Correction of Corrupted Columns through Fast Robust Hankel Matrix Completion

Shuai Zhang, *Student Member, IEEE*, Meng Wang, *Member, IEEE*

***Abstract*—This paper studies the robust matrix completion (RMC) problem with the objective to recover a low-rank matrix from partial observations that may contain significant errors. If all the observations in one column are erroneous, existing RMC methods can locate the corrupted column at best but cannot recover the actual data in that column. Low-rank Hankel matrices characterize the additional correlations among columns besides the low-rankness and exist in power system monitoring, magnetic resonance imaging (MRI) imaging, and array signal processing. Exploiting the low-rank Hankel property, this paper develops an alternating-projection-based fast algorithm to solve the nonconvex RMC problem. The algorithm converges to the ground-truth low-rank matrix with a linear rate even when all the measurements in a constant fraction of columns are corrupted. The required number of observations is significantly less than the existing bounds for the conventional RMC. Numerical results are reported to evaluate the proposed algorithm.**

***Index Terms*—matrix completion, low-rank Hankel matrix, matrix decomposition, non-convex method**

I. Introduction

Robust matrix completion (RMC) [5] aims to recover a low-rank matrix $\boldsymbol{X}^*$ in $\mathbb{C}^{n_c\times n}$ ($n_c \le n$) from partial observations of measurements $\boldsymbol{M} = \boldsymbol{X}^* + \boldsymbol{S}^*$, where the sparse matrix $\boldsymbol{S}^*$ in $\mathbb{C}^{n_c\times n}$ represents arbitrary errors. Due to the wide existence of low-rank matrices, RMC finds applications in areas like video surveillance [28], face recognition [2], MRI image processing [26], network traffic analysis [22], and power systems [10]. For instance, each row of $\boldsymbol{X}^*$ represents the measurements from one phasor measurement unit (PMU) in power systems, and each column corresponds to the time-synchronized measurements from multiple PMUs [11]. $\boldsymbol{S}^*$ represents the bad measurements.

Let $\widehat{\Omega} \subseteq \{1,\cdots,n_c\}\times\{1,\cdots,n\}$ contain the indices of the observed entries. If $\boldsymbol{X}^*$ is at most rank r and $\boldsymbol{S}^*$ contains at most s nonzero entries, RMC can be formulated as a nonconvex optimization problem,

$$\begin{aligned}\min_{\boldsymbol{X},\boldsymbol{S}\in\mathbb{C}^{n_c\times n}} &\sum_{(i,j)\in\widehat{\Omega}} |M_{i,j}-X_{i,j}-S_{i,j}|^2 \\ \text{s.t.}\quad &\text{rank}(\boldsymbol{X})\le r \quad\text{and}\quad \|\boldsymbol{S}\|_0\le s,\end{aligned} \tag{1}$$

where $\|\boldsymbol{S}\|_0$ measures the number of nonzero entries in $\boldsymbol{S}$. If all the entries are observed, i.e., $\widehat{\Omega}$ contains all the indices, (1) reduces to the robust principal component analysis (RPCA)[1] problem, which decomposes a low-rank matrix and a sparse matrix from their sum. If $\boldsymbol{S}^*$ is a zero matrix, (1) reduces to the low-rank matrix completion problem.

One line of research is to relax the nonconvex rank and ℓ_0-norm terms in (1) into the corresponding approximated convex nuclear norm and ℓ_1-norm. Under mild assumptions, $\boldsymbol{X}^*$ and $\boldsymbol{S}^*$ are indeed the solution to the convex relaxation (see e.g., [5], [18] for RMC and [2], [7], [14] for RPCA). Since the convex relaxation is still time-consuming to solve, fast algorithms based on alternating minimization or gradient descent are developed recently to solve the nonconvex problem directly, for example [12], [19] for RMC and [6], [23], [32] for RPCA. These approaches are more computationally efficient than the convex alternatives.

If the fraction of nonzeros in each column and row of $\boldsymbol{S}^*$ is at most $\Theta(\frac{1}{r})$, then both the convex method in [5] and the nonconvex method in [9] are proven to be able to recover $\boldsymbol{X}^*$ successfully[2]. If all the observations in a column are corrupted, however, with the prior assumption that the matrix is low-rank, one can locate the corrupted column at best but cannot recover the actual values in either RPCA [29] or RMC [7]. Since every column is a data point in the r-dimensional column subspace, even if the column subspace is correctly identified, at least r linearly independent equations, i.e., r entries of each column, are needed to determine the exact values of that column.

In many applications, the low-rank matrix has the additional low-rank Hankel property. For instance, if $\boldsymbol{X}^*$ contains the time series of n_c output channels in a dynamical system, then the Hankel matrix of $\boldsymbol{X}^*$ is approximately low-rank, provided that the dynamical system can be approximated by a reduced-order linear system. As demonstrated in [33], the Hankel matrix of the spatial-temporal blocks of PMU data in power systems is low-rank. In array signal processing, the Hankel matrix of a spectrally sparse signal is low-rank [1], [8], [30], and the rank depends on the number of sinusoidal components. The low-rank Hankel property also holds for a class of finite rate of innovation (FRI) signals, which are motivated by MRI imaging [13], [16], [24], [25], [31].

The low-rank Hankel property has been exploited for data recovery and error correction. Refs. [1], [8], [33] studied the low-rank Hankel matrix completion problem from missing data and proved analytically that the required number of measurements by their respective approaches are significantly

The authors are with the Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, 12180. Email: {zhangs21, wangm7}@rpi.edu.

[1]Preliminary results of the paper about RPCA will appear in IEEE International Symposium on Information Theory (ISIT), 2018 [34]. The papar builds upon [34] and extends it to RMC.

[2]$f(n)=O(g(n))$ means that if for some constant $C>0$, $f(n)\le Cg(n)$ holds when n is sufficiently large. $f(n)=\Theta(g(n))$ means that for some constants $C_1>0$ and $C_2>0$, $C_1g(n)\le f(n)\le C_2g(n)$ holds when n is sufficiently large.

smaller than that needed to recover a general low-rank matrix. Error correction by exploiting the low-rank Hankel structure has been exploited in RPCA [17] and RMC [8]. Ref. [8] provides the analytical guarantee of low-rank Hankel matrix recovery from randomly located data losses and corruptions. No analytical guarantee is provided for fully corrupted channels in [8]. Moreover, the recovery approach in [8] requires solving Semidefinite Programming (SDP), which is computationally expensive in large datasets.

This paper solves the RMC problem of a low-rank Hankel matrix. Extending from the methods in [9], [23], this paper develops an alternation-projection-based algorithm, and the iterates are proved to converge to the ground-truth data matrix linearly with a complexity of $O\big(r^2 n_c n \log(n)\log(1/\varepsilon)\big)$, where ε is the recovery error of $\boldsymbol{X}^*$. The computational cost is significantly smaller than the approach in [8]. The required number of observations for the successful recovery is $O(\mu^2 r^3 \log^2(n)\log(1/\varepsilon))$, where μ is the incoherence of the corresponding Hankel matrix. This number is significantly smaller than the existing bound of $O(rn\log^2(n))$ for recovering a general rank-r matrix [27].

Our data model follows the multi-channel Hankel matrix studied in [33], which models multiple signals with common sinusoidal components. The multi-channel Hankel matrix is different from the single-channel Hankel matrix studied in [1], [8], where the recovery of only one spectrally sparse signal is considered. Our work provides the first algorithmic development with the theoretical performance guarantee for multi-channel low-rank Hankel matrix recovery from corrupted measurements. Our method can tolerate up to $\Theta(\frac{1}{r})$ fraction of corruptions per row and does not have any constraint on the number of corruptions per column. In fact, our method can recover $\boldsymbol{X}^*$ accurately even if $\boldsymbol{S}^*$ contains a constant fraction of fully corrupted columns. Full corrupted columns happen in many applications. For example, simultaneous bad data across all channels can happen due to device malfunctions, communication errors, or cyber data attacks in power systems.

The rest of the paper is organized as follows. Sections II and III introduce the problem formulation and discuss the related work. Sections IV and V describe the proposed algorithm and the theoretical performance guarantee. Section VI shows the numerical results. Section VII concludes the paper.

Notation: Vectors are bold lowercase, matrices are bold uppercase, and scalars are in normal font. For instant, $\boldsymbol{Z}$ is a matrix, and $\boldsymbol{z}$ is vector. $\boldsymbol{Z}_{i*}$ denotes the i-th row of $\boldsymbol{Z}$, and Z_{ij} denotes the (i,j)-th entry of $\boldsymbol{Z}$. $\boldsymbol{I}$ and $\boldsymbol{e}_i$ denote the identity matrix and the i-th standard basis vector. $\boldsymbol{Z}^T$ and $\boldsymbol{Z}^H$ denote the transpose and conjugate transpose of $\boldsymbol{Z}$, so do $\boldsymbol{z}^T$ and $\boldsymbol{z}^H$. The inner product between two vectors is $\langle \boldsymbol{z}_1, \boldsymbol{z}_2\rangle = \boldsymbol{z}_2^H \boldsymbol{z}_1$, and corresponding ℓ_2 norm is $\|\boldsymbol{z}\|_2 = \langle \boldsymbol{z}, \boldsymbol{z}\rangle^{1/2}$. For matrices, the inner product is defined as $\langle \boldsymbol{Z}_1, \boldsymbol{Z}_2\rangle = \mathrm{Tr}(\boldsymbol{Z}_2^H \boldsymbol{Z}_1)$. $\|\boldsymbol{Z}\|_F$ stands for the Frobenius norm with $\|\boldsymbol{Z}\|_F = \langle \boldsymbol{Z}, \boldsymbol{Z}\rangle^{1/2}$. The spectral norm of matrix $\boldsymbol{Z}$ is denoted by $\|\boldsymbol{Z}\|_2$. The maximum entry (in absolute value) of $\boldsymbol{Z}$ is denoted as $\|\boldsymbol{Z}\|_\infty$. In addition, we use $\sigma_i(\boldsymbol{Z}_1)$ to denote the i-th largest singular value of $\boldsymbol{Z}_1$, and $\lambda_i(\boldsymbol{Z}_2)$ to denote the i-th largest eigenvalue (in absolute value) of a symmetric matrix $\boldsymbol{Z}_2$. Linear operators on matrix spaces will be denoted by calligraphic letters. In particular, $\mathcal{I}$ is the identity operator.

II. Problem formulation

Let $\boldsymbol{X}^* = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{C}^{n_c \times n}$ denote the actual data. We define a linear operator $\mathcal{H}_{n_1} : \mathbb{C}^{n_c \times n} \to \mathbb{C}^{n_c n_1 \times n_2}$ that maps a matrix into its corresponding Hankel matrix, i.e.,

$$\mathcal{H}_{n_1}(\boldsymbol{X}^*) = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{n_2} \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_{n_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{n_1} & \boldsymbol{x}_{n_1+1} & \cdots & \boldsymbol{x}_n \end{pmatrix} \in \mathbb{C}^{n_c n_1 \times n_2} \quad (2)$$

with $n_1 + n_2 = n + 1$. We say $\boldsymbol{X}^*$ satisfies the low-rank Hankel property if $\mathrm{rank}(\mathcal{H}_{n_1}(\boldsymbol{X}^*)) \le r$ for some $r \ll n$ and some integer n_1 in $[r, n+1-r]$. Throughout this paper, we assume $n_1 > 1$ is known and fixed and use $\mathcal{H}\boldsymbol{X}^*$ instead of $\mathcal{H}_{n_1}(\boldsymbol{X}^*)$ for simplicity.

Let $\boldsymbol{S}^*$ denote the additive errors in the measurements. We assume at most s measurements are corrupted, i.e., $\|\boldsymbol{S}^*\|_0 \le s$. The values of the nonzero entries can be arbitrary. The measurements are presented by

$$\boldsymbol{M} = \boldsymbol{X}^* + \boldsymbol{S}^*. \quad (3)$$

Define the operator $\mathcal{P}_{\widehat{\Omega}}$ such that $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M})_{i,j} = M_{i,j}$ if $(i,j) \in \widehat{\Omega}$, and 0 otherwise. The robust low-rank Hankel matrix completion problem aims to recover $\boldsymbol{X}^*$ from $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M})$. We formulate it as the following nonconvex optimization problem,

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{S}} \quad & \big\|\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M} - \boldsymbol{X} - \boldsymbol{S})\big\|_F \\ \text{s.t.} \quad & \mathrm{rank}(\mathcal{H}\boldsymbol{X}) \le r \quad \text{and} \quad \|\boldsymbol{S}\|_0 \le s, \end{aligned} \quad (4)$$

where the nonconvexity results from the rank and the sparsity constraints.

Definition 1. *A rank-r matrix $\boldsymbol{L} \in \mathbb{C}^{l_1 \times l_2}$, with its Singular Value Decomposition (SVD) $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, is μ-incoherent if*

$$\max_{1\le i\le l_1} \|\boldsymbol{e}_i^T \boldsymbol{U}\|^2 \le \frac{\mu r}{l_1}, \quad \max_{1\le j\le l_2} \|\boldsymbol{e}_j^T \boldsymbol{V}\|^2 \le \frac{\mu r}{l_2}. \quad (5)$$

The incoherence assumption is standard in analyzing RPCA and MC problem, see, e.g., [3], [29]. If a matrix is both low-rank and sparse, like $\boldsymbol{e}_i\boldsymbol{e}_j^T$ for any $1 \le i \le l_1$ and $1 \le j \le l_2$, then there is no way to separate the sparse component and the low-rank component. The incoherence assumption prevents the low-rank matrix L to be sparse itself. The incoherence measures the closeness of $\boldsymbol{L}$ to matrices like $\boldsymbol{e}_i\boldsymbol{e}_j^T$. If $\boldsymbol{L} = \boldsymbol{e}_i\boldsymbol{e}_j^T$, the corresponding μ is as large as $\max\{l_1, l_2\}/r$. When μ is small, the energy of $\boldsymbol{L}$ is spread over all its entries.

We assume $\boldsymbol{S}^*$ and $\boldsymbol{X}^*$ satisfy the following assumption throughout the paper. We will show our method can accurately recover $\boldsymbol{X}^*$ based on this assumption.

