Journal of Machine Learning Research 20 (2019) 1-33 Submitted 1/18; Revised 11/18; Published 2/19

# Accelerated Alternating Projections for Robust Principal Component Analysis

**HanQin Cai** HQCAI@MATH.UCLA.EDU
*Department of Mathematics*
*University of California, Los Angeles*
*Los Angeles, California, USA*

**Jian-Feng Cai** JFCAI@UST.HK
*Department of Mathematics*
*Hong Kong University of Science and Technology*
*Hong Kong SAR, China*

**Ke Wei** KEWEI@FUDAN.EDU.CN
*School of Data Science*
*Fudan University*
*Shanghai, China*

**Editor:** Sujay Sanghavi

## Abstract

We study robust PCA for the fully observed setting, which is about separating a low rank matrix $\boldsymbol{L}$ and a sparse matrix $\boldsymbol{S}$ from their sum $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$. In this paper, a new algorithm, dubbed accelerated alternating projections, is introduced for robust PCA which significantly improves the computational efficiency of the existing alternating projections proposed in (Netrapalli et al., 2014) when updating the low rank factor. The acceleration is achieved by first projecting a matrix onto some low dimensional subspace before obtaining a new estimate of the low rank matrix via truncated SVD. Exact recovery guarantee has been established which shows linear convergence of the proposed algorithm. Empirical performance evaluations establish the advantage of our algorithm over other state-of-the-art algorithms for robust PCA.

**Keywords:** Robust PCA, Alternating Projections, Matrix Manifold, Tangent Space, Subspace Projection

## 1. Introduction

Robust principal component analysis (RPCA) appears in a wide range of applications, including video and voice background subtraction (Li et al., 2004; Huang et al., 2012), sparse graphs clustering (Chen et al., 2012), 3D reconstruction (Mobahi et al., 2011), and fault isolation (Tharrault et al., 2008). Suppose we are given a sum of a low rank matrix and a sparse matrix, denoted $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$. The goal of RPCA is to reconstruct $\boldsymbol{L}$ and $\boldsymbol{S}$ simultaneously from $\boldsymbol{D}$. As a concrete example, for foreground-background separation in video processing, $\boldsymbol{L}$ represents static background through all the frames of a video which should be low rank while $\boldsymbol{S}$ represents moving objects which can be assumed to be sparse since typically they will not block a large portion of the screen for a long time.

©2019 HanQin Cai, Jian-Feng Cai and Ke Wei.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-022.html.

RPCA can be achieved by seeking a low rank matrix $\boldsymbol{L'}$ and a sparse matrix $\boldsymbol{S'}$ such that their sum fits the measurement matrix $\boldsymbol{D}$ as well as possible:

$$\min_{\boldsymbol{L'},\boldsymbol{S'}\in\mathbb{R}^{m\times n}} \|\boldsymbol{D}-\boldsymbol{L'}-\boldsymbol{S'}\|_F \quad \text{subject to } rank(\boldsymbol{L'})\le r \text{ and } \|\boldsymbol{S'}\|_0\le|\Omega|, \tag{1}$$

where $r$ denotes the rank of the underlying low rank matrix, $\Omega$ denotes the support set of the underlying sparse matrix, and $\|\boldsymbol{S'}\|_0$ counts the number of non-zero entries in $\boldsymbol{S'}$. Compared to the traditional principal component analysis (PCA) which computes a low rank approximation of a data matrix, RPCA is less sensitive to outliers since it includes a sparse part in the formulation.

Since the seminal works of (Wright et al., 2009; Candès et al., 2011; Chandrasekaran et al., 2011), RPCA has received intensive investigations both from theoretical and algorithmic aspects. Noticing that (1) is a non-convex problem, some of the earlier works focus on the following convex relaxation of RPCA:

$$\min_{\boldsymbol{L'},\boldsymbol{S'}\in\mathbb{R}^{m\times n}} \|\boldsymbol{L'}\|_*+\lambda\|\boldsymbol{S'}\|_1 \quad \text{subject to } \boldsymbol{L'}+\boldsymbol{S'}=\boldsymbol{D}, \tag{2}$$

where $\|\cdot\|_*$ is the nuclear norm (*viz.* trace norm) of matrices, $\lambda$ is the regularization parameter, and $\|\cdot\|_1$ denotes the $\ell_1$-norm of the vectors obtained by stacking the columns of associated matrices. Under some mild conditions, it has been proven that the RPCA problem can be solved exactly by the aforementioned convex relaxation Candès et al. (2011); Chandrasekaran et al. (2011). However, a limitation of the convex relaxation based approach is that the resulting semidefinite programming is computationally rather expensive to solve, even for medium size matrices. Alternative to the convex relaxation, many non-convex algorithms have been designed to target (1) directly. This line of research will be reviewed in more detail in Section 2.3 after our approach has been introduced.

This paper targets the non-convex optimization for RPCA directly. The main contributions of this work are two-fold. Firstly, we propose a new algorithm, accelerated alternating projections (AccAltProj), for RPCA, which is substantially faster than other state-of-the-art algorithms. Secondly, exact recovery of accelerated alternating projections has been established for the fixed sparsity model, where we assume the ratio of the number of non-zero entries in each row and column of $\boldsymbol{S}$ is less than a threshold.

### 1.1. Assumptions

It is clear that the RPCA problem is ill-posed without any additional conditions. Common assumptions are that $\boldsymbol{L}$ cannot be too sparse and $\boldsymbol{S}$ cannot be locally too dense, which are formalized in A1 and A2, respectively.

**A1** *The underlying low rank matrix $\boldsymbol{L}\in\mathbb{R}^{m\times n}$ is a rank-$r$ matrix with $\mu$-incoherence, that is*

$$\max_i \|\boldsymbol{e}_i^T\boldsymbol{U}\|_2\le\sqrt{\frac{\mu r}{m}}, \quad \text{and} \quad \max_j \|\boldsymbol{e}_j^T\boldsymbol{V}\|_2\le\sqrt{\frac{\mu r}{n}}$$

*hold for a positive numerical constant $1\le\mu\le\frac{\min\{m,n\}}{r}$, where $\boldsymbol{L}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{L}$.*

Assumption A1 was first introduced in (Candès and Recht, 2009) for low rank matrix completion, and now it is a very standard assumption for related low rank reconstruction problems. It basically states that the left and right singular vectors of $\boldsymbol{L}$ are weakly correlated with the canonical basis, which implies $\boldsymbol{L}$ cannot be a very sparse matrix.

**A2** *The underlying sparse matrix $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is $\alpha$-sparse. That is, $\boldsymbol{S}$ has at most $\alpha n$ non-zero entries in each row, and at most $\alpha m$ non-zero entries in each column. In the other words, for all $1 \le i \le m, 1 \le j \le n$,*

$$\|\boldsymbol{e}_i^T \boldsymbol{S}\|_0 \le \alpha n \quad \textit{and} \quad \|\boldsymbol{S}\boldsymbol{e}_j\|_0 \le \alpha m. \tag{3}$$

*In this paper, we assume*[1]

$$\alpha \lesssim \min\left\{\frac{1}{\mu r^2\kappa^3}, \frac{1}{\mu^{1.5} r^2 \kappa}, \frac{1}{\mu^2 r^2}\right\}, \tag{4}$$

*where $\kappa$ is the condition number of $\boldsymbol{L}$.*

Assumption A2 states that the non-zero entries of the sparse matrix $\boldsymbol{S}$ cannot concentrate in a few rows or columns, so there does not exist a low rank component in $\boldsymbol{S}$. If the indices of the support set $\Omega$ are sampled independently from the Bernoulli distribution with the associated parameter being slightly smaller than $\alpha$, by the Chernoff inequality, one can easily show that (3) holds with high probability.

### 1.2. Organization and Notation of the Paper

The rest of the paper is organized as follows. In the remainder of this section, we introduce standard notation that is used throughout the paper. Section 2.1 presents the proposed algorithm and discusses how to implement it efficiently. The theoretical recovery guarantee of the proposed algorithm is presented in Section 2.2, followed by a review of prior art for RPCA. In Section 3, we present the numerical simulations of our algorithm. Section 4 contains all the mathematical proofs of our main theoretical result. We conclude this paper with future directions in Section 5.

In this paper, vectors are denoted by bold lowercase letters (e.g., $\boldsymbol{x}$), matrices are denoted by bold capital letters (e.g., $\boldsymbol{X}$), and operators are denoted by calligraphic letters (e.g., $\mathcal{H}$). In particular, $\boldsymbol{e}_i$ denotes the $i^{th}$ canonical basis vector, $\boldsymbol{I}$ denotes the identity matrix, and $\mathcal{I}$ denotes the identity operator. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_0$ counts the number of non-zero entries in $\boldsymbol{x}$, and $\|\boldsymbol{x}\|_2$ denotes the $\ell_2$ norm of $\boldsymbol{x}$. For a matrix $\boldsymbol{X}$, $[\boldsymbol{X}]_{ij}$ denotes its $(i,j)^{th}$ entry, $\sigma_i(\boldsymbol{X})$ denotes its $i^{th}$ singular value, $\|\boldsymbol{X}\|_\infty = \max_{ij} |[\boldsymbol{X}]_{ij}|$ denotes the maximum magnitude of its entries, $\|\boldsymbol{X}\|_2 = \sigma_1(\boldsymbol{X})$ denotes its spectral norm, $\|\boldsymbol{X}\|_F = \sqrt{\sum_i \sigma_i^2(\boldsymbol{X})}$ denotes its Frobenius norm, and $\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X})$ denotes its nuclear norm. The inner product of two real valued vectors is defined as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^T\boldsymbol{y}$, and the inner product of two real valued matrices is defined as $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = Trace(\boldsymbol{X}^T\boldsymbol{Y})$, where $(\cdot)^T$ represents the transpose of a vector or matrix.

Additionally, we sometimes use the shorthand $\sigma_i^A$ to denote the $i^{th}$ singular value of a matrix $\boldsymbol{A}$. Note that $\kappa = \sigma_1^L/\sigma_r^L$ always denotes the condition number of the underlying

1. The standard notion "$\lesssim$" in (4) means there exists an absolute numerical constant $C > 0$ such that $\alpha$ can be upper bounded by $C$ times the right hand side.

rank-$r$ matrix $\boldsymbol{L}$, and $\Omega = supp(\boldsymbol{S})$ is always referred to as the support of the underlying sparse matrix $\boldsymbol{S}$. At the $k^{th}$ iteration of the proposed algorithm, the estimates of the low rank matrix and the sparse matrix are denoted by $\boldsymbol{L}_k$ and $\boldsymbol{S}_k$, respectively.

## 2. Algorithm and Theoretical Results

In this section, we present the new algorithm and its recovery guarantee. For ease of exposition, we assume all matrices are square (i.e., $m = n$), but emphasize that nothing is special about this assumption and all the results can be easily extended to rectangular matrices.

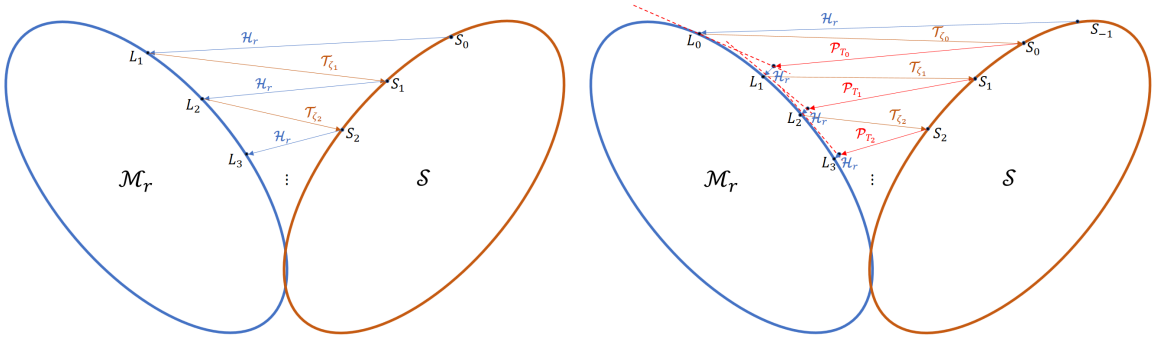
### 2.1. Proposed Algorithm

Alternating projections is a minimization approach that has been successfully used in many fields, including image processing (Wang et al., 2008; Chan and Wong, 2000; O'Sullivan and Benac, 2007), matrix completion (Keshavan et al., 2012; Jain et al., 2013; Hardt, 2013; Tanner and Wei, 2016), phase retrieval (Netrapalli et al., 2013; Cai et al., 2017; Zhang, 2017), and many others (Peters and Heath, 2009; Agarwal et al., 2014; Yu et al., 2016; Pu et al., 2017). A non-convex algorithm based on alternating projections, namely AltProj, is presented in (Netrapalli et al., 2014) for RPCA accompanied with a theoretical recovery guarantee. In each iteration, AltProj first updates $\boldsymbol{L}$ by projecting $\boldsymbol{D} - \boldsymbol{S}$ onto the space of rank-$r$ matrices, denoted $\mathcal{M}_r$, and then updates $\boldsymbol{S}$ by projecting $\boldsymbol{D} - \boldsymbol{L}$ onto the space of sparse matrices, denoted $\mathcal{S}$; see the left plot of Figure 1 for an illustration. Regarding to the implementation of AltProj, the projection of a matrix onto the space of low rank matrices can be computed by the singular value decomposition (SVD) followed by truncating out small singular values, while the projection of a matrix onto the space of sparse matrices can be computed by the hard thresholding operator. As a non-convex algorithm which targets (1) directly, AltProj is computationally much more efficient than solving the convex relaxation problem (2) using semidefinite programming (SDP). However, when projecting $\boldsymbol{D}-\boldsymbol{S}$ onto the low rank matrix manifold, AltProj requires to compute the SVD of a full size matrix, which is computationally expensive. Inspired by the work in (Vandereycken, 2013; Wei et al., 2016a,b), we propose an accelerated algorithm for RPCA, coined accelerated alternating projections (AccAltProj), to circumvent the high computational cost of the SVD. The new algorithm is able to reduce the per-iteration computational cost of AltProj significantly, while a theoretical guarantee can be similarly established.

Our algorithm consists of two phases: initialization and projections onto $\mathcal{M}_r$ and $\mathcal{S}$ alternatively. We begin our discussion with the second phase, which is described in Algorithm 1. For geometric comparison between AltProj and AccAltProj, see Figure 1.

