

Focusing on steps 5 and 6 of Algorithm 1, we perform the following motions:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F = \big\|\boldsymbol{W}(i)+\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^*-\boldsymbol{W}(i))+\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}-\boldsymbol{X}^*\big\|_F \tag{102}$$
$$= \big\|\big(I-\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\big)\big(\boldsymbol{W}(i)-\boldsymbol{X}^*\big)+\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F$$
$$= \big\|\big(I-\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\big)\big(\boldsymbol{W}(i)-\boldsymbol{X}^*\big)+\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{X}^*-\boldsymbol{W}(i))+\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{X}^*-\boldsymbol{W}(i))+\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F$$
$$\leq \big\|\big(I-\xi_i\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\big)\big(\boldsymbol{W}(i)-\boldsymbol{X}^*\big)\big\|_F+\xi_i\big\|\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{X}^*-\boldsymbol{W}(i))\big\|_F$$
$$+\|\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\xi_i\big\|\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F$$
$$\overset{(i)}{\leq} \frac{2\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\|\mathcal{P}_{\mathcal{W}_i}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\delta_{2k}(\boldsymbol{\mathcal{A}})\xi_i\|\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\|\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\xi_i\big\|\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F$$
$$= \frac{2\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\|\mathcal{P}_{\mathcal{W}_i}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\Big(1+\frac{\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\Big)\|\mathcal{P}_{\mathcal{W}_i}^{\perp}(\boldsymbol{W}(i)-\boldsymbol{X}^*)\|_F+\xi_i\big\|\mathcal{P}_{\mathcal{W}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F$$
$$\overset{(ii)}{\leq} \Big(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{W}(i)-\boldsymbol{X}^*\big\|_F+\frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2 \tag{103}$$

where (i) is due to Lemmas 3 and 4 and (ii) is due to Remark 6 and $\delta_{\alpha k}(\boldsymbol{\mathcal{A}}) \leq \delta_{\beta k}(\boldsymbol{\mathcal{A}})$ for $\alpha<\beta,\ \alpha,\beta\in\mathcal{Z}_+$. Combining the recursions in (101) and (103), we finally compute:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq \Big(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F$$
$$+\left(\Big(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(\frac{2\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}+\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}\Big)+\frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}\right)\big\|\boldsymbol{\varepsilon}\big\|_2. \tag{104}$$

For the convergence parameter ρ, further compute:

$$\Big(\frac{1+2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(\frac{4\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})} + (2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big) \leq \frac{1+2\delta_{3k}(\boldsymbol{\mathcal{A}})}{(1-\delta_{3k}(\boldsymbol{\mathcal{A}}))^2}\big(4\delta_{3k}(\boldsymbol{\mathcal{A}})+8\delta_{3k}^2(\boldsymbol{\mathcal{A}})\big) =: \rho. \tag{105}$$

for $\delta_k(\boldsymbol{\mathcal{A}}) \leq \delta_{2k}(\boldsymbol{\mathcal{A}}) \leq \delta_{3k}(\boldsymbol{\mathcal{A}})$. Calculating the roots of this expression, we easily observe that $\rho<1$ for $\delta_{3k}(\boldsymbol{\mathcal{A}})<0.1235$.

C. Proof of Theorem 2

Before we present the proof of Theorem 2, we list a series of lemmas that correspond to the motions Algorithm 2 performs—detailed proofs of the following results can be found in the Appendix.

Lemma 10. *[Error norm reduction via least-squares optimization] Let $\mathcal{S}_i$ be a set of orthonormal, rank-1 matrices that span a rank-2k subspace in $\mathcal{R}^{m\times n}$. Then, the least squares solution $\boldsymbol{V}(i)$ given by:*

$$\boldsymbol{V}(i) \leftarrow \underset{\boldsymbol{V}:\boldsymbol{V}\in span(\mathcal{S}_i)}{\arg\min}\ \big\|\boldsymbol{y}-\boldsymbol{\mathcal{A}}\boldsymbol{V}\big\|_2^2, \tag{106}$$

satisfies:

$$\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F \leq \frac{1}{\sqrt{1-\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F + \frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{107}$$

Proof: We observe that $\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F^2$ is decomposed as follows:

$$\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F^2 = \left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2 + \left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2. \tag{108}$$

In (106), $\boldsymbol{V}(i)$ is the minimizer over the low-rank subspace spanned by $\mathcal{S}_i$ with $\text{rank}(\text{span}(\mathcal{S}_i)) \leq 2k$. Using the optimality condition (Lemma 5) over the convex set $\Theta = \{\boldsymbol{X} : \text{span}(\boldsymbol{X}) \in \mathcal{S}_i\}$, we have:

$$\langle \nabla f(\boldsymbol{V}(i)), \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{V}(i))\rangle \geq 0 \Rightarrow \langle \boldsymbol{\mathcal{A}}\boldsymbol{V}(i) - \boldsymbol{y}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle \leq 0. \tag{109}$$

for $\mathcal{P}_{\mathcal{S}_i}\boldsymbol{X}^* \in \text{span}(\mathcal{S}_i)$. Given condition (109), the first term on the right hand side of (108) becomes:

$$\begin{aligned}
\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2 &= \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (110)\\
&\leq \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle - \langle \boldsymbol{\mathcal{A}}\boldsymbol{V}(i)-\boldsymbol{y}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (111)\\
&= \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle - \langle \boldsymbol{\mathcal{A}}\boldsymbol{V}(i)-\boldsymbol{\mathcal{A}}\boldsymbol{X}^*-\boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (112)\\
&= \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle - \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (113)\\
&= \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (114)\\
&\leq |\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (115)
\end{aligned}$$

