TARM: A Turbo-Type Algorithm for Affine Rank Minimization

Zhipeng Xue, Xiaojun Yuan, *Senior Member, IEEE*, Junjie Ma, and Yi Ma, *Fellow, IEEE*

***Abstract*—The affine rank minimization (ARM) problem arises in many real-world applications. The goal is to recover a low-rank matrix from a small amount of noisy affine measurements. The original problem is NP-hard, and so directly solving the problem is computationally prohibitive. Approximate low-complexity solutions for ARM have recently attracted much research interest. In this paper, we design an iterative algorithm for ARM based on message passing principles. The proposed algorithm is termed turbo-type ARM (TARM), as inspired by the recently developed turbo compressed sensing algorithm for sparse signal recovery. We show that, for right-orthogonally invariant linear (ROIL) operators, a scalar function called state evolution can be established to accurately predict the behaviour of the TARM algorithm. We also show that TARM converges faster than the counterpart algorithms when ROIL operators are used for low-rank matrix recovery. We further extend the TARM algorithm for matrix completion, where the measurement operator corresponds to a random selection matrix. Slight improvement of the matrix completion performance has been demonstrated for the TARM algorithm over the state-of-the-art algorithms.**

***Index Terms*—Low-rank matrix recovery, matrix completion, affine rank minimization, state evolution, low-rank matrix denoising.**

I. Introduction

Low-rank matrices have found extensive applications in real-world applications including but not limit to remote sensing [1], recommendation systems [2], global positioning [3], and system identification [4]. In these applications, a fundamental problem is to recover an unknown matrix from a small number of observations by exploiting its low-rank property [5], [6]. Specifically, we consider a rank-r matrix $\boldsymbol{X}_0 \in \mathbb{R}^{n_1 \times n_2}$ with the integers r, n_1, and n_2 satisfying $r \ll \min(n_1, n_2)$. We aim to recover $\boldsymbol{X}_0$ from an affine measurement given by

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) \in \mathbb{R}^m \tag{1}$$

where $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is a linear map with $m < n_1 n_2 = n$. When $\mathcal{A}$ is a general linear operator such as Gaussian operators and partial orthogonal operators, we refer to the problem as *low-rank matrix recovery*; when $\mathcal{A}$ is a selector that outputs a subset of the entries of $\boldsymbol{X}_0$, we refer to the problem as *matrix completion*.

The problem can be formally cast as affine rank minimization (ARM):

$$\min_{\boldsymbol{X}} \ \operatorname{rank}(\boldsymbol{X}) \quad \text{s.t.} \ \boldsymbol{y} = \mathcal{A}(\boldsymbol{X}). \tag{2}$$

Problem (2) is NP-hard, and so solving (2) is computationally prohibitive. To reduce complexity, a popular alternative to (2) is the following nuclear norm minimization (NNM) problem:

$$\min_{\boldsymbol{X}} \ \|\boldsymbol{X}\|_* \quad \text{s.t.} \ \boldsymbol{y} = \mathcal{A}(\boldsymbol{X}). \tag{3}$$

In [7], Recht *et al.* proved that when the restricted isometry property (RIP) holds for the linear operator $\mathcal{A}$, the ARM problem in (2) is equivalent to the NNM problem in (3). The NNM problem can be solved by semidefinite programing (SDP). Existing convex solvers, such as the interior point method [4], can be employed to find a solution in polynomial time. However, SDP is computationally heavy, especially when applied to large-scale problems with high dimensional data. To address this issue, low-cost iterative methods, such as the singular value thresholding (SVT) method [8] and the proximal gradient algorithm [9], have been proposed to further reduce complexity at the cost of a certain amount of performance degradation.

In real-world applications, perfect measurements are rare, and noise is naturally introduced in the measurement process. That is, we want to recover $\boldsymbol{X}_0$ from a noisy measurement of

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{n} \tag{4}$$

where $\boldsymbol{n} \in \mathbb{R}^m$ is a Gaussian noise with zero-mean and covariance $\sigma^2 \boldsymbol{I}$ and is independent of $\mathcal{A}(\boldsymbol{X}_0)$. To recover the low-rank matrix $\boldsymbol{X}_0$ from (4), we turn to the following formulation of the stable ARM problem:

$$\min_{\boldsymbol{X}} \ \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X})\|_2^2 \quad \text{s.t.} \ \operatorname{rank}(\boldsymbol{X}) \le r. \tag{5}$$

Manuscript received April 12, 2019; revised September 2, 2019; accepted September 13, 2019. Date of publication October 2, 2019; date of current version October 22, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yuejie Chi. This work was supported by the National Key R&D Program of China under Grant 2018YFB1801105. This paper was presented in part at the IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, AB, Canada, April 2018 [35]. *(Corresponding author: Xiaojun Yuan.)*

Z. Xue is with the Center for Intelligent Networking and Communications, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: xuezhp@shanghaitech.edu.cn).

X. Yuan is with the Center for Intelligent Networking and Communications, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: xjyuan@uestc.edu.cn).

J. Ma is with the Department of Statistics, Columbia University, New York, NY 10027 USA (e-mail: jm4520@columbia.edu).

Y. Ma is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: yima@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TSP.2019.2944740

The problem in (5) is still NP-hard and difficult to solve. Several suboptimal algorithms have been proposed to yield approximate solutions to (5). For example, the author in [10] proposed an alternating minimization method to factorize rank-r matrix $\boldsymbol{X}_0$ as the product of two matrices with dimension $n_1 \times r$ and $r \times n_2$ respectively. This method is more efficient in storage than SDP and SVT methods, especially when large-dimension low-rank matrices are involved. A second approach borrows the idea of iterative hard thresholding (IHT) for compressed sensing. For example, the singular value projection (SVP) algorithm [11] for stable ARM can be viewed as a counterpart of the IHT algorithm [12] for compressed sensing. SVP solves the stable ARM problem by combining the projected gradient method with singular value decomposition (SVD). An improved version of SVP, termed normalized IHT (NIHT) [13], adaptively selects the step size of the gradient descent step of SVP, rather than using a fixed step size. These algorithms involve a projection step which projects a matrix into a low-rank space using truncated SVD. In [14], a Riemannian method, termed RGrad, was proposed to extend NIHT by projecting the search direction of gradient descent into a low dimensional space. Compared with the alternating minimization method, these IHT-based algorithms exhibit better convergence performance with lower computational complexity. Furthermore, the convergence of these IHT-based algorithms is guaranteed when a certain restricted isometry property (RIP) holds [11], [13], [14].

In this paper, we aim to design low-complexity iterative algorithms to solve the stable ARM problem based on message-passing principles [15], a different perspective from the existing approaches mentioned above. Specifically, we present a turbo-type algorithm, termed turbo-type affine rank minimization (TARM), for solving the stable ARM problem, as inspired by the turbo compressed sensing (Turbo-CS) algorithm for sparse signal recovery [15], [16]. Interestingly, although TARM is designed based on the idea of message passing, the resulting algorithm bears a similar structure to the gradient-based algorithms such as SVP and NIHT. A key difference of TARM from SVP and NIHT resides in an extra step in TARM for the calculation of the so-called extrinsic messages. With this extra step, TARM is able to find a better descent direction for each iteration, so as to achieve a much higher convergence rate than SVP and NIHT. For low-rank matrix recovery, we establish a state evolution technique to characterize the behaviour of the TARM algorithm when the linear operator $\mathcal{A}$ is right-orthogonally invariant (ROIL). We show that the state evolution accurately predicts the performance of the TARM algorithm. We also show that TARM runs faster and achieves successful recovery for a broader range of parameter settings than those of the existing algorithms including SVP, NIHT, RGrad, Riemannian conjugate gradient descent (RCG) [14], algrebraic pursuits (ALPS) [18], Bayesian affine rank minimization (BARM) [19], iterative reweighted least squares (IRLS) [20], and low-rank matrix fitting (LMAFit) [21]. We further extend the TARM algorithm for matrix completion (when the linear operator is chosen as a random selector). We show that TARM with carefully tuned parameters outperforms the counterpart algorithms, especially when the measurement rate is relatively low.

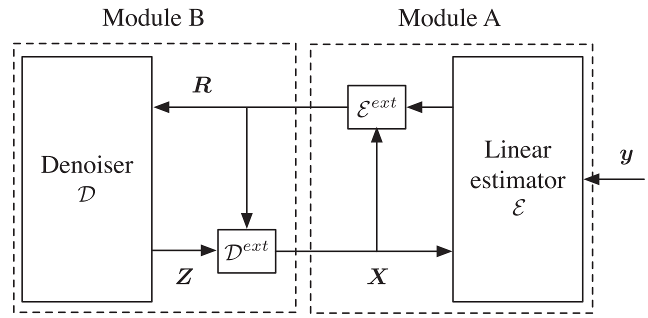

Fig. 1. The diagram of the TARM algorithm.

It is worthy of mentioning the early seminal work [22], [23] on message passing for solving compressed sensing problems. Particularly, the denoising-based approximated message passing (AMP) algorithm in [23] can be possibly used to recover signals with a general structure including low-rank matrices. It is known that the AMP algorithms perform well when the sensing matrix consists of independent and identically distributed (i.i.d.) elements; however, these algorithms suffer from considerable performance losses when applied to the case of non-i.i.d. sensing matrices. In this paper, the considered operator $\mathcal{A}$ is a ROIL operator for low-rank matrix recovery and a random selector for matrix completion. In both cases, the corresponding sensing matrices are partial orthogonal and thus far from being i.i.d. generated. As such, rather than following [22] and [23], we take the turbo message passing approach [15], [16] (designed to handle partial orthogonal sensing) in our algorithm design.

In this paper, we use bold capital letters to denote matrices and use bold lowercase letters to denote vectors. Denote by $\boldsymbol{X}^T$, $\mathrm{rank}(\boldsymbol{X})$, and $\mathrm{Tr}(\boldsymbol{X})$ the transpose, the rank, and the trace of matrix $\boldsymbol{X}$, respectively. Denote by $X_{i,j}$ the (i,j)-th entry of matrix $\boldsymbol{X}$, and by $\mathrm{vec}(\boldsymbol{X})$ the vector obtained by sequentially stacking the columns of $\boldsymbol{X}$. Denote by $\mathcal{A}$ a linear operator, and by $\mathcal{A}^T$ its adjoint linear operator. The inner product of two matrices is defined by $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{Y}^T)$. $\boldsymbol{I}$ denotes the identity matrix with an appropriate size. $\|\boldsymbol{X}\|_F$ and $\|\boldsymbol{X}\|_*$ denote the Frobenius norm and the nuclear norm of matrix $\boldsymbol{X}$ respectively. $\|\boldsymbol{x}\|_2$ denotes the l_2 norm of vector $\boldsymbol{x}$ and $\min(a, b)$ denotes the minimum of two numbers a and b.

II. The TARM Algorithm

As inspired by the success of the Turbo-CS algorithm for sparse signal recovery [15], [16], we borrow the idea of turbo message passing and present the TARM algorithm for the affine rank minimization problem in this section.

A. Algorithm Description

The diagram of TARM is illustrated in Fig. 1 and the detailed steps of TARM are presented in Algorithm 1. We use index t to denote the t-th iteration. There are two concatenated modules in TARM:

Algorithm 1: TARM for Affine Rank Minimization.

Input: $\mathcal{A}, \boldsymbol{y}, \boldsymbol{X}^{(0)} = \boldsymbol{0}, t = 0$
1: **while** the stopping criterion is not met **do**
2: $t = t + 1$
3: $\boldsymbol{R}^{(t)} = \boldsymbol{X}^{(t-1)} + \mu_t \mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)}))$
4: $\boldsymbol{Z}^{(t)} = \mathcal{D}(\boldsymbol{R}^{(t)})$
5: $\boldsymbol{X}^{(t)} = \mathcal{D}^{ext}(\boldsymbol{R}^{(t)}, \boldsymbol{Z}^{(t)}) = c_t(\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)})$
6: **end while**
Output: $\boldsymbol{Z}^{(t)}$

1) Module A:
   - *Step 1:* We estimate the low-rank matrix $\boldsymbol{X}_0$ via a linear estimator $\mathcal{E}(\cdot)$ based on the observation $\boldsymbol{y}$ and the input $\boldsymbol{X}^{(t-1)}$:
$$\mathcal{E}(\boldsymbol{X}^{(t-1)}) = \boldsymbol{X}^{(t-1)} + \gamma_t \mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)})) \quad (6)$$
   where γ_t is a certain given coefficient.
   - *Step 2:* The extrinsic estimate of $\boldsymbol{X}_0$ is then given by
$$\begin{aligned}\boldsymbol{R}^{(t)} &= \mathcal{E}^{ext}(\boldsymbol{X}^{(t-1)}, \mathcal{E}(\boldsymbol{X}^{(t-1)})) \\ &= \boldsymbol{X}^{(t-1)} + \frac{\mu_t}{\gamma_t}(\mathcal{E}(\boldsymbol{X}^{(t-1)}) - \boldsymbol{X}^{(t-1)}) \\ &= \boldsymbol{X}^{(t-1)} + \mu_t \mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)})) \quad (7)\end{aligned}$$
   where $\mathcal{E}^{ext}(\boldsymbol{X}^{(t-1)}, \mathcal{E}(\boldsymbol{X}^{(t-1)}))$ linearly combines the inputs $\boldsymbol{X}^{(t-1)}$ and $\mathcal{E}(\boldsymbol{X}^{(t-1)})$ with μ_t being a known coefficient.[1]

   We combine (6) and (7) into a linear estimation (Line 3 of Algorithm 1) since both operations are linear.
2) Module B:
   - *Step 1:* The output of Module A, i.e., $\boldsymbol{R}^{(t)}$, is passed to a denoiser $\mathcal{D}(\cdot)$ which suppresses the estimation error by exploiting the low-rank structure of $\boldsymbol{X}_0$ (Line 4 of Algorithm 1).
   - *Step 2:* The denoised output $\boldsymbol{Z}^{(t)}$ is passed to a linear function $\mathcal{D}^{ext}(\cdot,\cdot)$ which linearly combines $\boldsymbol{Z}^{(t)}$ and $\boldsymbol{R}^{(t)}$ (Line 5 of Algorithm 1).