Let $(\boldsymbol{L}_k, \boldsymbol{S}_k)$ be a pair of current estimates. At the $(k+1)^{th}$ iteration, AccAltProj first trims $\boldsymbol{L}_k$ into an incoherent matrix $\widetilde{\boldsymbol{L}}_k$ using Algorithm 2. Noting that $\widetilde{\boldsymbol{L}}_k$ is still a rank-$r$ matrix, so its left and right singular vectors define an $(2n-r)r$-dimensional subspace (Vandereycken, 2013),

$$\widetilde{T}_k = \{\widetilde{\boldsymbol{U}}_k \boldsymbol{A}^T + \boldsymbol{B}\widetilde{\boldsymbol{V}}_k^T \mid \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times r}\}, \tag{5}$$




(i) Illustration of AltProj (ii) Illustration of AccAltProj

Figure 1: Visual comparison between AltProj and AccAltProj, where $\mathcal{M}_r$ denotes the manifold of rank-$r$ matrices and $\mathcal{S}$ denotes the set of sparse matrices. The red dash line in (ii) represents the tangent space of $\mathcal{M}_r$ at $\boldsymbol{L}_k$. In fact, each circle represents a sum of a low rank matrix and a sparse matrix, but with the component on one circle fixed when projecting onto the other circle. For conciseness, the trim stage, i.e., $\widetilde{\boldsymbol{L}}_k$, is not included in the plot for AccAltProj.

**Algorithm 1** Robust PCA by Accelerated Alternating Projections (AccAltProj)

**Input:** $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$: matrix to be split; $r$: rank of $\boldsymbol{L}$; $\epsilon$: target precision level; $\beta$: thresholding parameter; $\gamma$: target converge rate; $\mu$: incoherence parameter of $\boldsymbol{L}$.
**Initialization**
$k = 0$
**while** $<\|\boldsymbol{D} - \boldsymbol{L}_k - \boldsymbol{S}_k\|_F / \|\boldsymbol{D}\|_F \geq \epsilon>$ **do**
  $\widetilde{\boldsymbol{L}}_k = \mathrm{Trim}(\boldsymbol{L}_k, \mu)$
  $\boldsymbol{L}_{k+1} = \mathcal{H}_r(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k))$
  $\zeta_{k+1} = \beta\left(\sigma_{r+1}\left(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)\right) + \gamma^{k+1}\sigma_1\left(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)\right)\right)$
  $\boldsymbol{S}_{k+1} = \mathcal{T}_{\zeta_{k+1}}(\boldsymbol{D} - \boldsymbol{L}_{k+1})$
  $k = k + 1$
**end while**
**Output:** $\boldsymbol{L}_k$, $\boldsymbol{S}_k$

where $\widetilde{\boldsymbol{L}}_k = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{V}}_k^T$ is the SVD of $\widetilde{\boldsymbol{L}}_k$[2]. Given a matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$, it can be easily verified that the projections of $\boldsymbol{Z}$ onto the subspace $\widetilde{T}_k$ and its orthogonal complement are given by

$$\mathcal{P}_{\widetilde{T}_k}\boldsymbol{Z} = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{Z} + \boldsymbol{Z}\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{Z}\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T \tag{6}$$

and

$$(\mathcal{I} - \mathcal{P}_{\widetilde{T}_k})\boldsymbol{Z} = (\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\boldsymbol{Z}(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T). \tag{7}$$

2. In practice, we only need the trimmed orthogonal matrices $\widetilde{\boldsymbol{U}}_k$ and $\widetilde{\boldsymbol{V}}_k$ for the projection $\mathcal{P}_{\widetilde{T}_k}$, and they can be computed efficiently via a QR decomposition. The entire matrix $\widetilde{\boldsymbol{L}}_k$ should never be formed in an efficient implementation of AccAltProj.

**Algorithm 2** Trim

**Input:** $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$: matrix to be trimmed; $\mu$: target incoherence level.
$c_\mu = \sqrt{\frac{\mu r}{n}}$
**for** $<i = 1$ `to` $m>$ **do**
    $\boldsymbol{A}^{(i)} = \min\{1, \frac{c_\mu}{\|\boldsymbol{U}^{(i)}\|}\}\boldsymbol{U}^{(i)}$
**end for**
**for** $<j = 1$ `to` $n>$ **do**
    $\boldsymbol{B}^{(j)} = \min\{1, \frac{c_\mu}{\|\boldsymbol{V}^{(j)}\|}\}\boldsymbol{V}^{(i)}$
**end for**
**Output:** $\widetilde{\boldsymbol{L}} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{B}$

As stated previously, AltProj truncates the SVD of $\boldsymbol{D}-\boldsymbol{S}_k$ directly to get a new estimate of $\boldsymbol{L}$. In contrast, AccAltProj first projects $\boldsymbol{D}-\boldsymbol{S}_k$ onto the low dimensional subspace $\widetilde{T}_k$, and then projects the intermediate matrix onto the rank-$r$ matrix manifold $\mathcal{M}_r$ using the truncated SVD. That is,

$$\boldsymbol{L}_{k+1} = \mathcal{H}_r(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)),$$

where $\mathcal{H}_r$ computes the best rank-$r$ approximation of a matrix,

$$\mathcal{H}_r(\boldsymbol{Z}) := \boldsymbol{Q}\boldsymbol{\Lambda}_r\boldsymbol{P}^T \text{ where } \boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{P}^T \text{ is its SVD and } [\boldsymbol{\Lambda}_r]_{ii} := \begin{cases} [\boldsymbol{\Lambda}]_{ii} & i \le r \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Before proceeding, it is worth noting that the set of rank-$r$ matrices $\mathcal{M}_r$ form a smooth manifold of dimension $(2n-r)r$, and $\widetilde{T}_k$ is indeed the tangent space of $\mathcal{M}_r$ at $\widetilde{\boldsymbol{L}}_k$ (Vandereycken, 2013). Matrix manifold algorithms based on the tangent space of low dimensional spaces have been widely studied in the literature, see for example (Ngo and Saad, 2012; Mishra et al., 2012; Vandereycken, 2013; Mishra and Sepulchre, 2014; Mishra et al., 2014; Wei et al., 2016a,b) and references therein. In particular, we invite readers to explore the book (Absil et al., 2009) for more details about the differential geometry ideas behind manifold algorithms.

One can see that a SVD is still needed to obtain the new estimate $\boldsymbol{L}_{k+1}$. Nevertheless, it can be computed in a very efficient way (Vandereycken, 2013; Wei et al., 2016a,b). Let $(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{D}-\boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k = \boldsymbol{Q}_1\boldsymbol{R}_1$ and $(\boldsymbol{I}-\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)(\boldsymbol{D}-\boldsymbol{S}_k)\widetilde{\boldsymbol{U}}_k = \boldsymbol{Q}_2\boldsymbol{R}_2$ be the QR decompositions of $(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k$ and $(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{U}}_k$, respectively. Note that $(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k$ and $(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{U}}_k$ can be computed by one matrix-matrix subtraction between an $n\times n$ matrix and an $n\times n$ matrix, two matrix-matrix multiplications between an $n \times n$ matrix and an $n \times r$ matrix, and a few matrix-matrix multiplications between a $r \times n$ and an $n \times r$ or between an $n \times r$ matrix and a $r \times r$ matrix. Moreover, A little algebra gives

$$\begin{aligned}
\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k) &= \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D} - \boldsymbol{S}_k) + (\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T \\
&= \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D} - \boldsymbol{S}_k)(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T) + (\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T + \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T \\
&= \widetilde{\boldsymbol{U}}_k\boldsymbol{R}_2^T\boldsymbol{Q}_2^T + \boldsymbol{Q}_1\boldsymbol{R}_1\widetilde{\boldsymbol{V}}_k^T + \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D} - \boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T
\end{aligned}$$

$$= \begin{bmatrix} \widetilde{\boldsymbol{U}}_k & \boldsymbol{Q}_1 \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{U}}_k^T(\boldsymbol{D}-\boldsymbol{S}_k)\widetilde{\boldsymbol{V}}_k & \boldsymbol{R}_2^T \\ \boldsymbol{R}_1 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{V}}_k^T \\ \boldsymbol{Q}_2^T \end{bmatrix}$$
$$:= \begin{bmatrix} \widetilde{\boldsymbol{U}}_k & \boldsymbol{Q}_1 \end{bmatrix} \boldsymbol{M}_k \begin{bmatrix} \widetilde{\boldsymbol{V}}_k^T \\ \boldsymbol{Q}_2^T \end{bmatrix},$$

where the fourth line follows from the fact $\widetilde{\boldsymbol{U}}_k^T\boldsymbol{Q}_1 = \widetilde{\boldsymbol{V}}_k^T\boldsymbol{Q}_2 = \boldsymbol{0}$. Let $\boldsymbol{M}_k = \boldsymbol{U}_{M_k}\boldsymbol{\Sigma}_{M_k}\boldsymbol{V}_{M_k}^T$ be the SVD of $\boldsymbol{M}_k$, which can be computed using $O(r^3)$ flops since $\boldsymbol{M}_k$ is a $2r \times 2r$ matrix. Then the SVD of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k) = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{V}}_k^T$ can be computed by

$$\widetilde{\boldsymbol{U}}_{k+1} = \begin{bmatrix} \widetilde{\boldsymbol{U}}_k & \boldsymbol{Q}_1 \end{bmatrix} \boldsymbol{U}_{M_k}, \quad \widetilde{\boldsymbol{\Sigma}}_{k+1} = \boldsymbol{\Sigma}_{M_k}, \quad \text{and} \quad \widetilde{\boldsymbol{V}}_{k+1} = \begin{bmatrix} \widetilde{\boldsymbol{V}}_k & \boldsymbol{Q}_2 \end{bmatrix} \boldsymbol{V}_{M_k} \tag{9}$$

since both the matrices $\begin{bmatrix} \widetilde{\boldsymbol{U}}_k & \boldsymbol{Q}_1 \end{bmatrix}$ and $\begin{bmatrix} \widetilde{\boldsymbol{V}}_k & \boldsymbol{Q}_2 \end{bmatrix}$ are orthogonal. In summary, the overall computational costs of $\mathcal{H}_r(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k))$ lie in one matrix-matrix subtraction between an $n \times n$ matrix and an $n \times n$ matrix, two matrix-matrix multiplications between an $n \times n$ matrix and an $n \times r$ matrix, the QR decomposition of two $n \times r$ matrices, an SVD of a $2r \times 2r$ matrix, and a few matrix-matrix multiplications between a $r \times n$ matrix and an $n \times r$ matrix or between an $n \times r$ matrix and a $r \times r$ matrix, leading to a total of $4n^2r + n^2 + O(nr^2 + r^3)$ flops. Thus, the dominant per iteration computational complexity of AccAltProj for updating the estimate of $\boldsymbol{L}$ is the same as the novel gradient descent based approach introduced in (Yi et al., 2016). In contrast, computing the best rank-$r$ approximation of a non-structured $n \times n$ matrix $\boldsymbol{D}-\boldsymbol{S}_k$ typically costs $O(n^2r) + n^2$ flops with a large hidden constant in front of $n^2r$.

After $\boldsymbol{L}_{k+1}$ is obtained, following the approach in (Netrapalli et al., 2014), we apply the hard thresholding operator to update the estimate of the sparse matrix,

$$\boldsymbol{S}_{k+1} = \mathcal{T}_{\zeta_{k+1}}(\boldsymbol{D}-\boldsymbol{L}_{k+1}),$$

where the thresholding operator $\mathcal{T}_{\zeta_{k+1}}$ is defined as

$$[\mathcal{T}_{\zeta_{k+1}}\boldsymbol{Z}]_{ij} = \begin{cases} [\boldsymbol{Z}]_{ij} & |[\boldsymbol{Z}]_{ij}| > \zeta_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

for any matrix $\boldsymbol{Z} \in \mathbb{R}^{m\times n}$. Notice that the thresholding value of $\zeta_{k+1}$ in Algorithm 1 is chosen as

$$\zeta_{k+1} = \beta\left(\sigma_{r+1}\left(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)\right) + \gamma^{k+1}\sigma_1\left(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)\right)\right),$$

which relies on a tuning parameter $\beta > 0$, a convergence rate parameter $0 \le \gamma < 1$, and the singular values of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)$. Since we have already obtained all the singular values of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)$ when computing $\boldsymbol{L}_{k+1}$, the extra cost of computing $\zeta_{k+1}$ is very marginal. Therefore, the cost of updating the estimate of $\boldsymbol{S}$ is very low and insensitive to the sparsity of $\boldsymbol{S}$.

In this paper, a good initialization is achieved by two steps of modified AltProj when setting the input rank to $r$, see Algorithm 3. With this initialization scheme, we can construct an initial guess that is sufficiently close to the ground truth and is inside the "basin of attraction" as detailed in the next subsection. Note that the thresholding parameter $\beta_{init}$ used in Algorithm 3 is different from that in Algorithm 1.

**Algorithm 3** Initialization by Two Steps of AltProj

1: **Input:** $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$: matrix to be split; $r$: rank of $\boldsymbol{L}$; $\beta_{init}, \beta$: thresholding parameters.
2: $\boldsymbol{L}_{-1} = \boldsymbol{0}$
3: $\zeta_{-1} = \beta_{init} \cdot \sigma_1^D$
4: $\boldsymbol{S}_{-1} = \mathcal{T}_{\zeta_{-1}}(\boldsymbol{D} - \boldsymbol{L}_{-1})$
5: $\boldsymbol{L}_0 = \mathcal{H}_r(\boldsymbol{D} - \boldsymbol{S}_{-1})$
6: $\zeta_0 = \beta \cdot \sigma_1(\boldsymbol{D} - \boldsymbol{S}_{-1})$
7: $\boldsymbol{S}_0 = \mathcal{T}_{\zeta_0}(\boldsymbol{D} - \boldsymbol{L}_0)$
8: **Output:** $\boldsymbol{L}_0$, $\boldsymbol{S}_0$

### 2.2. Theoretical Guarantee

In this subsection, we present the theoretical recovery guarantee of AccAltProj (Algorithm 1 together with Algorithm 3). The following theorem establishes the local convergence of AccAltProj.

**Theorem 1 (Local Convergence of AccAltProj)** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2. If the initial guesses $\boldsymbol{L}_0$ and $\boldsymbol{S}_0$ obey the following conditions:*

$$\|\boldsymbol{L} - \boldsymbol{L}_0\|_2 \le 8\alpha\mu r\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_0\|_\infty \le \frac{\mu r}{n}\sigma_1^L, \quad \text{and} \quad supp(\boldsymbol{S}_0) \subset \Omega,$$

*then the iterates of Algorithm 1 with parameters $\beta = \frac{\mu r}{2n}$ and $\gamma \in \left(\frac{1}{\sqrt{12}}, 1\right)$ satisfy*

$$\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 \le 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \le \frac{\mu r}{n}\gamma^k\sigma_1^L, \quad \text{and} \quad supp(\boldsymbol{S}_k) \subset \Omega.$$

The next theorem states that the initial guesses obtained from Algorithm 3 fulfill the conditions required in Theorem 1.