Focusing on the term $|\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle|$, we derive the following:

$$\begin{aligned}
&|\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (116)\\
&= |\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle - \langle \boldsymbol{V}(i)-\boldsymbol{X}^*, \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (117)\\
&\overset{(i)}{=} |\langle \mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*), \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle - \langle \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*), \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (118)\\
&\overset{(ii)}{=} |\langle \mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*), \mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && \\
&- \langle \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*), \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (119)\\
&= |\langle \mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*), (I-\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (120)\\
&= |\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| && (121)
\end{aligned}$$

where (i) follows from the facts that $\boldsymbol{V}(i)-\boldsymbol{X}^* \in \text{span}(\mathcal{S}_i \cup \mathcal{X}^*)$ and thus $\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}(\boldsymbol{V}(i)-\boldsymbol{X}^*) = \boldsymbol{V}(i)-\boldsymbol{X}^*$ and (ii) is due to $\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i} = \mathcal{P}_{\mathcal{S}_i}$ since $\text{span}(\mathcal{S}_i) \subseteq \text{span}(\mathcal{S}_i \cup \mathcal{X}^*)$. Then, (115) becomes:

$$\begin{aligned}
&\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2 && (122)\\
&\leq |\langle \boldsymbol{V}(i)-\boldsymbol{X}^*, (I-\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle| + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle && (123)\\
&\overset{(i)}{\leq} \left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F \left\|(I-\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\cup\mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F && (124)\\
&\overset{(ii)}{\leq} \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F \left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F + \sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F \left\|\boldsymbol{\varepsilon}\right\|_2, && (125)
\end{aligned}$$

where (i) comes from Cauchy-Swartz inequality and (ii) is due to Lemmas 1 and 3. Simplifying the above quadratic expression, we obtain:

$$\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F \leq \delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F + \sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{126}$$

As a consequence, (108) can be upper bounded by:

$$\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F^2 \leq \left(\delta_{3k}(\boldsymbol{\mathcal{A}})\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F + \sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{\varepsilon}\right\|_2\right)^2 + \left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2. \tag{127}$$

We form the quadratic polynomial for this inequality assuming as unknown variable the quantity $\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F$. Bounding by the largest root of the resulting polynomial, we get:

$$\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F \leq \frac{1}{\sqrt{1-\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F + \frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{128}$$

■

The following Lemma characterizes how subspace *pruning* affects the recovered energy:

Lemma 11. *[Best rank-k subspace selection] Let $\boldsymbol{V}(i) \in \mathcal{R}^{m\times n}$ be a rank-2k proxy matrix in the subspace spanned by $\mathcal{S}_i$ and let $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$ denote the best rank-k approximation to $\boldsymbol{V}(i)$, according to (13). Then:*

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{V}(i)\big\|_F \leq \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F \leq \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F. \tag{129}$$

Proof: Since $\boldsymbol{X}(i+1)$ denotes the best rank-k approximation to $\boldsymbol{V}(i)$, the following inequality holds for any rank-k matrix $\boldsymbol{X} \in \mathcal{R}^{m\times n}$ in the subspace spanned by $\mathcal{S}_i$, i.e. $\forall \boldsymbol{X} \in \text{span}(\mathcal{S}_i)$:

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{V}(i)\big\|_F \leq \big\|\boldsymbol{X} - \boldsymbol{V}(i)\big\|_F. \tag{130}$$

Since $\mathcal{P}_{\mathcal{S}_i}\boldsymbol{V}(i) = \boldsymbol{V}(i)$, the left inequality in (129) is satisfied for $\boldsymbol{X} := \mathcal{P}_{\mathcal{S}_i}\boldsymbol{X}^*$ in (130). The second inequality in (129) holds according to Remark 6. ∎

Lemma 12. *Let $\boldsymbol{V}(i)$ be the least squares solution in Step 2 of the ADMiRA algorithm and let $\boldsymbol{X}(i+1)$ be a proxy, rank-k matrix to $\boldsymbol{V}(i)$ according to: $\boldsymbol{X}(i+1) \leftarrow \mathcal{P}_k(\boldsymbol{V}(i))$. Then, $\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F$ can be expressed in terms of the distance from $\boldsymbol{V}(i)$ to $\boldsymbol{X}^*$ as follows:*

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F \leq \sqrt{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F + \sqrt{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}\sqrt{\frac{3(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\|\boldsymbol{\varepsilon}\|_2. \tag{131}$$

Proof: We observe the following

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F^2 &= \big\|\boldsymbol{X}(i+1) - \boldsymbol{V}(i) + \boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 && (132)\\
&= \big\|(\boldsymbol{V}(i) - \boldsymbol{X}^*) - (\boldsymbol{V}(i) - \boldsymbol{X}(i+1))\big\|_F^2 && (133)\\
&= \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 + \big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F^2 - 2\langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \boldsymbol{V}(i) - \boldsymbol{X}(i+1)\rangle. && (134)
\end{aligned}$$

Focusing on the right hand side of expression (134), $\langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \boldsymbol{V}(i) - \boldsymbol{X}(i+1)\rangle = \langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}(i+1))\rangle$ can be similarly analysed as (117)-(121) where we obtain the following expression:

$$\begin{aligned}
|\langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}(i+1))\rangle| \leq{}& \delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F\big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F\\
&+ \sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F\|\boldsymbol{\varepsilon}\|_2. && (135)
\end{aligned}$$

Now, expression (134) can be further transformed as:

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F^2 &= \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 + \big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F^2 - 2\langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \boldsymbol{V}(i) - \boldsymbol{X}(i+1)\rangle && (136)\\
&\leq \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 + \big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F^2 + 2|\langle \boldsymbol{V}(i) - \boldsymbol{X}^*, \boldsymbol{V}(i) - \boldsymbol{X}(i+1)\rangle| && (137)\\
&\overset{(i)}{\leq} \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 + \big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F^2 + 2(\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F\big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F && (138)\\
&\quad + \sqrt{1+\delta_{2k}}\big\|\boldsymbol{V}(i) - \boldsymbol{X}(i+1)\big\|_F\|\boldsymbol{\varepsilon}\|_2) && (139)
\end{aligned}$$

where (i) is due to (135). Using Lemma 11, we further have:

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F^2 \leq{}& \big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F^2 + \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F^2\\
&+ 2\Big(\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_F\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F + \sqrt{1+\delta_{2k}}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F\|\boldsymbol{\varepsilon}\|_2\Big) && (140)
\end{aligned}$$

Furthermore, replacing $\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}^* - \boldsymbol{V}(i))\big\|_F$ with its upper bound defined in (126), we get:

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_2^2 &= \big(1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})\big)\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_2^2 + 6\delta_{3k}(\boldsymbol{\mathcal{A}})\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_2\|\boldsymbol{\varepsilon}\|_2 + 3(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))\|\boldsymbol{\varepsilon}\|_2^2\\
&\overset{(i)}{\leq} \Big(1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})\Big)\left(\big\|\boldsymbol{V}(i) - \boldsymbol{X}^*\big\|_2 + \sqrt{\frac{3(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\|\boldsymbol{\varepsilon}\|\right)^2 && (141)
\end{aligned}$$

where (i) is obtained by completing the squares and eliminating negative terms. ∎

Applying basic algebra tools in (131) and (107), we get:

$$\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F \leq \sqrt{\frac{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\big\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i) - \boldsymbol{X}^*)\big\|_F + \Big(\frac{\sqrt{1+3\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + \sqrt{3}\Big)\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\varepsilon}\|_2. \tag{142}$$

Since $\boldsymbol{V}(i) \in \text{span}(\mathcal{S}_i)$, we observe $\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{V}(i) - \boldsymbol{X}^*) = -\mathcal{P}^{\perp}_{\mathcal{S}_i}\boldsymbol{X}^* = -\mathcal{P}_{\mathcal{X}^* \setminus (\mathcal{D}_i \cup \mathcal{X}_i)}\boldsymbol{X}^*$. Then, using Lemma 6, we obtain:

$$\begin{aligned}\big\|\boldsymbol{X}(i+1) - \boldsymbol{X}^*\big\|_F &\leq \sqrt{\frac{1+3\delta^2_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta^2_{3k}(\boldsymbol{\mathcal{A}})}}\Big[\big(2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}})\big)\big\|\boldsymbol{X}^* - \boldsymbol{X}(i)\big\|_F + \sqrt{2(1+\delta_{3k})}\big\|\boldsymbol{\varepsilon}\big\|_2\Big] \\ &\quad + \Big(\frac{\sqrt{1+3\delta^2_{3k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + \sqrt{3}\Big)\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2 \qquad (143) \\ &= \big(2\delta_{2k}(\boldsymbol{\mathcal{A}}) + 2\delta_{3k}(\boldsymbol{\mathcal{A}})\big)\sqrt{\frac{1+3\delta^2_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta^2_{3k}(\boldsymbol{\mathcal{A}})}}\big\|\boldsymbol{X}^* - \boldsymbol{X}(i)\big\|_F \\ &\quad + \left[\sqrt{\frac{1+3\delta^2_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta^2_{3k}(\boldsymbol{\mathcal{A}})}}\sqrt{2(1+\delta_{3k})} + \Big(\frac{\sqrt{1+3\delta^2_{3k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + \sqrt{3}\Big)\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\right]\big\|\boldsymbol{\varepsilon}\big\|_2 \qquad (144)\end{aligned}$$

Given $\delta_{2k}(\boldsymbol{\mathcal{A}}) \leq \delta_{3k}(\boldsymbol{\mathcal{A}})$, ρ is upper bounded by:

$$\rho < 4\delta_{3k}(\boldsymbol{\mathcal{A}})\sqrt{\frac{1+3\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta^2_{3k}(\boldsymbol{\mathcal{A}})}} \qquad (145)$$

Then,

$$4\delta_{3k}(\boldsymbol{\mathcal{A}})\sqrt{\frac{1+3\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta^2_{3k}(\boldsymbol{\mathcal{A}})}} < 1 \Leftrightarrow \delta_{3k}(\boldsymbol{\mathcal{A}}) < 0.2267. \qquad (146)$$

D. Proof of Lemma 7

Let $\mathcal{D}^{\epsilon}_i := \text{ortho}\Big(\mathcal{P}^{\epsilon}_k(\mathcal{P}^{\perp}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i)))\Big)$ and $\mathcal{D}_i := \text{ortho}\Big(\mathcal{P}_k(\mathcal{P}^{\perp}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i)))\Big)$. Using Definition 4, the following holds true:

$$\big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i)) - \nabla f(\boldsymbol{X}(i))\big\|^2_F \leq (1+\epsilon)\big\|\mathcal{P}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i)) - \nabla f(\boldsymbol{X}(i))\big\|^2_F \qquad (147)$$

Furthermore, we observe:

$$\begin{aligned}\big\|\nabla f(\boldsymbol{X}(i))\big\|^2_F &= \big\|\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \qquad (148) \\ \big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i)) + \mathcal{P}^{\perp}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &= \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i)) + \mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \qquad (149) \\ \big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}^{\perp}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &= \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \qquad (150)\end{aligned}$$

Since $\mathcal{P}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i))$ is the best rank-k approximation to $\nabla f(\boldsymbol{X}(i))$, we have:

$$\begin{aligned}\big\|\mathcal{P}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i)) - \nabla f(\boldsymbol{X}(i))\big\|^2_F &\leq \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i)) - \nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \qquad (151) \\ \big\|\mathcal{P}^{\perp}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\leq \big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \qquad (152) \\ (1+\epsilon)\big\|\mathcal{P}^{\perp}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\leq (1+\epsilon)\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \qquad (153)\end{aligned}$$

where $\text{rank}(\text{span}(\mathcal{X}^* \setminus \mathcal{X}_i)) \leq k$. Using (147) in (153), the following series of inequalities are observed:

$$\big\|\mathcal{P}^{\perp}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \leq (1+\epsilon)\big\|\mathcal{P}^{\perp}_{\mathcal{D}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \leq (1+\epsilon)\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \qquad (154)$$