   Denoiser $\mathcal{D}(\cdot)$ can be chosen as the best rank-r approximation [25] or the singular value thresholding (SVT) denoiser [26]. In this paper, we focus on the best rank-r approximation defined by
$$\mathcal{D}(\boldsymbol{R}) = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \quad (8)$$
   where σ_i, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ are respectively the i-th singular value and the corresponding left and right singular vectors of the input $\boldsymbol{R}$.

In the above, the superscript "ext" stands for *extrinsic message*. Step 2 of each module is dedicated to the calculation of extrinsic messages which is a major difference of TARM from its counterpart algorithms. In particular, we note that in TARM when $c_t = 1$ and $\alpha_t = 0$ for any t, the algorithm reduces to the SVP or NIHT algorithm (depending on the choice of μ_t). As such, the key difference of TARM from SVP and NITH resides in the choice of these parameters. By optimizing these parameters, the TARM algorithm aims to find a better descent direction in each iteration, so as to achieve a convergence rate much higher than SVP and NIHT.

[1]From the turbo principle [16], μ_t is chosen to ensure that the output error of Module A is uncorrelated with the input error, i.e. $\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0 \rangle = 0$. More detailed discussions can be found in Subsection II-B.

B. Determining the Parameters of TARM

In this subsection, we discuss how to determine the parameters $\{\mu_t\}$, $\{c_t\}$, and $\{\alpha_t\}$ based on turbo message passing. Turbo message passing was first applied to iterative decoding of turbo codes [24] and then extended for solving compressed sensing problems in [15], [16]. Following the turbo message passing rule in [16], three conditions for the calculation of extrinsic messages are presented:
- *Condition 1:*
$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0 \rangle = 0; \quad (9)$$
- *Condition 2:*
$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t)} - \boldsymbol{X}_0 \rangle = 0; \quad (10)$$
- *Condition 3:* For given $\boldsymbol{X}^{(t-1)}$,
$$\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 \text{ is minimized under (9) and (10).} \quad (11)$$

In the above, Conditions 1 and 2 follow from [16, Eq. 7, Eq. 14], and Condition 3 follows from Condition (ii) in [16]. Condition 1 ensures that the input and output estimation errors of Module A are uncorrelated. Similarly, Condition 2 ensures that the input and output estimation errors of Module B are uncorrelated. Condition 3 ensures that the output estimation error of Module B is minimized over $\{\mu_t, c_t, \alpha_t\}$ for each iteration t. Intuitively, in graphical-model based message passing, the out-going message on an edge is required to be independent of the incoming message on the edge (by excluding the incoming message in the calculation of the out-going message). Since uncorrelatedness implies independence for Gaussian random variables, (9) and (10) can be seen as necessary conditions for turbo Gaussian message passing. In this sense, the minimization in (11) can be interpreted as finding the best estimate of $\boldsymbol{X}^*$ for each iteration under the turbo Gaussian message passing framework.

We have the following lemma for $\{\mu_t, c_t, \alpha_t\}$, with the proof given in Appendix A.

Lemma 1: If Conditions 1–3 hold, then
$$\mu_t = \frac{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2}{\langle \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0) - \boldsymbol{n}, \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\rangle} \quad (12a)$$
$$\alpha_t = \frac{-b_t \pm \sqrt{b_t^2 - 4a_t d_t}}{2a_t} \quad (12b)$$
$$c_t = \frac{\langle \boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)} \rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}\|_F^2}, \quad (12c)$$
with
$$a_t = \|\boldsymbol{R}^{(t)}\|_F^2 \|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 \quad (13a)$$
$$b_t = -\|\boldsymbol{R}^{(t)}\|_F^2 \langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{Z}^{(t)} \rangle - \|\boldsymbol{Z}^{(t)}\|_F^2 \|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2$$

$$+\|\boldsymbol{Z}^{(t)}\|_F^2\langle \boldsymbol{R}^{(t)}-\boldsymbol{X}_0,\boldsymbol{X}_0\rangle \tag{13b}$$

$$d_t=\|\boldsymbol{Z}^{(t)}\|_F^2\langle \boldsymbol{R}^{(t)}-\boldsymbol{X}_0,\boldsymbol{Z}^{(t)}-\boldsymbol{X}_0\rangle. \tag{13c}$$

Remark 1: In (12b), α_t has two possible choices and only one of them minimizes the error in (11). From the discussion below (37), minimizing the square error in (11) is equivalent to minimizing $\|\boldsymbol{X}^{(t)}-\boldsymbol{R}^{(t)}\|_F^2$. We have

$$\begin{aligned}\left\|\boldsymbol{X}^{(t)}-\boldsymbol{R}^{(t)}\right\|_F^2 &= \left\|c_t(\boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)})-\boldsymbol{R}^{(t)}\right\|_F^2 &(14a)\\ &= -\frac{\langle \boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)},\boldsymbol{R}^{(t)}\rangle^2}{\|\boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)}\|_F^2}+\|\boldsymbol{R}^{(t)}\|_F^2 &(14b)\end{aligned}$$

where (14a) follows from substituting $\boldsymbol{X}^{(t)}$ in Line 5 of Algorithm 1, and (14b) follows by substituting c_t in (12c). Since $\|\boldsymbol{R}^{(t)}\|_F^2$ is invariant to α_t, minimizing $\|\boldsymbol{X}^{(t)}-\boldsymbol{R}^{(t)}\|_F^2$ is equivalent to maximizing $\frac{\langle \boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)},\boldsymbol{R}^{(t)}\rangle^2}{\|\boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)}\|_F^2}$. We choose α_t that gives a larger value of $\frac{\langle \boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)},\boldsymbol{R}^{(t)}\rangle^2}{\|\boldsymbol{Z}^{(t)}-\alpha_t\boldsymbol{R}^{(t)}\|_F^2}$.

Remark 2: Similarly to SVP [11] and NIHT [13], the convergence of TARM can be analyzed by assuming that the linear operator $\mathcal{A}$ satisfies the restricted isometry property (RIP). The convergence rate of TARM is much faster than those of NIHT and SVP (provided that $\{\alpha_t\}$ are sufficiently small). More detailed discussions are presented in Appendix B.

We emphasize that the parameters μ_t, α_t, and c_t in (12) are actually difficult to evaluate since $\boldsymbol{X}_0$ and $\boldsymbol{n}$ are unknown. This means that Algorithm 1 cannot rely on (12) to determine μ_t, α_t and c_t. In the following, we focus on how to approximately evaluate these parameters to yield practical algorithms. Based on different choices of the linear operator $\mathcal{A}$, our discussions are divided into two parts, namely, low-rank matrix recovery and matrix completion.

III. Low-Rank Matrix Recovery

A. Preliminaries

In this section, we consider recovering $\boldsymbol{X}_0$ from measurement in (4) when the linear operator $\mathcal{A}$ is right-orthogonally invariant and $\boldsymbol{X}_0$ is generated by following the random models described in [17].[2] Denote the vector form of an arbitrary matrix $\boldsymbol{X}\in\mathbb{R}^{n_1\times n_2}$ by $\boldsymbol{x}=\mathrm{vec}(\boldsymbol{X})=[\boldsymbol{x}_1^T,\boldsymbol{x}_2^T,\ldots,\boldsymbol{x}_n^T]^T$, where $\boldsymbol{x}_i$ is the ith column of $\boldsymbol{X}$. The linear operator $\mathcal{A}$ can be generally expressed as $\mathcal{A}(\boldsymbol{X})=\boldsymbol{A}\mathrm{vec}(\boldsymbol{X})=\boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ is a matrix representation of $\mathcal{A}$. The adjoint operator $\mathcal{A}^T:\mathbb{R}^m\to\mathbb{R}^{n_1\times n_2}$ is defined by the transpose of $\boldsymbol{A}$ with $\boldsymbol{x}'=\mathrm{vec}(\boldsymbol{X}')=\mathrm{vec}(\mathcal{A}^T(\boldsymbol{y}'))=\boldsymbol{A}^T\boldsymbol{y}'$. Consider a linear operator $\mathcal{A}$ with matrix form $\boldsymbol{A}$, the SVD of $\boldsymbol{A}$ is $\boldsymbol{A}=\boldsymbol{U}_A\boldsymbol{\Sigma}_A\boldsymbol{V}_A^T$, where $\boldsymbol{U}_A\in\mathbb{R}^{m\times m}$ and $\boldsymbol{V}_A\in\mathbb{R}^{n\times n}$ are orthogonal matrices and $\boldsymbol{\Sigma}_{\boldsymbol{A}}\in\mathbb{R}^{m\times n}$ is a diagonal matrix.

[2]The generation models of $\boldsymbol{X}_0$ can be found in the definitions before Assumption 2.4 in [17]. This choice allows us to borrow the results of [17] in our analysis; see (61).

Definition 1: If $\boldsymbol{V}_A$ is a Haar distributed random matrix [34] independent of $\boldsymbol{\Sigma}_A$, we say that $\mathcal{A}$ is a right-orthogonally invariant linear (ROIL) operator.[3]

We focus on two types of ROIL operators: partial orthogonal ROIL operators where the matrix form of $\mathcal{A}$ satisfies $\boldsymbol{A}\boldsymbol{A}^T=\boldsymbol{I}$, and Gaussian ROIL operators where the elements of $\boldsymbol{A}$ are i.i.d. Gaussian with zero mean. For convenience of discussion, the linear operator $\mathcal{A}$ is normalized such that the l_2-norm of each row of $\boldsymbol{A}$ is 1. It is worth noting that from the perspective of the algorithm, $\boldsymbol{A}$ is deterministic since $\boldsymbol{A}$ is known by the algorithm. However, the randomness of $\boldsymbol{A}$ has impact on parameter design and performance analysis, as detailed in what follows.

We now present two assumptions that are useful in determining the algorithm parameters in the following subsection.

Assumption 1: For each iteration t, Module A's input estimation error $\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0$ is independent of the orthogonal matrix $\boldsymbol{V}_A$ and the measurement noise $\boldsymbol{n}$.

Assumption 2: For each iteration t, the output error of Module A, given by $\boldsymbol{R}^{(t)}-\boldsymbol{X}_0$, is an i.i.d. Gaussian noise, i.e., the elements of $\boldsymbol{R}^{(t)}-\boldsymbol{X}_0$ are independently and identically drawn from $\mathcal{N}(0,v_t)$, where v_t is the output variance of Module A at iteration t.

The above two assumptions will be verified for ROIL operators by the numerical results presented in Subsection D. Similar assumptions have been introduced in the design of Turbo-CS in [15] (see also [27]). Later, these assumptions were rigorously analyzed in [28], [29] using the conditioning technique [30]. Based on that, state evolution was established to characterize the behavior of the Turbo-CS algorithm.

Assumptions 1 and 2 allow to decouple Module A and Module B in the analysis of the TARM algorithm. We will derive two mean square error (MSE) transfer functions, one for each module, to characterize the behavior of the TARM algorithm. The details will be presented in Subsection C.

B. Parameter Design

We now determine the parameters in (12) when ROIL operators are involved. We show that (12) can be approximately evaluated without the knowledge of $\boldsymbol{X}_0$. Since $\{c_t\}$ in (12c) can be readily computed given $\{\alpha_t\}$, we focus on the calculation of $\{\mu_t\}$ and $\{\alpha_t\}$.