**Theorem 2 (Guaranteed Initialization)** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. If the thresholding parameters obey $\frac{\mu r\sigma_1^L}{n\sigma_1^D} \le \beta_{init} \le \frac{3\mu r\sigma_1^L}{n\sigma_1^D}$ and $\beta = \frac{\mu r}{2n}$, then the outputs of Algorithm 3 satisfy*

$$\|\boldsymbol{L} - \boldsymbol{L}_0\|_2 \le 8\alpha\mu r\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_0\|_\infty \le \frac{\mu r}{n}\sigma_1^L, \quad \text{and} \quad supp(\boldsymbol{S}_0) \subset \Omega.$$

The proofs of Theorems 1 and 2 are presented in Section 4. The convergence of AccAltProj follows immediately by combining the above two theorems together.

For conciseness, the main theorems are stated for symmetric matrices. However, similar results can be established for nonsymmetric matrix recovery problems as they can be cast as problems with respect to symmetric augmented matrices, as suggested in (Netrapalli et al., 2014). Without loss of generality, assume $dm \le n < (d+1)m$ for some $d \ge 1$ and construct

$\overline{\boldsymbol{L}}$ and $\overline{\boldsymbol{S}}$ as

$$\overline{\boldsymbol{L}} := \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{L} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{L} \\ \boldsymbol{L}^T & \cdots & \boldsymbol{L}^T & \mathbf{0} \end{bmatrix}}_{d \text{ times}} \Bigg\} d \text{ times}, \qquad \overline{\boldsymbol{S}} := \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{S} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{S} \\ \boldsymbol{S}^T & \cdots & \boldsymbol{S}^T & \mathbf{0} \end{bmatrix}}_{d \text{ times}} \Bigg\} d \text{ times}.$$

Then it is not hard to see that $\overline{\boldsymbol{L}}$ is $O(\mu)$-incoherent, and $\overline{\boldsymbol{S}}$ is $O(\alpha)$-sparse, with the hidden constants being independent of $d$. Moreover, based on the connection between the SVD of the augmented matrix and that of the original one, it can be easily verified that at the $k^{th}$ iteration the estimates returned by AccAltProj with input $\overline{\boldsymbol{D}} = \overline{\boldsymbol{L}} + \overline{\boldsymbol{S}}$ have the form

$$\overline{\boldsymbol{L}}_k = \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{L}_k \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{L}_k \\ \boldsymbol{L}_k^T & \cdots & \boldsymbol{L}_k^T & \mathbf{0} \end{bmatrix}}_{d \text{ times}} \Bigg\} d \text{ times}, \qquad \overline{\boldsymbol{S}}_k = \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{S}_k \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{S}_k \\ \boldsymbol{S}_k^T & \cdots & \boldsymbol{S}_k^T & \mathbf{0} \end{bmatrix}}_{d \text{ times}} \Bigg\} d \text{ times},$$

where $\boldsymbol{L}_k, \boldsymbol{S}_k$ are the the $k^{th}$ estimates returned by AccAltProj with input $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$.

### 2.3. Related Work

As mentioned earlier, convex relaxation based methods for RPCA have higher computational complexity and slower convergence rate which are not applicable for high dimensional problems. In fact, the convergence rate of the algorithm for computing the solution to the SDP formulation of RPCA (Candès et al., 2011; Chandrasekaran et al., 2011; Xu et al., 2010) is sub-linear with the per iteration computational complexity being $O(n^3)$. By contrast, AccAltProj only requires $O(\log(1/\epsilon))$ iterations to achieve an accuracy of $\epsilon$, and the dominant per iteration computational cost is $O(rn^2)$.

There have been many other algorithms which are designed to solve the non-convex RPCA problem directly. In (Wang et al., 2013), an alternating minimization algorithm was proposed for (1) based on the factorization model of low rank matrices. However, only convergence to fixed points was established there. In (Gu et al., 2016), the authors developed an alternating minimization algorithm for RPCA, which allows the sparsity level $\alpha$ to be $O(1/(\mu^{2/3}r^{2/3}n))$ for successful recovery, which is more stringent than our result when $r \ll n$. A projected gradient descent algorithm was proposed in Chen and Wainwright (2015) for the special case of positive semidefinite matrices based on the $\ell_1$-norm of each row of the underlying sparse matrix, which is not very practical.

In Table 1, we compare AccAltProj with the other two competitive non-convex algorithms for RPCA: AltProj from (Netrapalli et al., 2014) and non-convex gradient descent

(GD) from (Yi et al., 2016). GD attempts to reconstruct the low rank matrix by minimizing an objective function which contains the prior knowledge of the sparse matrix. The table displays the computational complexity of each algorithm for updating the estimates of the low rank matrix and the sparse matrix, as well as the convergence rate and the theoretical tolerance for the number of non-zero entries in the sparse matrix.

From the table, we can see that AccAltProj achieves the same linear convergence rate as AltProj, which is faster than GD. Moreover, AccAltProj has the lowest per iteration computational complexity for updating both the estimates of $\boldsymbol{L}$ and $\boldsymbol{S}$ (ties with AltProj for updating the sparse part). It is worth emphasizing that the acceleration stage in AccAltProj which first projects $\boldsymbol{D}-\boldsymbol{S}_k$ onto a low dimensional subspace reduces the computational cost of the SVD in AltProj dramatically. Overall, AccAltProj will be substantially faster than AltProj and GD, as confirmed by our numerical simulations in next section. The table also shows that the theoretical sparsity level that can be tolerated by AccAltProj is lower than that of GD and AltProj. Our result looses an order in $r$ because we have replaced the spectral norm by the Frobenius norm when considering the reduction of the reconstruction error in terms of the spectral norm. In addition, the condition number of the target matrix appears in the theoretical result because the current version of AccAltProj deals with the fixed rank case which requires the initial guess is sufficiently close to the target matrix for the theoretical analysis. Nevertheless, we note that the sufficient condition regarding to $\alpha$ to guarantee the exact recovery of AccAltProj is highly pessimistic when compared with its empirical performance. Numerical investigations in next section show that AccAltProj can tolerate as large $\alpha$ as AltProj does under different energy levels.

Table 1: Comparison of AccAltProj, AltProj and GD.

| Algorithm | AccAltProj | AltProj | GD |
|---|---|---|---|
| Updating $\boldsymbol{S}$ | $\boldsymbol{O\left(n^2\right)}$ | $O\left(rn^2\right)$ | $O\left(n^2+\alpha n^2\log(\alpha n)\right)$ |
| Updating $\boldsymbol{L}$ | $\boldsymbol{O\left(rn^2\right)}$ | $O\left(r^2n^2\right)$ | $\boldsymbol{O\left(rn^2\right)}$ |
| Tolerance of $\alpha$ | $O\left(\frac{1}{\max\{\mu r^2\kappa^3,\mu^{1.5}r^2\kappa,\mu^2r^2\}}\right)$ | $\boldsymbol{O\left(\frac{1}{\mu r}\right)}$ | $O\left(\frac{1}{\max\{\mu r^{1.5}\kappa^{1.5},\mu r\kappa^2\}}\right)$ |
| Iterations needed | $\boldsymbol{O\left(\log\left(\frac{1}{\epsilon}\right)\right)}$ | $\boldsymbol{O\left(\log\left(\frac{1}{\epsilon}\right)\right)}$ | $O\left(\kappa\log(\frac{1}{\epsilon})\right)$ |

## 3. Numerical Expierments

In this section, we present the empirical performance of our AccAltProj algorithm and compare it with the state-of-the-art AltProj algorithm from (Netrapalli et al., 2014) and the leading gradient descent based algorithm (GD) from (Yi et al., 2016). The tests are conducted on a laptop equipped with 64-bit Windows 7, Intel i7-4712HQ (4 Cores at 2.3 GHz) and 16GB DDR3L-1600 RAM, and executed from MATLAB R2017a. We implement AltProj by ourselves, while the codes for GD are downloaded from the author's website[3]. Hand tuned parameters are used for these algorithms to achieve the best performance in the numerical comparison. The codes for AccAltProj can be found online:

3. Website: `www.yixinyang.org/code/RPCA_GD.zip`.

https://github.com/caesarcai/AccAltProj_for_RPCA.

Notice that the computation of an initial guess by Algorithm 3 requires the truncated SVD on a full size matrix. As is typical in the literature, we used the PROPACK library[4] for this task when the size of $\boldsymbol{D}$ is large and $r$ is relatively small. To reduce the dependence of the theoretical result on the condition number of the underlying low rank matrix, AltProj was originally designed to loop $r$ stages for the input rank increasing from 1 to $r$ and each stage contains a few number of iterations for a fixed rank. However, when the condition number is medium large which is the case in our tests, we have observed that AltProj achieves the best computational efficiency when fixing the rank to $r$. Thus, to make fair comparison, we test AltProj when input rank is fixed, the same as the other two algorithms.

**Synthetic Datasets** We follow the setup in (Netrapalli et al., 2014) and (Yi et al., 2016) for the random tests on synthetic data. An $n \times n$ rank $r$ matrix $\boldsymbol{L}$ is formed via $\boldsymbol{L} = \boldsymbol{P}\boldsymbol{Q}^T$, where $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{n\times r}$ are two random matrices having their entries drawn i.i.d from the standard normal distribution. The locations of the non-zero entries of the underlying sparse matrix $\boldsymbol{S}$ are sampled uniformly and independently without replacement, while the values of the non-zero entries are drawn i.i.d from the uniform distribution over the interval $[-c \cdot \mathbb{E}(|[\boldsymbol{L}]_{ij}|), c \cdot \mathbb{E}(|[\boldsymbol{L}]_{ij}|)]$ for some constant $c > 0$. The relative computing error at the $k^{th}$ iteration of a single test is defined as

$$err_k = \frac{\|\boldsymbol{D} - \boldsymbol{L}_k - \boldsymbol{S}_k\|_F}{\|\boldsymbol{D}\|_F}. \tag{11}$$

The test algorithms are terminated when either the relative computing error is smaller than a tolerance, $err_k < tol$, or a maximum number of 100 iterations is reached. Recall that $\mu$ is the incoherence parameter of the low rank matrix $\boldsymbol{L}$ and $\alpha$ is the sparsity parameter of the sparse matrix $\boldsymbol{S}$. In the random tests, we use $1.1\mu$ in AltProj and AccAltProj, and use $1.1\alpha$ in GD.

Though we are only able to provide a theoretical guarantee for AccAltProj with trim in this paper, it can be easily seen that AccAltProj can also be implemented without the trim step. Thus, both AccAltProj with and without trim are tested. The parameters $\beta$ and $\beta_{init}$ are set to be $\beta = \frac{1.1\mu r}{2\sqrt{mn}}$ and $\beta_{init} = \frac{1.1\mu r}{\sqrt{mn}}$ in our experiments, and $\gamma = 0.5$ is used when $\alpha < 0.55$ and $\gamma = 0.65$ is used when $\alpha \geq 0.55$.

We first test the performance of the algorithms under different values of $\alpha$ for fixed $n = 2500$ and $r = 5$. Three different values of $c$ are investigated: $c \in \{0.2, 1, 5\}$, which represent three different signal levels of $\boldsymbol{S}$. For each value of $c$, 10 different values of $\alpha$ from 0.3 to 0.75 are tested. We set $tol = 10^{-6}$ in the stopping condition for all the test algorithms. The backtracking line search has been used in GD which can improve its recovery performance substantially in our tests. An algorithm is considered to have successfully reconstructed $\boldsymbol{L}$ (or equivalently, $\boldsymbol{S}$) if the low rank output of the algorithm $\boldsymbol{L}_k$ satisfies

$$\frac{\|\boldsymbol{L}_k - \boldsymbol{L}\|_F}{\|\boldsymbol{L}\|_F} \leq 10^{-4}.$$

4. Website: sun.stanford.edu/~rmunk/PROPACK.

Table 2: Rate of success for AccAltProj with and without trim, AltProj, and GD for different values of $\alpha$.

| $c = 0.2$ | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 |
|---|---|---|---|---|---|---|---|---|---|---|
| AccAltProj w/ trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 4 | 0 |
| AccAltProj w/o trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 4 | 0 |
| AltProj | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 0 | 0 |
| GD | 10 | 10 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| $c = 1$ | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 |
|---|---|---|---|---|---|---|---|---|---|---|
| AccAltProj w/ trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 0 | 0 |
| AccAltProj w/o trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 0 | 0 |
| AltProj | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 8 | 0 | 0 |
| GD | 10 | 10 | 10 | 10 | 9 | 0 | 0 | 0 | 0 | 0 |

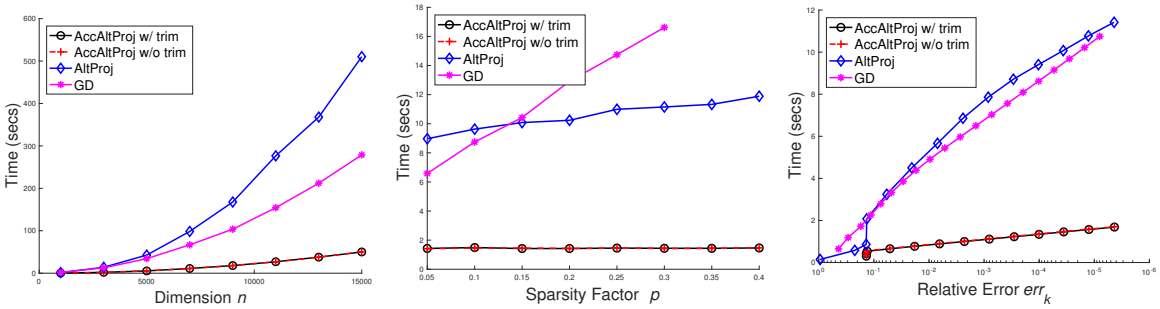
| $c = 5$ | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 |
|---|---|---|---|---|---|---|---|---|---|---|
| AccAltProj w/ trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 5 | 0 | 0 |
| AccAltProj w/o trim | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 5 | 0 | 0 |
| AltProj | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 3 | 0 | 0 |
| GD | 10 | 10 | 10 | 10 | 10 | 10 | 7 | 0 | 0 | 0 |

The number of successful reconstructions for each algorithm out of 10 random tests are presented in Table 2. It is clear that AccAltProj (with and without trim) and AltProj exhibit similar behavior even though the theoretical requirement of AccAltProj with trim is a bit more stringent than that of AltProj, and they can tolerate larger values of $\alpha$ than GD when $c$ is small.

Next, we evaluate the runtime of the test algorithms. The computational results are plotted in Figure 2 together with the setup corresponding to each plot. Figure 2(i) shows that AccAltProj is substantially faster than AltProj and GD. In particular, when $n$ is large, it achieves about $10\times$ speedup. Figure 2(ii) shows that AccAltProj and AltProj are less sensitive to the sparsity of $\boldsymbol{S}$. Notice that we have used a well-tuned fixed stepsize for GD here so that it can achieve the best computational efficiency. Thus, GD fails to converge when $\alpha \geq 0.35$ which is smaller than the largest value of $\alpha$ for successful recovery corresponding to $c = 1$ in Table 2. Lastly, Figure 2(iii) shows the lowest computational time of AccAltProj against the relative computing error.