Now, in (150), we have:

$$\begin{aligned}\big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}^{\perp}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &= \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \overset{(153)}{\Leftrightarrow} \\ \big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + (1+\epsilon)\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \\ \big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \epsilon\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \\ \big\|\mathcal{P}_{\mathcal{D}^{\epsilon}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \epsilon\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \big\|\mathcal{P}_{\mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \overset{(i)}{\Leftrightarrow} \\ \big\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \epsilon\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{S}^*_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \overset{(ii)}{\Leftrightarrow} \\ \big\|\mathcal{P}_{\mathcal{S}_i \setminus \mathcal{S}^*_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F + \epsilon\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{S}^*_i \setminus \mathcal{S}_i}\nabla f(\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \\ \big\|\mathcal{P}_{\mathcal{S}_i \setminus \mathcal{S}^*_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|^2_F + \epsilon\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|^2_F &\geq \big\|\mathcal{P}_{\mathcal{S}^*_i \setminus \mathcal{S}_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|^2_F \Leftrightarrow \\ \big\|\mathcal{P}_{\mathcal{S}_i \setminus \mathcal{S}^*_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|_F + \sqrt{\epsilon}\big\|\mathcal{P}^{\perp}_{\mathcal{X}^* \setminus \mathcal{X}_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|_F &\geq \big\|\mathcal{P}_{\mathcal{S}^*_i \setminus \mathcal{S}_i}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\boldsymbol{X}(i))\big\|_F \qquad (155)\end{aligned}$$

Focusing on $\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*(\boldsymbol{y}-\mathcal{A}\boldsymbol{X}(i))\right\|_F$, we observe:

$$\begin{aligned}
&\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*(\boldsymbol{y}-\mathcal{A}\boldsymbol{X}(i))\right\|_F = \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*(\mathcal{A}\boldsymbol{X}^*+\boldsymbol{\varepsilon}-\mathcal{A}\boldsymbol{X}(i))\right\|_F = \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\\
&= \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\\
&\overset{(i)}{=} \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}_i}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}^*\cup\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\\
&\leq \left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F+\left\|\mathcal{P}_{\mathcal{X}_i}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\\
&+\left\|\mathcal{P}_{\mathcal{X}^*\cup\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F+\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F
\end{aligned}\tag{156}$$

where (i) is due to $\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\boldsymbol{X}=[\mathcal{P}_{\mathcal{X}_i}+\mathcal{P}_{\mathcal{X}^*\cup\mathcal{X}_i}^{\perp}]\boldsymbol{X},\ \forall\boldsymbol{X}\in\mathcal{R}^{m\times n}$.

In (156), we further observe:

- $\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq\delta_{2k}(\mathcal{A})\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq\delta_{2k}(\mathcal{A})\left\|\boldsymbol{X}^*-\boldsymbol{X}(i)\right\|_F$ since rank(span$((\mathcal{X}^*\setminus\mathcal{X}_i)\cup\mathcal{X}_i\cup\mathcal{X}^*))\leq 2k$ and $\left\|\mathcal{P}_{\boldsymbol{X}^*\setminus\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq\left\|\boldsymbol{X}^*-\boldsymbol{X}(i)\right\|_F$.
- $\left\|\mathcal{P}_{\mathcal{X}_i}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq(1+\delta_k(\mathcal{A}))\left\|\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\right\|_F\leq(1+\delta_k(\mathcal{A}))\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F$.
- $\left\|\mathcal{P}_{\mathcal{X}^*\cup\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq\left\|\mathcal{P}_{\mathcal{X}_i}^{\perp}\mathcal{A}^*\mathcal{A}\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F\leq\delta_{2k}(\mathcal{A})\left\|\mathcal{P}_{\mathcal{X}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\right\|_F\leq\delta_{2k}(\mathcal{A})\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F$.

Then, (156) becomes:

$$\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*(\boldsymbol{y}-\mathcal{A}\boldsymbol{X}(i))\right\|_F\leq(1+2\delta_{2k}(\mathcal{A})+\delta_k(\mathcal{A}))\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F+\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\tag{157}$$

Moreover, we know the following hold true from Lemma 6:

$$\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\mathcal{A}^*\mathcal{A}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\leq 2\delta_{3k}(\mathcal{A})\left\|\boldsymbol{X}^*-\boldsymbol{X}(i)\right\|_F+\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\tag{158}$$

and

$$\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\mathcal{A}^*\mathcal{A}(\boldsymbol{X}^*-\boldsymbol{X}(i))+\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\geq\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\boldsymbol{X}^*-\boldsymbol{X}(i))\right\|_F-2\delta_{2k}(\mathcal{A})\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F-\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F\tag{159}$$

Combining (157)-(159) in (155), we obtain:

$$\begin{aligned}
\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\right\|_F=\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\right\|_F&\leq\big(2\delta_{2k}(\mathcal{A})+2\delta_{3k}(\mathcal{A})+\sqrt{\epsilon}(1+2\delta_{2k}(\mathcal{A})+\delta_k(\mathcal{A}))\big)\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F\\
&+\sqrt{2(1+\delta_{2k}(\mathcal{A}))}\left\|\boldsymbol{\varepsilon}\right\|_2+\sqrt{\epsilon}\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F.
\end{aligned}\tag{160}$$

E. Proof of Theorem 4

To prove Theorem 4, we combine the following series of lemmas for each step of Algorithm 1.