Assumption 1. *Each row of $\boldsymbol{S}^*$ contains at most α fraction of non-zero entries with $\alpha \le \frac{C_1}{\mu c_s r}$ for some small positive constant $C_1 \le \frac{1}{840}$,* [3] *where $c_s = \max\left(\frac{n}{n_1}, \frac{n}{n_2}\right)$; $\mathcal{H}\boldsymbol{X}^*$ is rank-r and μ-incoherent.*

In the successful recovery of $\boldsymbol{X}^*$ in conventional RPCA, $\boldsymbol{S}^*$ can have at most $\Theta(\frac{1}{r})$ fraction of nonzeros in each row

[3] The constant C_1 is derived from (69) and (74) in the Appendix.

and each column [23]. In contrast, Assumption 1 only requires an upper limit for each row, while the entries in one column of $\boldsymbol{S}^*$ can all be nonzero. In fact, $\boldsymbol{S}^*$ can contain α fraction of consecutive columns with all nonzero entries. If n_1 and n_2 are in the same order, i.e., both proportional to n, then c_s is a constant. α could be as large as $\Theta(\frac{1}{r})$. Thus, our method can handle bad data across all the channels consecutively for a nearly constant fraction of time if each row of $\boldsymbol{X}^*$ corresponding to a time series.

III. Applications and related work

A. Low-rank Hankel matrices

The low-rank Hankel property has been recently exploited in different areas including array signal processing [8], [30], dynamic system monitoring [33], and magnetic resonance imaging (MRI) [16], [25], [31].

One example of the low-rank Hankel property is the class of spectrally sparse signals [1], which are weighted sums of r damped or undamped sinusoids. The mathematical expression of an one-dimensional spectrally sparse signal is

$$g[t] = \sum_{i=1}^{r} d_i e^{(2\pi\imath f_i - \tau_i)t}, \quad t \in \mathbb{N}, \tag{6}$$

where f_i and d_i are the frequency and the normalized complex amplitude of the i-th sinusoid, respectively, and $\imath$ is the imaginary unit. As $g[t]$ is the sum of r sinusoids, its degree of freedom is $\Theta(r)$. The one-dimensional spectrally sparse signal $g[t]$ can be viewed as a special case of $\boldsymbol{X}^*$ in our paper. Specially, $\boldsymbol{X}^*$ only contains one row, i.e., $n_c = 1$, and let its i-th entry be $g[i]$. We follow [33] and refer to the resulting Hankel matrix as a single-channel Hankel matrix to differentiate from our general model of a multi-channel Hankel matrix with $n_c > 1$ in (2).

Ref. [8] also considers two-dimensional (2-D) and higher-dimensional spectrally sparse signals that are the sums of r 2-D or higher-dimensional sinusoids. The data matrix $\boldsymbol{X}^*$ of a 2-D spectrally sparse signal in [8] can be represented as

$$X^*_{t_1,t_2} = \sum_{i=1}^{r} d_i e^{(2\pi\imath f_{1i} - \tau_{1i})t_1 + (2\pi\imath f_{2i} - \tau_{2i})t_2}, \tag{7}$$

where X_{t_1,t_2} is the entry in row t_1 and column t_2. Note that the degree of freedom of $\boldsymbol{X}$ is still $\Theta(r)$ for a 2-D signal.

The second example of the low-rank Hankel property is the outputs of linear dynamic system discussed in [33]. Consider a discrete-time system with the state vector $\boldsymbol{s}_t \in \mathbb{C}^{n_p}$, and the observation vector $\boldsymbol{x}_t \in \mathbb{C}^{n_c}$,

$$\begin{aligned} \boldsymbol{s}_{t+1} &= \boldsymbol{A}\boldsymbol{s}_t, \\ \boldsymbol{x}_{t+1} &= \boldsymbol{C}\boldsymbol{s}_{t+1}, \quad t = 0, 1, \cdots, n. \end{aligned} \tag{8}$$

As described in [33], the data matrix $\boldsymbol{X}^* = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$ satisfies low-rank Hankel property. $\mathcal{H}_{n_1}(\boldsymbol{X}^*)$ is rank r for $n_1 \in [r, n+1-r]$, and the rank r is the number of the observed modes of the dynamical system. If $r < n_c$, $\mathcal{H}_{n_1}(\boldsymbol{X}^*)$ is also rank r for any $n_1 \in [1, n+1-r]$. Each row of $\boldsymbol{X}^*$ can be represented by a one-dimensional spectrally sparse signal. All rows share the same set of sinusoids but have different weights. The entry in row k and column t, denoted by $X^*_{k,t}$, can be written as

$$X^*_{k,t} = \sum_{i=1}^{r} d_{k,i} e^{(2\pi\imath f_i - \tau_i)t}, \quad k = 1, \cdots, n_c, \tag{9}$$

where $d_{k,i}$ is the normalized complex amplitude of the i-th sinusoid in the k-th signal. The degree of freedom of $\boldsymbol{X}^*$ is $\Theta(n_c r)$. We also remark that this paper considers the recovery of $\boldsymbol{X}^*$, which is irrelevant to the observability and identifiability of the linearly dynamical system in (8). Our method directly recovers the data from partial observations and does not need to estimate the system model.

In MRI imaging, a signal is called finite rate of innovation (FIR) [25] if there exists a finite sequence $\boldsymbol{h}[t]$ such that

$$(\boldsymbol{x}^* * \boldsymbol{h})[t] = 0, \quad \forall t, \tag{10}$$

where $(\cdot) * (\cdot)$ computes the convolution of two signals. Such $\boldsymbol{h}[t]$ is also known as annihilating filter. If the length of $\boldsymbol{h}[t]$ is $r+1$, then the Hankel matrix $\mathcal{H}_{n_1}(\boldsymbol{x})$ is a rank-r matrix for some $n_1 \in [r, n-r+1]$. The MRI images satisfy (10) after some transformation. The low-rank Hankel property has been exploited in MRI image recovery [16], [25], [31].

B. Robust matrix completion

When $n_1 = 1$, (4) reduces to the conventional RMC problem studied in [2], [4], [5], [7], [9], [12], [18], [19], [21]. If all the measurements are available, RMC reduces to the RPCA problem. The state-of-art RPCA algorithms such as [14], [23] can recover the low-rank matrix even if at most $O(\frac{1}{r})$ fraction of entries per row and per column are corrupted. This bound is also proved to be order-wise optimal [23]. If no corruptions exist, RMC reduces to the low-rank matrix completion problem, and $O(\mu_0 r n \log n)$ measurements are needed to recover an n_c-by-n ($n_c < n$) rank-r matrix with incoherence μ_0 [3].

For the general RMC problem, one approach is to relax the nonconvex rank and ℓ_0-norm into the convex nuclear norm and ℓ_1 norm and then solve the resulting convex optimization problem [2], [5], [18], [21]. Refs. [2], [5] show that the convex approach can correct a constant fraction of randomly distributed outliers, provided that a constant fraction of the matrix entries are observed. Based on a stronger requirement on the incoherence of the matrix, ref. [21] improves the theoretical bound such that only $O(\mu_0 r n \log^2(n))$ observed entries are required while tolerating a constant fraction of bad data. Although fully corrupted columns are considered in [7], [18], both papers cannot recover the corrupted columns. Ref. [7] shows that when $\boldsymbol{S}^*$ contains fully corrupted columns, the convex approach can recover noncorrupted columns and estimate the column subspace accurately. However, their approach can only locate the corrupted column but cannot recover its actual entries. Ref. [18] provides an upper bound of RMC when the measurements contain noise. The error bound is large when some columns are fully corrupted, because the recovery of corrupted columns is not accurate.

Fast algorithms to solve the nonconvex formulation directly have been recently developed. Ref. [4] adds a nonconvex

penalty function to speed up the minimization through a shrinkage operator, but no analytical analysis is reported. Ref. [19] proposes a projected gradient descent algorithm over the nonconvex sets. Ref. [12] proposes an alternating minimization algorithm. Both [12] and [19] prove the proposed algorithms converge under the assumption of Restrict Isometric Property (RIP), but no theoretical analyses of the recovery performance are provided.

The low-rank Hankel has been exploited in missing data recovery but not much in error corrections. Refs. [1] analyze the matrix completion performance for single-channel Hankel matrices, i.e., $n_c = 1$. Ref. [33] extends the analyses to multi-channel Hankel matrices with $n_c > 1$. If $\boldsymbol{S}^*$ is a zero matrix, one can recover $\boldsymbol{X}^*$ from $O(\mu r^3 \log n)$ observations [33], where μ is the incoherence of $\mathcal{H}\boldsymbol{X}^*$. Theorem 5 of [33] indicates that μ is a constant for a group of well separated frequencies f_i's and concentrated normalized amplitude $d_{k,i}$'s.

Only Refs. [8] and [17] consider the RMC problems for the low-rank Hankel matrix. The nonconvex rank and ℓ_0-norm are relaxed into the convex nuclear norm and ℓ_1-norm, respectively in both [8] and [17], and only Ref. [8] provides the theoretical guarantee. Although Ref. [8] consider high-dimensional spectral sparse signals, the degree of freedom of these signals is still $\Theta(r)$, which corresponds to single-channel Hankel matrices in our setup. We consider multi-channel Hankel matrix where $n_c > 1$ in this paper. Moreover, Ref. [8] assumes the locations of the corrupted entries are randomly distributed and does not provide any theoretical recovery guarantee when column-wise corruptions exist. This paper provides the first theoretical study of RMC and RPCA for multi-channel low-rank Hankel matrices with fully corrupted columns. Furthermore, the convex approach in [8] requires solving SDP, which is computationally challenging for large-scale problems. The computational complexity of solving the SDP to recover a Hankel matrix $\mathcal{H}\boldsymbol{X}^* \in \mathbb{C}^{n_c n_1 \times (n+1-n_1)}$ is $\mathcal{O}(n_c^3 n^3)$, while the computational complexity of our algorithm is $O\big(r^2 n_c n \log(n) \log(1/\varepsilon)\big)$, where ε is the approximation error.

C. Rank-based stagewise (R-RMC) algorithm

Ref. [9] proposed a nonconvex algorithm called Rank-based stagewise (R-RMC) algorithm to solve RMC. The R-RMC algorithm is directly extended from the AltProj algorithm in [23] for RPCA by adjusting to partial measurements. R-RMC contains two loops of iterations. In the k-th stage of the outer loop, it decomposes $\boldsymbol{M}$ into a rank-k matrix and a sparse matrix. The resulting matrices are used for initiation in the $(k+1)$-th stage. In the t-th iteration of the inner loop, it updates the sparse matrix $\boldsymbol{S}_t$ and the rank-k matrix $\boldsymbol{L}_{t+1}$ based on $\boldsymbol{S}_{t-1}$ and $\boldsymbol{L}_t$. $\boldsymbol{S}_t$ is obtained by a hard thresholding over the residual error between $\boldsymbol{M}$ and $\boldsymbol{L}_t$. $\boldsymbol{L}_{t+1}$ is updated by first moving along the gradient descent direction and then truncating it to a rank-k matrix. The reason of using an outer loop instead of directly decomposing into a rank-r and a sparse matrix is that by the initial thresholding, the remaining sparse corruptions in the residual is in the order of $\sigma_1(\boldsymbol{X}^*)$, the largest singular value of $\boldsymbol{X}^*$. These corruptions would lead to large errors in the estimation of the lower singular values of $\boldsymbol{X}^*$. Through the upper loop, the algorithm recovers the lower singular values after the corruptions at higher values are already removed. The computational complexity of R-RMC is $O\Big((mr^2+nr^3)\log(1/\varepsilon)\Big)$, where m is the number of observed measurements.

To achieve a recovery accuracy of ε, R-RMC requires at least $O(\mu_0^2 r^3 n \log^2(n) \log(1/\varepsilon))$ observed measurements. The percentage of outliers per row and per column is at most $O(\frac{1}{r})$.

This paper develops an algorithm based upon R-RMC [9] to solve the nonconvex problem (4). By exploiting the Hankel structure, our algorithm can correct fully corrupted columns, which cannot be corrected by R-RMC. Moreover, the required number of measurements by our method is significantly less than that by R-RMC.

IV. Structured Alternating Projection (SAP) Algorithm

Here we present the structured alternating projections (SAP) algorithm to solve (4). In the algorithm, $\boldsymbol{M}, \boldsymbol{X}_t, \boldsymbol{S}_t \in \mathbb{C}^{n_c \times n}$, and $\boldsymbol{W}_t, \boldsymbol{L}_t \in \mathbb{C}^{n_c n_1 \times n_2}$. $\mathcal{T}_\xi$ is the hard thresholding operator,

$$\mathcal{T}_\xi(\boldsymbol{Z})_{i,j} = Z_{ij} \quad \text{if} \quad |Z_{ij}| \geq \xi, \quad \text{and } 0 \quad \text{otherwise.} \tag{11}$$

Let $\boldsymbol{Z} = \sum_{i=1} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H$ denote the SVD of $\boldsymbol{Z}$ with $\sigma_1 \geq \sigma_2 \geq \cdots$. $\mathcal{Q}_k$ finds the best rank-k approximation to $\boldsymbol{Z}$, i.e.,

$$\mathcal{Q}_k(\boldsymbol{Z}) = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H. \tag{12}$$

$\mathcal{H}^\dagger$ denote the Moore-Penrose pseudoinverse of $\mathcal{H}$. Given any matrix $\boldsymbol{Z} \in \mathbb{C}^{n_c n_1 \times n_2}$, $\mathcal{H}^\dagger(\boldsymbol{Z}) \in \mathbb{C}^{n_c \times n}$ satisfies

$$(\mathcal{H}^\dagger(\boldsymbol{Z}))_{i,j} = \frac{1}{w_j} \sum_{k_1+k_2=j+1} Z_{(k_1-1)n_c+i,k_2}, \tag{13}$$

where w_j denotes the number of elements in the j-th anti-diagonal of an $n_1 \times n_2$ matrix.