**Video Background Subtraction** In this section, we compare the performance of AccAltProj with and without trim, AltProj and GD on video background subtraction, a real world benchmark problem for RPCA. The task in background subtraction is to separate moving foreground objects from a static background. The two videos we have used for this test are *Shoppingmall* and *Restaurant* which can be found online[5].The size of each frame of *Shoppingmall* is $256 \times 320$ and that of *Restaurant* is $120 \times 160$. The total number of frames are 1000 and 3055 in *Shoppingmall* and *Restaurant*, respectively. Each video can be represented by a matrix, where each column of the matrix is a vectorized frame of the

5. Website: `perception.i2r.a-star.edu.sg/bk_model/bk_index.html`.




(i) Varying Dimension vs Runtime (ii) Varying Sparsity vs Runtime (iii) Relative Error vs Runtime

Figure 2: Runtime for synthetic datasets: (i) Varying dimension $n$ vs runtime, where $r = 5$, $\alpha = 0.1$, $c = 1$, and $n$ varies from 1000 to 15000. The algorithms are terminated after $err_k < 10^{-4}$ is satisfied. (ii) Varying sparsity factor $\alpha$ vs runtime, where $r = 5$, $c = 1$ and $n = 2500$. The algorithms are terminated when either $err_k < 10^{-4}$ or 100 number of iterations is reached, whichever comes first. (iii) Relative error $err_k$ vs runtime, where $r = 5$, $\alpha = 0.1$, $c = 1$, and $n = 2500$. The algorithms are terminated after $err_k < 10^{-5}$ is satisfied so that we can observe more iterations.

Table 3: Computational results for video background subtraction. Here "S" represents *Shoppingmall*, "R" represents *Restaurant*, and $\mu$ is the incoherence parameter of the output low rank matrices along the time axis (i.e., among different frames).

| | AccAltProj w/ trim | | AccAltProj w/o trim | | AltProj | | GD | |
|---|---|---|---|---|---|---|---|---|
| | runtime | $\mu$ | runtime | $\mu$ | runtime | $\mu$ | runtime | $\mu$ |
| S | 38.98$s$ | 2.12 | 38.79$s$ | 2.26 | 82.97$s$ | 2.13 | 161.1$s$ | 2.85 |
| R | 28.09$s$ | 5.16 | 27.94$s$ | 5.25 | 69.12$s$ | 5.28 | 107.3$s$ | 6.07 |

video. Then, we apply each algorithm to decompose the matrix into a low rank part which represents the static background of the video and a sparse part which represents the moving objects in the video. Since there is no ground truth for the incoherence parameter and the sparsity parameter, their values are estimated by trial-and-error in the tests. We set $\gamma = 0.7$ and $r = 2$ in the decomposition of both videos, and *tol* is set to $10^{-4}$ in the stopping criteria. All the four algorithms can achieve desirable visual performance for the two tested videos and we only present the decomposition results of three selected frames for both AccAltProj with trim and without trim in Figure 3.

Table 3 contains the runtime of each algorithm. We can see that AccAltProj with and without trim are also faster than AltProj and GD for the background subtraction experiments conducted here. We also include the incoherence values of the output low rank matrices along the time axis. It is worth noting that the incoherence parameter value of the low rank output from AccAltProj with trim are smaller than that from AccAltProj without trim, which suggests the output backgrounds from AccAltProj with trim are more consistent through all the frames. Additionally, AccAltProj and AltProj have comparable output incoherence.

Figure 3: Video background subtraction: The top three rows correspond to three different frames from the video *Shoppingmall*, while the bottom three rows are frames from the video *Restaurant*. The first column contains the original frames, the middle two columns are the separated background and foreground outputs of AccAltProj with trim, and the right two columns are the separated background and foreground outputs of AccAltProj without trim.

## 4. Proofs

### 4.1. Proof of Theorem 1

The proof of Theorem 1 follows a route established in (Netrapalli et al., 2014). Despite this, the details of the proof itself are nevertheless quite involved because there are two more operations (i.e., projection onto a tangent space and trim) in AccAltProj than in AltProj. Overall, the proof consists of two steps:

- When $\|\boldsymbol{L}-\boldsymbol{L}_k\|_2$ and $\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty$ are sufficiently small, and $supp(\boldsymbol{S}_k)\subset\Omega$, then $\|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_2$ deceases in some sense by a constant factor (see Lemma 12) and $\|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_\infty$ is small (see Lemma 13).
- When $\|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_\infty$ is sufficiently small, we can choose $\zeta_{k+1}$ such that $supp(\boldsymbol{S}_{k+1})\subset\Omega$ and $\|\boldsymbol{S}-\boldsymbol{S}_{k+1}\|_\infty$ is small (see Lemma 14).

These results will be presented in a set of lemmas. For ease of notation we define $\tau := 4\alpha\mu r\kappa$ and $\upsilon := \tau(48\sqrt{\mu}r\kappa+\mu r)$ in the sequel.

**Lemma 3** ***(Weyl's inequality)*** *Let $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\in\mathbb{R}^{n\times n}$ be the symmetric matrices such that $\boldsymbol{A}=\boldsymbol{B}+\boldsymbol{C}$. Then the inequality*

$$|\sigma_i^A-\sigma_i^B|\le\|\boldsymbol{C}\|_2$$

*holds for all $i$, where $\sigma_i^A$ and $\sigma_i^B$ represent the $i^{th}$ singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively.*

**Proof** This is a well-known result and the proof can be found in many standard textbooks, see for example Bhatia (2013). ■

**Lemma 4** *Let $\boldsymbol{S}\in\mathbb{R}^{n\times n}$ be a symmetric sparse matrix which satisfies Assumption A2. Then, the inequality*

$$\|\boldsymbol{S}\|_2\le\alpha n\|\boldsymbol{S}\|_\infty$$

*holds, where $\alpha$ is the sparsity level of $\boldsymbol{S}$.*

**Proof** The proof can be found in (Netrapalli et al., 2014, Lemma 4). ■

**Lemma 5** *Let* Trim *be the algorithm defined by Algorithm 2. If $\boldsymbol{L}_k\in\mathbb{R}^{n\times n}$ is a rank-$r$ matrix with*

$$\|\boldsymbol{L}_k-\boldsymbol{L}\|_2\le\frac{\sigma_r^L}{20\sqrt{r}},$$

*then the trim output with the level $\sqrt{\frac{\mu r}{n}}$ satisfies*

$$\|\widetilde{\boldsymbol{L}}_k-\boldsymbol{L}\|_F\le 8\kappa\|\boldsymbol{L}_k-\boldsymbol{L}\|_F, \tag{12}$$

$$\max_i\|\boldsymbol{e}_i^T\widetilde{\boldsymbol{U}}_k\|_2\le\frac{10}{9}\sqrt{\frac{\mu r}{n}},\quad\text{and}\quad\max_j\|\boldsymbol{e}_j^T\widetilde{\boldsymbol{V}}_k\|_2\le\frac{10}{9}\sqrt{\frac{\mu r}{n}}, \tag{13}$$

*where* $\widetilde{\boldsymbol{L}}_k = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{V}}_k^T$ *is the SVD of* $\widetilde{\boldsymbol{L}}_k$*. Furthermore, it follows that*

$$\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2 \leq 8\sqrt{2r}\kappa\|\boldsymbol{L}_k - \boldsymbol{L}\|_2. \tag{14}$$

**Proof** Since both $\boldsymbol{L}$ and $\boldsymbol{L}_k$ are rank-$r$ matrices, $\boldsymbol{L}_k - \boldsymbol{L}$ is rank at most $2r$. So

$$\|\boldsymbol{L}_k - \boldsymbol{L}\|_F \leq \sqrt{2r}\|\boldsymbol{L}_k - \boldsymbol{L}\|_2 \leq \sqrt{2r}\frac{\sigma_r^L}{20\sqrt{r}} = \frac{\sigma_r^L}{10\sqrt{2}}.$$

Then, the first two parts of the lemma, i.e., (12) and (13), follow from (Wei et al., 2016a, Lemma 4.10). Noting that $\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2 \leq \|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_F$, (14) follows immediately. ■

**Lemma 6** *Let* $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ *and* $\widetilde{\boldsymbol{L}}_k = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{V}}_k^T$ *be the SVD of two rank-r matrices, then*

$$\|\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^T\|_2 \leq \frac{\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2}{\sigma_r^L}, \qquad \|\boldsymbol{V}\boldsymbol{V}^T - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^T\|_2 \leq \frac{\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2}{\sigma_r^L}, \tag{15}$$

*and*

$$\|(\mathcal{I} - \mathcal{P}_{\widetilde{T}_k})\boldsymbol{L}\|_2 \leq \frac{\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2^2}{\sigma_r^L}. \tag{16}$$

**Proof** The proof of (15) can be found in (Wei et al., 2016b, Lemma 4.2). The Frobenius norm version of (16) can also be found in (Wei et al., 2016a,b). Here we only need to prove the spectral norm version, i.e., (16). Since $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{L}$ and $\widetilde{\boldsymbol{L}}_k(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T) = \boldsymbol{0}$, we have

$$\begin{aligned}
\|(\mathcal{I} - \mathcal{P}_{\widetilde{T}_k})\boldsymbol{L}\|_2 &= \|(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\boldsymbol{L}(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&= \|(\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{L}(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&= \|(\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{L}(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&= \|(\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\boldsymbol{L}(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&= \|(\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{L} - \widetilde{\boldsymbol{L}}_k)(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&\leq \|(\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\|_2\|(\boldsymbol{L} - \widetilde{\boldsymbol{L}}_k)\|_2\|(\boldsymbol{I} - \widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T)\|_2 \\
&\leq \frac{\|\widetilde{\boldsymbol{L}}_k - \boldsymbol{L}\|_2^2}{\sigma_r^L},
\end{aligned}$$

where the last inequality follows from (15). ■

**Lemma 7** *Let* $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ *be a symmetric matrix satisfying Assumption A2. Let* $\widetilde{\boldsymbol{L}}_k \in \mathbb{R}^{n\times n}$ *be a rank-r matrix with* $\frac{100}{81}\mu$*-incoherence. That is,*

$$\max_i \|\boldsymbol{e}_i^T\widetilde{\boldsymbol{U}}_k\|_2 \leq \frac{10}{9}\sqrt{\frac{\mu r}{n}} \quad \text{and} \quad \max_j \|\boldsymbol{e}_j^T\widetilde{\boldsymbol{V}}_k\|_2 \leq \frac{10}{9}\sqrt{\frac{\mu r}{n}},$$

*where* $\widetilde{\boldsymbol{L}} = \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{V}}_k^T$ *is the SVD of* $\widetilde{\boldsymbol{L}}_k$*. If supp*$(\boldsymbol{S}_k) \subset \Omega$*, then*

$$\|\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k)\|_\infty \leq 4\alpha\mu r\|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty. \tag{17}$$

**Proof** By the incoherence assumption of $\widetilde{\boldsymbol{L}}_k$ and the sparsity assumption of $\boldsymbol{S}-\boldsymbol{S}_k$, we have

$$\begin{aligned}
[\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)]_{ab} &= \langle \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k), \boldsymbol{e}_a\boldsymbol{e}_b^T\rangle \\
&= \langle \boldsymbol{S}-\boldsymbol{S}_k, \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{e}_a\boldsymbol{e}_b^T)\rangle \\
&= \langle \boldsymbol{S}-\boldsymbol{S}_k, \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T + \boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\rangle \\
&= \langle (\boldsymbol{S}-\boldsymbol{S}_k)\boldsymbol{e}_b, \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\rangle + \langle \boldsymbol{e}_a^T(\boldsymbol{S}-\boldsymbol{S}_k), \boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\rangle - \langle \boldsymbol{S}-\boldsymbol{S}_k, \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\rangle \\
&\le \|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty \left( \sum_{i|(i,b)\in\Omega} |\boldsymbol{e}_i^T\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a| + \sum_{j|(a,j)\in\Omega} |\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\boldsymbol{e}_j| \right) \\
&\quad + \|\boldsymbol{S}-\boldsymbol{S}_k\|_2 \, \|\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\|_* \\
&\le 2\alpha n\frac{100\mu r}{81n}\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty + \alpha n\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty\|\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T\|_F \\
&\le \frac{200}{81}\alpha\mu r\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty + \alpha n\frac{\mu r}{n}\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty \\
&= 4\alpha\mu r\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty,
\end{aligned}$$

where the first inequality uses Hölder's inequality and the second inequality uses Lemma 4. We also use the fact $\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\boldsymbol{e}_a\boldsymbol{e}_b^T\widetilde{\boldsymbol{V}}_k\widetilde{\boldsymbol{V}}_k^T$ is a rank-1 matrix to bound its nuclear norm. ■

**Lemma 8** *Under the symmetric setting, i.e., $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{V}\boldsymbol{V}^T$ where $\boldsymbol{U}\in\mathbb{R}^{n\times r}$ and $\boldsymbol{V}\in\mathbb{R}^{n\times r}$ are two orthogonal matrices, we have*

$$\|\mathcal{P}_T\boldsymbol{Z}\|_2 \le \sqrt{\frac{4}{3}}\|\boldsymbol{Z}\|_2$$

*for any symmetric matrix $\boldsymbol{Z}\in\mathbb{R}^{n\times n}$. Moreover, the upper bound is tight.*

**Proof** First notice that

$$\mathcal{P}_T\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{U}\boldsymbol{U}^T - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{Z}\boldsymbol{U}\boldsymbol{U}^T$$

is symmetric. Let $\boldsymbol{y}\in\mathbb{R}^n$ be a unit vector such that $\|\mathcal{P}_T\boldsymbol{Z}\|_2 = |\boldsymbol{y}^T(\mathcal{P}_T\boldsymbol{Z})\boldsymbol{y}|$. Denote $\boldsymbol{y}_1 = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{y}$ and $\boldsymbol{y}_2 = (\boldsymbol{I}-\boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{y}$. Then,

$$\begin{aligned}
\|\mathcal{P}_T\boldsymbol{Z}\|_2 &= |\boldsymbol{y}^T(\mathcal{P}_T\boldsymbol{Z})\boldsymbol{y}| \\
&= |\boldsymbol{y}_1^T\boldsymbol{Z}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{Z}\boldsymbol{y}_1 - \boldsymbol{y}_1^T\boldsymbol{Z}\boldsymbol{y}_1| \\
&= |\boldsymbol{y}_1^T\boldsymbol{Z}\boldsymbol{y}_1 + \boldsymbol{y}_1^T\boldsymbol{Z}\boldsymbol{y}_2 + \boldsymbol{y}_2^T\boldsymbol{Z}\boldsymbol{y}_1| \\
&= |\langle \boldsymbol{y}_1\boldsymbol{y}_1^T + \boldsymbol{y}_1\boldsymbol{y}_2^T + \boldsymbol{y}_2\boldsymbol{y}_1^T, \boldsymbol{Z}\rangle| \\
&\le \|\boldsymbol{y}_1\boldsymbol{y}_1^T + \boldsymbol{y}_1\boldsymbol{y}_2^T + \boldsymbol{y}_2\boldsymbol{y}_1^T\|_*\|\boldsymbol{Z}\|_2.
\end{aligned}$$