We start with μ_t. From (12a), we have

$$\begin{aligned}\mu_t &= \frac{\|\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0\|_F^2}{\langle \mathcal{A}(\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0)-\boldsymbol{n},\mathcal{A}(\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0)\rangle} &(15a)\\ &\approx \frac{\|\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0\|_F^2}{\|\mathcal{A}(\boldsymbol{X}^{(t-1)}-\boldsymbol{X}_0)\|_2^2} &(15b)\\ &= \frac{1}{\tilde{\boldsymbol{x}}^T\boldsymbol{V}_A\boldsymbol{\Sigma}_A^T\boldsymbol{\Sigma}_A\boldsymbol{V}_A^T\tilde{\boldsymbol{x}}} &(15c)\\ &= \frac{1}{\boldsymbol{v}_A^T\boldsymbol{\Sigma}_A^T\boldsymbol{\Sigma}_A\boldsymbol{v}_A}\approx\frac{n}{m} &(15d)\end{aligned}$$

[3]In this section, we focus on ROIL operators so that the algorithm parameters can be determined by following the discussion in Subsection B. However, we emphasize that the proposed TARM algorithm applies to low-rank matrix recovery even when $\mathcal{A}$ is not a ROIL operator. In this case, the only difference is that the algorithm parameters shall be determined by following the heuristic methods described in Section III.

where (15b) holds approximately for a relatively large matrix size since $\langle \boldsymbol{n}, \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\rangle \approx 0$ from Assumption 1, (15c) follows by utilizing the matrix form of $\mathcal{A}$ and $\tilde{\boldsymbol{x}} = \frac{\text{vec}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)}{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F}$, and (15d) follows by letting $\boldsymbol{v}_A = \boldsymbol{V}_A^T \tilde{\boldsymbol{x}}$. $\boldsymbol{V}_A$ is Haar distributed and from Assumption 1 is independent of $\tilde{\boldsymbol{x}}$, implying that $\boldsymbol{v}_A$ is a unit vector uniformly distributed over the sphere $\|\boldsymbol{v}_A\|_2 = 1$. Then, the approximation in (15d) follows by noting $\text{Tr}(\boldsymbol{\Sigma}_A^T \boldsymbol{\Sigma}_A) = m$.

We next consider the approximation of α_t. We first note

$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t)} - \boldsymbol{X}_0\rangle \tag{16a}$$

$$= \langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, c_t(\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}) - \boldsymbol{X}_0\rangle \tag{16b}$$

$$\approx c_t \langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}\rangle \tag{16c}$$

where (16a) follows by substituting $\boldsymbol{X}^{(t)}$ in line 5 of Algorithm 1, and (16b) follows from $\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0\rangle \approx 0$ (implying that the error $\boldsymbol{R}^{(t)} - \boldsymbol{X}_0$ is uncorrelated with the original signal $\boldsymbol{X}_0$). Combining (16) and Condition 2 in (10), we have

$$\alpha_t = \frac{\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{Z}^{(t)}\rangle}{\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{R}^{(t)}\rangle} \tag{17a}$$

$$\approx \frac{\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \mathcal{D}(\boldsymbol{R}^{(t)})\rangle}{\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{R}^{(t)} - \boldsymbol{X}_0\rangle} \tag{17b}$$

$$\approx \frac{\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \mathcal{D}(\boldsymbol{R}^{(t)})\rangle}{n v_t} \tag{17c}$$

$$\approx \frac{1}{n}\sum_{i,j} \frac{\partial \mathcal{D}(\boldsymbol{R}^{(t)})_{i,j}}{\partial R_{i,j}^{(t)}} = \frac{1}{n}\text{div}(\mathcal{D}(\boldsymbol{R}^{(t)})) \tag{17d}$$

where (17b) follows from $\boldsymbol{Z}^{(t)} = \mathcal{D}(\boldsymbol{R}^{(t)})$ and $\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0\rangle \approx 0$, (17c) follows from the Assumption 2 that the elements of $\boldsymbol{R}^{(t)} - \boldsymbol{X}_0$ are i.i.d. Gaussian with zero mean and variance v_t, (17d) follows from Stein's lemma [31] since we approximate the entries of $\boldsymbol{R}^{(t)} - \boldsymbol{X}_0$ as i.i.d. Gaussian distributed.

C. State Evolution

We now characterize the performance of TARM for low-rank matrix recovery based on Assumptions 1 and 2.

We first consider the MSE behavior of Module A. Denote the output MSE of Module A at iteration t by

$$MSE_A^{(t)} = \frac{1}{n}\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2. \tag{18}$$

The following theorem gives the asymptotic MSE of Module A when the dimension of $\boldsymbol{X}_0$ goes to infinity, with the proof given in Appendix C.

Theorem 1: Assume that Assumption 1 holds, and let $\mu = \frac{n}{m}$. Then,

$$MSE_A^{(t)} \xrightarrow{\text{a.s.}} f(\tau_t) \tag{19}$$

as $m, n \to \infty$ with $\frac{m}{n} \to \delta$, where $\frac{1}{n}\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 \to \tau_t$ as $n \to \infty$. For partial orthogonal ROIL operator $\mathcal{A}$,

$$f(\tau) = \left(\frac{1}{\delta} - 1\right)\tau + \sigma^2 \tag{20a}$$

and for Gaussian ROIL operator $\mathcal{A}$,

$$f(\tau) = \frac{1}{\delta}\tau + \sigma^2. \tag{20b}$$

We now consider the MSE behavior of Module B. We start with the following useful lemma, with the proof given in Appendix D.

Lemma 2: Assume that $\boldsymbol{R}^{(t)}$ satisfies Assumption 2, $\|\boldsymbol{X}_0\|_F^2 = n$, and the empirical distribution of eigenvalue θ of $\frac{1}{n_2}\boldsymbol{X}_0^T \boldsymbol{X}_0$ converges almost surely to the density function $p(\theta)$ as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$, $\frac{r}{n_2} \to \lambda$. Then,

$$\alpha_t \xrightarrow{\text{a.s.}} \alpha(v_t) \tag{21a}$$

$$c_t \xrightarrow{\text{a.s.}} c(v_t) \tag{21b}$$

as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$, $\frac{r}{n_2} \to \lambda$, where

$$\alpha(v) = \left|1 - \frac{1}{\rho}\right|\lambda + \frac{1}{\rho}\lambda^2 + 2\left(\min\left(1, \frac{1}{\rho}\right) - \frac{\lambda}{\rho}\right)\lambda\Delta_1(v) \tag{22a}$$

$$c(v) = \frac{1 + \lambda(1+\frac{1}{\rho})v + \lambda v^2\Delta_2 - \alpha(v)(1+v)}{(1-2\alpha(v))(1+\lambda(1+\frac{1}{\rho})v + \lambda v^2\Delta_2) + \alpha(v)^2(1+v)} \tag{22b}$$

with Δ_1 and Δ_2 defined by

$$\Delta_1(v) = \int_0^\infty \frac{(v+\theta^2)(\rho v + \theta^2)}{(\sqrt{\rho}v - \theta^2)^2} p(\theta) d\theta \tag{23a}$$

$$\Delta_2 = \int_0^\infty \frac{1}{\theta^2} p(\theta) d\theta. \tag{23b}$$

Denote the output MSE of Module B at iteration t by

$$MSE_B^{(t)} = \frac{1}{n}\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2. \tag{24}$$

The output MSE of Module B is characterized by the following theorem.

Theorem 2: Assume that Assumption 2 holds, and let $\|\boldsymbol{X}_0\|_F^2 = n$. Then, the output MSE of Module B

$$MSE_B^{(t)} \xrightarrow{\text{a.s.}} g(v_t) \tag{25}$$

as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$, $\frac{r}{n_2} \to \lambda$, where

$$g(v_t) \triangleq \frac{v_t - \lambda\left(1+\frac{1}{\rho}\right)v_t - \lambda v_t^2\Delta_2}{\frac{v_t - \lambda\left(1+\frac{1}{\rho}\right)v_t - \lambda v_t^2\Delta_2}{1+\lambda\left(1+\frac{1}{\rho}\right)v_t + \lambda v_t^2\Delta_2}\alpha(v_t)^2 + (1-\alpha(v_t))^2} - v_t \tag{26}$$

α and Δ_2 are given in Lemma 2, and $\frac{1}{n}\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 \xrightarrow{\text{a.s.}} v_t$.

Remark 3: Δ_1 and Δ_2 in (23) may be difficult to obtain since $p(\theta)$ is usually unknown in practical scenarios. We now

TABLE I
μ_t CALCULATED BY (10A) FOR THE 1ST TO 8TH ITERATIONS OF ONE RANDOM REALIZATION OF THE ALGORITHM WITH A PARTIAL ORTHOGONAL ROIL OPERATOR. $n_1 = n_2 = 1000$, $r = 30$, $\sigma = 10^{-2}$

iteration t	1	2	3	4	5	6	7	8
$\frac{n}{m} = 2.5$	2.5005	2.4833	2.4735	2.4440	2.4455	2.4236	2.4465	2.4244
$\frac{n}{m} = 3.3333$	3.3474	3.3332	3.3045	3.2734	3.2512	3.2112	3.2593	3.2833
$\frac{n}{m} = 5$	5.0171	4.9500	4.9204	4.9141	4.8563	4.8102	4.7864	4.8230

introduce an approximate MSE expression that does not depend on $p(\theta)$:

$$g(v_t) \approx \bar{g}(v_t) \triangleq \frac{v_t - \lambda(1+\frac{1}{\rho})v_t}{(1-\alpha)^2} - v_t \tag{27}$$

where $\alpha = \alpha(0) = |1 - \frac{1}{\rho}|\lambda - \frac{1}{\rho}\lambda^2 + 2\min(1, \frac{1}{\rho})\lambda$. Compared with $g(v_t)$, $\bar{g}(v_t)$ omits two terms $-\lambda v_t^2 \Delta$ and $\frac{v_t - \lambda(1+\frac{1}{\rho})v_t - \lambda v_t^2\Delta}{1+\lambda(1+\frac{1}{\rho})v_t + \lambda v_t^2 \Delta}\alpha(v_t)^2$ and replaces $\alpha(v_t)$ by α. Recall that v_t is the mean square error at the t-iteration. As the iteration proceeds, we have $v_t \ll 1$, and hence $g(v_t)$ can be well approximated by $\bar{g}(v_t)$, as seen later from Fig. 3.

Combining Theorems 1 and 2, we can characterize the MSE evolution of TARM by

$$v_t = f(\tau_t) \tag{28a}$$

$$\tau_{t+1} = g(v_t). \tag{28b}$$

The fixed point of TARM's MSE evolution in (28) is given by

$$\tau^* = g(f(\tau^*)). \tag{29}$$

The above fixed point equation can be used to analysis the phase transition curves of the TARM algorithm. It is clear that the fixed point τ^* of (29) is a function of $\{\delta, \rho, \lambda, \Delta, \sigma\}$. For any given $\{\delta, \rho, \lambda, \Delta, \sigma\}$, we say that the TARM algorithm is successful if the corresponding τ^* is below a certain predetermined threshold. The critical values of $\{\delta, \rho, \lambda, \Delta, \sigma\}$ define the phase transition curves of the TARM algorithm.

D. *Numerical Results*

Simulation settings are as follows. For the case of partial orthogonal ROIL operators, we generate a partial orthogonal ROIL operator with the matrix form

$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{W}\boldsymbol{\Theta} \tag{30}$$

where $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is a random selection matrix, $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is a discrete cosine transform (DCT) matrix, and $\boldsymbol{\Theta}$ is a diagonal matrix with diagonal entries being 1 or -1 randomly. For the case of Gaussian ROIL operators, we generate an i.i.d. Gaussian random matrix of size $m \times n$ with elements drawn from $\mathcal{N}(0, \frac{1}{n})$. The rank-$r$ matrix $\boldsymbol{X}_0 \in \mathbb{R}^{n_1\times n_2}$ is generated by the product of two i.i.d. Gaussian matrices of size $n_1 \times r$ and $r \times n_2$.

1) Verification of the Assumptions: We first verify Assumption 1 using Table I. Recall that if Assumption 1 holds, the approximations in the calculation of μ_t in (15) become accurate. Thus, we compare the value of μ_t calculated by (12a) with $\mu_t = \frac{n}{m}$ by (15). We record the μ_t of the first 8 iterations of TARM in Table I for low-rank matrix recovery with a partial orthogonal ROIL operator. As shown in Table I, the approximation $\mu_t = \frac{n}{m}$ is close to the real value calculated by (12a) which serves as an evidence of the validity of Assumption 1. We then verify Assumption 2 using Fig. 2, where we plot the QQplots of the input estimation errors of Module A with partial orthogonal and Gaussian ROIL operators. The QQplots show that the output errors of Module A closely follow a Gaussian distribution, which agrees with Assumption 2.

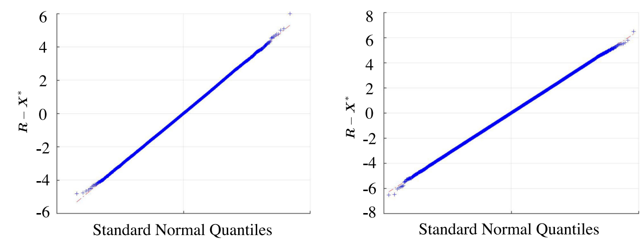

Fig. 2. The QQplots of the output error of Module B in the 2nd iteration of TARM. Left: $\mathcal{A}$ is a Gaussian ROIL operator. Right: $\mathcal{A}$ is a partial orthogonal ROIL operator. Simulation settings: $n_1 = 100$, $n_2 = 120$, $\frac{m}{n_1 n_2} = 0.3$, $\frac{r}{n_2} = 0.25$, $\sigma^2 = 0$.