Lemma 13. *[Error norm reduction via gradient descent] Let $\mathcal{S}_i:=\mathcal{X}_i\cup\mathcal{D}_i^{\epsilon}$ be a set of orthonormal, rank-1 matrices that span a rank-2k subspace in $\mathcal{R}^{m\times n}$. Then:*

$$\begin{aligned}
\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F\leq&\left[\left(1+\frac{\delta_{3k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\right)\left(2\delta_{2k}(\mathcal{A})+2\delta_{3k}(\mathcal{A})+\sqrt{\epsilon}(1+2\delta_{2k}(\mathcal{A})+\delta_k(\mathcal{A}))\right)+\frac{2\delta_{2k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\right]\left\|\boldsymbol{X}(i)-\boldsymbol{X}^*\right\|_F\\
&+\left[\big(1+\frac{\delta_{3k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\big)\sqrt{2(1+\delta_{2k}(\mathcal{A}))}+\frac{\sqrt{1+\delta_{2k}(\mathcal{A})}}{1-\delta_{2k}(\mathcal{A})}\right]\left\|\boldsymbol{\varepsilon}\right\|_2\\
&+\big(1+\frac{\delta_{3k}(\mathcal{A})}{1-\delta_{2k}(\mathcal{A})}\big)\sqrt{\epsilon}\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{X}_i}^{\perp}\mathcal{A}^*\boldsymbol{\varepsilon}\right\|_F.
\end{aligned}\tag{161}$$

Proof: We observe the following:

$$\left\|\boldsymbol{V}(i)-\boldsymbol{X}^*\right\|_F^2=\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2+\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2\tag{162}$$

The following equations hold true:

$$\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\right\|_F^2=\left\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\boldsymbol{X}^*\right\|_F^2=\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\boldsymbol{X}^*\right\|_F^2=\left\|\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i^{\epsilon}\cup\mathcal{X}_i)}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\right\|_F^2\tag{163}$$

Furthermore, we compute:

$$\begin{aligned}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F &= \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\boldsymbol{X}(i))-\boldsymbol{X}^*)\big\|_F\\ &= \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}(i)-\boldsymbol{X}^*)+\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F\\ &= \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)+\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F\\ &\le \big\|(I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F+\mu_i\big\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F+\mu_i\big\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F\\ &\overset{(i)}{\le} \frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F+\frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2\end{aligned} \tag{164}$$

where (i) is due to Lemmas 1, 3, 4 and $\frac{1}{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\le\mu_i\le\frac{1}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}$.

Using the subadditivity property of the square root in (162), we have:

$$\begin{aligned}\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F &\le \big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F+\big\|\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F\\ &\overset{(i)}{\le} \frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F\\ &+\frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2+\big\|\mathcal{P}^{\perp}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F\\ &\overset{(ii)}{\le}\left[\Big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(2\delta_{2k}(\boldsymbol{\mathcal{A}})+2\delta_{3k}(\boldsymbol{\mathcal{A}})+\sqrt{\epsilon}(1+2\delta_{2k}(\boldsymbol{\mathcal{A}})+\delta_k(\boldsymbol{\mathcal{A}}))\Big)+\frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right]\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F\\ &+\left[\big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big)\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}+\frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right]\big\|\boldsymbol{\varepsilon}\big\|_2\\ &+\big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big)\sqrt{\epsilon}\big\|\mathcal{P}^{\perp}_{\mathcal{X}^*\setminus\mathcal{X}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F.\end{aligned} \tag{165}$$

where (i) is due to (164) and (ii) is due to Lemma 7 and $\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{X}(i)-\boldsymbol{X}^*)\big\|_F\le\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F$. ■

We exploit Lemma 8 to obtain the following inequalities:

$$\begin{aligned}\big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*\big\|_F=\big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{V}(i)+\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F &\le \big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{V}(i)\big\|_F+\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\\ &\le (1+\epsilon)\big\|\boldsymbol{W}(i)-\boldsymbol{V}(i)\big\|_F+\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\\ &\le (2+\epsilon)\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\end{aligned} \tag{166}$$

where the last inequality holds since $\boldsymbol{W}(i)$ is the best rank-k matrix estimate of $\boldsymbol{V}(i)$ and, thus, $\big\|\boldsymbol{W}(i)-\boldsymbol{V}(i)\big\|_F\le\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F$.

Furthermore, Step 7 in Matrix ALPS I satisfies the following:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2=\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F^2+\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F^2\Rightarrow \tag{167}$$

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F\le\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F+\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F \tag{168}$$

In (165), we observe $\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F=\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F$ and

$$\begin{aligned}&\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*)\big\|_F\\ &=\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i+\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^*-\widehat{\boldsymbol{W}}_i)+\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}-\boldsymbol{X}^*)\big\|_F\\ &=\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)-\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)-\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\boldsymbol{W}(i)-\boldsymbol{X}^*)+\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F\\ &\le\big\|(I-\xi_i\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\widehat{\mathcal{W}}_i})\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F+\xi_i\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F+\xi_i\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F\\ &\le\frac{2\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F+\frac{\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta k(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F+\frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2\\ &=\big(\frac{2\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}+\frac{\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta k(\boldsymbol{\mathcal{A}})}\big)\big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*\big\|_F+\frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{\varepsilon}\big\|_2\end{aligned} \tag{169}$$

since $\frac{1}{1+\delta_k(\boldsymbol{\mathcal{A}})}\le\xi_i\le\frac{1}{1-\delta_k(\boldsymbol{\mathcal{A}})}$ and both $\big\|\mathcal{P}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F$ and $\big\|\mathcal{P}^{\perp}_{\widehat{\mathcal{W}}_i}(\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*)\big\|_F$ are less than or equal to $\big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*\big\|_F$.

Using the above in (168), we get:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq \big(1+\frac{2\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}+\frac{\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big)\big\|\widehat{\boldsymbol{W}}_i-\boldsymbol{X}^*\big\|_F+\frac{\sqrt{1+\delta_k(\boldsymbol{\mathcal{A}})}}{1-\delta_k(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\varepsilon}\|_2 \tag{170}$$

Furthermore, combining (170), (166) and (165), we obtain:

$$\begin{aligned}
&\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq \\
&\big(1+\frac{3\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big)\big(2+\epsilon\big)\left[\Big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big)\Big(4\delta_{3k}(\boldsymbol{\mathcal{A}})+\sqrt{\epsilon}(1+3\delta_{2k}(\boldsymbol{\mathcal{A}}))\Big)+\frac{2\delta_{2k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\right]\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F \\
&+\big(1+\frac{3\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big)\big(2+\epsilon\big)\Big[\big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big)\sqrt{2(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))}+2\frac{\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\Big]\|\boldsymbol{\varepsilon}\|_2 \\
&+\big(1+\frac{3\delta_k(\boldsymbol{\mathcal{A}})}{1-\delta_k(\boldsymbol{\mathcal{A}})}\big)\big(2+\epsilon\big)\big(1+\frac{\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{2k}(\boldsymbol{\mathcal{A}})}\big)\sqrt{\epsilon}\big\|\mathcal{P}^{\perp}_{\mathcal{X}^*\setminus\mathcal{X}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\big\|_F
\end{aligned} \tag{171}$$

since $\delta_k(\boldsymbol{\mathcal{A}}) \leq \delta_{2k}(\boldsymbol{\mathcal{A}}) \leq \delta_{3k}(\boldsymbol{\mathcal{A}})$.