Let $a = \|\boldsymbol{y}_1\|_2^2$. Since $\boldsymbol{y}_1\perp\boldsymbol{y}_2$, we have $\|\boldsymbol{y}_1\|_2^2 + \|\boldsymbol{y}_2\|_2^2 = 1$, which implies $\|\boldsymbol{y}_2\|_2^2 = 1-a$ and

$$\boldsymbol{y}_1\boldsymbol{y}_1^T + \boldsymbol{y}_1\boldsymbol{y}_2^T + \boldsymbol{y}_2\boldsymbol{y}_1^T = \begin{bmatrix}\boldsymbol{y}_1 & \boldsymbol{y}_2\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{y}_1 & \boldsymbol{y}_2\end{bmatrix}^T$$

$$= \begin{bmatrix} \frac{\boldsymbol{y}_1}{\sqrt{a}} & \frac{\boldsymbol{y}_2}{\sqrt{1-a}} \end{bmatrix} \begin{bmatrix} a & \sqrt{a(1-a)} \\ \sqrt{a(1-a)} & 0 \end{bmatrix} \begin{bmatrix} \frac{\boldsymbol{y}_1}{\sqrt{a}} & \frac{\boldsymbol{y}_2}{\sqrt{1-a}} \end{bmatrix}^T.$$

Since $\begin{bmatrix} \frac{\boldsymbol{y}_1}{\sqrt{a}} & \frac{\boldsymbol{y}_2}{\sqrt{1-a}} \end{bmatrix}$ is an orthogonal matrix, one has

$$\begin{aligned} \|\boldsymbol{y}_1\boldsymbol{y}_1^T + \boldsymbol{y}_1\boldsymbol{y}_2^T + \boldsymbol{y}_2\boldsymbol{y}_1^T\|_* &= \left\| \begin{bmatrix} a & \sqrt{a(1-a)} \\ \sqrt{a(1-a)} & 0 \end{bmatrix} \right\|_* \\ &= \sqrt{a^2 + 4a(1-a)} \\ &= \sqrt{\frac{4}{3} - 3\left(a - \frac{2}{3}\right)^2} \\ &\leq \sqrt{\frac{4}{3}}, \end{aligned}$$

which complete the proof for the upper bound.

To show the tightness of the bound, let $\boldsymbol{U} = \boldsymbol{V} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\boldsymbol{Z} = \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & -1 \end{bmatrix}$. It can be easily verified that $\|\mathcal{P}_T\boldsymbol{Z}\|_2 = \sqrt{\frac{4}{3}}\|\boldsymbol{Z}\|_2$. ■

**Lemma 9** *Let $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ be an orthogonal matrix with $\mu$-incoherence, i.e., $\|\boldsymbol{e}_i^T\boldsymbol{U}\|_2 \leq \sqrt{\frac{\mu r}{n}}$ for all $i$. Then, for any $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$, the inequality*

$$\|\boldsymbol{e}_i^T\boldsymbol{Z}^a\boldsymbol{U}\|_2 \leq \max_l \sqrt{\frac{\mu r}{n}}(\sqrt{n}\|\boldsymbol{e}_l^T\boldsymbol{Z}\|_2)^a$$

*holds for all $i$ and $a \geq 0$.*

**Proof** This proof is done by mathematical induction.
**Base case:** When $a = 0$, $\|\boldsymbol{e}_i^T\boldsymbol{U}\| \leq \sqrt{\frac{\mu r}{n}}$ is satisfied following from the assumption.
**Induction Hypothesis:** $\|\boldsymbol{e}_i^T(\boldsymbol{Z})^a\boldsymbol{U}\|_2 \leq \max_l \sqrt{\frac{\mu r}{n}}(\sqrt{n}\|\boldsymbol{e}_l^T\boldsymbol{Z}\|_2)^a$ for all $i$ at the $a^{th}$ power.
**Induction Step:** We have

$$\begin{aligned} \|\boldsymbol{e}_i^T\boldsymbol{Z}^{a+1}\boldsymbol{U}\|_2^2 &= \|\boldsymbol{e}_i^T\boldsymbol{Z}\boldsymbol{Z}^a\boldsymbol{U}\|_2^2 \\ &= \sum_j \left( \sum_k [\boldsymbol{Z}]_{ik}[\boldsymbol{Z}^a\boldsymbol{U}]_{kj} \right)^2 \\ &= \sum_{k_1k_2} [\boldsymbol{Z}]_{ik_1}[\boldsymbol{Z}]_{ik_2} \sum_j [\boldsymbol{Z}^a\boldsymbol{U}]_{k_1j}[\boldsymbol{Z}^a\boldsymbol{U}]_{k_2j} \\ &= \sum_{k_1k_2} [\boldsymbol{Z}]_{ik_1}[\boldsymbol{Z}]_{ik_2} \langle \boldsymbol{e}_{k_1}^T\boldsymbol{Z}^a\boldsymbol{U}, \boldsymbol{e}_{k_2}^T\boldsymbol{Z}^a\boldsymbol{U} \rangle \\ &\leq \sum_{k_1k_2} |[\boldsymbol{Z}]_{ik_1}[\boldsymbol{Z}]_{ik_2}| \|\boldsymbol{e}_{k_1}^T\boldsymbol{Z}^a\boldsymbol{U}\|_2 \ \|\boldsymbol{e}_{k_2}^T\boldsymbol{Z}^a\boldsymbol{U}\|_2 \end{aligned}$$

$$\begin{aligned}
&\leq \max_l \frac{\mu r}{n}(\sqrt{n}\|\boldsymbol{e}_l^T\boldsymbol{Z}\|_2)^{2a}\sum_{k_1k_2}|[\boldsymbol{Z}]_{ik_1}[\boldsymbol{Z}]_{ik_2}|\\
&\leq \max_l \frac{\mu r}{n}(\sqrt{n}\|\boldsymbol{e}_l^T\boldsymbol{Z}\|_2)^{2a}(\sqrt{n}\|\boldsymbol{e}_i^T\boldsymbol{Z}\|_2)^2\\
&\leq \max_l \frac{\mu r}{n}(\sqrt{n}\|\boldsymbol{e}_l^T\boldsymbol{Z}\|_2)^{2a+2}.
\end{aligned}$$

The proof is complete by taking a square root from both sides. ■

**Lemma 10** *Let $\boldsymbol{L}\in\mathbb{R}^{n\times n}$ and $\boldsymbol{S}\in\mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. Let $\widetilde{\boldsymbol{L}}_k\in\mathbb{R}^{n\times n}$ be the trim output of $\boldsymbol{L}_k$. If*

$$\|\boldsymbol{L}-\boldsymbol{L}_k\|_2\leq 8\alpha\mu r\gamma^k\sigma_1^L,\quad \|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty\leq\frac{\mu r}{n}\gamma^k\sigma_1^L,\text{ and } supp(\boldsymbol{S}_k)\subset\Omega,$$

*then*

$$\|(\mathcal{P}_{\widetilde{T}_k}-\mathcal{I})\boldsymbol{L}+\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)\|_2\leq\tau\gamma^{k+1}\sigma_r^L \tag{18}$$

*and*

$$\max_l\sqrt{n}\|\boldsymbol{e}_l^T[(\mathcal{P}_{\widetilde{T}_k}-\mathcal{I})\boldsymbol{L}+\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)]\|_2\leq\upsilon\gamma^k\sigma_r^L \tag{19}$$

*hold for all $k\geq 0$, provided $1>\gamma\geq 512\tau r\kappa^2+\frac{1}{\sqrt{12}}$. Here recall that $\tau=4\alpha\mu r\kappa$ and $\upsilon=\tau(48\sqrt{\mu}r\kappa+\mu r)$.*

**Proof** For all $k\geq 0$, we get

$$\begin{aligned}
\|(\mathcal{P}_{\widetilde{T}_k}-\mathcal{I})\boldsymbol{L}+\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)\|_2&\leq\|(\mathcal{P}_{\widetilde{T}_k}-\mathcal{I})\boldsymbol{L}\|_2+\|\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)\|_2\\
&\leq\frac{\|\boldsymbol{L}-\widetilde{\boldsymbol{L}}_k\|_2^2}{\sigma_r^L}+\sqrt{\frac{4}{3}}\|\boldsymbol{S}-\boldsymbol{S}_k\|_2\\
&\leq\frac{(8\sqrt{2r}\kappa)^2\|\boldsymbol{L}-\boldsymbol{L}_k\|_2^2}{\sigma_r^L}+\sqrt{\frac{4}{3}}\alpha n\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty\\
&\leq 128\cdot 8\alpha\mu r^2\kappa^3\|\boldsymbol{L}-\boldsymbol{L}_k\|_2+\sqrt{\frac{4}{3}}\alpha n\|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty\\
&\leq\left(512\tau r\kappa^2+\frac{1}{4}\sqrt{\frac{4}{3}}\right)4\alpha\mu r\gamma^k\sigma_1^L\\
&\leq 4\alpha\mu r\gamma^{k+1}\sigma_1^L,\\
&=\tau\gamma^{k+1}\sigma_r^L
\end{aligned}$$

where the second inequality uses Lemma 6 and 8, the third inequality uses Lemma 4 and 5, the fourth inequality follows from $\frac{\|\boldsymbol{L}-\boldsymbol{L}_k\|_2}{\sigma_r^L}\leq 8\alpha\mu r\kappa$, and the last inequality uses the bound of $\gamma$.

To compute the bound of $\max_l\sqrt{n}\|\boldsymbol{e}_l^T[(\mathcal{P}_{\widetilde{T}_k}-\mathcal{I})\boldsymbol{L}+\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)]\|_2$, first note that

$$\max_l\|\boldsymbol{e}_l^T(\mathcal{I}-\mathcal{P}_{\widetilde{T}_k})\boldsymbol{L}\|_2=\max_l\|\boldsymbol{e}_l^T(\boldsymbol{U}\boldsymbol{U}^T-\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)(\boldsymbol{L}-\widetilde{\boldsymbol{L}}_k)(\boldsymbol{I}-\widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\|_2$$

$$\begin{aligned}
&\leq \max_l \|\boldsymbol{e}_l^T(\boldsymbol{U}\boldsymbol{U}^T - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T)\|_2 \|\boldsymbol{L} - \widetilde{\boldsymbol{L}}_k\|_2 \|\boldsymbol{I} - \widetilde{\boldsymbol{U}}_k\widetilde{\boldsymbol{U}}_k^T\|_2 \\
&\leq \left(\frac{19}{9}\sqrt{\frac{\mu r}{n}}\right) \|\boldsymbol{L} - \widetilde{\boldsymbol{L}}_k\|_2,
\end{aligned}$$

where the last inequality follows from the fact $\boldsymbol{L}$ is $\mu$-incoherent and $\widetilde{\boldsymbol{L}}_k$ is $\frac{100}{81}\mu$-incoherent. Hence, for all $k \geq 0$, we have

$$\begin{aligned}
\max_l \sqrt{n}\|\boldsymbol{e}_l^T((\mathcal{P}_{\widetilde{T}_k} - \mathcal{I})\boldsymbol{L} + \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k))\|_2 &\leq \max_l \sqrt{n}\|\boldsymbol{e}_l^T(\mathcal{I} - \mathcal{P}_{\widetilde{T}_k})\boldsymbol{L}\|_2 + \sqrt{n}\|\boldsymbol{e}_l^T\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k)\|_2 \\
&\leq \frac{19\sqrt{n}}{9}\sqrt{\frac{\mu r}{n}}\|\boldsymbol{L} - \widetilde{\boldsymbol{L}}_k\|_2 + n\|\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k)\|_\infty \\
&\leq \frac{19}{9}8\sqrt{2\mu r}r\kappa\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 + 4n\alpha\mu r\|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \\
&\leq 24\sqrt{\mu}r\kappa \cdot 8\alpha\mu r\gamma^k\sigma_1^L + 4n\alpha\mu r \cdot \frac{\mu r}{n}\gamma^k\sigma_1^L \\
&= \upsilon\gamma^k\sigma_r^L,
\end{aligned}$$

where the third inequality uses Lemma 5 and 7. ■

**Lemma 11** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. Let $\widetilde{\boldsymbol{L}}_k \in \mathbb{R}^{n\times n}$ be the trim output of $\boldsymbol{L}_k$. If*

$$\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 \leq 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \leq \frac{\mu r}{n}\gamma^k\sigma_1^L, \text{ and } supp(\boldsymbol{S}_k) \subset \Omega,$$

*then*

$$|\sigma_i^L - |\lambda_i^{(k)}|| \leq \tau\sigma_r^L \tag{20}$$

*and*

$$(1 - 2\tau)\gamma^j\sigma_1^L \leq |\lambda_{r+1}^{(k)}| + \gamma^j|\lambda_1^{(k)}| \leq (1 + 2\tau)\gamma^j\sigma_1^L \tag{21}$$

*hold for all $k \geq 0$ and $j \leq k+1$, provided $1 > \gamma \geq 512\tau r\kappa^2 + \frac{1}{\sqrt{12}}$. Here $|\lambda_i^{(k)}|$ is the $i^{th}$ singular value of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)$.*

**Proof** Since $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$, we have

$$\begin{aligned}
\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k) &= \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{S}_k) \\
&= \boldsymbol{L} + (\mathcal{P}_{\widetilde{T}_k} - \mathcal{I})\boldsymbol{L} + \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k).
\end{aligned}$$

Hence, by Weyl's inequality and (18) in Lemma 10, we can see that

$$\begin{aligned}
|\sigma_i^L - |\lambda_i^{(k)}|| &\leq \|(\mathcal{P}_{\widetilde{T}_k} - \mathcal{I})\boldsymbol{L} + \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k)\|_2 \\
&\leq \tau\gamma^{k+1}\sigma_r^L
\end{aligned}$$

hold for all $i$ and $k \geq 0$. So the first claim is proved since $\gamma < 1$.

Notice that $\boldsymbol{L}$ is a rank-$r$ matrix, which implies $\sigma_{r+1}^L = 0$, so we have

$$\begin{aligned}||\lambda_{r+1}^{(k)}| + \gamma^j|\lambda_1^{(k)}| - \gamma^j\sigma_1^L| &= ||\lambda_{r+1}^{(k)}| - \sigma_{r+1}^L + \gamma^j|\lambda_1^{(k)}| - \gamma^j\sigma_1^L| \\ &\le \tau\gamma^{k+1}\sigma_r^L + \tau\gamma^{j+k+1}\sigma_r^L \\ &\le \left(1+\gamma^{k+1}\right)\tau\gamma^j\sigma_r^L \\ &\le 2\tau\gamma^j\sigma_1^L\end{aligned}$$

for all $j \le k+1$. This completes the proof of the second claim. ■

**Lemma 12** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. Let $\widetilde{\boldsymbol{L}}_k \in \mathbb{R}^{n\times n}$ be the trim output of $\boldsymbol{L}_k$. If*

$$\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 \le 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \le \frac{\mu r}{n}\gamma^k\sigma_1^L, \text{ and } supp(\boldsymbol{S}_k) \subset \Omega,$$

*then we have*

$$\|\boldsymbol{L} - \boldsymbol{L}_{k+1}\|_2 \le 8\alpha\mu r\gamma^{k+1}\sigma_1^L,$$

*provided* $1 > \gamma \ge 512\tau r\kappa^2 + \frac{1}{\sqrt{12}}$.