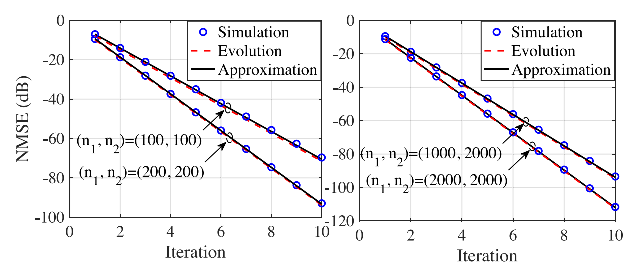

Fig. 3. Left: State evolution of TARM for partial orthogonal ROIL operator. $r = 40, m/n = m/(n_1 n_2) = 0.35, \sigma^2 = 0$. The size of $\boldsymbol{X}_0$ is shown in the plot. Right: State evolution of TARM for Gaussian ROIL operator. $r = 4, m/n = 0.35, \sigma^2 = 0$. The size of $\boldsymbol{X}_0$ is shown in the plot.

2) State Evolution: We now verify the state evolution of TARM given in (28). We plot the simulation performance of TARM and the predicted performance by the state evolution in Fig. 3. From the two subfigures in Fig. 3, we see that the state evolution of TARM is accurate when the dimension of $\boldsymbol{X}_0$ is large enough for both partial orthogonal and Gaussian ROIL operators. We also see that the state evolution with $g(\cdot)$ replaced by the approximation in (27) (referred to as "Approximation" in Fig. 3) provides reasonably accurate performance predictions. This makes the upper bound very useful since it does not require the knowledge of the singular value distribution of $\boldsymbol{X}_0$.

3) Performance Comparisons: We compare TARM with the existing algorithms for low-rank matrix recovery problems with

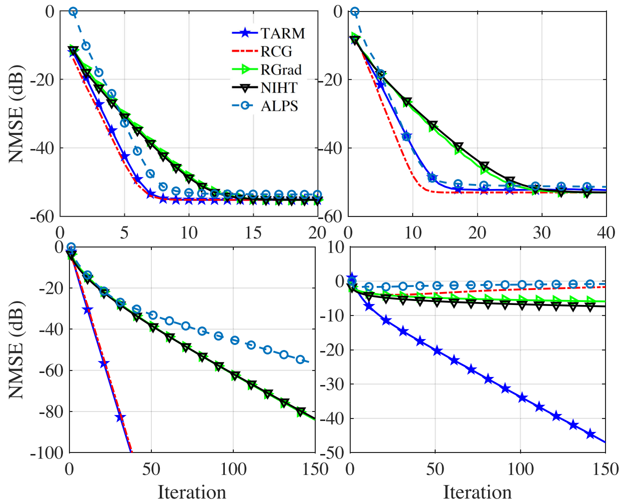

Fig. 4. Comparison of algorithms. Top left: $\mathcal{A}$ is a partial orthogonal ROIL operator with $n_1 = n_2 = 1000, r = 50, m/n = 0.39, \sigma^2 = 10^{-5}$. Top right: $\mathcal{A}$ is a Gaussian ROIL operator with $n_1 = n_2 = 80, r = 10, p = (n_1 + n_2 - r) \times r, m/p = 3, \sigma^2 = 10^{-5}$. Bottom left: $\mathcal{A}$ is a partial orthogonal ROIL operator with $n_1 = n_2 = 1000, r = 20, m/n = 0.07, \sigma^2 = 0$. Bottom right: $\mathcal{A}$ is a partial orthogonal ROIL operator with $n_1 = n_2 = 1000, r = 20, m/n = 0.042, \sigma^2 = 0$.

partial orthogonal and Gaussian ROIL operators.[4] The following algorithms are involved: Normalized Iterative Hard Thresholding (NIHT) [13], Riemannian Gradient Descent (RGrad) [14], Riemannian Conjugate Gradient Descent (RCG) [14], and ALPS [18]. We compare these algorithms under the same settings. We plot the per iteration normalized mean square error (NMSE) defined by $\frac{\|\boldsymbol{X}^{out} - \boldsymbol{X}_0\|_F}{\|\boldsymbol{X}_0\|_F}$ where $\boldsymbol{X}^{out}$ is the output of an algorithm in Fig. 4. From Fig. 4, we see that TARM converges much faster than NIHT, RGrad, and ALPS for both Gaussian ROIL operators and partial orthogonal ROIL operators. Moreover, from the last plot in Fig. 4, we see TARM converges under extremely low measurement rate while the other algorithms diverge. More detailed performance comparisons are given in Section V.

4) Empirical Phase Transition: The phase transition curve characterized the tradeoff between measurement rate δ and the largest rank r that an algorithm succeeds in the recovery of $\boldsymbol{X}_0$. Throughout the paper, we consider an algorithm to be successful in recovering the low-rank matrix $\boldsymbol{X}_0$ when the following conditions are satisfied: 1) the normalized mean square error $\frac{\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2}{\|\boldsymbol{X}_0\|_F^2} \leq 10^{-6}$; 2) the iteration number $t < 1000$. The dimension of the manifold of $n_1 \times n_2$ matrices of rank r is $r(n_1 + n_2 - r)$ [32]. Thus, for any algorithm, the minimal number of measurements for successful recovery is $r(n_1 + n_2 - r)$, i.e., $m \geq r(n_1 + n_2 - r)$. Then, an upper bound for successful recovery is $r \leq \frac{n_1 + n_2 - \sqrt{(n_1+n_2)^2 - 4m}}{2}$. In Fig. 5, we plot the phase transition curves of the algorithms mentioned before. From Fig. 5, we see that the phase transition curve of TARM is the closest to the upper bound and considerably higher than the curves of NIHT and RGrad.

[4]The code of TARM is available at https://github.com/xuezhp/tarm.

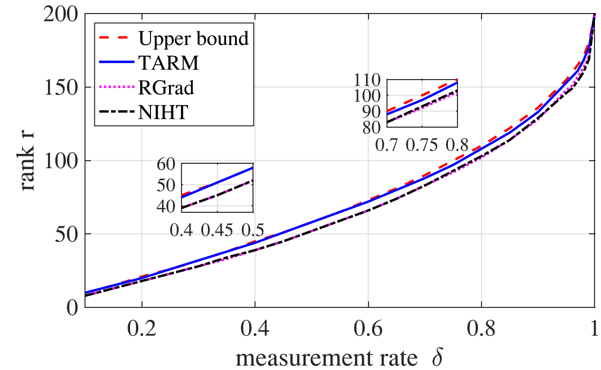

Fig. 5. The phase transition curves of various low-rank matrix recovery algorithms with a partial orthogonal ROIL operator. $n_1 = n_2 = 200, \sigma^2 = 0$. The region below each phase transition curve corresponds to the situation that the corresponding algorithm successfully recovers $\boldsymbol{X}_0$.

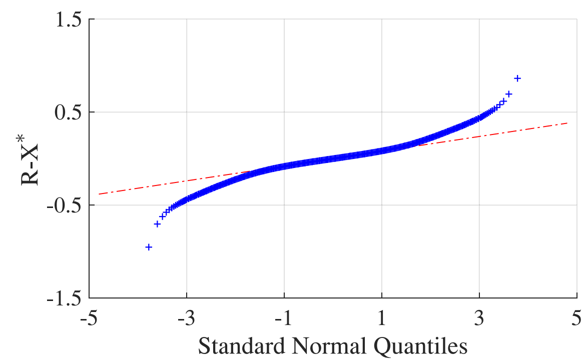

Fig. 6. The QQplots of the output error of Module A in the 5th iteration of TARM for matrix completion. Simulation settings: $n_1 = 800, n_2 = 800, r = 50, \frac{m}{n_1 n_2} = 0.3, \sigma^2 = 0$.

IV. Matrix Completion

In this section, we consider TARM for the matrix completion problem, where the linear operator $\mathcal{A}$ is a selector which selects a subset of the elements of the low-rank matrix $\boldsymbol{X}_0$. With such a choice of $\mathcal{A}$, the two assumptions in Section III for low-rank matrix recovery do not hold any more; see, e.g., Fig. 6. Thus, μ_t given in (15) and α_t in (17) cannot be used for matrix completion. We next discuss how to design μ_t and α_t for matrix completion.[5]

A. Determining μ_t

The TARM algorithm is similar to SVP and NIHT as aforementioned. These three algorithms are all SVD based and a gradient descent step is involved at each iteration. The choice of descent step size μ_t is of key importance. In [13], [14], μ_t are chosen adaptively based on the idea of the steepest descent. Due to the similarity between TARM and NIHT, we follow the

[5]We emphasize that the approaches described in Subsection IV-A and IV-B can also be used to determine the algorithm parameters for low-rank matrix recovery when $\mathcal{A}$ is not a ROIL operator.

methods in [13], [14] and choose μ_t as

$$\mu_t = \frac{\|\mathcal{P}_{\mathcal{S}}^{(t)}(\mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t)})))\|_F^2}{\|\mathcal{A}(\mathcal{P}_{\mathcal{S}}^{(t)}(\mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t)}))))\|_2^2} \tag{31}$$

where $\mathcal{P}_{\mathcal{S}}^{(t)} : \mathbb{R}^{n_1 \times n_2} \to \mathcal{S}$ denotes a projection operator with $\mathcal{S}$ being a predetermined subspace of $\boldsymbol{R}^{n_1 \times n_2}$. The subspace $\mathcal{S}$ can be chosen as the left singular vector space of $\boldsymbol{X}^{(t)}$, the right singular vector space of $\boldsymbol{X}^{(t)}$, or the direct sum of the two subspaces [14]. Let the SVD of $\boldsymbol{X}^{(t)}$ be $\boldsymbol{X}^{(t)} = \boldsymbol{U}^{(t)}\boldsymbol{\Sigma}^{(t)}(\boldsymbol{V}^{(t)})^T$. Then, the corresponding three projection operators are given respectively by

$$\mathcal{P}_{\mathcal{S}_2}^{(t)}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{V}^{(t)}(\boldsymbol{V}^{(t)})^T \tag{32a}$$

$$\mathcal{P}_{\mathcal{S}_1}^{(t)}(\boldsymbol{X}) = \boldsymbol{U}^{(t)}(\boldsymbol{U}^{(t)})^T\boldsymbol{X} \tag{32b}$$

$$\begin{aligned}\mathcal{P}_{\mathcal{S}_3}^{(t)}(\boldsymbol{X}) &= \boldsymbol{U}^{(t)}(\boldsymbol{U}^{(t)})^T\boldsymbol{X} + \boldsymbol{X}\boldsymbol{V}^{(t)}(\boldsymbol{V}^{(t)})^T \\ &\quad - \boldsymbol{U}^{(t)}(\boldsymbol{U}^{(t)})^T\boldsymbol{X}\boldsymbol{V}^{(t)}(\boldsymbol{V}^{(t)})^T.\end{aligned} \tag{32c}$$

By combining (32) with (31), we obtain three different choices of μ_t. Later, we present numerical results to compare the impact of different choices of μ_t on the performance of TARM.

B. Determining α_t and c_t

The linear combination parameters α_t and c_t in TARM is difficult to evaluate since Assumptions 1 and 2 do not hold for TARM in the matrix completion problem. Recall that c_t is determined by α_t through (12c). So, we only need to determine α_t. In the following, we propose three different approaches to evaluate α_t.

The first approach is to choose α_t as in (17):

$$\alpha_t = \frac{\operatorname{div}(\mathcal{D}(\boldsymbol{R}^{(t)}))}{n}. \tag{33}$$

We use the Monte Carlo method to compute the divergence. Specifically, the divergence of $\mathcal{D}(\boldsymbol{R}^{(t)})$ can be estimated by [23]

$$\operatorname{div}(\mathcal{D}(\boldsymbol{R}^{(t)})) = \mathrm{E}_{\boldsymbol{N}}\left[\left\langle \frac{\mathcal{D}(\boldsymbol{R}^{(t)} + \epsilon\boldsymbol{N}) - \mathcal{D}(\boldsymbol{R}^{(t)})}{\epsilon}, \boldsymbol{N}\right\rangle\right] \tag{34}$$

where $\boldsymbol{N} \in \mathbb{R}^{n_1 \times n_2}$ is a random Gaussian matrix with zero mean and unit variance entries, and ϵ is a small real number. The expectation in (34) can be approximated by sample mean. When the size of $\boldsymbol{R}^{(t)}$ is large, one sample is good enough for approximation. In our following simulations, we choose $\epsilon = 0.001$ and use one sample to approximate (34).