F. Proof of Lemma 9

Using ineq. (139) and Lemma 8, we compute the following:

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2 \leq \big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F^2 &+ (1+\epsilon)\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\|_F^2 \\
&+ 2\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\sqrt{1+\epsilon}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F \\
&+ 2\sqrt{1+\epsilon}\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\big\|_F\|\boldsymbol{\varepsilon}\|_2
\end{aligned} \tag{172}$$

Furthermore, from (126), we further obtain in (172):

$$\begin{aligned}
\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2 &\leq \big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F^2+(1+\epsilon)\big(\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F+\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\varepsilon}\|_2\big)^2 \\
&+2\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\sqrt{1+\epsilon}\big(\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F+\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\varepsilon}\|_2\big) \\
&+2\sqrt{1+\epsilon}\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big(\delta_{3k}(\boldsymbol{\mathcal{A}})\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F+\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\varepsilon}\|_2\big)\|\boldsymbol{\varepsilon}\|_2 \qquad (173)\\
&\leq \big(1+((1+\epsilon)+2\sqrt{1+\epsilon})\delta_{3k}^2(\boldsymbol{\mathcal{A}})\big)\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F^2 \\
&+\big((1+\epsilon)+2\sqrt{1+\epsilon}\big)2\delta_{3k}(\boldsymbol{\mathcal{A}})\sqrt{1+\delta_{2k}(\boldsymbol{\mathcal{A}})}\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F\|\boldsymbol{\varepsilon}\|_2 \\
&+\big((1+\epsilon)+2\sqrt{1+\epsilon}\big)(1+\delta_{2k}(\boldsymbol{\mathcal{A}}))\|\boldsymbol{\varepsilon}\|_2^2 \qquad (174)
\end{aligned}$$

Completing the squares in (174) and eliminating some negative terms, we obtain:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2 \leq \big(1+\big((1+\epsilon)+2\sqrt{1+\epsilon}\big)\delta_{3k}^2(\boldsymbol{\mathcal{A}})\big)\left(\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F+\sqrt{\frac{\big((1+\epsilon)+2\sqrt{1+\epsilon}\big)\big(1+\delta_{2k}(\boldsymbol{\mathcal{A}})\big)}{1+\big((1+\epsilon)+2\sqrt{1+\epsilon}\big)\delta_{3k}^2(\boldsymbol{\mathcal{A}})}}\|\boldsymbol{\varepsilon}\|_2\right)^2 \tag{175}$$

Computing the square root on both sides of the above expression, we obtain the desired inequality.

G. Proof of Lemma 3

Let $\mathcal{X}^* \leftarrow \text{ortho}(\boldsymbol{X}^*)$ be a set of orthonormal, rank-1 matrices that span the range of $\boldsymbol{X}^*$. In Algorithm 3, $\boldsymbol{X}(i+1)$ is the best rank-k approximation of $\boldsymbol{V}(i)$. Thus:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{V}(i)\big\|_F^2 \leq \big\|\boldsymbol{X}^*-\boldsymbol{V}(i)\big\|_F^2 \Rightarrow \tag{176}$$

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*+\boldsymbol{X}^*-\boldsymbol{V}(i)\big\|_F^2 \leq \big\|\boldsymbol{X}^*-\boldsymbol{V}(i)\big\|_F^2 \Rightarrow \tag{177}$$

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2+\big\|\boldsymbol{V}(i)-\boldsymbol{X}^*\big\|_F^2+2\langle\boldsymbol{X}(i+1)-\boldsymbol{X}^*,\boldsymbol{X}^*-\boldsymbol{V}(i)\rangle \leq \big\|\boldsymbol{X}^*-\boldsymbol{V}(i)\big\|_F^2 \Rightarrow \tag{178}$$

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2 \leq 2\langle\boldsymbol{X}(i+1)-\boldsymbol{X}^*,\boldsymbol{V}(i)-\boldsymbol{X}^*\rangle \tag{179}$$

From Algorithm 3, it is obvious that *i)* $\boldsymbol{V}(i) \in \text{span}(\mathcal{S}_i)$, *ii)* $\boldsymbol{Q}_i \in \text{span}(\mathcal{S}_i)$ and *iii)* $\boldsymbol{W}(i) \in \text{span}(\mathcal{S}_i)$. We define $\mathcal{E} := \mathcal{S}_i \cup \mathcal{X}^*$ where $\text{rank}(\text{span}(\mathcal{E})) \leq 4k$ and let $\mathcal{P}_{\mathcal{E}}$ be the orthogonal projection onto the subspace defined by $\mathcal{E}$.

Since $\boldsymbol{X}(i+1)-\boldsymbol{X}^* \in \text{span}(\mathcal{E})$ and $\boldsymbol{V}(i)-\boldsymbol{X}^* \in \text{span}(\mathcal{E})$, the following hold true:

$$\boldsymbol{X}(i+1)-\boldsymbol{X}^* = \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*) \;\text{ and }\; \boldsymbol{V}(i)-\boldsymbol{X}^* = \mathcal{P}_{\mathcal{E}}(\boldsymbol{V}(i)-\boldsymbol{X}^*) \tag{180}$$

due to Remark 3.