Algorithm 1 Structured Alternating Projections (SAP)

Input Observations $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M})$, thresholding parameter ε, the largest singular value $\sigma_1 = \sigma_1(\mathcal{H}\boldsymbol{X}^*)$, and convergence criterion $\eta = \frac{4\mu c_s r}{\sqrt{n_c n}}$.
Initialization $\boldsymbol{X}_0 = \boldsymbol{0}$, $\xi_0 = \eta\sigma_1$.
Partition $\widehat{\Omega}$ into disjoint subsets $\widehat{\Omega}_{k,t}$ $(1 \leq k \leq r, 0 \leq t \leq T)$ of equal size $\widehat{m}$, let $\hat{p} = \frac{\widehat{m}}{n_c n}$.
for Stage $k = 1, 2, \cdots, r$ **do**
  for $t = 0, 1, \cdots, T = \log(\eta\sqrt{n_c} n \sigma_1/\varepsilon)$ **do**
    $\boldsymbol{S}_t = \mathcal{T}_{\xi_t}\big(\mathcal{P}_{\widehat{\Omega}_{k,t}}(\boldsymbol{M} - \boldsymbol{X}_t)\big)$;
    $\boldsymbol{W}_t = \mathcal{H}\Big(\boldsymbol{X}_t + \hat{p}^{-1}\big(\mathcal{P}_{\widehat{\Omega}_{k,t}}(\boldsymbol{M} - \boldsymbol{X}_t) - \boldsymbol{S}_t\big)\Big)$;
    $\xi_{t+1} = \eta\big(\sigma_{k+1}(\boldsymbol{W}_t) + (\frac{1}{2})^t \sigma_k(\boldsymbol{W}_t)\big)$;
    $\boldsymbol{L}_{t+1} = \mathcal{Q}_k(\boldsymbol{W}_t)$;
    $\boldsymbol{X}_{t+1} = \mathcal{H}^\dagger \boldsymbol{L}_{t+1}$;
  end for
  if $\eta\sigma_{k+1}(\boldsymbol{W}_T) \leq \frac{\varepsilon}{\sqrt{n_c n}}$ **then**
    Return $\boldsymbol{X}_{T+1}$;
  end if
  $\boldsymbol{X}_0 = \boldsymbol{X}_{T+1}$, $\xi_0 = \xi_{T+1}$;
end for

SAP is built upon Rank-based stagewise (R-RMC) in [9]. The major differences of SAP from R-RMC are the additional Hankel structure. The main contribution of this paper is the analytical performance guarantee of SAP, which we defer to Section V. The key steps are summarized as follows. Similar to R-RMC [9], SAP also contains two stages of iterations. In the t-th iteration of the inner loop, it updates the estimated sparse error matrix $\boldsymbol{S}_t$ and data matrix $\boldsymbol{X}_{t+1}$ based on $\boldsymbol{S}_{t-1}$ and $\boldsymbol{X}_t$. $\boldsymbol{S}_t$ is obtained by a hard thresholding over the residual error between $\boldsymbol{M}$ and $\boldsymbol{X}_t$. The thresholding ξ_t decreases as t increases. The entire sampling set $\widehat{\Omega}$ is first divided into several disjoint subsets. The disjointness guarantees the independence across $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+1}$, which is a standard analysis trick in solving RMC (see [27]). To obtain $\boldsymbol{X}_{t+1}$, we first updated X_t by moving along the gradient descent direction with a step size $\hat{p}^{-1} = \frac{n_c n}{|\widehat{\Omega}_{k,t}|}$. Then, $\boldsymbol{W}_t$ is calculated as the projection of the updated $\boldsymbol{X}_t$ to the Hankel matrix space. Finally, $\boldsymbol{X}_{t+1}$ is obtained by $\mathcal{H}^\dagger \boldsymbol{L}_{t+1}$, and $\boldsymbol{L}_{t+1}$ is updated by truncating $\boldsymbol{W}_t$ to a rank-k matrix. The maximum number of iterations in each inner loop, denoted as T, is set as $\log(\eta\sqrt{n_c}n\sigma_1/\varepsilon)$. In practice, the algorithm can exit the loop before reaching the maximum number of iterations if $\boldsymbol{X}_{t+1}$ is already very close to $\boldsymbol{X}_t$. In the k-th iteration of the outer loop, the target rank increases from 1 gradually, and the resulting matrices are used as the initialization in the $(k+1)$-th stage.

The reason of using an outer loop instead of directly applying rank-r approximation when calculating $\boldsymbol{L}_t$ is the same as that in R-RMC [9] and AltProj [23]. By the initial thresholding, the remaining sparse corruptions in the residual is in the order of $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$, the largest singular value of $\mathcal{H}\boldsymbol{X}^*$. These corruptions would lead to large errors in the estimation of the lower singular values of $\mathcal{H}\boldsymbol{X}^*$. Through the outer loop, the algorithm recovers the lower singular values after the corruptions with higher values are already removed.

Calculating the best rank-k approximation in line 9 dominates the computation complexity. Generally for a matrix $\boldsymbol{W}_t \in \mathbb{C}^{n_c n_1 \times n_2}$, the best rank-$k$ approximation can be solved in $O(kn_c n^2)$, and $n_c n^2$ results from calculating $\boldsymbol{W}_t\boldsymbol{z}$ for $\boldsymbol{z} \in \mathbb{C}^{n_2}$. Here, due to the Hankel structure of $\boldsymbol{W}_t$, a fast convolution algorithm (see [1], [33]) only requires computational complexity at $O(n_c n\log(n))$ to compute $(\mathcal{H}\boldsymbol{Z})\boldsymbol{z}$ for any $\boldsymbol{Z} \in \mathbb{C}^{n_c \times n}$ and $\boldsymbol{z} \in \mathbb{C}^{n_2}$. The fast convolution can also applied to reduce the computational time to $O\big(rn_c n\log(n)\big)$ when calculating $\boldsymbol{X}_{t+1} = \mathcal{H}^\dagger \boldsymbol{L}_{t+1}$ with stored SVD of $\boldsymbol{L}_{t+1}$[1], [33]. Hence, the computational complexity per iteration is $O\big(rn_c n\log(n)\big)$, and the total computational complexity is $O\big(r^2 n_c n\log(n)\log(1/\varepsilon)\big)$.

One can directly apply R-RMC on the structured Hankel matrix. The resulting algorithm differs from SAP in line 7 and 10 that the updated rank-k matrix is not projected to the Hankel matrix space. Based on the analysis in [9], the computational time per iteration of R-RMC on Hankel matrix is $O(m_s r + n_c n r^2)$, and m_s is the number of observed measurements in the structured Hankel matrix. With full observations, the computational complexity per iteration of R-RMC on Hankel matrices is as large as $O(n_c n^2 r)$. By downsampling the observation set to its theoretical limit in Theorem 2 of [9], the computational complexity per iteration can be reduced to $O(\mu^2 r^3 n_c n\log^2(n))$. However, it is still larger than $O(rn_c n\log(n))$ of SAP. Moreover, the constant item of the theoretical limit in theorem 2 [9] is hard to determine in practice. Furthermore, downsampling will increases the iteration number numerically. Though the complexity per iteration is reduced by downsampling, the computational time may increase, which is reported in Fig. 1(b) [9] as well.

We remark that $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$ and μ may not be computed directly. $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$ is only used to obtain the initial estimate of the sparse matrix. In practice, we use $p^{-1}(\mathcal{H}\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M}))$ to estimate $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$. This estimation idea is borrowed from [9], [23]. As long as the estimated value is in the same order as $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$, all the theoretical results in the following Theorem 1 still hold, with a different constant C_1 in Assumption 1. μ is only used in η as $\eta = \frac{\mu c_s r}{\sqrt{n_c n}}$. If the estimated incoherence is in the same order as μ, all the results still hold with a different constant C_1 in Assumption 1 and a different constant C_2 in (14). In practice, one can estimate η by $\frac{r}{\sqrt{n_c n_1 n_2}}$ for incoherent matrices without computing μ. This idea has been used in [2], [9], [23], which all require μ in their algorithms but do not actually compute it. Thus, we present Algorithm 1 using $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$ and μ to simply the following theoretical analysis, and one can replace them with estimated values in implementation.

We also note that the recovery of $\boldsymbol{X}^*$ is irrelevant to the observability and identifiability of the linearly dynamical system in (8), when $\boldsymbol{X}^*$ contains the output time series of (8). Our method directly recovers the data from partial observations and does not need to estimate the system model.

V. Recovery Guarantee of SAP

The recovery guarantee of SAP is summarized in Theorem 1, and the proof is deferred to the Appendix.

Theorem 1. *Suppose $\boldsymbol{X}^*$, $\boldsymbol{S}^*$ satisfy the Assumption 1, and the support of the sampling set $\widehat{\Omega}$ is randomly selected. Let $\eta = \frac{4\mu c_s r}{\sqrt{n_c n}}$ and $T = \log(\eta\sqrt{n_c}n\sigma_1/\varepsilon)$ in Algorithm 1. If*

$$m \geq C_2 \mu^2 r^3 \log^2(n)\log\Big(\frac{\mu c_s r\sigma_1}{\varepsilon}\Big), \tag{14}$$

with probability at least $1 - \frac{rn_c T\log^3(n_c n)}{n^2}$, its output $\boldsymbol{X}$ and $\boldsymbol{S}$ satisfy:

$$\begin{aligned} &\|\boldsymbol{X} - \boldsymbol{X}^*\|_F \leq \varepsilon \\ &\|\boldsymbol{S} - \mathcal{P}_{\widehat{\Omega}}(\boldsymbol{S}^*)\|_F \leq \varepsilon, \quad Supp(\boldsymbol{S}) \subseteq Supp\big(\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{S}^*)\big) \end{aligned} \tag{15}$$

for some large constant $C_2 > 0$ [4].

Theorem 1 indicates that the resulting $\boldsymbol{X}$ returned by SAP can be arbitrarily close to the ground truth $\boldsymbol{X}^*$ as long as the number of observations exceeds $O\Big(\mu^2 r^3\log^2(n)\log(\mu c_s r\sigma_1/\varepsilon)\Big)$, and each row of $\boldsymbol{S}^*$ has at most $\Theta(\frac{1}{\mu r})$ fraction of outliers. If $\boldsymbol{X}^*$ contains spectrally sparse signals as shown in (9), then $\boldsymbol{X}^*$ is also rank r. If we directly apply a low-rank MC method via convex

[4]The constant $C_2 = \max(C_4, C_5)$, and C_4 is derived from (63) in the proof of Lemma 5, C_5 is derived from (74) in the proof of Lemma 6.

relaxation [27] to recover $\boldsymbol{X}^*$ from $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{X}^*)$, the required number of observations is at least $O(\mu_0 rn\log^2(n))$. Since $n \gg r$, SAP reduces the required number of observations significantly by exploiting the Hankel structure. Moreover, SAP can identity and correct fully corrupted columns up to a fraction at $\Theta(1/\mu r)$. In contrast, traditional RMC methods can locate the fully corrupted columns but cannot recover the corrupted columns [7], [29]. The number of iterations rT depends on $\log(1/\varepsilon)$, where ε is desired accuracy. Therefore, the algorithm also enjoys a linear convergent rate.

If there is no bad data, i.e. $\boldsymbol{S}^* = \boldsymbol{0}$, (4) is reduced to the MC problem. Under the setup of spectrally sparse signals in (9), according to the Theorem 5 in [33], a group of well separated frequencies f_i's can guarantee that the incoherence $\mu < O(n_c)$. If we further assume on the normalized amplitude $d_{k,i}$, say that $d_{k,i}$'s are close to each other, the incoherence μ is a constant. The degree of freedom depends linearly on the rank r, while the theoretical bound in (14) relies on (r^3). When r is small, the theoretical bound in (14) is nearly optimal. Compared with our algorithm AM-FIHT in [33], SAP does not have the heavy-ball step and increases the rank gradually instead of keeping fixed rank. To achieve a recovery error of ε, AM-FIHT requires $O\Big(\mu\kappa^6 r^2\log(n)\log\left(\frac{\sigma_1}{\kappa^3\varepsilon}\right)\Big)$ observations. In contrast, SAP depends on r^3 but does not rely on the conditional number κ, where κ is defined as the ratio of the largest to smallest singular values of $\mathcal{H}\boldsymbol{X}^*$.

If there is no missing data, i.e. $\widehat{\Omega} = \{(k,t)|1 \le k \le n_c, 1 \le t \le n\}$, (4) is reduced to RPCA problem. Each row of $\boldsymbol{S}^*$ can have up to $\alpha \le \frac{C_1}{\mu c_s r}$ fraction of corrupted entries. If we choose $n_1 = n_2$, c_s is constant. The existing results in [14], [23] for RPCA can tolerate at most $\Theta(\frac{1}{r})$ fraction of outliers per row and per column. SAP also tolerates at most $\Theta(\frac{1}{r})$ fraction of outliers in each row. Moreover, SAP can recover fully corrupted columns. There is no upper bound of the number of corruptions per column. One can directly a general RPCA algorithm such as AltProj [23] on the structured Hankel matrix $\mathcal{H}(\boldsymbol{M})$, Altproj can recover the corrupted data correctly based on the same analysis as in [23]. However, the computational time per iteration of Altproj is $O(rn_cn^2)$, which is much large than $O(rn_cn\log(n))$ by SAP.

VI. Numerical Results

We evaluate the performance of SAP numerically. The experiments are implemented in MATLAB 2015 on a desktop with 3.4GHz Intel Core i7-4770 CPU. Here, we study several modes of missing data and bad data as shown in Figs. 1 and 2. For each pair of data loss and bad data modes, the supports of the bad data matrix $\boldsymbol{S}^*$ and the observed indices $\widehat{\Omega}$ are generated independently. The models are summarized as:

- M1/B1: Missing data or bad data occur randomly across the all channels and times;
- M2/B2: Missing data or bad data occur in all channel simultaneously s at randomly selected time indices;
- B3: Bad data occurs simultaneously and consecutively in all the channels. The starting point is selected randomly.

The performance is tested on the spectrally sparse signals as shown in (9). Each f_i in (9) is randomly selected from $(0,1)$.

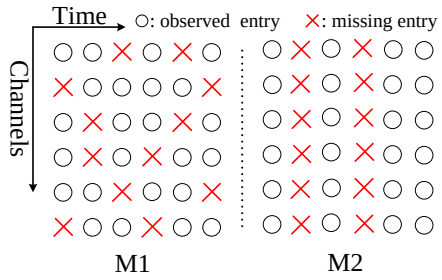

Fig. 1: Two modes of missing data

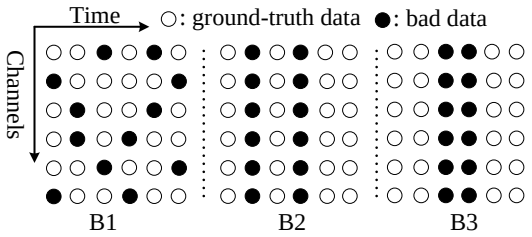

Fig. 2: Three modes of bad data

τ_i is set as 0 for all i. For the complex coefficient $d_{k,i}$, its angle is randomly selected from $(0, 2\pi)$, and its magnitude is set as $1+10^{0.5a_{k,i}}$, where $a_{k,i}$ is randomly selected from $(0,1)$. For each non-zero entry in the bad data matrix $\boldsymbol{S}^*$, its angle is randomly selected from $(0, 2\pi)$ (except for Fig. 4), and its magnitude is randomly selected from $(\bar{X}^*, 5\bar{X}^*)$, where $\bar{X}^* = \|\boldsymbol{X}^*\|_F/\sqrt{n_cn}$ is the average energy of $\boldsymbol{X}^*$. Unless otherwise stated, the size of the data matrix $\boldsymbol{X}^* \in \mathbb{C}^{n_c\times n}$ is set as $n_c = 30$ and $n = 300$, and $n_1 = n/2 = 150$.