We now describe the second approach. Recall that we choose c_t according to (12c) to satisfy Condition 2: $\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t)} - \boldsymbol{X}_0\rangle = 0$. Since $\boldsymbol{X}_0$ is unknown, finding α_t to satisfy Condition 2 is difficult. Instead, we try to find α_t that minimizes the transformed correlation of the two estimation errors:

$$\left|\langle \mathcal{A}(\boldsymbol{R}^{(t)} - \boldsymbol{X}_0), \mathcal{A}(\boldsymbol{X}^{(t)} - \boldsymbol{X}_0)\rangle\right| \tag{35a}$$

$$= \left|\langle \mathcal{A}(\boldsymbol{R}^{(t)}) - \boldsymbol{y}, \mathcal{A}(\boldsymbol{X}^{(t)}) - \boldsymbol{y}\rangle\right| \tag{35b}$$

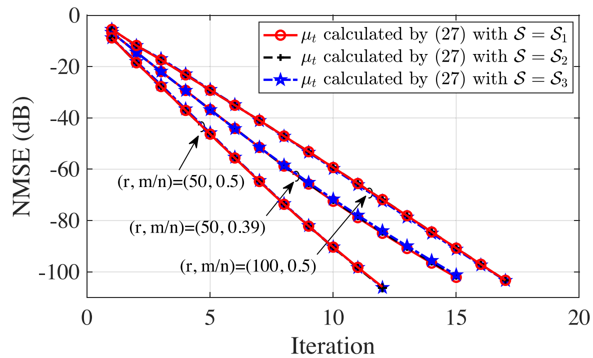

Fig. 7. Comparison of the TARM algorithms for matrix completion with different choices of μ_t. $n_1 = n_2 = 1000, \sigma^2 = 0$.

$$= \left|\left\langle \frac{\langle \boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)}\rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2}\mathcal{A}(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}) - \boldsymbol{y}, \mathcal{A}(\boldsymbol{R}^{(t)}) - \boldsymbol{y}\right\rangle\right|. \tag{35c}$$

The minimization of (35d) over α_t can be done by an exhaustive search over a small neighbourhood of zero.

The third approach is to set α_t as the asymptotic limit given in (21a). We next provide numerical simulations to show the impact of the above three different choices of α_t on the performance of TARM.

C. Numerical Results

In this subsection, we compare the performance of TARM algorithms with different choices of μ_t and α_t. We also compare TARM with the existing matrix completion algorithms, including RCG [14], RGrad [14], NIHT [13], ALPS [18], LMAFit [21], and LRGeomCG [32].[6] The matrix form $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of the matrix completion operator $\mathcal{A}$ is chosen as a random selection matrix (with randomly selected rows from a permutation matrix). The low-rank matrix $\boldsymbol{X}_0 \in \mathbb{R}^{n_1 \times n_2}$ is generated by the multiplication of two random Gaussian matrices of size $n_1 \times r$ and $r \times n_2$.

1) Non-Gaussianity of the Output Error of Module A: In Fig. 6, we plot the QQplot of the input estimation errors of Module A of TARM for matrix completion. The QQplot shows that the distribution of the estimation errors of Module A is non-Gaussian. Thus, Assumption 2 does not hold for matrix completion.

2) Comparisons of Different Choices of μ_t: We compare the TARM algorithms with μ_t in (31) and the subspace $\mathcal{S}$ given by (32), as shown in Fig. 7. We see that the performance of TARM

[6]The codes of these algorithms are available at https://github.com/xuezhp/tarm. All these codes are implemented purely on Matlab 2018 platform for a fair comparison. Specifically, compared with the publicly available codes for LMAFit and LRGeomCG, our codes do not use MEX files to speed up low-rank factorizations. This will slow down the LMAFit and LRGeomCG algorithms by approximately an order of magnitude in the simulation results presented later in Table III.

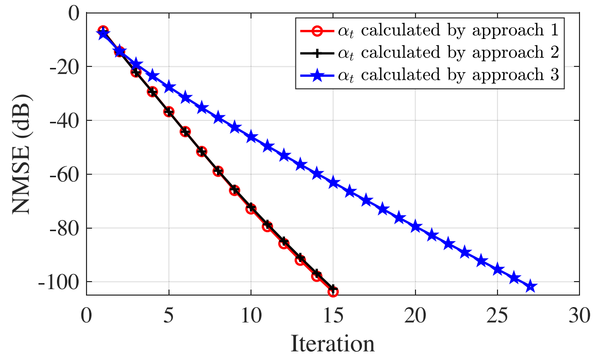

Fig. 8. Comparison of the TARM algorithms for matrix completion with different choices of α_t. $n_1 = n_2 = 1000, r = 50, \frac{m}{n_1 n_2} = 0.39, \sigma^2 = 0$.

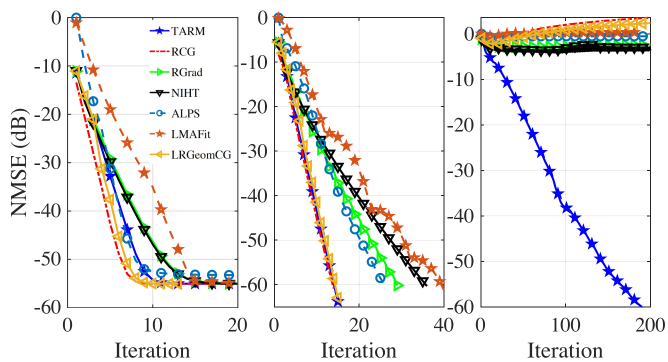

Fig. 9. Comparison of algorithms for matrix completion. Left: $n_1 = n_2 = 1000, r = 50, m/n = 0.39, \sigma^2 = 10^{-5}$. Middle: $n_1 = n_2 = 1000, r = 20, m/n = 0.12, \sigma^2 = 0$. Right: $n_1 = n_2 = 1000, r = 20, m/n = 0.045, \sigma = 0$.

is not sensitive to the three choices of $\mathcal{S}$ in (32). In the following, we always choose μ_t with $\mathcal{S}$ given by (32a).

3) Comparisons of Different Choices of α_t: We compare the TARM algorithms with α_t given by the three different approaches in Subsection B. As shown in Fig. 8, the first two approaches perform close to each other; the third approach performs considerably worse than the first two. Note that the first approach involves the computation of the divergence in (34), which is computationally demanding. Thus, we henceforth choose α_t based on the second approach in (35).

4) Performance Comparisons: We compare TARM with the existing algorithms for matrix completion in Fig. 9. We see that in the left and middle plots (with measurement rate $m/n = 0.39$ and 0.12), TARM performs close to LRGeomCG and RCG, while in the right plot (with $m/n = 0.045$), TARM significantly outperforms the other algorithms. More detailed performance comparisons are given in Section V.

5) Empirical Phase Transition: Similar to the case of low-rank matrix recovery. We consider an algorithm to be successful in recovering the low-rank matrix $\boldsymbol{X}_0$ when the following conditions are satisfied: 1) the normalized mean square error $\frac{\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2}{\|\boldsymbol{X}_0\|_F^2} \leq 10^{-6}$; 2) the iteration number $t < 1000$. In Fig. 5, we plot the phase transition curves of the algorithms mentioned before. From Fig. 10, we see that the phase transition of TARM

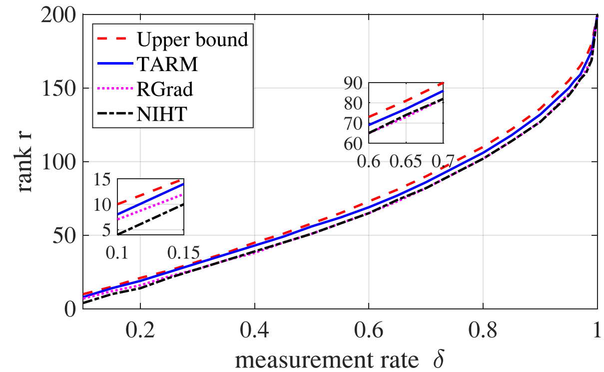

Fig. 10. The phase transition curves of various matrix completion algorithms. $n_1 = n_2 = 200, \sigma^2 = 0$. For each algorithm, the region below the phase transition curve corresponds to the successful recovery of $\boldsymbol{X}_0$.

is the closest to the upper bound and considerably higher than the curves of NIHT and RGrad.

V. More Numerical Results

We now compare the performance of TARM with other existing algorithms including LMAFit [21], RCG [14], ALPS [18], LRGeomCG [32], BARM [19], and IRLS0 [20] for low-rank matrix recovery and matrix completion problems. In our comparisons, we set $n_1 = n_2$, $n = n_1 n_2$, and $\sigma = 0$. All experiments are conducted in Matlab on an Intel 2.3 GHz Core i5 Quad-core processor with 16 GB RAM on MacOS. The implementation of ALPS, BARM, and IRLS0 are from public sources and the implementation of LMAFit, RCG, RGrad, LRGeomCG, and NIHT are from our own codes (available for downloading at GitHub). For a fair comparison, all the codes are realized in a pure Matlab environment without using MEX files for acceleration. The comparison results are shown in Tables II–IV. In these tables, "#iter" denotes the average iteration times of a successful recovery, "NS" denotes the number of successful recovery out of 10 trials for each settings and "Time" denotes the average running time of successful recoveries.

A. Low-Rank Matrix Recovery

1) Algorithm Comparison: In Table II we compare the performance of algorithms for low-rank matrix recovery with different settings. The linear operator is chosen as the partial orthogonal operator in (30). LRGeomCG is not included for that it only applies to matrix completion. From Table II, we see that TARM has the best performance (with the least running time and the highest success rate), and LMAFit does not work well in the considered settings. It is worth noting that TARM works well at low measurement rates when the other algorithms fail in recovery.

2) Impact of the Singular Value Distribution of the Low-Rank Matrix on TARM: The low-rank matrix generated by the product of two Gaussian matrices has a clear singular-value gap, i.e., the smallest singular σ_r is not close to zero. We now discuss the impact of the singular value distribution of the low-rank matrix on the performance of TARM.

TABLE II
COMPARISONS OF ALGORITHMS FOR LOW-RANK MATRIX RECOVERY

Parameters			TARM			LMAFit			RCG			ALPS			NIHT			RGrad		
n_1	m/n	r	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time
1000	0.1	20	14	10	**4.18**	/	0	/	14	10	6.65	9	10	4.55	38.2	10	9.54	38	10	8.66
1000	0.07	20	23	10	**6.98**	/	0	/	23.7	10	11.12	30.6	10	15.75	96.8	10	24.48	95.9	10	21.57
1000	0.05	20	54.4	10	**15.89**	/	0	/	60.2	10	29.05	/	0	/	/	0	/	/	0	/
1000	0.045	20	96.2	10	**28.43**	/	0	/	119.4	10	55.69	/	0	/	/	0	/	/	0	/
1000	0.042	20	198	9	**61.90**	/	0	/	/	0	/	/	0	/	/	0	/	/	0	/
1000	0.1	30	24	10	**9.10**	/	0	/	25	10	12.14	34	10	23.29	105	10	32.40	105	10	25.36
1000	0.2	50	16	10	**10.62**	/	0	/	17	10	10.13	12	10	12.83	51	10	22.25	51	10	13.97

TABLE III
COMPARISONS OF TARM WITH LMAFIT, RCG AND ALPS FOR MATRIX COMPLETION

Parameters			TARM			LMAFit			RCG			ALPS			NIHT			RGrad			LRGeomCG		
n_1	m/n	r	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time
500	0.2	10	10.2	10	0.24	21.4	10	**0.076**	9.6	10	0.33	11.5	10	0.212	18.7	10	0.192	17.5	10	0.295	10	10	0.119
500	0.12	10	20.7	10	0.41	48	4	**0.13**	17.5	10	0.54	33.6	10	0.66	49.8	10	0.47	32.2	10	0.53	17.2	10	0.16
500	0.05	10	199.3	6	**9.05**	/	0	/	/	0	/	/	0	/	0	/	0	/	0	/	/	0	/
1000	0.12	10	10	10	0.97	21.7	10	**0.208**	9	10	1.99	13.4	10	0.93	17.2	10	0.69	15.1	10	1.55	9.4	10	0.46
1000	0.06	10	21.6	10	2.08	/	0	/	18.4	10	4.18	46.1	10	3.65	52.1	8	1.89	37	10	3.88	18.2	10	**0.78**
1000	0.026	10	195.5	8	**21.82**	/	0	/	/	0	/	/	0	/	0	/	0	/	0	/	/	0	/
1000	0.12	20	14.4	10	1.57	39.7	10	**0.41**	14	10	3.18	25.8	10	3.02	35.7	10	1.79	28.5	10	3.10	14.2	10	0.77
1000	0.06	20	48.1	10	5.37	/	0	/	48.9	10	11.55	/	0	/	295	2	12.89	185	9	20.08	48.1	10	**2.23**
1000	0.045	20	185	7	**30.48**	/	0	/	/	0	/	/	0	/	/	0	/	/	0	/	/	0	/
2000	0.12	40	15	10	11.78	33.4	10	**1.77**	13	10	23.6	20	10	17.10	31.7	10	11.26	29	10	23.37	13	10	4.78
2000	0.06	40	45.4	10	34.98	/	0	/	38.6	10	73.3	/	0	/	230	8	74.04	174.6	10	140.06	39	10	**12.31**
2000	0.044	40	145.5	10	**116.07**	/	0	/	/	0	/	/	0	/	0	/	0	/	0	/	/	0	/