Then, (179) can be written as:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F^2 \tag{181}$$
$$\leq 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}(\boldsymbol{V}(i)-\boldsymbol{X}^*)\rangle \tag{182}$$
$$= 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}\left(\boldsymbol{Q}_i+\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{X}^*-\boldsymbol{Q}_i)-\boldsymbol{X}^*\right)\rangle \tag{183}$$
$$= 2\langle \mathcal{P}_{\mathcal{E}}(\boldsymbol{X}(i+1)-\boldsymbol{X}^*), \mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\boldsymbol{Q}_i-\boldsymbol{X}^*))\rangle \tag{184}$$
$$= 2\langle \boldsymbol{X}(i+1)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\rangle \tag{185}$$
$$= 2\langle \boldsymbol{X}(i+1)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\rangle \tag{186}$$
$$\overset{(i)}{=} 2\langle \boldsymbol{X}(i+1)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\big[\mathcal{P}_{\mathcal{S}_i}+\mathcal{P}_{\mathcal{S}_i}^{\perp}\big]\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\rangle \tag{187}$$
$$= 2\langle \boldsymbol{X}(i+1)-\boldsymbol{X}^*, (I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\rangle - 2\mu_i\langle \boldsymbol{X}(i+1)-\boldsymbol{X}^*, \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\rangle \tag{188}$$
$$\overset{(ii)}{\leq} 2\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F\big\|(I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F + 2\mu_i\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F\big\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F \tag{189}$$

where (i) is due to $\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*) := \mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)+\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)$ and (ii) follows from Cauchy-Schwarz inequality. Since $\frac{1}{1+\delta_{3k}(\boldsymbol{\mathcal{A}})} \leq \mu_i \leq \frac{1}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}$, Lemma 3 implies:

$$\lambda(I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}) \in \left[1-\frac{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}{1+\delta_{3k}(\boldsymbol{\mathcal{A}})}, \frac{1+\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}-1\right] \leq \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}. \tag{190}$$

and thus:

$$\big\|(I-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F \leq \frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F. \tag{191}$$

Furthermore, according to Lemma 4:

$$\big\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F \leq \delta_{4k}(\boldsymbol{\mathcal{A}})\big\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F \tag{192}$$

since $\text{rank}(\mathcal{P}_{\mathcal{E}\cup\mathcal{S}_i}\boldsymbol{Q}) \leq 4k,\ \forall \boldsymbol{Q} \in \mathcal{R}^{m\times n}$. Since $\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*) = \mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\boldsymbol{X}^*$ where $\mathcal{D}_i := \mathcal{P}_k\left(\mathcal{P}_{\mathcal{X}_i}^{\perp}\nabla f(\boldsymbol{Q}_i)\right)$, then:

$$\big\|\mathcal{P}_{\mathcal{S}_i}^{\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\boldsymbol{X}^*)\big\|_F = \big\|\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\boldsymbol{X}^*\big\|_F \leq (2\delta_{3k}(\boldsymbol{\mathcal{A}})+2\delta_{4k}(\boldsymbol{\mathcal{A}}))\big\|\boldsymbol{Q}_i-\boldsymbol{X}^*\big\|_F, \tag{193}$$

using Lemma 6. Using the above in (189), we compute:

$$\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq \Big(\frac{4\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + (2\delta_{3k}(\boldsymbol{\mathcal{A}})+2\delta_{4k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{Q}_i-\boldsymbol{X}^*\big\|_F \tag{194}$$

Furthermore:

$$\begin{aligned}\big\|\boldsymbol{Q}_i-\boldsymbol{X}^*\big\|_F &= \big\|\boldsymbol{X}(i)+\tau_i(\boldsymbol{X}(i)-\boldsymbol{X}(i-1))\big\|_F \\ &= \big\|(1+\tau_i)(\boldsymbol{X}(i)-\boldsymbol{X}^*)+\tau_i(\boldsymbol{X}^*-\boldsymbol{X}(i-1))\big\|_F \\ &\leq (1+\tau_i)\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F + \tau_i\big\|\boldsymbol{X}(i-1)-\boldsymbol{X}^*\big\|_F\end{aligned} \tag{195}$$

Combining (194) and (195), we get:

$$\begin{aligned}\big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F \leq{}& (1+\tau_i)\Big(\frac{4\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + (2\delta_{3k}(\boldsymbol{\mathcal{A}})+2\delta_{4k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{X}(i)-\boldsymbol{X}^*\big\|_F \\ &+ \tau_i\Big(\frac{4\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + (2\delta_{3k}(\boldsymbol{\mathcal{A}})+2\delta_{4k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}\Big)\big\|\boldsymbol{X}(i-1)-\boldsymbol{X}^*\big\|_F\end{aligned} \tag{196}$$

Let $\alpha := \frac{4\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})} + (2\delta_{3k}(\boldsymbol{\mathcal{A}})+2\delta_{4k}(\boldsymbol{\mathcal{A}}))\frac{2\delta_{3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}$ and $g(i) := \big\|\boldsymbol{X}(i+1)-\boldsymbol{X}^*\big\|_F$. Then, (196) defines the following homogeneous recurrence:

$$g(i+1)-\alpha(1+\tau_i)g(i)+\alpha\tau_i g(i-1) \leq 0 \tag{197}$$

Using the *method of characteristic roots* to solve the above recurrence, we assume that the homogeneous linear recursion has solution of the form $g(i) = r^i$ for $r \in \mathcal{R}$. Thus, replacing $g(i) = r^i$ in (197) and factoring out $r^{(i-2)}$, we form the following characteristic polynomial:

$$r^2-\alpha(1+\tau_i)r-\alpha\tau_i \leq 0 \tag{198}$$

Focusing on the worst case where (198) is satisfied with equality, we compute the roots $r_{1,2}$ of the quadratic characteristic polynomial as:

$$r_{1,2} = \frac{\alpha(1+\tau_i) \pm \sqrt{\Delta}}{2}, \text{ where } \Delta := \alpha^2(1+\tau_i)^2 + 4\alpha\tau_i. \tag{199}$$