In SAP, the SVD algorithm for a structured Hankel matrix is computed via PROPACK [20]. PROPACK provides a general framework to compute the partial SVD of a structured matrix that denoted by $\boldsymbol{A}$, and the user is required to implement the functions to compute $\boldsymbol{A}\boldsymbol{y}_1$ and $\boldsymbol{A}^H\boldsymbol{y}_2$. For a Hankel matrix $\mathcal{H}\boldsymbol{Z}_1$, the function to compute $(\mathcal{H}\boldsymbol{Z}_1)\boldsymbol{z}_2$ is implemented by calculating the convolution of $\boldsymbol{z}_2$ and each row of $\boldsymbol{Z}_1$. Since $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$ is unknown, we use $p^{-1}(\mathcal{H}\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M}))$ to approximate $\sigma_1(\mathcal{H}\boldsymbol{X}^*)$ in our experiments following the same idea in [9], [23]. Also, during each iteration, we use the entire observed set rather than the disjoint subsets as shown in line 3 of Alg. 1. In each inner loop, instead of keeping a fixed number of iterations, SAP will jump out of the current inner loop if

$$\frac{\|\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{X}_{t+1} - \boldsymbol{X}_t)\|_F}{\|\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{X}_t)\|_F} \le 10^{-3} \tag{16}$$

before reaching the maximum iteration number, which is set as 200. The algorithm finally terminates if $\sigma_{k+1}(\boldsymbol{W}_t) \le 10^{-3}$ holds.

The results in Figs. 3-10 are all obtained by averaging over 100 independent trials for each block. We say that the trial is successful if the returned $\boldsymbol{X}$ satisfies that

$$\frac{\|\mathcal{P}_{\widehat{\Omega}^c}(\boldsymbol{X} - \boldsymbol{X}^*)\|_F}{\|\mathcal{P}_{\widehat{\Omega}^c}(\boldsymbol{X}^*)\|_F} \le 10^{-2}, \tag{17}$$

where $\widehat{\Omega}^c$ is the complementary set of $\widehat{\Omega}$ over $\{1,2,\cdots,n_c\}\times\{1,2,\cdots,n\}$. A white block means that all 100 trials are successful, while all trials fail in a black block.

A. Performance of SAP

In this experiment, we vary the rank and bad data percentage to test the performance of SAP for several combined modes, M1×B1, M2×B2, and M2×B3. M1×B1 means missing data model M1 and bad data mode B1. We only provide the simulation results of these three combined modes because the performances of SAP are almost the same under modes M1×B1, M2×B1, M1×B2 and M2×B2. The data loss percentage is fixed as 50%.

Fig. 3 shows the recovery performance when the angles of nonzero entries in $\boldsymbol{S}^*$ is randomly selected from $(0, 2\pi)$. The x-axis is the bad data percentage, and the y-axis is the rank. The results under M1×B1 and M2×B2 are included in Fig. 3 to illustrate the similarity of SAP under these modes, and the similarity also shows that columnwise corruptions and missing entries do not affect the performance of SAP. Under mode M2×B3, we test the performance of SAP under simultaneous and consecutive bad data. It can tolerate 9% outliers for a rank-17 matrix, and 27 out of 300 consecutive columns are corrupted.

Fig. 4 shows the recovery performance when the angles of nonzero entries in $\boldsymbol{S}^*$ is randomly selected from $(0, \pi/2)$ such that both the real and imaginary parts of $\boldsymbol{S}^*$ are positive. Comparing Figs. 3 and 4, one can see that SAP performs very similar when the corruptions have random signs and when the corruptions have positive signs. The recovery performance with random signs is slightly better in all three modes.

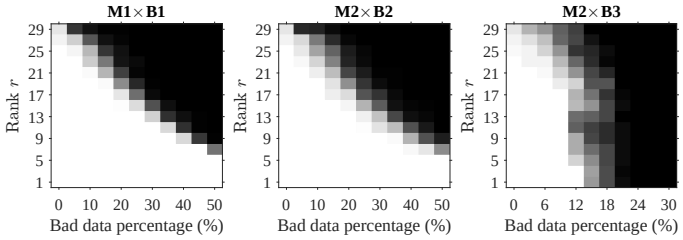

Fig. 3: Phase transition of SAP with random outliers

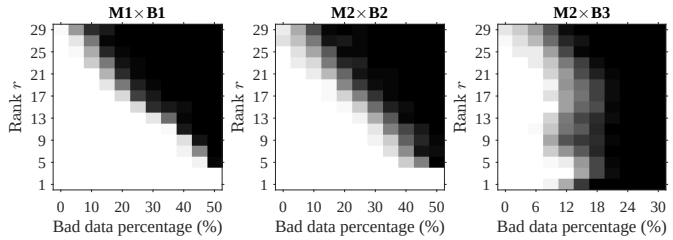

Fig. 4: Phase transition of SAP with outliers restricted in Quadrant I

B. Comparison with existing RMC methods

We compare SAP with two other RMC methods to recover $\boldsymbol{X}^*$ from $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M})$. One is R-RMC [9], and the other is the convex relaxation of (4) by relaxing rank and ℓ_0-norm to the approximated convex nuclear norm and ℓ_0-norm, and the convex optimization is solved by Alternating Direction Method of Multipliers (ADMM) [2]. [5] Under M1×B1, we apply ADMM and R-RMC on both $\mathcal{P}_{\Omega}(\boldsymbol{M})$ and the Hankel matrix $\mathcal{H}\big(\mathcal{P}_{\Omega}(\boldsymbol{M})\big)$. Since ADMM and R-RMC cannot tolerate columnwise data losses or corruptions, they can not recover $\boldsymbol{X}^*$ under M2×B2 and M2×B3. Hence, we only test ADMM and R-RMC on $\mathcal{H}\mathcal{P}_{\Omega}(\boldsymbol{M})$ under these two modes. The phase transitions in Fig. 5 are obtained by varying the data loss and bad data percentages, and the rank is set as 5 throughout this simulation.

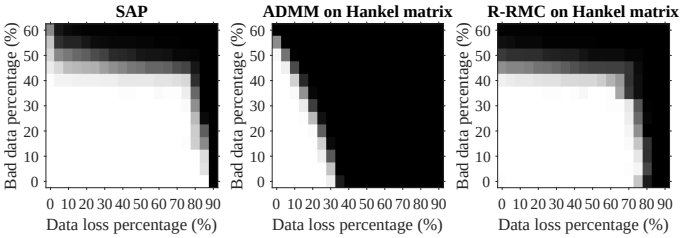

Fig. 6: Phase transition of SAP, ADMM and R-RMC on Hankel matrix under mode M2×B2

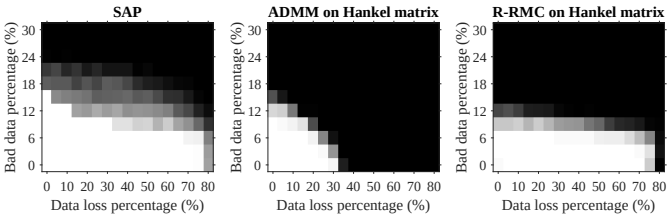

Fig. 7: Phase transition of SAP, ADMM and R-RMC on Hankel matrix under mode M2×B3

From the results shown in Figs. 5-7, under all these three modes, SAP performs the best among all methods. In Fig. 5, ADMM and R-RMC are both applied on the original observed data matrix $\mathcal{P}_{\widehat{\Omega}}(\boldsymbol{M})$, and the performances are much worse than SAP. When applying ADMM and R-RMC on the structured Hankel matrices, they both achieve higher success rates as shown in Figs. 5, 6 and 7. However, under modes M1×B1 and M2×B2, ADMM can only handle up to 30% data loss even on the structured Hankel matrix, while SAP

[5] We downloaded the codes from *https://github.com/andrewssobral* for R-RMC and *https://github.com/dlaptev/RobustPCA* for ADMM

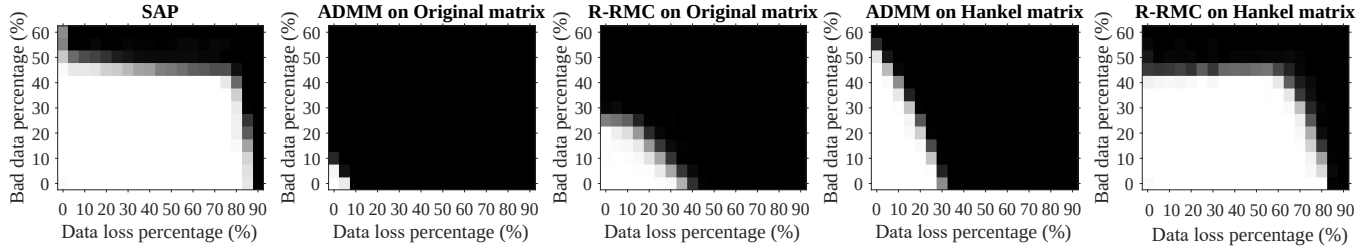

Fig. 5: Phase transition of SAP, ADMM and R-RMC under mode M1×B1

can still recover the data matrix under 80% data loss. R-RMC performs slightly worse than SAP when applying on the structured Hankel matrix in modes 1 and 2, but SAP obtains a much larger rate of success in mode M2×B3.

Moreover, SAP is significantly faster than ADMM and R-RMC on structured Hankel matrix as shown in Fig. 8. We vary number of columns n from 2000 to 8000 with a step size of 1000, and the results are averaging over 100 successful independent trials for each n. Since the computational complexities of all these methods depend linearly on n_c, we keep $n_c = 1$. The rank is fixed as 5, and n_1 is set as $n/2$ throughout the experiments. The size of the Hankel matrix is approximately $\frac{n}{2} \times \frac{n}{2}$. We consider the mode of M1×B1 where the locations of both bad data and miss data are generated randomly, and the bad data percentage is set as 20%. Since the computational complexity of R-RMC depends on the size of observed set, we study both 50% and 95% data loss percentages. The computational time of ADMM with 95% data loss is not included since it does not converge in this setting.

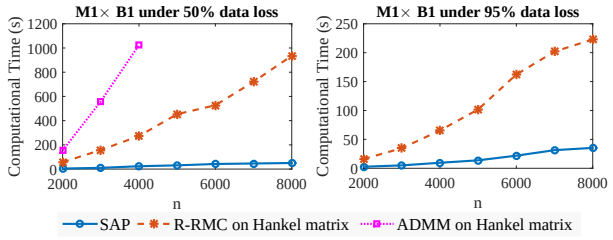

Fig. 8: Computational time of SAP, ADMM and R-RMC on the structured Hankel matrix

The computational time of SAP increases the least as the matrix size increases among all the methods. The convex method ADMM is the slowest with 50% data loss. ADMM takes over 1000 seconds to recover the Hankel matrix of size 2000×2000. R-RMC takes around 935 seconds to recover the Hankel matrix of size 4000×4000. With 95% data loss, the computational time of R-RMC decreases by applying fast algorithms to compute the sparse matrix multiplication. It takes much more time than SAP. For example, SAP takes less than 40 seconds to recover the Hankel matrix of size 4000×4000, while R-RMC takes around 227 seconds in the same setting.

C. Comparison with AM-FIHT in MC

In this experiment, we compare SAP with AM-FIHT in [33] to solve MC problem. We do not include other MC methods, such as SVT, because AM-FIHT is demonstrated to outperform other methods in both recovering accuracy and computational time [33]. We fix rank as 5. Since the number of observed entries for successful recovery depends on the conditional number κ, we consider both well-conditioned matrices, where κ is small, and ill-conditioned matrices, where κ is larger. To generate a well-conditioned matrix, we just follow the same setup for generating $\boldsymbol{X}^*$ in the previous experiments. To generate a ill-conditioned matrix, we enlarges the amplitude of the first sinusoid $d_{1,i}$ by a factor of r in all channels.

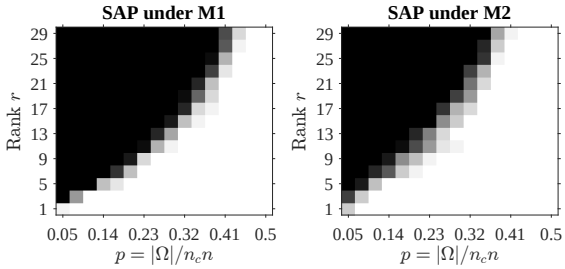

Fig. 9: Phase transition of SAP for ill-conditioned matrix

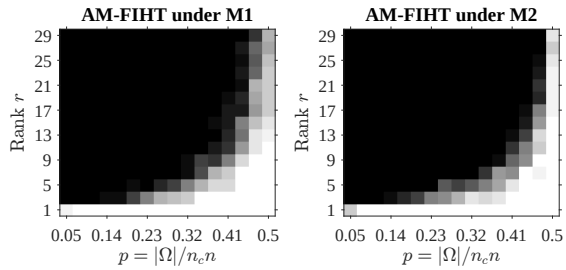

Fig. 10: Phase transition of AM-FIHT for ill-conditioned matrix

Fig. 9 shows the performance of SAP when recovering ill-conditioned matrices. When the matrix is well-conditioned, both SAP and AM-FIHT perform very similarly. Moreover, SAP performs similarly on both well-conditioned and ill-conditioned matrices. This verify our result in (14) that the performance of SAP does not depend on κ. We do not include the results of SAP and AM-FIHT in well-conditioned matrices because they are both similar to Fig. 9. When the matrix is ill-conditioned, AM-FIHT is much worse than SAP.

VII. Conclusion and Discussions

The multi-channel low-rank Hankel matrix naturally characterizes the correlations among columns of a matrix in addition to the low-rankness. Exploiting the low-rank Hankel structure, this paper develops a non-convex approach to recover the low-rank matrix from partial observations, even when a constant fraction of the columns are all corrupted simultaneously and consecutively. The proposed algorithm converges to the ground-truth matrix linearly. The required number of observations is significantly smaller than all the existing bounds for robust matrix completion. Our method applies to power system monitoring, MRI imaging, and array signal processing.