TABLE IV
COMPARISONS OF TARM WITH BARM AND IRLS0 FOR MATRIX COMPLETION

Parameters			TARM			BRAM			IRLS0		
n_1	m/n	r	#iter	NS	Time	#iter	NS	Time	#iter	NS	Time
100	0.2	5	44	10	1.43	45	10	9.43	547	10	**0.58**
100	0.2	8	157	10	5.39	66	10	13.83	6102	10	**4.1373**
150	0.2	10	47	10	1.85	48	10	62.36	917	10	**1.08**
150	0.15	10	159	10	**5.93**	94	10	78.25	/	0	/
200	0.12	10	295.5	9	**12.01**	/	0	/	/	0	/

To this end, we generate two matrices $\boldsymbol{G}_1 \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{G}_2 \in \mathbb{R}^{r \times n_2}$ with the elements independently drawn from $\mathcal{N}(0,1)$. Let the SVDs of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ be $\boldsymbol{G}_1 = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^T$, $\boldsymbol{G}_2 = \boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^T$, where $\boldsymbol{U}_i \in \mathbb{R}^{n_i \times r}$ and $\boldsymbol{V}_i \in \mathbb{R}^{r \times n_i}$ are the left and right singular vector matrices respectively, and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{r \times r}$ is the singular value matrix for $i = 1, 2$. The low-rank matrix $\boldsymbol{X}_0$ is then generated as

$$\tilde{\boldsymbol{X}} = \boldsymbol{U}_1 \mathrm{diag}(\exp(-k), \exp(-2\,k), \ldots, \exp(-rk))\boldsymbol{U}_2^T \quad (36)$$

where k controls the rate of decay. The matrix $\tilde{\boldsymbol{X}}$ is normalized to yield $\boldsymbol{X}_0 = \frac{\sqrt{n}\tilde{\boldsymbol{X}}}{\|\tilde{\boldsymbol{X}}\|_F}$. It is readily seen that the singular values of $\boldsymbol{X}_0$ do not have a clear gap away from zero, provided that k is sufficiently large.

In simulation, we consider the low-rank matrix recovery problem with the following three decay rates: $k = 0.1$, $k = 0.5$, and $k = 1$. We choose matrix dimensions $n_1 = n_2 = 1000$, rank $r = 20$, and measurement rate $\frac{m}{n} = 0.08$. The singular value distributions of low-rank matrix $\boldsymbol{X}_0$ in the three cases are given in the left figure of Fig. 11. We see that the singular values for $k = 0.1$ have a clear gap away from 0, while the singular values for $k = 0.5$ and 1 decay to 0 rapidly. In TARM, the target rank is set to the real rank value of $\boldsymbol{X}_0$. The NMSE of TARM against the iteration number is plotted on the right part of Fig. 11. We see that TARM converges the fastest for $k = 0.1$, and it works still well for $k = 0.5$ and 1 but with reduced convergence rates. Generally speaking, the convergence rate of TARM becomes slower as the increase of k.

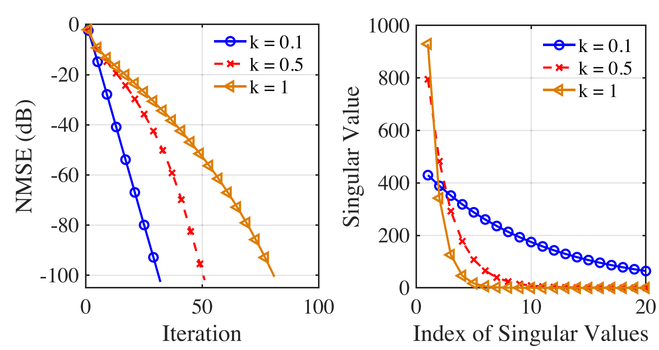

Fig. 11. The performance of the TARM algorithm when the low-rank matrices to be recovered do not have a clear singular-value gap away from zero.

B. Matrix Completion

In Table III, we compare the performance of algorithms for matrix completion in various settings. The linear operator is chosen as the random selection operator. We see from Table III that LMAFit runs the fastest when the measurement rate m/n is relatively high (say, $m/n = 0.2$ with $n_1 = n_2 = 500$); LRGeomCG performs the best when the measurement rate is medium (say, $m/n = 0.06$ with $n_1 = n_2 = 1000$); and TARM works the best when the measurement rate is relatively low (say, $m/n = 0.026$ with $n_1 = n_2 = 1000$).

Since both BARM and IRLS0 involves matrix inversions, these algorithms cannot handle the ARM problem with a relatively large size. Therefore, we set a relatively small size in our comparisons. The comparison results of TARM with BARM and IRLS0 on matrix completion are presented in Table IV. From the table, we see that TARM generally requires a shorter recovery time and has a higher recovery success rate than the other two counterparts.

VI. Conclusions

In this paper, we proposed a low-complexity iterative algorithm termed TARM for solving the stable ARM problem. The proposed algorithm can be applied to both low-rank matrix recovery and matrix completion. For low-rank matrix recovery, the performance of TARM can be accurately characterized by the state evolution technique when ROIL operators are involved. For matrix completion, we showed that, although state evolution is not accurate, the parameters of TARM can be carefully tuned to achieve good performance. Numerical results demonstrate that TARM has competitive performance compared to the existing algorithms for low-rank matrix recovery problems with ROIL operators and random selection operators.

Appendix A
Proof of Lemma 1

We first determine μ_t. We have

$$\begin{aligned}
&\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\rangle \\
&= \langle \boldsymbol{X}^{(t-1)} + \mu_t \mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)})) - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\rangle \quad (37a)\\
&= \langle \boldsymbol{X}^{(t-1)} + \mu_t \mathcal{A}^T(\mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{n} - \mathcal{A}(\boldsymbol{X}^{(t-1)}))\\
&\quad - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\rangle \quad (37b)\\
&= \langle \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\rangle + \mu_t\langle \boldsymbol{n}, \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\rangle\\
&\quad - \mu_t\langle \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0), \mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\rangle \quad (37c)
\end{aligned}$$

where step (37a) follows by substituting $\boldsymbol{R}^{(t)}$ in Line 3 of Algorithm 1, and step (37c) follows by noting

$$\langle \mathcal{A}(\boldsymbol{B}), \boldsymbol{c}\rangle = \langle \boldsymbol{B}, \mathcal{A}^T(\boldsymbol{c})\rangle \quad (38)$$

for any matrix $\boldsymbol{B}$ and vector $\boldsymbol{c}$ of appropriate sizes. Together with Condition 1, we obtain (12a).

We next determine α_t and c_t. First note

$$\begin{aligned}
\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 &= \|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 + \|\boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2\\
&\quad + 2\langle \boldsymbol{X}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\rangle \quad (39a)\\
&= \|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 + \|\boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2 \quad (39b)
\end{aligned}$$

where (39b) is from Condition 2 in (10). Recall that in the t-th iteration $\boldsymbol{R}^{(t)}$ is a function of μ_t but not of α_t and c_t. Thus, minimizing $\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2$ over α_t and c_t is equivalent to minimizing $\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2$ over α_t and c_t. For any given α_t, the optimal c_t to minimize $\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 = \|c_t(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}) - \boldsymbol{R}^{(t)}\|_F^2$ is given by

$$c_t = \frac{\langle \boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)}\rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2}. \quad (40)$$

Then,

$$\begin{aligned}
&\langle \boldsymbol{X}^{(t)} - \boldsymbol{X}_0, \boldsymbol{R}^{(t)} - \boldsymbol{X}_0\rangle\\
&= \langle c_t(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}) - \boldsymbol{X}_0, \boldsymbol{R}^{(t)} - \boldsymbol{X}_0\rangle \quad (41a)\\
&= \left\langle \frac{\langle \boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)}\rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2}(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}) - \boldsymbol{X}_0, \boldsymbol{R}^{(t)} - \boldsymbol{X}_0\right\rangle \quad (41b)
\end{aligned}$$

where (41a) follows by substituting $\boldsymbol{X}^{(t)}$ in Line 5 of Algorithm 1, and (41b) by substituting c_t in (40). Combining (41) and Condition 2, we see that α_t is the solution of the following quadratic equation:

$$a_t\alpha_t^2 + b_t\alpha_t + d_t = 0 \quad (42)$$

where a_t, b_t, and d_t are defined in (13). Therefore, α_t is given by (12b). With the above choice of c_t, we have

$$\begin{aligned}
&\langle \boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}, \boldsymbol{X}^{(t)}\rangle\\
&= \langle c_t(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}) - \boldsymbol{R}^{(t)}, c_t(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)})\rangle = 0. \quad (43)
\end{aligned}$$

This orthogonality is useful in analyzing the performance of Module B.

Appendix B
Convergence Analysis of TARM Based on RIP

Without loss of generality, we assume $n_1 \leq n_2$ in this appendix. Following the convention in [13], we focus our discussion on the noiseless case, i.e., $\boldsymbol{n} = \boldsymbol{0}$.

Definition 2: (Restricted Isometry Property). Given a linear operator $\mathcal{A}: \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$, a minimum constant called the rank restricted isometry constant (RIC) $\delta_r(\mathcal{A}) \in (0,1)$ exists such that

$$(1 - \delta_r(\mathcal{A}))\|\boldsymbol{X}\|_F^2 \leq \|\gamma\mathcal{A}(\boldsymbol{X})\|_2^2 \leq (1 + \delta_r(\mathcal{A}))\|\boldsymbol{X}\|_F^2 \quad (44)$$

for all $\boldsymbol{X} \in \mathbb{R}^{n_1\times n_2}$ with $\mathrm{rank}(\boldsymbol{X}) \leq r$, where $\gamma > 0$ is a constant scaling factor.

We now introduce two useful lemmas.

Lemma 3: Assume that α_{t+1} and c_{t+1} satisfy Condition 2 and Condition 3. Then,

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 = \frac{\|\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}\|_F^2}{\frac{\|\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2}\alpha_t^2 + (1-\alpha_t)^2}. \quad (45)$$

Lemma 4: Let $\boldsymbol{Z}^{(t)}$ be the best rank-r approximation of $\boldsymbol{R}^{(t)}$. Then,

$$\|\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}\|_F^2 \leq \|\boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2. \quad (46)$$

The proof of Lemma 3 is given in Appendix E. Lemma 4 is straightforward from the definition of the best rank-r approximation [25, p. 211–218].

Theorem 3: Assume that μ_t, α_t, c_t satisfy Conditions 1–3, and the linear operator $\mathcal{A}$ satisfies the RIP with rank n_1 and RIC δ_{n_1}. Then,

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 \le \left(\frac{1}{(1-\alpha_t)^2} - 1\right)\left(\frac{1+\delta_{n_1}}{1-\delta_{n_1}} - 1\right)^2 \times \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 \quad (47)$$

TARM guarantees to converge when RIC satisfies $\alpha_t \neq 1, \forall t$, and

$$\delta_{n_1} < \frac{1}{1 + 2\sqrt{\frac{1}{\xi}\left(\frac{1}{(1-\alpha_{\max})^2} - 1\right)}} \quad (48)$$

where the constant ξ satisfies $0 < \xi < 1$, and $\alpha_{\max} = \sup\{\alpha_t\}$.

Proof: Since $\boldsymbol{Z}^{(t)}$ is the best rank-r approximation of $\boldsymbol{R}^{(t)}$, we have $\|\boldsymbol{R}^{(t)}\|_F^2 \ge \|\boldsymbol{Z}^{(t)}\|_F^2$. Then, from Lemma 3, we obtain

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 \le \frac{\|\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}\|_F^2}{(1-\alpha_t)^2}. \quad (49)$$

Then, we have

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 = \|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0 + \boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2 \quad (50a)$$

$$= \|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 + \|\boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2 + 2\langle \boldsymbol{X}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\rangle \quad (50b)$$

$$= \|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 + \|\boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\|_F^2 \quad (50c)$$

where (50c) follows from $\langle \boldsymbol{X}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0 - \boldsymbol{R}^{(t)}\rangle = 0$ in Condition 2. Combining (4), (49), and (50), we obtain

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 \le \left(\frac{1}{(1-\alpha_t)^2} - 1\right)\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 \quad (51a)$$

$$= \left(\frac{1}{(1-\alpha_t)^2} - 1\right) \|\boldsymbol{X}^{(t-1)} + \mu_t\mathcal{A}^*(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)})) - \boldsymbol{X}_0\|_F^2 \quad (51b)$$

$$= \left(\frac{1}{(1-\alpha_t)^2} - 1\right)\|(\mathcal{I} - \mu_t\mathcal{A}^*\mathcal{A})(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2. \quad (51c)$$

Since $\mathcal{A}$ has RIP with rank n_1 and RIC δ_{n_1}, we obtain the following inequality from [18]:

$$\|(\mathcal{I} - \mu_t\mathcal{A}^*\mathcal{A})(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 \le \max\left((\mu_t(1+\delta_{n_1}) - 1)^2, (\mu_t(1-\delta_{n_1}) - 1)^2\right) \times \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2. \quad (52)$$

Recall that $\mu_t = \frac{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2}{\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_2^2}$ obtained by letting $\boldsymbol{n} = \boldsymbol{0}$ in (12a). From RIP, we have

$$\frac{1}{1+\delta_{n_1}} \le \mu_t = \frac{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2}{\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_2^2} \le \frac{1}{1-\delta_{n_1}}. \quad (53)$$

Then, combining (52) and (53), we have

$$\|(\mathcal{I} - \mu_t\mathcal{A}^*\mathcal{A})(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 \le \left(\frac{1+\delta_{n_1}}{1-\delta_{n_1}} - 1\right)^2 \times \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2. \quad (54)$$

Combining (54) and (51), we arrive at (47).