Then, as a general solution, we combine the above roots with unknown coefficients b_1, b_2 to obtain:

$$\begin{aligned} g(i+1) &\leq \left[b_1 \Big(\frac{\alpha(1+\tau_i) + \sqrt{\Delta}}{2} \Big)^{i+1} + b_2 \Big(\frac{\alpha(1+\tau_i) - \sqrt{\Delta}}{2} \Big)^{i+1} \right] \left\| \boldsymbol{X}(0) - \boldsymbol{X}^* \right\|_F \\ &\leq \left[(b_1 + b_2) \Big(\frac{\alpha(1+\tau_i) + \sqrt{\Delta}}{2} \Big)^{i+1} \right] \left\| \boldsymbol{X}(0) - \boldsymbol{X}^* \right\|_F \end{aligned} \tag{200}$$

Using the initial condition $g(0) := \left\| \boldsymbol{X}(0) - \boldsymbol{X}^* \right\|_F \overset{\boldsymbol{X}(0)=0}{=} \left\| \boldsymbol{X}^* \right\|_F = 1$, we get $b_1 + b_2 = 1$. Thus, we conclude to the following recurrence:

$$\left\| \boldsymbol{X}(i+1) - \boldsymbol{X}^* \right\|_F \leq \Big(\frac{\alpha(1+\tau_i) + \sqrt{\Delta}}{2} \Big)^{i+1}. \tag{201}$$

REFERENCES

[1] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289 – 1306, April 2006.

[2] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[3] R.G. Baraniuk, V. Cevher, and M.B. Wakin. Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of the IEEE*, 98(6):959–971, 2010.

[4] E.J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[5] R. Meka, P. Jain, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS Workshop on Discrete Optimization in Machine Learning*, 2010.

[6] H. Tyagi and V. Cevher. Learning ridge functions with randomized sampling in high dimensions. *Technical report, EPFL*, 2011.

[7] H. Tyagi and V. Cevher. Learning non-parametric basis independent models from point queries via low-rank methods. *Technical report, EPFL*, 2012.

[8] J. Bennett and S. Lanning. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[9] A. Kyrillidis and V. Cevher. Matrix alps: Accelerated low rank and sparse matrix reconstruction. *Technical report, EPFL*, 2012.

[10] A.E. Waters, A.C. Sankaranarayanan, and R.G. Baraniuk. Sparcs: Recovering low-rank and sparse matrices from compressive measurements. In *NIPS*, 2011.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[12] M. Fazel, B. Recht, and P. A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[13] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B*, 58(1):267–288, 1996.

[14] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31:1235–1256, November 2009.

[15] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *American Control Conference (ACC)*. IEEE, 2010.

[16] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20:1956–1982, March 2010.

[17] B. Recht and C. Re. Parallel stochastic gradient algorithms for large-scale matrix completion. *Preprint*, 2011.

[18] L. Wu Z. Lin, M. Chen and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report UILU-ENG-09-2215*.

[19] J. Wright L. Wu M. Chen Z. Lin, A. Ganesh and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *UIUC Technical Report UILU-ENG-09-2214*.

[20] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[21] K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. *IEEE Trans. on Information Theory*, 56(9):4402–4416, 2010.

[22] D. Goldfarb and S. Ma. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Found. Comput. Math.*, 11:183–210, April 2011.

[23] A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In *Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2011.

[24] A. Kyrillidis and V. Cevher. Sublinear time, approximate model-based sparse recovery for all. *Technical report, EPFL*, 2011.

[25] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53:217–288, May 2011.

[26] P.A. Parrilo, A.S. Willsky, V. Chandrasekaran, and B. Recht. The convex geometry of linear inverse problems, *Preprint*, 2010.

[27] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge Univ. Press, 1990.

[28] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, 22(1):211–231, 2009.

[29] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. on Information Theory*, 50(10):2231–2242, Oct. 2004.

[30] D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1995.

[31] V. Cevher. An alps view of sparse recovery. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5808–5811. IEEE, 2011.

[32] D. Needell and J.A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[33] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. on Information Theory*, 55:2230–2249, May 2009.

[34] S. Foucart. Hard thresholding pursuit: an algorithm for compressed sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[35] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Appl. Comp. Harm. Anal*, 27(3):265–274, 2009.

[36] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *ICML*. ACM, 2009.

[37] T. Blumensath and M. E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *J. Sel. Topics Signal Processing*, 4(2):298–309, 2010.

[38] T. Blumensath. Accelerated iterative hard thresholding. *Signal Process.*, 92:752–756, March 2012.

[39] S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Proceedings of the 13th International Conference on Approximation Theory*, 2010.

[40] R. Coifman, F. Geshwind, and Y. Meyer. Noiselets. *Applied and Computational Harmonic Analysis*, 10(1):27–44, 2001.

[41] V. Cevher. On accelerated hard thresholding methods for sparse approximation. *Technical report, EPFL*, 2011.

[42] Y. Nesterov. *Introductory lectures on convex optimization*. Kluwer Academic Publishers, 1996.

[43] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1):9–33, 2004.

[44] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36:158–183, July 2006.

[45] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, pages 1117–1126, New York, NY, USA, 2006. ACM.

[46] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(042), 2006.

[47] R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. on Information Theory*, 56(6):2980–2998, 2010.

[48] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

[49] J. He, L. Balzano, and J. C. S. Lui. Online robust subspace tracking from partial information. *arXiv:1109.3827*, 2011.

[50] N. Boumal and P.A. Absil. Rtrmc: A riemannian trust-region method for low-rank matrix completion. In *NIPS*, 2011.

[51] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Rice University CAAM Technical Report TR10-07. Submitted*, 2010.

[52] R. M. Larsen. Propack: Software for large and sparse svd calculations. http://soi.stanford.edu/r̃munk/PROPACK.

[53] X. Shi and P.S. Yu. Limitations of matrix completion via trace norm minimization. *ACM SIGKDD Explorations Newsletter*, 12(2):16–20, 2011.