**Proof** A direct calculation yields

$$\begin{aligned}\|\boldsymbol{L} - \boldsymbol{L}_{k+1}\|_2 &\le \|\boldsymbol{L} - \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)\|_2 + \|\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k) - \boldsymbol{L}_{k+1}\|_2 \\ &\le 2\|\boldsymbol{L} - \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)\|_2 \\ &= 2\|\boldsymbol{L} - \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{S}_k)\|_2 \\ &= 2\|(\mathcal{P}_{\widetilde{T}_k} - \mathcal{I})\boldsymbol{L} + \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S} - \boldsymbol{S}_k)\|_2 \\ &\le 2 \cdot \tau\gamma^{k+1}\sigma_r^L \\ &= 8\alpha\mu r\gamma^{k+1}\sigma_1^L,\end{aligned}$$

where the second inequality follows from the fact $\boldsymbol{L}_{k+1} = \mathcal{H}_r(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k))$ is the best rank-$r$ approximation of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D} - \boldsymbol{S}_k)$, and the last inequality uses (18) in Lemma 10. ■

**Lemma 13** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. Let $\widetilde{\boldsymbol{L}}_k \in \mathbb{R}^{n\times n}$ be the trim output of $\boldsymbol{L}_k$. If*

$$\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 \le 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \le \frac{\mu r}{n}\gamma^k\sigma_1^L, \text{ and } supp(\boldsymbol{S}_k) \subset \Omega,$$

*then we have*

$$\|\boldsymbol{L} - \boldsymbol{L}_{k+1}\|_\infty \le \left(\frac{1}{2} - \tau\right)\frac{\mu r}{n}\gamma^{k+1}\sigma_1^L,$$

*provided* $1 > \gamma \ge \max\{512\tau r\kappa^2 + \frac{1}{\sqrt{12}}, \frac{2\upsilon}{(1-12\tau)(1-\tau-\upsilon)^2}\}$ *and* $\tau < \frac{1}{12}$.

**Proof** Let $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k) = \begin{bmatrix}\boldsymbol{U}_{k+1} & \ddot{\boldsymbol{U}}_{k+1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda} & \boldsymbol{0}\\ \boldsymbol{0} & \ddot{\boldsymbol{\Lambda}}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_{k+1}^T\\ \ddot{\boldsymbol{U}}_{k+1}^T\end{bmatrix} = \boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T$ be its eigenvalue decomposition. We use the lighter notation $\lambda_i$ $(1 \le i \le n)$ for the eigenvalues of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)$ at the $k$-th iteration and assume they are ordered by $|\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_n|$. Moreover, $\boldsymbol{\Lambda}$ has its $r$ largest eigenvalues in magnitude, $\boldsymbol{U}_{k+1}$ contains the first $r$ eigenvectors, and $\ddot{\boldsymbol{U}}_{k+1}$ has the rest. It follows that $\boldsymbol{L}_{k+1} = \mathcal{H}_r(\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)) = \boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T$.

Denote $\boldsymbol{Z} = \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k) - \boldsymbol{L} = (\mathcal{P}_{\widetilde{T}_k} - \mathcal{I})\boldsymbol{L} + \mathcal{P}_{\widetilde{T}_k}(\boldsymbol{S}-\boldsymbol{S}_k)$. Let $\boldsymbol{u}_i$ be the $i^{th}$ eigenvector of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)$. Noting that $(\lambda_i\boldsymbol{I}-\boldsymbol{Z})\boldsymbol{u}_i = \boldsymbol{L}\boldsymbol{u}_i$, we have

$$\boldsymbol{u}_i = \left(\boldsymbol{I}-\frac{\boldsymbol{Z}}{\lambda_i}\right)^{-1}\frac{\boldsymbol{L}}{\lambda_i}\boldsymbol{u}_i = \left(\boldsymbol{I}+\frac{\boldsymbol{Z}}{\lambda_i}+\left(\frac{\boldsymbol{Z}}{\lambda_i}\right)^2+\cdots\right)\frac{\boldsymbol{L}}{\lambda_i}\boldsymbol{u}_i$$

for all $\boldsymbol{u}_i$ with $1 \le i \le r$, where the expansion is valid because

$$\frac{\|\boldsymbol{Z}\|_2}{\lambda_i} \le \frac{\|\boldsymbol{Z}\|_2}{\lambda_r} \le \frac{\tau}{1-\tau} < 1$$

following from (18) in Lemma 10 and (20) in Lemma 11. This implies

$$\begin{aligned}
\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T &= \sum_{i=1}^{r}\boldsymbol{u}_i\lambda_i\boldsymbol{u}_i^T\\
&= \sum_{i=1}^{r}\left(\sum_{a\ge0}\left(\frac{\boldsymbol{Z}}{\lambda_i}\right)^a\frac{\boldsymbol{L}}{\lambda_i}\right)\boldsymbol{u}_i\lambda_i\boldsymbol{u}_i^T\left(\sum_{b\ge0}\left(\frac{\boldsymbol{Z}}{\lambda_i}\right)^b\frac{\boldsymbol{L}}{\lambda_i}\right)^T\\
&= \sum_{a\ge0}\boldsymbol{Z}^a\boldsymbol{L}\sum_{i=1}^{r}\left(\boldsymbol{u}_i\frac{1}{\lambda_i^{a+b+1}}\boldsymbol{u}_i^T\right)\boldsymbol{L}\sum_{b\ge0}\boldsymbol{Z}^b\\
&= \sum_{a,b\ge0}\boldsymbol{Z}^a\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\boldsymbol{Z}^b.
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\|\boldsymbol{L}_{k+1}-\boldsymbol{L}\|_\infty &= \|\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T-\boldsymbol{L}\|_\infty\\
&= \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L}-\boldsymbol{L}+\sum_{a+b>0}\boldsymbol{Z}^a\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\boldsymbol{Z}^b\|_\infty\\
&\le \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L}-\boldsymbol{L}\|_\infty+\sum_{a+b>0}\|\boldsymbol{Z}^a\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\boldsymbol{Z}^b\|_\infty\\
&:= \boldsymbol{Y}_0+\sum_{a+b>0}\boldsymbol{Y}_{ab}.
\end{aligned}$$

We will handle $\boldsymbol{Y}_0$ first. Recall that $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of the symmetric matrix $\boldsymbol{L}$ which obeys $\mu$-incoherence, i.e., $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{V}\boldsymbol{V}^T$ and $\|\boldsymbol{e}_i^T\boldsymbol{U}\boldsymbol{U}^T\|_2 \le \sqrt{\frac{\mu r}{n}}$ for all $i$. So, for each $(i,j)$ entry of $\boldsymbol{Y}_0$, one has

$$\boldsymbol{Y}_0 = \max_{ij}|\boldsymbol{e}_i^T(\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L}-\boldsymbol{L})\boldsymbol{e}_j|$$

$$
\begin{aligned}
&= \max_{ij} |\boldsymbol{e}_i^T \boldsymbol{U}\boldsymbol{U}^T (\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L} - \boldsymbol{L})\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{e}_j| \\
&\le \max_{ij} \|\boldsymbol{e}_i^T \boldsymbol{U}\boldsymbol{U}^T\|_2 \, \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L} - \boldsymbol{L}\|_2 \, \|\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{e}_j\|_2 \\
&\le \frac{\mu r}{n} \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L} - \boldsymbol{L}\|_2,
\end{aligned}
$$

where the second equation follows from the fact $\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{L} = \boldsymbol{L}\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{L}$. Since $\boldsymbol{L} = \boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z}$, there hold

$$
\begin{aligned}
&\|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{L} - \boldsymbol{L}\|_2 \\
&= \|(\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z})\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T(\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z}) - \boldsymbol{L}\|_2 \\
&= \|\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T - \boldsymbol{L} - \boldsymbol{U}_{k+1}\boldsymbol{U}_{k+1}^T\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{U}_{k+1}\boldsymbol{U}_{k+1}^T - \boldsymbol{Z}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_{k+1}^T\boldsymbol{Z}\|_2 \\
&\le \|\boldsymbol{Z} - \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T\|_2 + 2\|\boldsymbol{Z}\|_2 + \frac{\|\boldsymbol{Z}\|_2^2}{|\lambda_r|} \\
&\le \|\ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T\|_2 + 4\|\boldsymbol{Z}\|_2 \\
&\le |\lambda_{r+1}| + 4\|\boldsymbol{Z}\|_2 \\
&\le 5\|\boldsymbol{Z}\|_2 \\
&\le 5\tau\gamma^{k+1}\sigma_1^L,
\end{aligned}
$$

where the fifth inequality uses (18) in Lemma 10, and notice that $\frac{\|\boldsymbol{Z}\|_2}{|\lambda_r|} \le \frac{\tau}{1-\tau} < 1$ since $\tau < \frac{1}{2}$ and $|\lambda_{r+1}| \le \|\boldsymbol{Z}\|_2$ since $\boldsymbol{L}$ is a rank-$r$ matrix. Thus, we have

$$
\boldsymbol{Y}_0 \le \frac{\mu r}{n} 5\tau\gamma^{k+1}\sigma_1^L. \tag{22}
$$

Next, we derive an upper bound for the rest part. Note that

$$
\begin{aligned}
\boldsymbol{Y}_{ab} &= \max_{ij} |\boldsymbol{e}_i^T \boldsymbol{Z}^a \boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\boldsymbol{Z}^b\boldsymbol{e}_j| \\
&= \max_{ij} |(\boldsymbol{e}_i^T \boldsymbol{Z}^a \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}(\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{Z}^b\boldsymbol{e}_j)| \\
&\le \max_{ij} \|\boldsymbol{e}_i^T \boldsymbol{Z}^a \boldsymbol{U}\|_2 \, \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\|_2 \, \|\boldsymbol{U}^T\boldsymbol{Z}^b\boldsymbol{e}_j\|_2 \\
&\le \max_{l} \frac{\mu r}{n} (\sqrt{n}\|\boldsymbol{e}_l^T \boldsymbol{Z}\|_2)^{a+b} \|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\|_2,
\end{aligned}
$$

where the last inequality uses Lemma 9. Furthermore, by using $\boldsymbol{L} = \boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z}$ again, we get

$$
\begin{aligned}
&\|\boldsymbol{L}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{L}\|_2 \\
&= \|(\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z})\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T(\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}\boldsymbol{U}_{k+1}^T + \ddot{\boldsymbol{U}}_{k+1}\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_{k+1}^T - \boldsymbol{Z})\|_2 \\
&= \|\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b-1)}\boldsymbol{U}_{k+1}^T - \boldsymbol{Z}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b)}\boldsymbol{U}_{k+1}^T - \boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b)}\boldsymbol{U}_{k+1}^T\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{U}_{k+1}\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_{k+1}^T\boldsymbol{Z}\|_2 \\
&\le |\lambda_r|^{-(a+b-1)} + |\lambda_r|^{-(a+b)}\|\boldsymbol{Z}\|_2 + |\lambda_r|^{-(a+b)}\|\boldsymbol{Z}\|_2 + |\lambda_r|^{-(a+b+1)}\|\boldsymbol{Z}\|_2^2 \\
&= |\lambda_r|^{-(a+b-1)} \left(1 + \frac{2\|\boldsymbol{Z}\|_2}{|\lambda_r|} + \left(\frac{\|\boldsymbol{Z}\|_2}{|\lambda_r|}\right)^2\right)
\end{aligned}
$$

$$
\begin{aligned}
&= |\lambda_r|^{-(a+b-1)} \left(1 + \frac{\|\boldsymbol{Z}\|_2}{|\lambda_r|}\right)^2 \\
&\leq |\lambda_r|^{-(a+b-1)} \left(\frac{1}{1-\tau}\right)^2 \\
&\leq \left(\frac{1}{1-\tau}\right)^2 \left((1-\tau)\sigma_r^L\right)^{-(a+b-1)},
\end{aligned}
$$

where the second inequality follows from $\frac{\|\boldsymbol{Z}\|_2}{|\lambda_r|} \leq \frac{\tau}{1-\tau}$, and the last inequality follows from Lemma 11. Together with (19) in Lemma 10, we have

$$
\begin{aligned}
\sum_{a+b>0} \boldsymbol{Y}_{ab} &\leq \sum_{a+b>0} \frac{\mu r}{n} \left(\frac{1}{1-\tau}\right)^2 \upsilon\gamma^k \sigma_r^L \left(\frac{\upsilon\gamma^k\sigma_r^L}{(1-\tau)\sigma_r^L}\right)^{a+b-1} \\
&\leq \frac{\mu r}{n} \left(\frac{1}{1-\tau}\right)^2 \upsilon\gamma^k \sigma_1^L \sum_{a+b>0} \left(\frac{\upsilon}{1-\tau}\right)^{a+b-1} \\
&\leq \frac{\mu r}{n} \left(\frac{1}{1-\tau}\right)^2 \upsilon\gamma^k \sigma_1^L \left(\frac{1}{1-\frac{\upsilon}{1-\tau}}\right)^2 \\
&\leq \frac{\mu r}{n} \left(\frac{1}{1-\tau-\upsilon}\right)^2 \upsilon\gamma^k \sigma_1^L. \qquad (23)
\end{aligned}
$$

Finally, combining (22) and (23) together gives

$$
\begin{aligned}
\|\boldsymbol{L}_{k+1} - \boldsymbol{L}\|_\infty &= \boldsymbol{Y}_0 + \sum_{a+b>0} \boldsymbol{Y}_{ab} \\
&\leq \frac{\mu r}{n} 5\tau\gamma^{k+1}\sigma_1^L + \frac{\mu r}{n} \left(\frac{1}{1-\tau-\upsilon}\right)^2 \upsilon\gamma^k\sigma_1^L \\
&\leq \left(\frac{1}{2} - \tau\right) \frac{\mu r}{n} \gamma^{k+1}\sigma_1^L,
\end{aligned}
$$

where the last inequality follows from $\gamma \geq \frac{2\upsilon}{(1-12\tau)(1-\tau-\upsilon)^2}$. ■

**Lemma 14** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be two symmetric matrices satisfying Assumptions A1 and A2, respectively. Let $\widetilde{\boldsymbol{L}}_k \in \mathbb{R}^{n\times n}$ be the trim output of $\boldsymbol{L}_k$. Recall that $\beta = \frac{\mu r}{2n}$. If*

$$
\|\boldsymbol{L} - \boldsymbol{L}_k\|_2 \leq 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S} - \boldsymbol{S}_k\|_\infty \leq \frac{\mu r}{n}\gamma^k\sigma_1^L, \text{ and } supp(\boldsymbol{S}_k) \subset \Omega
$$

*then we have*

$$
supp(\boldsymbol{S}_{k+1}) \subset \Omega \quad and \quad \|\boldsymbol{S} - \boldsymbol{S}_{k+1}\|_\infty \leq \frac{\mu r}{n}\gamma^{k+1}\sigma_1^L,
$$

*provided* $1 > \gamma \geq \max\{512\tau r\kappa^2 + \frac{1}{\sqrt{12}}, \frac{2\upsilon}{(1-12\tau)(1-\tau-\upsilon)^2}\}$ *and* $\tau < \frac{1}{12}$.