When δ_{n_1} satisfies (48), we have

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 < \xi\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 \quad (55)$$

at each iteration t. Then, TARM converges exponentially to $\boldsymbol{X}_0$. ■

We now compare the convergence rate of TARM with those of SVP and NIHT. Compared with [13, Eq. 2.11–2.14], (47) contains an extra term $\frac{1}{(1-\alpha_t)^2} - 1$. From numerical experiments, α_t is usually close to zero, implying that TARM converges faster than SVP and NIHT.

APPENDIX C
PROOF OF THEOREM 1

For a partial orthogonal ROIL operator $\mathcal{A}$, the following properties hold:

$$\mathcal{A}(\mathcal{A}^T(\boldsymbol{a})) = \boldsymbol{a} \quad (56a)$$

$$\langle \mathcal{A}^T(\boldsymbol{a}), \mathcal{A}^T(\boldsymbol{b})\rangle = \langle \boldsymbol{a}, \boldsymbol{b}\rangle. \quad (56b)$$

Then as $m, n \to \infty$ with $\frac{m}{n} \to \delta$, we have

$$\left\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\right\|_F^2 = \left\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0 - \frac{1}{\delta}\mathcal{A}^T\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0) + \frac{1}{\delta}\mathcal{A}^T(\boldsymbol{n})\right\|_F^2 \quad (57a)$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \quad (57b)$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta}\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 - 2\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \quad (57c)$$

$$= \left(\frac{1}{\delta} - 1\right)\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + n\sigma^2 \quad (57d)$$

where (57a) is obtained by substituting $\boldsymbol{R}^{(t)} = \boldsymbol{X}^{(t-1)} + \mu_t\mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)}))$ and $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{n}$ with $\mu_t = \delta^{-1}$, (57b) is obtained by noting that $\boldsymbol{n}$ is independent of $\mathcal{A}(\boldsymbol{X}^{(t)} - \boldsymbol{X}_0)$ (ensured by Assumption 1) and (56b), and (57c) follows from $\frac{\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_2^2}{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2} \to \delta$ (see (15)). When $\frac{1}{n}\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 \to \tau$, we have

$$\frac{1}{n}\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 \to \left(\frac{1}{\delta} - 1\right)\tau + \sigma^2. \quad (58)$$

We now consider the case of Gaussian ROIL operators. As $m, n \to \infty$ with $\frac{m}{n} \to \delta$, we have

$$\left\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\right\|_F^2 = \left\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0 - \frac{1}{\delta}\mathcal{A}^T\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0) + \frac{1}{\delta}\mathcal{A}^T(\boldsymbol{n})\right\|_F^2 \tag{59a}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\|\mathcal{A}^T\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59b}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{A}^T\boldsymbol{A}\mathrm{vec}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59c}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\frac{\|\boldsymbol{A}^T\boldsymbol{A}\|_F^2}{mn}\|\mathrm{vec}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_2^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59d}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\frac{\mathrm{Tr}((\boldsymbol{A}^T\boldsymbol{A})^2)}{mn}\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59e}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \left(1 + \frac{1}{\delta}\right)\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 - \frac{2}{\delta}\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0)\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59f}$$

$$= \|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \left(1 + \frac{1}{\delta}\right)\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 - 2\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + \frac{1}{\delta^2}\|\boldsymbol{n}\|_2^2 \tag{59g}$$

$$= \frac{1}{\delta}\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0\|_F^2 + n\sigma^2 \tag{59h}$$

where (59a) is obtained by substituting $\boldsymbol{R}^{(t)} = \boldsymbol{X}^{(t-1)} + \mu_t\mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)}))$ and $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{n}$ with $\mu_t = \delta^{-1}$, (59b) is obtained by noting that $\boldsymbol{n}$ is independent of $\mathcal{A}(\boldsymbol{X}^{(t)} - \boldsymbol{X}_0)$ (from Assumption 1), (59c) follows by utilizing the matrix form $\boldsymbol{A}$ of $\mathcal{A}$, (59d) follows from the fact that $\boldsymbol{V}_A$ (i.e. the right singular-vector matrix of $\boldsymbol{A}$) is a Haar distributed orthogonal matrix independent of $\boldsymbol{X}^{(t-1)} - \boldsymbol{X}_0$, (59e) follows from $\mathrm{Tr}((\boldsymbol{A}^T\boldsymbol{A})^2) = \|\boldsymbol{A}^T\boldsymbol{A}\|_F^2$, (59f) is obtained by noting that $\frac{1}{mn}\mathrm{Tr}((\boldsymbol{A}^T\boldsymbol{A})^2) \to \delta + \delta^2$ since $\boldsymbol{A}^T\boldsymbol{A}$ is a Wishart matrix with variance $\frac{1}{n}$ [34, p. 26], and (59g) follows by noting $\frac{\|\mathcal{A}(\boldsymbol{X}^{(t)} - \boldsymbol{X}_0)\|_2^2}{\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2} \to \delta$. When $\frac{1}{n}\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2 \to \tau$, we have

$$\frac{1}{n}\|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 \to \frac{1}{\delta}\tau + \sigma^2. \tag{60}$$

Appendix D
Proof of Lemma 2

We first introduce two useful facts.

Fact 1: When $n_2 \to \infty$ with fixed $n_1/n_2 = \rho$, the i-th singular value σ_i of the Gaussian noise corrupted matrix $\boldsymbol{R}^{(t)}$ is given by [17, Eq. 9]

$$\frac{1}{\sqrt{n_2}}\sigma_i \xrightarrow{\text{a.s.}} \begin{cases} \sqrt{\dfrac{(v_t + \theta_i^2)(\rho v_t + \theta_i^2)}{\theta_i^2}} & \text{if } i \le r \text{ and } \theta_i > \rho^{\frac{1}{4}} \\ \sqrt{v_t}(1 + \sqrt{\rho}) & \text{otherwise} \end{cases} \tag{61}$$

where v_t is the variance of the Gaussian noise, and θ_i is the i-th largest singular value of $\frac{1}{\sqrt{n_2}}\boldsymbol{X}_0$.

Fact 2: From [33, Eq. 9], the divergence of a spectral function $h(\boldsymbol{R})$ is given by

$$\mathrm{div}(h(\boldsymbol{R})) = |n_1 - n_2|\sum_{i=1}^{\min(n_1,n_2)}\frac{h_i(\sigma_i)}{\sigma_i} + \sum_{i=1}^{\min(n_1,n_2)}h_i'(\sigma_i) + 2\sum_{i\neq j, i,j=1}^{\min(n_1,n_2)}\frac{\sigma_i h_i(\sigma_i)}{\sigma_i^2 - \sigma_j^2}. \tag{62}$$

The best rank-r approximation denoiser $\mathcal{D}(\boldsymbol{R})$ is a spectral function with

$$\begin{cases} h_i(\sigma_i) = \sigma_i & i \le r; \\ h_i(\sigma_i) = 0 & i > r. \end{cases} \tag{63}$$

Combining (62) and (63), the divergence of $\mathcal{D}(\boldsymbol{R}^{(t)})$ is given by

$$\mathrm{div}(\mathcal{D}(\boldsymbol{R}^{(t)})) = |n_1 - n_2|r + r^2 + 2\sum_{i=1}^{r}\sum_{j=r+1}^{\min(n_1,n_2)}\frac{\sigma_i^2}{\sigma_i^2 - \sigma_j^2}. \tag{64}$$

Further, we have

$$\sum_{i=1}^{r}\sum_{j=r+1}^{\min(n_1,n_2)}\frac{\sigma_i^2}{\sigma_i^2 - \sigma_j^2} \xrightarrow{\text{a.s.}} (\min(n_1,n_2) - r)\sum_{i=1}^{r}\frac{\sigma_i^2}{\sigma_i^2 - (\sqrt{n_2 v_t}(1 + \sqrt{\rho}))^2} \tag{65a}$$

$$= (\min(n_1,n_2) - r)\sum_{i=1}^{r}\frac{n_2\frac{(v_t+\theta_i^2)(\rho v_t+\theta_i^2)}{\theta_i^2}}{\frac{n_2(v_t+\theta_i^2)(\rho v_t+\theta_i^2)}{\theta_i^2} - n_2 v_t(1+\sqrt{\rho})^2} \tag{65b}$$

$$= (\min(n_1,n_2) - r)\sum_{i=1}^{r}\frac{(v_t + \theta_i^2)(\rho v_t + \theta_i^2)}{(\sqrt{\rho}v_t - \theta_i^2)^2} \tag{65c}$$

$$\xrightarrow{\text{a.s.}} (\min(n_1,n_2) - r)r\int_0^\infty\frac{(v_t + \theta^2)(\rho v_t + \theta^2)}{(\sqrt{\rho}v_t - \theta^2)^2}p(\theta)d\theta \tag{65d}$$

$$= (\min(n_1,n_2) - r)r\Delta_1(v_t) \tag{65e}$$

where both (65a) and (65b) are from (61), and (65e) follows by the definition of $\Delta_1(v_t)$. Combining (64) and (65), we obtain the asymptotic divergence of $\mathcal{D}(\boldsymbol{R})$ given by

$$\mathrm{div}(\mathcal{D}(\boldsymbol{R})) \overset{\text{a.s.}}{\to} |n_1 - n_2|r + r^2 + 2(\min(n_1, n_2) - r)r\Delta_1(v_t) \tag{66}$$

and

$$\begin{aligned} \alpha_t &= \frac{1}{n}\mathrm{div}(f(\boldsymbol{R}^{(t)})) && (67a)\\ &\overset{\text{a.s.}}{\to} \left|1 - \frac{1}{\rho}\right|\lambda + \frac{1}{\rho}\lambda^2 + 2\left(\min\left(1, \frac{1}{\rho}\right) - \frac{\lambda}{\rho}\right)\lambda\Delta_1(v_t) && (67b)\\ &= \alpha(v_t) && (67c) \end{aligned}$$

with $\lambda = r/n_2$.

Recall that $\boldsymbol{Z}^{(t)}$ is the best rank-r approximation of $\boldsymbol{R}^{(t)}$ satisfying

$$\|\boldsymbol{Z}^{(t)}\|_F^2 = \sum_{i=1}^{r}\sigma_i^2 \tag{68a}$$

$$\|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2 = \sum_{i=r+1}^{n_1}\sigma_i^2. \tag{68b}$$

Then, when $m, n \to \infty$ with $\frac{m}{n} \to \delta$, we have

$$\begin{aligned} \|\boldsymbol{Z}^{(t)}\|_F^2 &= \sum_{i=1}^{r}\sigma_i^2 && (69a)\\ &\overset{\text{a.s.}}{\longrightarrow} n_2\sum_{i=1}^{r}\frac{(v+\theta_i^2)(\rho v+\theta_i^2)}{\theta_i^2} && (69b)\\ &= n + \lambda\left(1+\frac{1}{\rho}\right)nv + \lambda nv^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2} && (69c) \end{aligned}$$

and

$$\begin{aligned} &\|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2\\ &\quad = \|\boldsymbol{X}_0\|_F^2 + nv_t - \|\boldsymbol{Z}^{(t)}\|_F^2 && (70a)\\ &\quad \overset{\text{a.s.}}{\longrightarrow} nv_t - \lambda\left(1+\frac{1}{\rho}\right)nv_t - \lambda nv_t^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2} && (70b) \end{aligned}$$

where (69b) is from (61), (70a) is from Assumption 2, and (70b) is from (69). Then, Eq. (71a)–(71g) shown at bottom of this page, where (71a) is from (12c), (71c) follows from Assumption 2 that $\boldsymbol{R}^{(t)} = \boldsymbol{X}_0 + \sqrt{v_t}\boldsymbol{N}$ with $\|\boldsymbol{X}_0\|_F^2 = n$ and the elements of $\boldsymbol{N}$ independently drawn from $\mathcal{N}(0,1)$, (71d) is from (68), and (71f) is from the definition of Δ_2.