Acknowledgement

This research is supported in part by NSF 1508875, ARO W911NF-17-1-0407 and EPRI #1007316.

References

[1] J.-F. Cai, T. Wang, and K. Wei, "Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank hankel matrix completion," *Applied and Computational Harmonic Analysis*, vol. 46, no. 1, pp. 94–121, 2017.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[3] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[4] R. Chartrand, "Nonconvex splitting for regularized low-rank+ sparse decomposition," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5810–5819, 2012.
[5] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4324–4337, 2013.
[6] Y. Chen and M. J. Wainwright, "Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees," *arxiv preprint*, 2015, http://arxiv.org/abs/1509.03025.
[7] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Robust matrix completion with corrupted columns," in *Proc. International Conference on Machine Learning*, 2011.
[8] Y. Chen and Y. Chi, "Robust spectral compressed sensing via structured matrix completion," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6576–6601, 2014.
[9] Y. Cherapanamjeri, K. Gupta, and P. Jain, "Nearly-optimal robust matrix completion," *arXiv preprint arXiv:1606.07315*, 2016.
[10] P. Gao, M. Wang, J. H. Chow, S. G. Ghiocel, B. Fardanesh, G. Stefopoulos, and M. P. Razanousky, "Identification of successive "unobservable" cyber data attacks in power systems." *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5557–5570, 2016.
[11] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1006–1013, 2016.
[12] Q. Gu, Z. Wang, and H. Liu, "Low-rank and sparse structure pursuit via alternating minimization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, vol. 51, 2016, pp. 600–609.
[13] J. P. Haldar, "Low-rank modeling of local k-space neighborhoods (loraks) for constrained mri," *IEEE Trans. Med. Imag.*, vol. 33, no. 3, pp. 668–681, 2014.
[14] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7221–7234, 2011.
[15] P. Jain and P. Netrapalli, "Fast exact matrix completion with finite samples," in *Conference on Learning Theory*, 2015, pp. 1007–1034.
[16] K. H. Jin, D. Lee, and J. C. Ye, "A general framework for compressed sensing and parallel mri using annihilating filter based low-rank hankel matrix," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 480–495, 2016.
[17] K. H. Jin and J. C. Ye, "Sparse and low-rank decomposition of a hankel structured matrix for impulse noise removal," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1448–1461, 2018.
[18] O. Klopp, K. Lounici, and A. B. Tsybakov, "Robust matrix completion," *Probability Theory and Related Fields*, vol. 169, no. 1, pp. 523–564, Oct. 2017.
[19] A. Kyrillidis and V. Cevher, "Matrix alps: Accelerated low rank and sparse matrix reconstruction," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, 2012, pp. 185–188.
[20] R. M. Larsen, "Propack-software for large and sparse svd calculations," *Available online. URL http://sun. stanford. edu/rmunk/PROPACK*, pp. 2008–2009, 2004.
[21] X. Li, "Compressed sensing and matrix completion with constant proportion of corruptions," *Constructive Approximation*, vol. 37, no. 1, pp. 73–99, Feb 2013.
[22] M. Mardani, G. Mateos, and G. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5186–5205, 2013.
[23] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust PCA," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 1107–1115.
[24] G. Ongie, S. Biswas, and M. Jacob, "Convex recovery of continuous domain piecewise constant images from nonuniform fourier samples," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 236–250, 2018.
[25] G. Ongie and M. Jacob, "Off-the-grid recovery of piecewise constant images from few fourier samples," *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 1004–1041, 2016.
[26] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.
[27] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learn. Res.*, pp. 3413–3430, 2011.
[28] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Adv. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.
[29] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3047–3064, May 2012.
[30] Z. Yang and L. Xie, "Exact joint sparse frequency recovery via optimization methods," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5145–5157, 2016.
[31] J. C. Ye, J. M. Kim, K. H. Jin, and K. Lee, "Compressive sampling using annihilating filter-based low-rank interpolation," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 777–801, 2017.
[32] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust PCA via gradient descent," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4152–4160.
[33] S. Zhang, Y. Hao, M. Wang, and J. H. Chow, "Multi-channel hankel matrix completion through nonconvex optimization," *IEEE J. Sel. Topics Signal Process., Special Issue on Signal and Information Processing for Critical Infrastructures*, vol. 12, no. 4, pp. 617–632, 2018.
[34] S. Zhang and M. Wang, "Correction of simultaneous bad measurements by exploiting the low-rank hankel structure," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 646–650.

APPENDIX

A. Notations and technical assumptions

Sampling model with replacement. As a standard technique in solving RMC problem [27], the model of sampling with replacement assumes that every entry is sampled independently with replacement. In this model, one entry can be sampled multiple times. To distinguish from $\widehat{\Omega}$ defined in Section II, let Ω be the union of indices that uniformly sampled from $\{1,2,\cdots,n_c\}\times\{1,2,\cdots,n\}$ following the sampling model with replacement. Due to the repetitions in the sampling model with replacement, $|\Omega|\geq|\widehat{\Omega}|$ should hold for successful recovery [27]. Hence, the required number of observations for successful recovery under sampling model with replacement is sufficient to guarantee successful recovery under sampling model without replacement.

Symmetric Hankel Operator. Here, we introduce the operator $\widetilde{\mathcal{H}}$, which is the symmetric extension of Hankel operator $\mathcal{H}$. For any $\boldsymbol{Z}\in\mathbb{C}^{n_c\times n}$, $\widetilde{\mathcal{H}}(\boldsymbol{Z})\in\mathbb{C}^{n_c(n_1+n_2)\times n_c(n_1+n_2)}$ is defined as

$$\widetilde{\mathcal{H}}(\boldsymbol{Z})=\begin{pmatrix}\underbrace{\begin{matrix}\mathbf{0}&\cdots&\mathbf{0}\\ \vdots&\ddots&\vdots\\ \mathbf{0}&\cdots&\mathbf{0}\\ \mathcal{H}(\boldsymbol{Z})&\cdots&\mathcal{H}(\boldsymbol{Z})\end{matrix}}_{n_c\text{ copies}}&\begin{matrix}(\mathcal{H}(\boldsymbol{Z}))^H\\ \vdots\\ (\mathcal{H}(\boldsymbol{Z}))^H\\ \mathbf{0}\end{matrix}\end{pmatrix}. \tag{18}$$

Define $\mathcal{H}\boldsymbol{X}^*=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$ as the SVD of $\mathcal{H}\boldsymbol{X}^*$, then $\widetilde{\mathcal{H}}\boldsymbol{X}^*$ can be written as

$$\widetilde{\mathcal{H}}\boldsymbol{X}^*=\frac{1}{\sqrt{2}}\begin{pmatrix}\widetilde{\boldsymbol{V}}&\widetilde{\boldsymbol{V}}\\ \boldsymbol{U}&\boldsymbol{U}\end{pmatrix}\begin{pmatrix}\sqrt{n_c}\boldsymbol{\Sigma}&\mathbf{0}\\ \mathbf{0}&-\sqrt{n_c}\boldsymbol{\Sigma}\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}\widetilde{\boldsymbol{V}}&\widetilde{\boldsymbol{V}}\\ \boldsymbol{U}&\boldsymbol{U}\end{pmatrix}^H, \tag{19}$$

where $\widetilde{\boldsymbol{V}}=\frac{1}{\sqrt{n_c}}[\boldsymbol{V}^H\quad\cdots\quad\boldsymbol{V}^H]^H$. Therefore, $\widetilde{\mathcal{H}}\boldsymbol{X}$ is a rank-$2r$ matrix. Moreover, if $\mathcal{H}\boldsymbol{X}^*$ is μ-incoherent, one can easily check that the incoherence $\tilde{\mu}$ of $\widetilde{\mathcal{H}}\boldsymbol{X}$ satisfies $\tilde{\mu}\leq\frac{c_s}{2}\mu$. When n_1 and n_2 are in the same order, c_s is a constant.

The key steps (lines 5-9) of Alg. 1 can be represented equivalently based on $\widetilde{\mathcal{H}}$ as:

$$
\begin{aligned}
&\widetilde{\boldsymbol{S}}_t = \mathcal{T}_{\xi_t}(\boldsymbol{M}-\boldsymbol{X}_t);\\
&\widetilde{\boldsymbol{W}}_t = \widetilde{\mathcal{H}}\big(\boldsymbol{X}_t + p^{-1}\mathcal{P}_{\Omega_{k,t}}(\boldsymbol{M}-\boldsymbol{X}_t-\widetilde{\boldsymbol{S}}_t)\big);\\
&\xi_{t+1} = \frac{\eta}{\sqrt{n_c}}\Big(|\lambda_{2k}(\widetilde{\boldsymbol{W}}_t)| + \big(\frac{1}{2}\big)^t|\lambda_{2k+2}(\widetilde{\boldsymbol{W}}_t)|\Big);\\
&\widetilde{\boldsymbol{L}}_{t+1} = \mathcal{Q}_{2k}(\widetilde{\boldsymbol{W}}_t);\\
&\boldsymbol{X}_{t+1} = \widetilde{\mathcal{H}}^{\dagger}(\widetilde{\boldsymbol{L}}_{t+1});
\end{aligned}
\tag{20}
$$

The Pseudoinverse operator $\widetilde{\mathcal{H}}^{\dagger}$ can be calculated from

$$
(\widetilde{\mathcal{H}}^{\dagger}(\boldsymbol{Z}))_{i,j} = \frac{1}{n_c w_j}\langle \widetilde{\mathcal{H}}(\boldsymbol{e}_i\boldsymbol{e}_j^T), \boldsymbol{Z}\rangle. \tag{21}
$$

In fact, (20) generates the same $\boldsymbol{X}_t$'s as lines 5-9 in Alg. 1. (20) differs from lines 5-9 in Alg. 1 mainly in two aspects : (1) $\widetilde{\boldsymbol{S}}_t$ is updated based on the full observation of $\boldsymbol{M}$; (2) $\widetilde{\boldsymbol{W}}_t$ lies in the space defined by $\widetilde{\mathcal{H}}$. Though we cannot calculate $\widetilde{\boldsymbol{S}}_t$ from $\mathcal{P}_{\Omega}(\boldsymbol{M})$ in practice, $\widetilde{\boldsymbol{S}}_t$ is introduced to simplify our analysis and does not affect the update of $\boldsymbol{X}_t$. To see this, we first assume the values of $\boldsymbol{X}_{t-1}$ are the same for (20) and lines 5-9 in Alg. 1. Then, the threshold ξ_t remains the same as well. Next, we have

$$
\mathcal{P}_{\Omega_{k,t}}(\widetilde{\boldsymbol{S}}_t) = \boldsymbol{S}_t, \tag{22}
$$

which suggests $\mathcal{P}_{\Omega_{k,t}}(\boldsymbol{M}-\boldsymbol{X}_t) - \boldsymbol{S}_t = \mathcal{P}_{\Omega_{k,t}}(\boldsymbol{M}-\boldsymbol{X}_t - \widetilde{\boldsymbol{S}}_t)$. Operator $\widetilde{\mathcal{H}}$ does not affect the update rule of $\boldsymbol{X}_t$, either. Similarly, suppose $\boldsymbol{X}_{t-1}$ remains the same for some t, then it is easy to verify that

$$
\widetilde{\boldsymbol{L}}_t = \begin{pmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{L}_t^H \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{L}_t^H \\ \underbrace{\boldsymbol{L}_t \quad \cdots \quad \boldsymbol{L}_t}_{n_c \text{ copies}} & & & \boldsymbol{0} \end{pmatrix} \in \mathbb{C}^{n_c(n_1+n_2)\times n_c(n_1+n_2)}.
$$

(Since $n+1 = n_1+n_2$, we use n to replace n_1+n_2 for convenience in all the sections of Appendix.) Moreover, $\widetilde{\boldsymbol{L}}_t$ has duplicated eigenvalues as $|\lambda_{2i-1}(\widetilde{\boldsymbol{L}}_t)| = |\lambda_{2i}(\widetilde{\boldsymbol{L}}_t)|$ for $1 \le i \le r$, where $\lambda_i(\widetilde{\boldsymbol{L}}_t)$ is the i-th largest eigenvalues (in absolute value) of $\widetilde{\boldsymbol{L}}_t$. Furthermore, let $\sigma_i(\boldsymbol{L}_t)$ be the i-th largest singular value of $\boldsymbol{L}_t$, from (19) we have

$$
\sigma_i(\boldsymbol{L}_t) = \frac{1}{\sqrt{n_c}}|\lambda_{2i-1}(\widetilde{\boldsymbol{L}}_t)| = \frac{1}{\sqrt{n_c}}|\lambda_{2i}(\widetilde{\boldsymbol{L}}_t)|. \tag{23}
$$

Similar results can be derived for $\widetilde{\boldsymbol{W}}_t$. From the definition of $\widetilde{\mathcal{H}}^{\dagger}$ and the structure of $\widetilde{\boldsymbol{L}}_t$, it is straightforward that $\boldsymbol{X}_{t+1}$ returned by lines 5-9 in Alg. 1 and (20) are equivalent. In conclusion, if we start with the same initial point $\boldsymbol{X}_0 = \boldsymbol{0}$, the update rule in (20) will generate the same $\boldsymbol{X}_t$'s as those by lines 5-9 in Alg. 1, and we also have $\mathcal{P}_{\Omega_{k,t}}(\widetilde{\boldsymbol{S}}_t) = \boldsymbol{S}_t$.