**Proof** We first notice that

$$[\boldsymbol{S}_{k+1}]_{ij} = [\mathcal{T}_{\zeta_{k+1}}(\boldsymbol{D}-\boldsymbol{L}_{k+1})]_{ij} = [\mathcal{T}_{\zeta_{k+1}}(\boldsymbol{S}+\boldsymbol{L}-\boldsymbol{L}_{k+1})]_{ij} = \begin{cases} \mathcal{T}_{\zeta_{k+1}}([\boldsymbol{S}+\boldsymbol{L}-\boldsymbol{L}_{k+1}]_{ij}) & (i,j)\in\Omega \\ \mathcal{T}_{\zeta_{k+1}}([\boldsymbol{L}-\boldsymbol{L}_{k+1}]_{ij}) & (i,j)\in\Omega^c \end{cases}.$$

Let $|\lambda_i^{(k)}|$ denote $i^{th}$ singular value of $\mathcal{P}_{\widetilde{T}_k}(\boldsymbol{D}-\boldsymbol{S}_k)$. By Lemmas 11 and 13, we have

$$\begin{aligned} |[\boldsymbol{L}-\boldsymbol{L}_{k+1}]_{ij}| \le \|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_\infty &\le \left(\frac{1}{2}-\tau\right)\frac{\mu r}{n}\gamma^{k+1}\sigma_1^L \\ &\le \left(\frac{1}{2}-\tau\right)\frac{\mu r}{n}\frac{1}{1-2\tau}\left(|\lambda_{r+1}^{(k)}|+\gamma^{k+1}|\lambda_1^{(k)}|\right) \\ &= \zeta_{k+1} \end{aligned}$$

for any entry of $\boldsymbol{L}-\boldsymbol{L}_{k+1}$. Hence, $[\boldsymbol{S}_{k+1}]_{ij}=0$ for all $(i,j)\in\Omega^c$, i.e., $supp(\boldsymbol{S}_{k+1})\subset\Omega$.

Denote $\Omega_{k+1} := supp(\boldsymbol{S}_{k+1}) = \{(i,j) \mid [(\boldsymbol{D}-\boldsymbol{L}_{k+1})]_{ij} > \zeta_k\}$. Then, for any entry of $\boldsymbol{S}-\boldsymbol{S}_{k+1}$, there hold

$$[\boldsymbol{S}-\boldsymbol{S}_{k+1}]_{ij} = \begin{cases} 0 \\ [\boldsymbol{L}_{k+1}-\boldsymbol{L}]_{ij} \\ [\boldsymbol{S}]_{ij} \end{cases} \le \begin{cases} 0 \\ \|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_\infty \\ \|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_\infty+\zeta_{k+1} \end{cases} \le \begin{cases} 0 & (i,j)\in\Omega^c \\ \left(\frac{1}{2}-\tau\right)\frac{\mu r}{n}\gamma^{k+1}\sigma_1^L & (i,j)\in\Omega_{k+1} \\ \frac{\mu r}{n}\gamma^{k+1}\sigma_1^L & (i,j)\in\Omega\backslash\Omega_{k+1}. \end{cases}$$

Here the last step follows from Lemma 11 which implies $\zeta_{k+1} = \frac{\mu r}{2n}(|\lambda_{r+1}^{(k)}| + \gamma^{k+1}|\lambda_1^{(k)}|) \le \left(\frac{1}{2}+\tau\right)\frac{\mu r}{n}\gamma^{k+1}\sigma_1^L$. Therefore, $\|\boldsymbol{S}-\boldsymbol{S}_{k+1}\|_\infty \le \frac{\mu r}{n}\gamma^{k+1}\sigma_1^L$. ■

Now, we have all the ingredients for the proof of Theorem 1.

**Proof** [Proof of Theorem 1] This theorem will be proved by mathematical induction.
**Base Case:** When $k=0$, the base case is satisfied by the assumption on the intialization.
**Induction Step:** Assume we have

$$\|\boldsymbol{L}-\boldsymbol{L}_k\|_2 \le 8\alpha\mu r\gamma^k\sigma_1^L, \quad \|\boldsymbol{S}-\boldsymbol{S}_k\|_\infty \le \frac{\mu r}{n}\gamma^k\sigma_1^L, \quad \text{and} \quad supp(\boldsymbol{S}_k)\subset\Omega$$

at the $k^{th}$ iteration. At the $(k+1)^{th}$ iteration. If follows directly from Lemmas 12 and 14 that

$$\|\boldsymbol{L}-\boldsymbol{L}_{k+1}\|_2 \le 8\alpha\mu r\gamma^{k+1}\sigma_1^L, \quad \|\boldsymbol{S}-\boldsymbol{S}_{k+1}\|_\infty \le \frac{\mu r}{n}\gamma^{k+1}\sigma_1^L \quad \text{and} \quad supp(\boldsymbol{S}_{k+1})\subset\Omega,$$

which completes the proof.

Additionally, notice that we overall require $1>\gamma\ge\max\{512\tau r\kappa^2+\frac{1}{\sqrt{12}}, \frac{2\upsilon}{(1-12\tau)(1-\tau-\upsilon)^2}\}$. By the definition of $\tau$ and $\upsilon$, one can easily see that the lower bound approaches $\frac{1}{\sqrt{12}}$ when the constant hidden in (4) is sufficiently large. Therefore, the theorem can be proved for any $\gamma\in\left(\frac{1}{\sqrt{12}},1\right)$. ■

### 4.2. Proof of Theorem 2

We first present a lemma which is a variant of Lemma 9 and also appears in (Netrapalli et al., 2014, Lemma 5). The lemma can be similarly proved by mathematical induction.

**Lemma 15** *Let $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ be a sparse matrix satisfying Assumption A2. Let $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ be an orthogonal matrix with $\mu$-incoherence, i.e., $\|\boldsymbol{e}_i^T\boldsymbol{U}\|_2 \le \sqrt{\frac{\mu r}{n}}$ for all $i$. Then*

$$\|\boldsymbol{e}_i^T\boldsymbol{S}^a\boldsymbol{U}\|_2 \le \sqrt{\frac{\mu r}{n}}(\alpha n\|\boldsymbol{S}\|_\infty)^a$$

*for all $i$ and $a \ge 0$.*

Though the proposed initialization scheme (i.e., Algorithms 3) basically consists of two steps of AltProj (Netrapalli et al., 2014), we provide an independent proof for Theorem 2 here because we bound the approximation errors of the low rank matrices using the spectral norm rather than the infinity norm. The proof of Theorem 2 follows a similar structure to that of Theorem 1, but without the projection onto a low dimensional tangent space. Instead of first presenting several auxiliary lemmas, we give a single proof by putting all the elements together.

**Proof** [Proof of Theorem 2] The proof can be partitioned into several parts.

(i) Note that $\boldsymbol{L}_{-1} = 0$ and

$$\|\boldsymbol{L} - \boldsymbol{L}_{-1}\|_\infty = \|\boldsymbol{L}\|_\infty = \max_{ij}\left|\boldsymbol{e}_i^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{e}_j\right| \le \max_{ij}\|\boldsymbol{e}_i^T\boldsymbol{U}\|_2\|\boldsymbol{\Sigma}\|_2\|\boldsymbol{U}^T\boldsymbol{e}_j\|_2 \le \frac{\mu r}{n}\sigma_1^L,$$

where the last inequality follows from Assumption A1, i.e., $\boldsymbol{L}$ is $\mu$-incoherent. Thus, with the choice of $\beta_{init} \ge \frac{\mu r\sigma_1^L}{n\sigma_1^D}$, we have

$$\|\boldsymbol{L} - \boldsymbol{L}_{-1}\|_\infty \le \beta_{init}\sigma_1^D = \zeta_{-1}. \tag{24}$$

Since

$$[\boldsymbol{S}_{-1}]_{ij} = [\mathcal{T}_{\zeta_{-1}}(\boldsymbol{S} + \boldsymbol{L} - \boldsymbol{L}_{-1})]_{ij} = \begin{cases}\mathcal{T}_{\zeta_{-1}}([\boldsymbol{S} + \boldsymbol{L} - \boldsymbol{L}_{-1}]_{ij}) & (i,j) \in \Omega \\ \mathcal{T}_{\zeta_{-1}}([\boldsymbol{L} - \boldsymbol{L}_{-1}]_{ij}) & (i,j) \in \Omega^c,\end{cases}$$

it follows that $[\boldsymbol{S}_{-1}]_{ij} = 0$ for all $(i,j) \in \Omega^c$, i.e. $\Omega_{-1} := supp(\boldsymbol{S}_{-1}) \subset \Omega$. Moreover, for any entries of $\boldsymbol{S} - \boldsymbol{S}_{-1}$, we have

$$[\boldsymbol{S} - \boldsymbol{S}_{-1}]_{ij} = \begin{cases}0 \\ [\boldsymbol{L}_{-1} - \boldsymbol{L}]_{ij} \\ [\boldsymbol{S}]_{ij}\end{cases} \le \begin{cases}0 \\ \|\boldsymbol{L} - \boldsymbol{L}_{-1}\|_\infty \\ \|\boldsymbol{L} - \boldsymbol{L}_{-1}\|_\infty + \zeta_{-1}\end{cases} \le \begin{cases}0 & (i,j) \in \Omega^c \\ \frac{\mu r}{n}\sigma_1^L & (i,j) \in \Omega_{-1} \\ \frac{4\mu r}{n}\sigma_1^L & (i,j) \in \Omega\backslash\Omega_{-1}\end{cases},$$

where the last inequality follows from $\beta_{init} \le \frac{3\mu r\sigma_1^L}{n\sigma_1^D}$, so that $\zeta_{-1} \le \frac{3\mu r}{n}\sigma_1^L$. Therefore, if follows that

$$supp(\boldsymbol{S}_{-1}) \subset \Omega \quad \text{and} \quad \|\boldsymbol{S} - \boldsymbol{S}_{-1}\|_\infty \le \frac{4\mu r}{n}\sigma_1^L. \tag{25}$$

By Lemma 4, we also have

$$\|\boldsymbol{S}-\boldsymbol{S}_{-1}\|_2 \le \alpha n\|\boldsymbol{S}-\boldsymbol{S}_{-1}\|_\infty \le 4\alpha\mu r\sigma_1^L.$$

(ii) To bound the approximation error of $\boldsymbol{L}_0$ to $\boldsymbol{L}$ in terms of the spectral norm, note that

$$\begin{aligned}\|\boldsymbol{L}-\boldsymbol{L}_0\|_2 &\le \|\boldsymbol{L}-(\boldsymbol{D}-\boldsymbol{S}_{-1})\|_2 + \|(\boldsymbol{D}-\boldsymbol{S}_{-1})-\boldsymbol{L}_0\|_2\\ &\le 2\|\boldsymbol{L}-(\boldsymbol{D}-\boldsymbol{S}_{-1})\|_2\\ &= 2\|\boldsymbol{L}-(\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{S}_{-1})\|_2\\ &= 2\|\boldsymbol{S}-\boldsymbol{S}_{-1}\|_2,\end{aligned}$$

where the second inequality follows from the fact $\boldsymbol{L}_0 = \mathcal{H}_r(\boldsymbol{D}-\boldsymbol{S}_{-1})$ is the best rank-$r$ approximation of $\boldsymbol{D}-\boldsymbol{S}_{-1}$. It follows immediately that

$$\|\boldsymbol{L}-\boldsymbol{L}_0\|_2 \le 8\alpha\mu r\sigma_1^L. \tag{26}$$

(iii) Since $\boldsymbol{D}=\boldsymbol{L}+\boldsymbol{S}$, we have $\boldsymbol{D}-\boldsymbol{S}_{-1}=\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{S}_{-1}$. Let $\lambda_i$ denotes the $i^{th}$ eigenvalue of $\boldsymbol{D}-\boldsymbol{S}_{-1}$ ordered by $|\lambda_1|\ge|\lambda_2|\ge\cdots\ge|\lambda_n|$. The application of Weyl's inequality together with the bound of $\alpha$ in Assumption A2 implies that

$$|\sigma_i^L - |\lambda_i|| \le \|\boldsymbol{S}-\boldsymbol{S}_{-1}\|_2 \le \frac{\sigma_r^L}{8} \tag{27}$$

holds for all $i$. Consequently, we have

$$\frac{7}{8}\sigma_i^L \le |\lambda_i| \le \frac{9}{8}\sigma_i^L, \qquad \forall 1\le i\le r, \tag{28}$$

$$\frac{\|\boldsymbol{S}-\boldsymbol{S}_{-1}\|_2}{|\lambda_r|} \le \frac{\frac{\sigma_r^L}{8}}{\frac{7\sigma_r^L}{8}} = \frac{1}{7}. \tag{29}$$

Let $\boldsymbol{D}-\boldsymbol{S}_{-1} = [\boldsymbol{U}_0,\ddot{\boldsymbol{U}}_0]\begin{bmatrix}\boldsymbol{\Lambda} & \boldsymbol{0}\\ \boldsymbol{0} & \ddot{\boldsymbol{\Lambda}}\end{bmatrix}[\boldsymbol{U}_0,\ddot{\boldsymbol{U}}_0]^T = \boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T$ be its eigenvalue decomposition, where $\boldsymbol{\Lambda}$ has the $r$ largest eigenvalues in magnitude and $\ddot{\boldsymbol{\Lambda}}$ contains the rest eigenvalues. Also, $\boldsymbol{U}_0$ contains the first $r$ eigenvectors, and $\ddot{\boldsymbol{U}}_0$ has the rest. Notice that $\boldsymbol{L}_0 = \mathcal{H}_r(\boldsymbol{D}-\boldsymbol{S}_{-1}) = \boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T$ due to the symmetric setting. Denote $\boldsymbol{E} = \boldsymbol{D}-\boldsymbol{S_{-1}}-\boldsymbol{L} = \boldsymbol{S}-\boldsymbol{S}_{-1}$. Let $\boldsymbol{u}_i$ be the $i^{th}$ eigenvector of $\boldsymbol{D}-\boldsymbol{S}_{-1} = \boldsymbol{L}+\boldsymbol{E}$. For $1\le i\le r$, since $(\boldsymbol{L}+\boldsymbol{E})\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i$, we have

$$\boldsymbol{u}_i = \left(\boldsymbol{I}-\frac{\boldsymbol{E}}{\lambda_i}\right)^{-1}\frac{\boldsymbol{L}}{\lambda_i}\boldsymbol{u}_i = \left(\boldsymbol{I}+\frac{\boldsymbol{E}}{\lambda_i}+\left(\frac{\boldsymbol{E}}{\lambda_i}\right)^2+\cdots\right)\frac{\boldsymbol{L}}{\lambda_i}\boldsymbol{u}_i$$

for each $\boldsymbol{u}_i$, where the expansion in the last equality is valid because $\frac{\|\boldsymbol{E}\|_2}{|\lambda_i|} \le \frac{1}{7}$ for all $1\le i\le r$ following from (29). Therefore,

$$\|\boldsymbol{L}_0-\boldsymbol{L}\|_\infty = \|\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T-\boldsymbol{L}\|_\infty$$

$$
\begin{aligned}
&= \|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L} + \sum_{a+b>0} \boldsymbol{E}^a\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\boldsymbol{E}^b\|_\infty \\
&\le \|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L}\|_\infty + \sum_{a+b>0} \|\boldsymbol{E}^a\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\boldsymbol{E}^b\|_\infty \\
&:= \boldsymbol{Y}_0 + \sum_{a+b>0} \boldsymbol{Y}_{ab}.
\end{aligned}
$$