Definition of $\boldsymbol{H}_{1,t}$, $\boldsymbol{H}_{2,t}$ and $\boldsymbol{H}_t$. From (20), we know that

$$
\begin{aligned}
\widetilde{\boldsymbol{L}}_{t+1} =& \mathcal{Q}_{2k}\big(\widetilde{\mathcal{H}}\boldsymbol{X}_t + \hat{p}^{-1}\widetilde{\mathcal{H}}\mathcal{P}_{\Omega_{k,t}}(\boldsymbol{M}-\boldsymbol{X}_t-\widetilde{\boldsymbol{S}}_t)\big)\\
=& \mathcal{Q}_{2k}\big(\widetilde{\mathcal{H}}\boldsymbol{X}_t + \hat{p}^{-1}\widetilde{\mathcal{H}}\mathcal{P}_{\Omega_{k,t}}(\boldsymbol{X}^*+\boldsymbol{S}^*-\boldsymbol{X}_t-\widetilde{\boldsymbol{S}}_t)\big)\\
=& \mathcal{Q}_{2k}\big(\widetilde{\mathcal{H}}\boldsymbol{X}^* + \widetilde{\mathcal{H}}(\mathcal{I}-\hat{p}^{-1}\mathcal{P}_{\Omega_{k,t}})(\boldsymbol{X}_t+\widetilde{\boldsymbol{S}}_t-\boldsymbol{X}^*-\boldsymbol{S}^*)\\
&+\widetilde{\mathcal{H}}(\boldsymbol{S}^*-\widetilde{\boldsymbol{S}}_t)\big).
\end{aligned}
$$

Let $\boldsymbol{H}_t = \boldsymbol{H}_{1,t} + \boldsymbol{H}_{2,t}$, where

$$
\boldsymbol{H}_{1,t} = \widetilde{\mathcal{H}}(\boldsymbol{S}^*-\widetilde{\boldsymbol{S}}_t), \tag{24}
$$

$$
\boldsymbol{H}_{2,t} = \widetilde{\mathcal{H}}(\mathcal{I}-\hat{p}^{-1}\mathcal{P}_{\Omega_{k,t}})(\boldsymbol{X}_t+\widetilde{\boldsymbol{S}}_t-\boldsymbol{X}^*-\boldsymbol{S}^*). \tag{25}
$$

Then, we have

$$
\widetilde{\boldsymbol{L}}_t = \mathcal{Q}_{2k}(\widetilde{\mathcal{H}}\boldsymbol{X}^* + \boldsymbol{H}_t) = \mathcal{Q}_{2k}(\widetilde{\mathcal{H}}\boldsymbol{X}^* + \boldsymbol{H}_{1,t} + \boldsymbol{H}_{2,t}).
$$

B. Key lemma in proving Theorem 1

We first introduce the most critical lemma in the whole proof. Lemma 1 is presented to bound the ℓ_2-norm of $\boldsymbol{e}_i^T(\boldsymbol{H}_t)^a\boldsymbol{Z}$ by the ℓ_2-norm of $\boldsymbol{e}_i^T\boldsymbol{Z}$ for all $1 \le a \le \log(n)$. Although Lemma 1 is not directly used in proving Theorem 1, Lemma 1 is paramount important in showing the recovery error of $\boldsymbol{X}_t$ decreases as t increases that summarized in Lemma 6.

Lemma 15 in [9] provides a similar result for general matrix but with a more complicated proof. Ref. [9] focused on bounding all entries of $\boldsymbol{e}_i^T(\boldsymbol{H}_{1,t}+\boldsymbol{H}_{2,t})^a\boldsymbol{Z}$, so Ref. [9] needed to write the closed forms of all entries in $(\boldsymbol{H}_{1,t}+\boldsymbol{H}_{2,t})^a$. The closed forms are hard to determine, and several cases should be discussed separately. However, we will prove (26) by mathematical induction over a. Only two items, $\|\boldsymbol{e}_i^T\boldsymbol{H}_{1,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\|_2$ and $\|\boldsymbol{e}_i^T\boldsymbol{H}_{2,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\|_2$, need to be bounded in the inductive step. Also, the conclusion of Lemma 1 in (26) can be extended to general matrices, though $\boldsymbol{H}_t = \boldsymbol{H}_{1,t}+\boldsymbol{H}_{2,t}$ is the Hankel matrix as defined in (24) and (25).

There are two lemmas used in the inductive steps of proving Lemma 1. [6] Lemma 2 is established on the sparsity assumption with respect to $\boldsymbol{H}_{1,t}$. Moreover, Lemma 2 is a special case that $a=1$ of Lemma 5 in [23], and all the steps are straightforward from [23]. However, instead of discussing a special $\boldsymbol{U}$ like in [23], we consider a general matrix $\boldsymbol{Z}$ here. Lemma 3 provides similar result for matrices with zero mean and bounded high moments, and it is used to bound $\boldsymbol{H}_{2,t}$. The technique used in Lemma 3 is similar as that of Lemma 9 in [15]. Rather than bounding each entry of $\boldsymbol{e}_i^T(\widetilde{\mathcal{H}}\boldsymbol{Y})\boldsymbol{Z}$ separately as [15], we bound the ℓ_2 norm of $\boldsymbol{e}_i^T(\widetilde{\mathcal{H}}\boldsymbol{Y})\boldsymbol{Z}$ directly, which leads to a tighter bound by a factor of r^{-1}. The same trick is applied in [9] as well.

Lemma 1. *Suppose the assumptions in Theorem 1. If we further assume that $Supp(\widetilde{\boldsymbol{S}}_t - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*)$, then for $1 \le a \le \log(n_c n)$ and any $\boldsymbol{Z} \in \mathbb{C}^{n_c n\times l}$, with probability at least $1-\frac{n_c\log(n_c n)}{n^2}$, we have*

$$
\begin{aligned}
&\max_i \|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^a\boldsymbol{Z}\|_2\\
\le&\Big(C_3\beta_t\log(n) + \alpha n_c n\|\boldsymbol{H}_{1,t}\|_{\infty}\Big)^a \max_i\|\boldsymbol{e}_i^T\boldsymbol{Z}\|_2,
\end{aligned}
\tag{26}
$$

where $\beta_t = \sqrt{\frac{n_c n}{\hat{p}}}\Big\|\boldsymbol{X}_t+\widetilde{\boldsymbol{S}}_t-\boldsymbol{X}-\boldsymbol{S}\Big\|_{\infty}$ and C_3 is a constant that greater than e^4.

Lemma 2. *Assume each row and column of $\boldsymbol{H} \in \mathbb{C}^{n_c n\times n_c n}$ has at most s nonzero entries, then for any $\boldsymbol{Z} \in \mathbb{C}^{n_c n\times l}$,*

$$
\max_{1\le i\le n_c n}\|\boldsymbol{e}_i^T\boldsymbol{H}\boldsymbol{Z}\|_2 \le (s\|\boldsymbol{H}\|_{\infty})\max_{1\le j\le n_c n}\|\boldsymbol{e}_j^T\boldsymbol{Z}\|_2. \tag{27}
$$

Lemma 3. *Assume each entry of $\boldsymbol{Y} \in \mathbb{C}^{n_c\times n}$ is drawn independently with*

$$
\mathbb{E}(Y_{i,j}) = 0, \qquad \mathbb{E}(|Y_{i,j}|^k) \le \frac{1}{n_c n} \tag{28}
$$

for all $1 \le i \le n_c$, $1 \le j \le n$ and $k \ge 2$. Then, for any $\boldsymbol{Z} \in \mathbb{C}^{n_c n\times l}$, we have

$$
\max_{1\le i\le n_c n}\|\boldsymbol{e}_i^T(\widetilde{\mathcal{H}}\boldsymbol{Y})\boldsymbol{Z}\|_2 \le C_3\log(n)\max_{1\le j\le n_c n}\|\boldsymbol{e}_j^T\boldsymbol{Z}\|_2, \tag{29}
$$

[6] The proof of these two lemmas are presented in the supplementary material

with probability $1-n_cn^{-3}$.

Proof of Lemma 1. From the assumption, we know each row of $\boldsymbol{S}^*$ has at most α fraction of nonzero entries. Since each row of $\mathcal{H}\boldsymbol{S}^*$ is a subset of the corresponding row in $\boldsymbol{S}^*$, then the number of nonzero entries in each row of $\mathcal{H}\boldsymbol{S}^*$ is bounded by αn. Similarly, the nonzero entries in each column of $\mathcal{H}\boldsymbol{S}^*$ is bounded by $\alpha n_c n$. By the definition of $\widetilde{\mathcal{H}}$ in (18), we know that each row or column of $\widetilde{\mathcal{H}}\boldsymbol{S}^*$ has at most $\alpha n_c n$ nonzero entries. On the other hand, $\frac{1}{\beta_t}(\mathcal{I}-\hat{p}^{-1}\mathcal{P}_{\Omega_{k,t}})(\boldsymbol{X}_t+\widetilde{\boldsymbol{S}}_t-\boldsymbol{X}^*-\boldsymbol{S}^*)$ satisfies (28) in Lemma 3. The property of zero mean in (28) is trivial. For bounded high moment, with $k\geq 2$,

$$\begin{aligned}
&\mathbb{E}\Big[\Big|\frac{1}{\beta_t}(\mathcal{I}-p^{-1}\mathcal{P}_\Omega)(\boldsymbol{X}_t+\widetilde{\boldsymbol{S}}_t-\boldsymbol{X}^*-\boldsymbol{S}^*)\Big|^k\Big]\\
\leq&\Big(\frac{p}{n_cn}\Big)^{\frac{k}{2}}\Big(p(1-p^{-1})^k+(1-p)\Big)\\
=&\Big(\frac{p}{n_cn}\Big)^{\frac{k}{2}}\cdot\frac{(1-p)\big((1-p)^{k-1}+p^{k-1}\big)}{p^{k-1}}\\
\leq&\Big(\frac{p}{n_cn}\Big)^{\frac{k}{2}}\cdot\frac{1}{p^{k-1}}=\frac{1}{n_cn}\cdot\frac{1}{(n_cnp)^{\frac{k}{2}-1}}\\
\leq&\frac{1}{n_cn}.
\end{aligned}\tag{30}$$

Since $\boldsymbol{H}_t=\boldsymbol{H}_{1,t}+\boldsymbol{H}_{2,t}$, we have

$$\begin{aligned}
&\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^a\boldsymbol{Z}\big\|_2=\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_{1,t}+\boldsymbol{H}_{2,t})(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2\\
\leq&\big\|\boldsymbol{e}_i^T\boldsymbol{H}_{1,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2+\big\|\boldsymbol{e}_i^T\boldsymbol{H}_{2,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2.
\end{aligned}\tag{31}$$

By Lemma 2, we have

$$\big\|\boldsymbol{e}_i^T\boldsymbol{H}_{1,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2\leq\alpha n_cn\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2.\tag{32}$$

By Lemma 3, we have

$$\Big\|\boldsymbol{e}_i^T\boldsymbol{H}_{2,t}(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\Big\|_2\leq C_3\beta_t\log(n)\Big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\Big\|_2.\tag{33}$$

with high probability.
Hence, (32) and (33) suggests

$$\begin{aligned}
&\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^a\boldsymbol{Z}\big\|_2\\
\leq&\big(C_3\beta_t\log(n)+\alpha n_cn\|\boldsymbol{H}_{1,t}\|_\infty\big)\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^{a-1}\boldsymbol{Z}\big\|_2
\end{aligned}\tag{34}$$

with high probability.
Then, by applying (34) multiple times, with high probability we have

$$\begin{aligned}
&\big\|\boldsymbol{e}_i^T(\boldsymbol{H}_t)^a\boldsymbol{Z}\big\|_2\\
\leq&\big(C_3\beta_t\log(n)+\alpha n_cn\|\boldsymbol{H}_{1,t}\|_\infty\big)^a\big\|\boldsymbol{e}_i^T\boldsymbol{Z}\big\|_2.
\end{aligned}\tag{35}$$

Taking a union bound over all i completes the whole proof. □

C. Supporting Lemmas for Theorem 1

In the following lemmas, $\widetilde{\boldsymbol{S}}_t$, $\boldsymbol{X}_t$, $\boldsymbol{H}_t$, $\widetilde{\boldsymbol{W}}_t$ and $\widetilde{\boldsymbol{L}}_t$ are generated in the k-th outer loop unless otherwise specified. For convenience, we use λ_i^* to denote $\lambda_{2i-1}(\widetilde{\mathcal{H}}\boldsymbol{X}^*)$, which is the $(2i-1)$-th largest eigenvalue (in absolute value) of $\widetilde{\mathcal{H}}\boldsymbol{X}^*$. Similarly, $\lambda_i^{(t)}$ stands for $\lambda_{2i-1}(\widetilde{\boldsymbol{W}}_t)$, which is the $(2i-1)$-th largest eigenvalue (in absolute value) of $\widetilde{\boldsymbol{W}}_t$.

Lemma 5 proves that the assumptions (40) and (42) are equivalent. Lemma 6 shows the reduction of $\|\boldsymbol{X}_{t+1}-\boldsymbol{X}^*\|_\infty$ as t increases. Moreover, the error bound of $\|\widetilde{\boldsymbol{S}}_{t+1}-\boldsymbol{S}^*\|_\infty$ is given in Lemma 7 based on the bound of $\|\boldsymbol{X}_{t+1}-\boldsymbol{X}^*\|_\infty$.

Lemma 4 (Weyl's inequality)**.** *Suppose $\boldsymbol{A}$ and $\boldsymbol{B}$ are two $n\times n$ symmetric matrices satisfying $\boldsymbol{B}=\boldsymbol{A}+\boldsymbol{E}$. Let $\lambda_1\geq\lambda_2\geq\cdots\lambda_n$ be the eigenvalues of $\boldsymbol{A}$, denoted by $\lambda_i(\boldsymbol{A})=\lambda_i$. Then,*

$$|\lambda_i(\boldsymbol{B})-\lambda_i(\boldsymbol{A})|\leq\|\boldsymbol{E}\|_2,\quad 1\leq i\leq n.\tag{36}$$

Lemma 5. *Suppose the assumptions in Theorem 1 and*

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_t-\boldsymbol{S}^*\|_\infty\leq\frac{7\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_t-\boldsymbol{S}^*)\subseteq Supp(\boldsymbol{S}^*),
\end{aligned}\tag{37}$$

$$\|\boldsymbol{X}_t-\boldsymbol{X}^*\|_\infty\leq\frac{2\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big).\tag{38}$$

With probability at least $1-n_cn^{-2}$, we have

$$\|\boldsymbol{H}_t\|_2\leq\frac{1}{60}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big).\tag{39}$$

provided that $\widehat{m}\geq C_4\tilde{\mu}^2\tilde{r}^2\log(n)$.

Lemma 6. *Suppose the assumptions in Theorem 1 and*

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_t-\boldsymbol{S}\|_\infty^*\leq\frac{8\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_t)\subseteq Supp(\boldsymbol{S}^*),\\
&\|\boldsymbol{X}_t-\boldsymbol{X}^*\|_\infty\leq\frac{2\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big).
\end{aligned}\tag{40}$$

With probability at least $1-\frac{n_c\log^3(n_cn)}{n^2}$, we have

$$\|\boldsymbol{X}_{t+1}-\boldsymbol{X}^*\|_\infty\leq\frac{2\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t}|\lambda_k^*|\Big)\tag{41}$$

provided that $\widehat{m}\geq C_5\tilde{\mu}^2\tilde{r}^2\log^2(n)$.

Lemma 7. *Suppose the assumptions in Theorem 1 and*

$$\begin{aligned}
&\|\boldsymbol{H}_t\|_2\leq\frac{1}{60}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big),\\
&\|\boldsymbol{X}_{t+1}-\boldsymbol{X}^*\|_\infty\leq\frac{2\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t}|\lambda_k^*|\Big).
\end{aligned}\tag{42}$$

Then, we have

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_{t+1}-\boldsymbol{S}^*\|_\infty\leq\frac{7\tilde{\mu}\tilde{r}}{n_cn}\Big(|\lambda_{k+1}^*|+\Big(\frac{1}{2}\Big)^{t}|\lambda_k^*|\Big),\\
&and\qquad Supp(\widetilde{\boldsymbol{S}}_{t+1}-\boldsymbol{S}^*)\subseteq Supp(\boldsymbol{S}^*).
\end{aligned}\tag{43}$$

D. Proof of Theorem 1

The proof of Theorem 1 follows the similar framework established in AltProj [23] by inductions over k and t. Here, we are mainly focused on the inductions over k and t for (44). The induction over k follows naturally for the selected T, which is the iteration number of inner loop. The key part is the induction over t, and the critical steps are verified by applying Lemmas 5, 6 and 7 recursively. Lemmas 6 and 7 play the similar roles as Lemmas 7 and 9 in [23]. However, we need an extra lemma 5 to handle the additional item $\boldsymbol{H}_t$

caused by the partial observation since AltProj only considers the case of full observation.

Proof of Theorem 1. The proof is based on induction over t and k for the following equation:

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_t^{(k)} - \boldsymbol{S}^*\|_\infty \le \frac{7\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_t^{(k)} - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*),\\
&\|\boldsymbol{X}_t^{(k)} - \boldsymbol{X}^*\|_\infty \le \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big),
\end{aligned} \tag{44}$$

where we use $\boldsymbol{X}_t^{(k)}$ to represent the iteration $\boldsymbol{X}_t$ generated in the k-th outer loop, similar for $\widetilde{\boldsymbol{S}}_t^{(k)}$, $\boldsymbol{H}_t^{(k)}$ and $\xi_t^{(k)}$.

Base Case: When $k = 1$ and $t = 0$, $\boldsymbol{X}_0^{(1)}$ is initialized as $\boldsymbol{0}$. Since $\widetilde{\mathcal{H}}\boldsymbol{X}^*$ is $\tilde{\mu}$-incoherent, we have

$$\|\boldsymbol{X}^* - \boldsymbol{X}_0^{(1)}\|_\infty = \|\boldsymbol{X}^*\|_\infty = \|\widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty \le \frac{\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^*. \tag{45}$$

Note that the hard thresholding $\xi_0^{(1)}$ is initialized as $\frac{4\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^*$, for $\widetilde{\boldsymbol{S}}_0^{(1)}$, we consider three cases:

Case 1: If $S_{i,j}^* = 0$, then $(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} = \mathcal{T}_{\xi_0^{(1)}}(X_{i,j}^*)$.

$$|X_{i,j}^*| \le \frac{\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^* \le \xi_0^{(1)}. \tag{46}$$

Hence, $(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} = 0$.

Case 2: If $S_{i,j}^* \neq 0$ and $|M_{i,j}| > \xi_0^{(1)}$, then $(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} = S_{i,j}^* + X_{i,j}^*$.

$$|(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} - S_{i,j}^*| = |X_{i,j}^*| \le \frac{\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^*. \tag{47}$$

Case 3: If $S_{i,j}^* \neq 0$ and $|M_{i,j}| \le \xi_0^{(1)}$, then $(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} = 0$.

$$|(\widetilde{\boldsymbol{S}}_0^{(1)})_{i,j} - S_{i,j}^*| = |S_{i,j}^*| \le \xi_0^{(1)} + |X_{i,j}^*| \le \frac{5\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^*. \tag{48}$$

Hence, we have

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_0^{(1)} - \boldsymbol{S}^*\|_\infty \le \frac{5\tilde{\mu}\tilde{r}}{n_c n}\lambda_1^*,\\
&Supp(\widetilde{\boldsymbol{S}}_0^{(1)} - \boldsymbol{S}^*) = Supp(\boldsymbol{S}^*).
\end{aligned} \tag{49}$$

From (45) and (49), we know that (44) is true in the base case.

Induction over t: For any fixed $k \ge 0$, suppose that $\widetilde{\boldsymbol{S}}_t^{(k)}$ and $\boldsymbol{X}_t^{(k)}$ satisfy (44) for some $t \ge 0$. Then, according to Lemma 6, we have

$$\|\boldsymbol{X}_{t+1}^{(k)} - \boldsymbol{X}^*\|_\infty \le \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t}|\lambda_k^*|\Big). \tag{50}$$

Note in Lemma 5, (44) suggests that

$$\|\boldsymbol{H}_t^{(k)}\|_2 \le \frac{1}{60}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big), \tag{51}$$

with high probability. By Lemma 7, (50) and (51) give that

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_{t+1}^{(k)} - \boldsymbol{S}^*\|_\infty \le \frac{7\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t}|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_{t+1}^{(k)} - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*).
\end{aligned} \tag{52}$$

Hence, (44) is still valid for $\widetilde{\boldsymbol{S}}_{t+1}^{(k)}$ and $\boldsymbol{X}_{t+1}^{(k)}$.

Induction over k: Suppose at k^{th} stage, the initialization $\boldsymbol{X}_0^{(k)}$ and $\widetilde{\boldsymbol{S}}_0^{(k)}$ satisfy (44), that is

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_0^{(k)} - \boldsymbol{S}^*\|_\infty \le \frac{7\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + 2|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_0^{(k)} - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*),\\
&and \quad \|\boldsymbol{X}_0^{(k)} - \boldsymbol{X}^*\|_\infty \le \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + 2|\lambda_k^*|\Big).
\end{aligned} \tag{53}$$

From the discussion of induction over t above, we know that,

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_T^{(k)} - \boldsymbol{S}^*\|_\infty \le \frac{7\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{T-1}|\lambda_k^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_T^{(k)} - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*),\\
&\|\boldsymbol{X}_{T+1}^{(k)} - \boldsymbol{X}^*\|_\infty \le \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{T}|\lambda_k^*|\Big),
\end{aligned} \tag{54}$$

where $T = \log(4\tilde{\mu}\tilde{r}|\lambda_1^*|/\varepsilon)$.

Then, from Lemmas 4 and 5, we have

$$\begin{aligned}
\Big||\lambda_{k+1}^{(T)}| - |\lambda_{k+1}^*|\Big| \le& \|\boldsymbol{H}_T\|_2\\
\le& \frac{1}{60}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{T-1}|\lambda_k^*|\Big)\\
\le& \frac{1}{60}\Big(|\lambda_{k+1}^*| + \frac{\varepsilon}{2\tilde{\mu}\tilde{r}}\Big).
\end{aligned} \tag{55}$$

Now, we consider two cases,

Case 1: if $\frac{\eta}{\sqrt{n_c}}\lambda_{k+1}^{(T)} \le \frac{\varepsilon}{n_c n}$, (55) implies that $|\lambda_{k+1}^*| \le \frac{\varepsilon}{2\tilde{\mu}\tilde{r}}$. Hence,

$$\|\boldsymbol{X}_{T+1}^{(k)} - \boldsymbol{X}^*\|_\infty \le \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{T}|\lambda_k^*|\Big) \le \frac{\varepsilon}{n_c n}. \tag{56}$$

Similar results can be established for $\boldsymbol{S}_T$. Therefore, $\boldsymbol{X} = \boldsymbol{X}_{T+1}^{(k)}$ and $\boldsymbol{S} = \boldsymbol{S}_T^{(k)}$ satisfy (15) in Theorem 1.

Case 2: if $\frac{\eta}{\sqrt{n_c}}\lambda_{k+1}^{(T)} > \frac{\varepsilon}{n_c n}$, then (55) implies that $|\lambda_{k+1}^*| \ge \frac{\varepsilon}{6\tilde{\mu}\tilde{r}}$. Hence,

$$\begin{aligned}
\|\boldsymbol{X}_{T+1}^{(k)} - \boldsymbol{X}^*\|_\infty \le& \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{T}|\lambda_k^*|\Big)\\
\le& \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \frac{\varepsilon}{4\tilde{\mu}\tilde{r}}\Big)\\
\le& \frac{2\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+2}^*| + 2|\lambda_{k+1}^*|\Big).
\end{aligned} \tag{57}$$

Suppose we have an extra step

$$\widetilde{\boldsymbol{S}}_{T+1}^{(k)} = \mathcal{T}_{\xi_{T+1}^{(k)}}(\boldsymbol{M} - \boldsymbol{X}_{T+1}).$$

From Lemma 7, we have

$$\begin{aligned}
&\|\widetilde{\boldsymbol{S}}_{T+1}^{(k)} - \boldsymbol{S}^*\|_\infty \le \frac{7\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+2}^*| + 2|\lambda_{k+1}^*|\Big),\\
&Supp(\widetilde{\boldsymbol{S}}_{T+1}^{(k)} - \boldsymbol{S}^*) \subseteq Supp(\boldsymbol{S}^*).
\end{aligned} \tag{58}$$

$\boldsymbol{X}_0^{(k+1)} = \boldsymbol{X}_{T+1}^{(k)}$ is clear from Alg. 1, and it can also be verified that $\widetilde{\boldsymbol{S}}_0^{(k+1)} = \widetilde{\boldsymbol{S}}_{T+1}^{(k)}$. Hence, $\boldsymbol{X}_0^{(k+1)}$ and $\widetilde{\boldsymbol{S}}_0^{(k+1)}$ satisfy (44) as well.

Note that $\widetilde{\mathcal{H}}\boldsymbol{X}^*$ is at most rank-$2r$, we will meet the terminating condition anyway. If not, from case 2, we have the contradiction

$$0 = |\lambda_{r+1}^*| \ge \frac{\varepsilon}{6\tilde{\mu}\tilde{r}} > 0. \tag{59}$$

Hence, the algorithm has at most $r \cdot T$ iterations, and we need the size of samplings satisfies

$$m \geq rT\widehat{m} \geq \max(C_4, C_5)\mu^2 c_s^2 r^3 \log^2(n) T, \tag{60}$$

where the requirement on $\widehat{m}$ comes from Lemmas 5 and 6. □

E. Proof of Lemma 5

We first bound the spectral norm of $\boldsymbol{H}_{1,t}$ and $\boldsymbol{H}_{2,t}$ in (65) and (63), respectively. Then, the theoretical bound will be directly obtained by applying the triangle inequality in (38). Lemma 8 is a direct application of the standard Bernstein inequality, which shows that the operator $\hat{p}^{-1}\widetilde{\mathcal{H}}\mathcal{P}_{\Omega_{k,t}}$ can be close enough to its mean $\widetilde{\mathcal{H}}$. Though the definition of $\widetilde{\mathcal{H}}$ is different from that in [33], (61) still holds and can be proved by following the same framework in [33].

Lemma 8 ([33], Lemma 12)**.** *Let $\boldsymbol{H}_{2,t}$ satisfy the definition in* (25)*. Then, with probability at least $1 - n_c n^{-2}$, we have*

$$\|\boldsymbol{H}_{2,t}\|_2 \leq \sqrt{16\log(n)}\beta_t \tag{61}$$

provided that $\widehat{m} \geq 16\log(n)$.

Proof of Lemma 5. Since $\boldsymbol{H}_t = \boldsymbol{H}_{1,t} + \boldsymbol{H}_{2,t}$, we have $\|\boldsymbol{H}_t\|_2 \leq \|\boldsymbol{H}_{1,t}\|_2 + \|\boldsymbol{H}_{2,t}\|_2$.
From Lemma 8, we know that

$$\begin{aligned}\|\boldsymbol{H}_{2,t}\|_2 \leq& \sqrt{16\log(n)}\beta_t \\ \leq& \sqrt{\frac{16\log(n)}{\widehat{m}}} n_c n \|\boldsymbol{X}_t + \boldsymbol{S}_t - \boldsymbol{X}^* - \boldsymbol{S}^*\|_\infty.\end{aligned} \tag{62}$$

From the assumption, we know that

$$\begin{aligned}\|\boldsymbol{X}_t + \boldsymbol{S}_t - \boldsymbol{X}^* - \boldsymbol{S}^*\|_\infty \leq& \|\boldsymbol{X}_t - \boldsymbol{X}^*\|_\infty + \|\boldsymbol{S}_t - \boldsymbol{S}^*\|_\infty \\ \leq& \frac{9\tilde{\mu}\tilde{r}}{n_c n}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big).\end{aligned}$$

Hence, if $\widehat{m} \geq C_4 \tilde{\mu}^2 \tilde{r}^2 \log(n)$ with $C_4 \geq 4800^2$, we have

$$\|\boldsymbol{H}_{2,t}\|_2 \leq \frac{1}{120}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big). \tag{63}$$

For $\|\boldsymbol{H}_{1,t}\|_2$, from the assumption, we know that each row or column of $\boldsymbol{H}_{1,t}$ has at most $\alpha n_c n$ nonzero entries. Then, for any pair of unit vectors $\boldsymbol{z}$, $\boldsymbol{w} \in \mathbb{C}^{n_c n}$, we have

$$\begin{aligned}&|\boldsymbol{z}\boldsymbol{H}_{1,t}\boldsymbol{w}^H| \\ =&|\sum_{i_1,i_2} z_{i_1} w_{i_2} (\boldsymbol{H}_{1,t})_{i,j}| \leq \sum_{i_1,i_2} |z_{i_1} w_{i_2}| \cdot |(\boldsymbol{H}_{1,t})_{i,j}| \\ \leq& \frac{1}{2}\sum_{i_1,i_2}\big(|z_{i_1}|^2 + |w_{i_2}|^2\big)|(\boldsymbol{H}_{1,t})_{i,j}| \leq \alpha n_c n \|\boldsymbol{H}_{1,t}\|_\infty \\ \leq& \frac{1}{120}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big)\end{aligned} \tag{64}$$

with $\alpha \leq \frac{1}{840\tilde{\mu}\tilde{r}}$. Since (64) holds for any pair of unit vectors $\boldsymbol{z}$ and $\boldsymbol{w}$, we know that

$$\|\boldsymbol{H}_{1,t}\|_2 \leq \frac{1}{120}\Big(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|\Big), \tag{65}$$

which completes the whole proof. □

F. Proof of Lemma 6

The proof of Lemma 6 is built upon exploiting the Taylor expansion of the eigenvectors of $\widetilde{\mathcal{H}}\boldsymbol{X}^*$, where similar proof structures are presented in both [23] and [15]. However, neither [23] nor [15] considers missing data and bad data simultaneously, so the perturbation item $\boldsymbol{H}_t$ in our paper is different since both $\boldsymbol{H}_{1,t}$ and $\boldsymbol{H}_{2,t}$ are nonzero.