We will handle $\boldsymbol{Y}_0$ first. Recall that $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of the symmetric matrix $\boldsymbol{L}$ which is $\mu$-incoherence, i.e., $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{V}\boldsymbol{V}^T$ and $\|\boldsymbol{e}_i^T\boldsymbol{U}\boldsymbol{U}^T\|_2 \le \sqrt{\frac{\mu r}{n}}$ for all $i$. For each $(i,j)$ entry of $\boldsymbol{Y}_0$, we have

$$
\begin{aligned}
\boldsymbol{Y}_0 &= \max_{ij} |\boldsymbol{e}_i^T(\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L})\boldsymbol{e}_j| \\
&= \max_{ij} |\boldsymbol{e}_i^T\boldsymbol{U}\boldsymbol{U}^T(\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L})\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{e}_j| \\
&\le \max_{ij} \|\boldsymbol{e}_i^T\boldsymbol{U}\boldsymbol{U}^T\|_2 \, \|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L}\|_2 \, \|\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{e}_j\|_2 \\
&\le \frac{\mu r}{n}\|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L}\|_2,
\end{aligned}
$$

where the second equation follows from the fact $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{L} = \boldsymbol{L}\boldsymbol{U}\boldsymbol{U}^T$. Since $\boldsymbol{L} = \boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E}$,

$$
\begin{aligned}
&\|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{L} - \boldsymbol{L}\|_2 \\
&= \|(\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E})\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T(\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E}) - \boldsymbol{L}\|_2 \\
&= \|\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T - \boldsymbol{L} - \boldsymbol{U}_0\boldsymbol{U}_0^T\boldsymbol{E} - \boldsymbol{E}\boldsymbol{U}_0\boldsymbol{U}_0^T - \boldsymbol{E}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-1}\boldsymbol{U}_0^T\boldsymbol{E}\|_2 \\
&\le \|\boldsymbol{E} - \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T\|_2 + 2\|\boldsymbol{E}\|_2 + \frac{\|\boldsymbol{E}\|_2^2}{|\lambda_r|} \\
&\le \|\ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T\|_2 + 4\|\boldsymbol{E}\|_2 \\
&\le |\lambda_{r+1}| + 4\|\boldsymbol{E}\|_2 \\
&\le 5\|\boldsymbol{E}\|_2,
\end{aligned}
$$

where the first and fourth inequality follow from (27) and (29), and $|\lambda_{r+1}| \le \|\boldsymbol{E}\|_2$ since $\sigma_{r+1}^L = 0$. Together, we have

$$
\boldsymbol{Y}_0 \le \frac{5\mu r}{n}\|\boldsymbol{E}\|_2 \le 5\alpha\mu r\|\boldsymbol{E}\|_\infty, \tag{30}
$$

where the last inequality follows from Lemma 4.

Next, we will find an upper bound for the rest part. Note that

$$
\begin{aligned}
\boldsymbol{Y}_{ab} &= \max_{ij} |\boldsymbol{e}_i^T\boldsymbol{E}^a\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\boldsymbol{E}^b\boldsymbol{e}_j| \\
&= \max_{ij} |(\boldsymbol{e}_i^T\boldsymbol{E}^a\boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}(\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{E}^b\boldsymbol{e}_j)| \\
&\le \max_{ij} \|\boldsymbol{e}_i^T\boldsymbol{E}^a\boldsymbol{U}\|_2 \, \|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\|_2 \, \|\boldsymbol{U}^T\boldsymbol{E}^b\boldsymbol{e}_j\|_2
\end{aligned}
$$

$$\leq \frac{\mu r}{n}(\alpha n\|\boldsymbol{E}\|_\infty)^{a+b}\|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\|_2$$
$$\leq \alpha\mu r\|\boldsymbol{E}\|_\infty\left(\frac{\sigma_r^L}{8}\right)^{a+b-1}\|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\|_2,$$

where the second inequality uses Lemma 15. Furthermore, by using $\boldsymbol{L} = \boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E}$ again, we have

$$\begin{aligned}
&\|\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{L}\|_2\\
&= \|(\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E})\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T(\boldsymbol{U}_0\boldsymbol{\Lambda}\boldsymbol{U}_0^T + \ddot{\boldsymbol{U}}_0\ddot{\boldsymbol{\Lambda}}\ddot{\boldsymbol{U}}_0^T - \boldsymbol{E})\|_2\\
&= \|\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b-1)}\boldsymbol{U}_0^T - \boldsymbol{E}\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b)}\boldsymbol{U}_0^T - \boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b)}\boldsymbol{U}_0^T\boldsymbol{E} + \boldsymbol{E}\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{\Lambda}^{-(a+b+1)}\boldsymbol{U}_0^T\boldsymbol{E}\|_2\\
&\leq |\lambda_r|^{-(a+b-1)} + |\lambda_r|^{-(a+b)}\|\boldsymbol{E}\|_2 + |\lambda_r|^{-(a+b)}\|\boldsymbol{E}\|_2 + |\lambda_r|^{-(a+b+1)}\|\boldsymbol{E}\|_2^2\\
&= |\lambda_r|^{-(a+b-1)}\left(1 + \frac{2\|\boldsymbol{E}\|_2}{|\lambda_r|} + \left(\frac{\|\boldsymbol{E}\|_2}{|\lambda_r|}\right)^2\right)\\
&= |\lambda_r|^{-(a+b-1)}\left(1 + \frac{\|\boldsymbol{E}\|_2}{|\lambda_r|}\right)^2\\
&\leq 2|\lambda_r|^{-(a+b-1)}\\
&\leq 2\left(\frac{7}{8}\sigma_r^L\right)^{-(a+b-1)},
\end{aligned}$$

where the second inequality follows from (29) and the last inequality follows from (28). Together, we have

$$\begin{aligned}
\sum_{a+b>0}\boldsymbol{Y}_{ab} &\leq \sum_{a+b>0} 2\alpha\mu r\|\boldsymbol{E}\|_\infty\left(\frac{\frac{1}{8}\sigma_r^L}{\frac{7}{8}\sigma_r^L}\right)^{a+b-1}\\
&\leq 2\alpha\mu r\|\boldsymbol{E}\|_\infty\sum_{a+b>0}\left(\frac{1}{7}\right)^{a+b-1}\\
&\leq 2\alpha\mu r\|\boldsymbol{E}\|_\infty\left(\frac{1}{1-\frac{1}{7}}\right)^2\\
&\leq 3\alpha\mu r\|\boldsymbol{E}\|_\infty. \qquad (31)
\end{aligned}$$

Finally, combining (30)) and (31)) together yields

$$\begin{aligned}
\|\boldsymbol{L}_0 - \boldsymbol{L}\|_\infty &= \boldsymbol{Y}_0 + \sum_{a+b>0}\boldsymbol{Y}_{ab}\\
&\leq 5\alpha\mu r\|\boldsymbol{E}\|_\infty + 3\alpha\mu r\|\boldsymbol{E}\|_\infty\\
&\leq \frac{\mu r}{4n}\sigma_1^L, \qquad (32)
\end{aligned}$$

where the last step uses (25) and the bound of $\alpha$ in Assumption A2.

(iv) From the thresholding rule, we know that

$$[\boldsymbol{S}_0]_{ij} = [\mathcal{T}_{\zeta_0}(\boldsymbol{S}+\boldsymbol{L}-\boldsymbol{L}_0)]_{ij} = \begin{cases} \mathcal{T}_{\zeta_0}([\boldsymbol{S}+\boldsymbol{L}-\boldsymbol{L}_0]_{ij}) & (i,j)\in\Omega \\ \mathcal{T}_{\zeta_0}([\boldsymbol{L}-\boldsymbol{L}_0]_{ij}) & (i,j)\in\Omega^c \end{cases}.$$

So (28), (32) and $\zeta_0 = \frac{\mu r}{2n}\lambda_1$ imply $[\boldsymbol{S}_0]_{ij} = 0$ for all $(i,j)\in\Omega^c$, i.e., $supp(\boldsymbol{S}_0) := \Omega_0 \subset \Omega$. Also, for any entries of $\boldsymbol{S}-\boldsymbol{S}_0$, there hold

$$[\boldsymbol{S}-\boldsymbol{S}_0]_{ij} = \begin{cases} 0 \\ [\boldsymbol{L}_0-\boldsymbol{L}]_{ij} \\ [\boldsymbol{S}]_{ij} \end{cases} \le \begin{cases} 0 \\ \|\boldsymbol{L}-\boldsymbol{L}_0\|_\infty \\ \|\boldsymbol{L}-\boldsymbol{L}_0\|_\infty + \zeta_0 \end{cases} \le \begin{cases} 0 & (i,j)\in\Omega^c \\ \frac{\mu r}{4n}\sigma_1^L & (i,j)\in\Omega_0 \\ \frac{\mu r}{n}\sigma_1^L & (i,j)\in\Omega\backslash\Omega_0. \end{cases}$$

Here the last inequality follows from (28) which implies $\zeta_0 = \frac{\mu r}{2n}\lambda_1 \le \frac{3\mu r}{4n}\sigma_1^L$. Therefore, we have

$$supp(\boldsymbol{S}_0)\subset\Omega \quad \text{and} \quad \|\boldsymbol{S}-\boldsymbol{S}_0\|_\infty \le \frac{\mu r}{n}\sigma_1^L.$$

The proof is compete by noting (26) and the above results. ■

## 5. Discussion and Future Direction

We have presented a highly efficient algorithm AccAltProj for robust principal component analysis. The algorithm is developed by introducing a novel subspace projection step before the SVD truncation, which reduces the per iteration computational complexity of the algorithm of alternating projections significantly. Theoretical recovery guarantee has been established for the new algorithm, while numerical simulations show that our algorithm is superior to other state-of-the-art algorithms.

There are three lines of research for future work. Firstly, the theoretical number of the non-zero entries in a sparse matrix below which AccAltProj can achieve successful recovery is highly pessimistic compared with our numerical findings. This suggests the possibility of improving the theoretical result. Secondly, recovery stability of the proposed algorithm to additive noise will be investigated in the future. Finally, this paper focuses on the fully observed setting. The proposed algorithm might be similarly extended to the partially observed setting where only partial entries of a matrix are observed. It is also interesting to study the recovery guarantee of the proposed algorithm under this partial observed setting.

## Acknowledgments

This work was supported in part by grants HKRGC 16306317, NSFC 11801088, and Shanghai Sailing Program 18YF1401600.

## References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon. Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137, 2014.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Jian-Feng Cai, Haixia Liu, and Yang Wang. Fast rank one alternating minimization algorithm for phase retrieval. *arXiv preprint arXiv:1708.08751*, 2017.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

Tony F Chan and Chiu-Kwong Wong. Convergence of the alternating minimization algorithm for blind deconvolution. *Linear Algebra and its Applications*, 316(1-3):259–285, 2000.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2): 572–596, 2011.

Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in neural information processing systems*, pages 2204–2212, 2012.

Quanquan Gu, Zhaoran Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In *Artificial Intelligence and Statistics*, pages 600–609, 2016.

Moritz Hardt. On the provable convergence of alternating minimization for matrix completion. *arXiv preprint arXiv:1312.0925*, 2013.

Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 57–60. IEEE, 2012.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

Raghunandan Hulikal Keshavan et al. *Efficient algorithms for collaborative filtering*. PhD thesis, Stanford University, 2012.

Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

Bamdev Mishra and Rodolphe Sepulchre. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1137–1142. IEEE, 2014.

Bamdev Mishra, K Adithya Apuroop, and Rodolphe Sepulchre. A Riemannian geometry for low-rank matrix completion. *arXiv preprint arXiv:1211.1550*, 2012.

Bamdev Mishra, Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3-4):591–621, 2014.

Hossein Mobahi, Zihan Zhou, Allen Y Yang, and Yi Ma. Holistic 3D reconstruction of urban structures from low-rank textures. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 593–600. IEEE, 2011.

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems*, pages 2796–2804, 2013.

Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

Thanh Ngo and Yousef Saad. Scaled gradients on Grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, pages 1412–1420, 2012.

Joseph A O'Sullivan and Jasenka Benac. Alternating minimization algorithms for transmission tomography. *IEEE Transactions on Medical Imaging*, 26(3):283–297, 2007.

Steven W Peters and Robert W Heath. Interference alignment via alternating minimization. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 2445–2448. IEEE, 2009.

Ye Pu, Melanie N Zeilinger, and Colin N Jones. Complexity certification of the fast alternating minimization algorithm for linear MPC. *IEEE Transactions on Automatic Control*, 62(2):888–893, 2017.

Jared Tanner and Ke Wei. Low rank matrix completion by alternating steepest descent methods. *Applied and Computational Harmonic Analysis*, 40(2):417–429, 2016.

Yvon Tharrault, Gilles Mourot, José Ragot, and Didier Maquin. Fault detection and isolation with robust principal component analysis. *International Journal of Applied Mathematics and Computer Science*, 18(4):429–442, 2008.

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

Yi Wang, Arthur Szlam, and Gilad Lerman. Robust locally linear analysis with applications to image denosing and blind inpainting. *SIAM Journal on Imaging Sciences*, 6(1):526–562, 2013.

Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix completion. *arXiv preprint arXiv:1603.06610*, 2016a.

Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1198–1222, 2016b.

John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in neural information processing systems*, pages 4152–4160, 2016.

Xianghao Yu, Juei-Chin Shen, Jun Zhang, and Khaled B Letaief. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems. *IEEE Journal of Selected Topics in Signal Processing*, 10(3):485–500, 2016.

Teng Zhang. Phase retrieval using alternating minimization in a batch setting. *arXiv preprint arXiv:1706.08167*, 2017.