The following two lemmas are introduced to condense our proof of Lemma 6. Lemma 9 illustrates the relationship between the infinity norm and the spectral norm of a matrix, and it is a direct corollary of the incoherence definition. (66) in Lemma 10 first appears in the proof of Lemma 7 [23] and later is summarized in Lemma 13 [15].

Lemma 9 ([15], Lemma 12)**.** *Suppose $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is a symmetric matrix with rank r and incoherence μ, then for any symmetric matrix $\boldsymbol{B} \in \mathbb{C}^{n\times n}$, we have*

$$\|\boldsymbol{A}\boldsymbol{B}\boldsymbol{A} - \boldsymbol{A}\|_\infty \leq \frac{\mu r}{n}\|\boldsymbol{A}\boldsymbol{B}\boldsymbol{A} - \boldsymbol{A}\|_2.$$

Lemma 10 ([15], Lemma 13)**.** *Suppose $\boldsymbol{A}, \boldsymbol{E} \in \mathbb{C}^{n\times n}$ are two symmetric matrices. Let $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{E}$ and $\mathcal{Q}_k(\boldsymbol{B}) = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$ be the eigenvalue decomposition of the best rank-k approximation of $\boldsymbol{B}$. Then, if $\boldsymbol{\Lambda}^{-1}$ exists, we have*

$$\begin{aligned}&\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^H\boldsymbol{A}\|_2 \leq 3\|\boldsymbol{E}\|_2 + \frac{\|\boldsymbol{E}\|_2^2}{|\lambda_k(\boldsymbol{B})|} + |\lambda_{k+1}(\boldsymbol{B})|, \\ &\|\boldsymbol{A}\boldsymbol{U}\boldsymbol{\Lambda}^{-a}\boldsymbol{U}^H\boldsymbol{A}\|_2 \leq |\lambda_k(\boldsymbol{B})|^{-a}\Big(\|\boldsymbol{E}\|_2 + |\lambda_k(\boldsymbol{B})|\Big)^2, \forall a \geq 2.\end{aligned} \tag{66}$$

Proof of Lemma 6. Since $\|\boldsymbol{X}_{t+1} - \boldsymbol{X}^*\|_\infty = \|\widetilde{\mathcal{H}}^\dagger(\widetilde{\boldsymbol{L}}_{t+1} - \widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_\infty \leq \|\widetilde{\boldsymbol{L}}_{t+1} - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty$, it is sufficient to bound $\|\widetilde{\boldsymbol{L}}_{t+1} - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty$. Recall that $\widetilde{\boldsymbol{L}}_{t+1} = \mathcal{Q}_{2k}(\widetilde{\boldsymbol{W}}_t)$ is a rank-$2k$ symmetric matrix, let $\widetilde{\boldsymbol{L}}_{t+1} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{U}}^H$ be the eigen decomposition of $\widetilde{\boldsymbol{L}}_{t+1}$. On the other hand,

$$\widetilde{\boldsymbol{W}}_t = \widetilde{\mathcal{H}}\boldsymbol{X}^* + \boldsymbol{H}_t, \tag{67}$$

then for each eigenvector $\widetilde{\boldsymbol{u}}_i$ of $\widetilde{\boldsymbol{L}}_{t+1}$, we have

$$\big(\widetilde{\mathcal{H}}\boldsymbol{X}^* + \boldsymbol{H}_t\big)\widetilde{\boldsymbol{u}}_i = \lambda_i(\widetilde{\boldsymbol{W}}_t)\widetilde{\boldsymbol{u}}_i.$$

For $\forall\ i \leq k \leq r$, we know that $|\lambda_{2i}(\widetilde{\boldsymbol{W}}_t)| = |\lambda_{2i-1}(\widetilde{\boldsymbol{W}}_t)| = |\lambda_i^{(t)}| \geq |\lambda_k^{(t)}|$. From Lemma 5 and (67), we know that

$$|\lambda_k^{(t)} - \lambda_k^*| \leq \|\boldsymbol{H}_t\|_2 \leq \frac{1}{20}\lambda_k^*, \tag{68}$$

that is

$$\lambda_k^{(t)} \geq \frac{19}{20}\lambda_k^* > 0. \tag{69}$$

Then dividing by $\lambda_i(\widetilde{\boldsymbol{W}}_t)$ on both sides,

$$\left(\boldsymbol{I} - \frac{\boldsymbol{H}_t}{\lambda_i(\widetilde{\boldsymbol{W}}_t)}\right)\widetilde{\boldsymbol{u}}_i = \frac{1}{\lambda_i(\widetilde{\boldsymbol{W}}_t)}(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{u}}_i.$$

Then, with Taylor expansion,

$$\begin{aligned}\widetilde{\boldsymbol{u}}_i =& \left(\boldsymbol{I} - \frac{\boldsymbol{H}_t}{\lambda_i(\widetilde{\boldsymbol{W}}_t)}\right)^{-1}\frac{(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{u}}_i}{\lambda_i(\widetilde{\boldsymbol{W}}_t)} \\ =& \left(\boldsymbol{I} + \frac{\boldsymbol{H}_t}{\lambda_i(\widetilde{\boldsymbol{W}}_t)} + \left(\frac{\boldsymbol{H}_t}{\lambda_i(\widetilde{\boldsymbol{W}}_t)}\right)^2 + \cdots\right)\frac{(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{u}}_i}{\lambda_i(\widetilde{\boldsymbol{W}}_t)}.\end{aligned}$$

Hence,

$$\begin{aligned}
&\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{U}}^H - \widetilde{\mathcal{H}}\boldsymbol{X}^* \\
=&\sum_{a=0}^{\infty}\sum_{b=0}^{\infty}\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+1)}\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{\Lambda}}^{-(b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b \\
&- \widetilde{\mathcal{H}}\boldsymbol{X}^* \\
=&\sum_{a=0}^{\infty}\sum_{b=0}^{\infty}\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b - \widetilde{\mathcal{H}}\boldsymbol{X}^* \\
=&\big((\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-1}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*) - \widetilde{\mathcal{H}}\boldsymbol{X}^*\big) \\
&+ \sum_{a+b\geq 1}\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b.
\end{aligned}$$

Then,

$$\begin{aligned}
&\|\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{U}}^H - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty \\
\leq&\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-1}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*) - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty \\
&+ \sum_{a+b\geq 1}\|\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b\|_\infty \\
:=&I_0 + \sum_{a+b\geq 1} I_{a,b}.
\end{aligned}$$

For I_0, since $\widetilde{\boldsymbol{W}}_t - \widetilde{\mathcal{H}}\boldsymbol{X}^* = \boldsymbol{H}_t$, we have

$$\begin{aligned}
I_0 \overset{(a)}{\leq}& \frac{\tilde{\mu}\tilde{r}}{n_c n}\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-1}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*) - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_2 \\
\overset{(b)}{\leq}& \frac{\tilde{\mu}\tilde{r}}{n_c n}\Big(3\|\boldsymbol{H}_t\|_2 + \frac{\|\boldsymbol{H}_t\|_2^2}{|\lambda_k^{(t)}|} + |\lambda_{k+1}^{(t)}|\Big).
\end{aligned} \tag{70}$$

where (a) holds due to Lemma 9, and (b) comes from the first inequality in Lemma 10.

Again from $\widetilde{\boldsymbol{W}}_t - \widetilde{\mathcal{H}}\boldsymbol{X}^* = \boldsymbol{H}_t$, Lemma 4 tells us that

$$|\lambda_{k+1}^{(t)}| \leq \|\boldsymbol{H}_t\|_2 + |\lambda_{k+1}^*|. \tag{71}$$

On the other hand, from (69), we have

$$\frac{\|\boldsymbol{H}_t\|_2}{\lambda_k^{(t)}} \leq \frac{1/20}{1-1/20} \leq \frac{1}{19}. \tag{72}$$

Then,

$$I_0 \leq \frac{\tilde{\mu}\tilde{r}}{n_c n}\Big(5\|\boldsymbol{H}_t\|_2 + |\lambda_{k+1}^*|\Big). \tag{73}$$

For $I_{a,b}$ and $a+b \leq \log(n_c n)$, we have

$$\begin{aligned}
&I_{a,b} \\
=&\max_{i_1,i_2}\Big|\boldsymbol{e}_{i_1}^T\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b\boldsymbol{e}_{i_2}\Big| \\
\leq&\max_{i_1,i_2}\|\boldsymbol{e}_{i_1}^T\big(\boldsymbol{H}_t\big)^a\widetilde{\boldsymbol{U}}\|_2\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
&\cdot\|\boldsymbol{e}_{i_2}^T\big(\boldsymbol{H}_t\big)^b\widetilde{\boldsymbol{U}}\|_2 \\
\leq&\Big(C_3\beta_t\log(n) + \alpha n_c n\|\boldsymbol{H}_{1,t}\|_\infty\Big)^{a+b} \\
&\cdot\max_{i_1,i_2}\|\boldsymbol{e}_{i_1}^T\widetilde{\boldsymbol{U}}\|_2\|\boldsymbol{e}_{i_2}^T\widetilde{\boldsymbol{U}}\|_2\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\leq&\frac{\tilde{\mu}\tilde{r}}{n_c n}\Big(C_3\beta_t\log(n) + \alpha n_c n\|\boldsymbol{H}_{1,t}\|_\infty\Big)^{a+b} \\
&\cdot\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\overset{(c)}{\leq}&\frac{\tilde{\mu}\tilde{r}}{n_c n}\Big(\frac{1}{60}\nu_t\Big)^{a+b}\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2
\end{aligned}$$

with high probability, where $\nu_t = |\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^{t-1}|\lambda_k^*|$. Moreover, (c) holds since

$$C_3\beta_t\log(n) \leq \frac{1}{120}\nu_t, \quad \alpha n_c n\|\boldsymbol{H}_{1,t}\|_\infty \leq \frac{1}{120}\nu_t \tag{74}$$

provided that $\widehat{m} \geq C_5\tilde{\mu}^2\tilde{r}^2\log^2(n)$ and $\alpha \leq \frac{1}{840\tilde{\mu}\tilde{r}}$, where $C_5 = (1200C_3)^2$.

When $a+b \geq \log(n_c n)$, we have

$$\begin{aligned}
&I_{a,b} \\
=&\max_{i_1,i_2}|\boldsymbol{e}_{i_1}^T\big(\boldsymbol{H}_t\big)^a(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\big(\boldsymbol{H}_t\big)^b\boldsymbol{e}_{i_2}| \\
\leq&\max_{i_1,i_2}\|\boldsymbol{e}_{i_1}^T\big(\boldsymbol{H}_t\big)^a\widetilde{\boldsymbol{U}}\|_2\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
&\cdot\|\boldsymbol{e}_{i_2}^T\big(\boldsymbol{H}_t\big)^b\widetilde{\boldsymbol{U}}\|_2 \\
\leq&\|\boldsymbol{H}_t\|_2^{a+b}\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\overset{(d)}{\leq}&\Big(\frac{1}{60}\nu_t\Big)^{a+b}\|\boldsymbol{H}_t\|_2^{a+b}\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\leq&\Big(\frac{1}{2}\Big)^{a+b}\Big(\frac{1}{30}\nu_t\Big)^{a+b}\|\boldsymbol{H}_t\|_2^{a+b} \\
&\cdot\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\leq&\frac{\tilde{\mu}\tilde{r}}{n_c n}\Big(\frac{1}{30}\nu_t\Big)^{a+b}\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2,
\end{aligned}$$

where (d) holds from Lemma 5.

Next, using Lemma 10, we have

$$\begin{aligned}
&\|(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}^{-(a+b+1)}\widetilde{\boldsymbol{U}}^H(\widetilde{\mathcal{H}}\boldsymbol{X}^*)\|_2 \\
\leq&|\lambda_k^{(t)}|^{-(a+b-1)}\left(1 + \frac{\|\boldsymbol{H}_t\|_2}{|\lambda_k^{(t)}|}\right)^2 \leq 3|\lambda_k^{(t)}|^{-(a+b-1)},
\end{aligned} \tag{75}$$

where the last inequality comes from (72).

Since $\nu_t \leq 3|\lambda_k^*|$ for $t \geq 0$, we have

$$\begin{aligned}
\sum_{a+b\geq 1} I_{a,b} \leq& \sum_{a+b\geq 1}\frac{\tilde{\mu}\tilde{r}}{10n_c n}\left(\frac{30|\lambda_k^{(t)}|}{\nu_t}\right)^{-(a+b-1)}\nu_t \\
\leq& \sum_{a+b\geq 1}\frac{\tilde{\mu}\tilde{r}}{10n_c n}\left(\frac{10|\lambda_k^{(t)}|}{|\lambda_t^*|}\right)^{-(a+b-1)}\nu_t \\
\leq&\frac{\tilde{\mu}\tilde{r}}{2n_c n}\nu_t.
\end{aligned} \tag{76}$$

Hence, (73) and (76) suggest

$$\begin{aligned}
\|\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{U}} - \widetilde{\mathcal{H}}\boldsymbol{X}^*\|_\infty \leq&\frac{\tilde{\mu}\tilde{r}}{n_c n}\big(5\|\boldsymbol{H}_t\|_2 + |\lambda_{k+1}^*|\big) + \frac{\tilde{\mu}\tilde{r}}{2n_c n}\nu_t \\
\leq&\frac{\tilde{\mu}\tilde{r}}{n_c n}\big(\nu_t + |\lambda_{k+1}^*|\big) \\
\leq&\frac{2\tilde{\mu}\tilde{r}}{n_c n}\left(|\lambda_{k+1}^*| + \Big(\frac{1}{2}\Big)^t|\lambda_k^*|\right),
\end{aligned}$$

which completes the whole proof. □