Appendix E
Proof of Lemma 3

Assume that $\boldsymbol{R}^{(t)} = \sum_{i=1}^{\min(m,n)}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T$, where σ_i, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the i-th singular value, left singular vector and right singular vectors respectively. Then, $\boldsymbol{Z}^{(t)} = \sum_{i=1}^{r}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T$ since that $\boldsymbol{Z}^{(t)}$ is the best rank-r approximation of $\boldsymbol{R}^{(t)}$, and

$$\begin{aligned} \langle\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}, \boldsymbol{Z}^{(t)}\rangle &= \left\langle\sum_{j=r+1}^{\min(m,n)}\sigma_j\boldsymbol{u}_j\boldsymbol{v}_j^T, \sum_{i=1}^{r}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\rangle && (72a)\\ &= \sum_{i=1}^{r}\sum_{j=r+1}^{\min(m,n)}\sigma_i\sigma_j\langle\boldsymbol{u}_i\boldsymbol{v}_i^T, \boldsymbol{u}_j\boldsymbol{v}_j^T\rangle && (72b)\\ &= \sum_{i=1}^{r}\sum_{j=r+1}^{\min(m,n)}\sigma_i\sigma_j\mathrm{Tr}(\boldsymbol{v}_i\boldsymbol{u}_i^T\boldsymbol{u}_j\boldsymbol{v}_j^T) && (72c)\\ &= 0 && (72d) \end{aligned}$$

$$\begin{aligned} c_t &= \frac{\langle\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)}\rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2} && (71a)\\ &= \frac{\langle\boldsymbol{Z}^{(t)}, \boldsymbol{R}^{(t)}\rangle - \alpha_t\|\boldsymbol{R}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2 - 2\alpha_t\langle\boldsymbol{Z}^{(t)}, \boldsymbol{R}^{(t)}\rangle + \alpha_t^2\|\boldsymbol{R}^{(t)}\|_F^2} && (71b)\\ &\overset{\text{a.s.}}{\longrightarrow} \frac{\|\boldsymbol{Z}^{(t)}\|_F^2 - \alpha_t(n + v_t n)}{\|\boldsymbol{Z}^{(t)}\|_F^2 - 2\alpha_t\|\boldsymbol{Z}^{(t)}\|_F^2 + \alpha_t^2(n + v_t n)} && (71c)\\ &= \frac{n + \lambda(1+\frac{1}{\rho})nv_t + \lambda nv_t^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2} - \alpha_t(n + v_t n)}{(1-2\alpha_t)(n + \lambda(1+\frac{1}{\rho})nv_t + \lambda nv_t^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2}) + \alpha_t^2(n + v_t n)} && (71d)\\ &= \frac{1 + \lambda(1+\frac{1}{\rho})v_t + \lambda v_t^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2} - \alpha_t(1 + v_t)}{(1-2\alpha_t)(1 + \lambda(1+\frac{1}{\rho})v_t + \lambda v_t^2\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\theta_i^2}) + \alpha_t^2(1 + v_t)} && (71e)\\ &\overset{\text{a.s.}}{\longrightarrow} \frac{1 + \lambda(1+\frac{1}{\rho})v_t + \lambda v_t^2\Delta_2 - \alpha(v_t)(1 + v_t)}{(1-2\alpha(v_t))(1 + \lambda(1+\frac{1}{\rho})v_t + \lambda v_t^2\Delta_2) + (\alpha(v_t))^2(1 + v_t)} && (71f)\\ &= c(v_t) && (71g) \end{aligned}$$

where (72d) follows from that $\boldsymbol{u}_i^T\boldsymbol{u}_j = 0$, $\forall i, j$, and $i \neq j$. Recall from (43) that:

$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^{(t)}, \boldsymbol{X}^{(t)}\rangle = 0. \tag{73}$$

With the above orthogonalities, we have

$$\begin{aligned}
\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 &= \|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{X}^{(t)}\|_F^2 && \text{(74a)}\\
&= \|\boldsymbol{R}^{(t)}\|_F^2 - \left\|c_t(\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)})\right\|_F^2 && \text{(74b)}\\
&= \|\boldsymbol{R}^{(t)}\|_F^2 - \frac{\langle \boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)}\rangle^2}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2} && \text{(74c)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2\|\boldsymbol{Z}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^4}{\|\boldsymbol{Z}^{(t)} - \alpha_t\boldsymbol{R}^{(t)}\|_F^2} && \text{(74d)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2\|\boldsymbol{Z}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^4}{\|\boldsymbol{Z}^{(t)}\|_F^2 - 2\alpha_t\langle\boldsymbol{R}^{(t)}, \boldsymbol{Z}^{(t)}\rangle + \alpha_t^2\|\boldsymbol{R}^{(t)}\|_F^2} && \text{(74e)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2\|\boldsymbol{Z}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^4}{\|\boldsymbol{Z}^{(t)}\|_F^2 - 2\alpha_t\|\boldsymbol{Z}^{(t)}\|_F^2 + \alpha_t^2\|\boldsymbol{R}^{(t)}\|_F^2} && \text{(74f)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{1 - 2\alpha_t + \alpha_t^2 + \alpha_t^2\frac{\|\boldsymbol{R}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2} - \alpha_t^2} && \text{(74g)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{\frac{\|\boldsymbol{R}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2}\alpha_t^2 + (1-\alpha_t)^2} && \text{(74h)}\\
&= \frac{\|\boldsymbol{R}^{(t)} - \boldsymbol{Z}^{(t)}\|_F^2}{\frac{\|\boldsymbol{R}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2}\alpha_t^2 + (1-\alpha_t)^2} && \text{(74i)}
\end{aligned}$$

where (74a) follows from (73), (74b) follows by substituting $\boldsymbol{X}^{(t)}$ in Line 5 of Algorithm 1, (74c) follows by substituting c_t in (12c), and (74d)–(74i) follow from (72). This concludes the proof of Lemma 3.

Appendix F
Proof of Theorem 2

From Condition 2 in (10) and Assumption 2, we have[7]

$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}_0\rangle = 0 \tag{75a}$$

$$\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}_0, \boldsymbol{X}^{(t)} - \boldsymbol{X}_0\rangle = 0. \tag{75b}$$

Then,

$$\begin{aligned}
&\|\boldsymbol{X}^{(t)} - \boldsymbol{X}_0\|_F^2\\
&= \|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 - 2\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^{(t)}, \boldsymbol{R}^{(t)} - \boldsymbol{X}_0\rangle\\
&\quad + \|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 && \text{(76a)}\\
&= \|\boldsymbol{R}^{(t)} - \boldsymbol{X}^{(t)}\|_F^2 - \|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 && \text{(76b)}\\
&= \frac{\|\boldsymbol{R}^{(t)}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{\frac{\|\boldsymbol{R}\|_F^2 - \|\boldsymbol{Z}^{(t)}\|_F^2}{\|\boldsymbol{Z}^{(t)}\|_F^2}\alpha_t^2 + (1-\alpha_t)^2} - \|\boldsymbol{R}^{(t)} - \boldsymbol{X}_0\|_F^2 && \text{(76c)}\\
&\xrightarrow{\text{a.s.}} \frac{nv_t - \lambda\left(1 + \frac{1}{\rho}\right)nv_t - \lambda n v_t^2\Delta_2}{\frac{v_t - \lambda\left(1+\frac{1}{\rho}\right)v_t - \lambda v_t^2\Delta_2}{1+\lambda\left(1+\frac{1}{\rho}\right)v_t + \lambda v_t^2\Delta_2}(\alpha(v_t))^2 + (1-\alpha(v_t))^2} - nv_t && \text{(76d)}
\end{aligned}$$

where (76b) is from (75b), (76c) follows from (74), and (76d) follows from (69) and (70) and Assumption 2. Therefore, (25) holds, which concludes the proof of Theorem 2.

[7] In fact, as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$ and $\frac{r}{n_2} \to \lambda$, the approximation in (16) become accurate, i.e. $\alpha_t = \frac{1}{n}\mathrm{div}(\mathcal{D}(\boldsymbol{R}^{(t)}))$ asymptotically satisfies Condition 2. Thus, (75b) asymptotically holds.

References

[1] S. Ralph, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[2] G. David, N. David, O. Brian, and T. Douglas, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–71, Sep. 1992.

[3] A. Singer, "A remark on global positioning from local distances," *Proc. Nat. Acad. Sci.*, vol. 105, no. 28, pp. 9507–9511, Jul. 2008.

[4] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1235–1256, Aug. 2009.

[5] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[6] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.

[7] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.

[8] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[9] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific J. Optim.*, vol. 6, no. 3, pp. 615–640, Mar. 2010.

[10] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, Palo Alto, CA, USA, Jun. 2013, pp. 665–674.

[11] P. Jain, R. Meka, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, Vancouver, BC, Canada, Dec. 2010, pp. 937–945.

[12] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[13] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. S104–S125, Oct. 2013.

[14] K. Wei, J.-F. Cai, T. F. Chan, and S. Leung, "Guarantees of Riemannian optimization for low rank matrix recovery," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 3, pp. 1198–1222, Sep. 2016.

[15] J. Ma, X. Yuan, and L. Ping, "Turbo compressed sensing with partial DFT sensing matrix," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 158–161, Feb. 2015.

[16] Z. Xue, J. Ma, and X. Yuan, "Denoising-based turbo compressed sensing," *IEEE Access*, vol. 5, pp. 7193–7204, 2017.

[17] F. Benaych-Georges and R. R. Nadakuditi, "The singular values and vectors of low rank perturbations of large rectangular random matrices," *J. Multivariate Anal.*, vol. 111, pp. 120–135, Oct. 2012.

[18] A. Kyrillidis and V. Cevher, "Matrix recipes for hard thresholding methods," *J. Math. Imag. Vis.*, vol. 48, no. 2, pp. 235–265, Feb. 2014.

[19] X. Bo and W. David, "Pushing the limits of affine rank minimization by adapting probabilistic PCA," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 419–427.

[20] M. Karthik and F. Maryam, "Iterative reweighted algorithms for matrix rank minimization," *J. Mach. Learn. Res.*, vol. 31, no. 1, pp. 3441–3473, Nov. 2012.

[21] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, Jul. 2012.

[22] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.* vol. 106, no. 45, pp. 18914–18919, Sep. 2009.

[23] C. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.
[24] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.* vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
[25] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Mar. 1936.
[26] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.
[27] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.
[28] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory*, Aachen, Germany, Aug. 2017, pp. 1588–1592.
[29] K. Takeuchi, "Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements," 2017, *arXiv:1701.05284*.
[30] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.
[31] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981.
[32] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.
[33] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko, "Unbiased risk estimates for singular value thresholding and spectral estimators," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4643–4657, Oct. 2013.
[34] A. M. Tulino and S. Verdú, "Random matrix theory and wireless communications," *Found. Trends Commun. Inf. Theory*, vol. 1, no. 14, pp. 1–182, Jun. 2004.
[35] Z. Xue, X. Yuan, and J. Ma, "TARM: A turbo-type algorithm for low-rank matrix recovery," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 4614–4618.

Zhipeng Xue received the B.E. degree in communication engineering from Southwest Jiaotong University, Chengdu, China, in 2015. He is currently working toward the Ph.D. degree at the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests include statistical signal processing and machine learning.

Xiaojun Yuan (S'04–M'09–SM'15) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2008. From 2009 to 2011, he was a Research Fellow with the Department of Electronic Engineering, the City University of Hong Kong. He was also a Visiting Scholar with the Department of Electrical Engineering, the University of Hawaii at Manoa in spring and summer 2009, as well as during the same period of 2010. From 2011 to 2014, he was a Research Assistant Professor with the Institute of Network Coding, The Chinese University of Hong Kong, Hong Kong. From 2014 to 2017, he was an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He is now a Professor with the National Key Laboratory of Science and Technology on Communications, the University of Electronic Science and Technology of China, supported by the Thousand Youth Talents Plan in China. He has authored or coauthored more than 110 peer-reviewed research papers in leading international journals and conferences in the related areas of his research. His research interests cover a broad range of statistical signal processing, machine learning, and information theory, including but not limited to multi-antenna and cooperative communications, sparse and structured signal recovery, Bayesian approximate inference, network coding, etc.

Prof. Yuan was a corecipient of the Best Paper Award of the IEEE International Conference on Communications (ICC) 2014 and also the Best Journal Paper Award of IEEE Technical Committee on Green Communications and Computing (TCGCC) 2017. He has been an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS since 2017. He has also been the member of a number of technical programs for international conferences.

Junjie Ma received the B.E. degree from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2015. He was a Postdoctoral Researcher with the City University of Hong Kong, from 2015 to 2016, and with Columbia University, from 2016 to 2019. Since July 2019, he has been a Postdoctoral Fellow with the Department of Electrical Engineering, Harvard University, Cambridge, MA, USA. His current research interest includes message passing approaches for high-dimensional signal processing.

Yi Ma (F'13) received the bachelor's degree in automation and applied mathematics from Tsinghua University, Beijing, China, in 1995, and two master's degrees in EECS and mathematics, and the Ph.D. degree in EECS in 2000, all from the University of California, Berkeley, CA, USA. Since January 2018, he has been a Professor with the EECS Department, UC Berkeley. From 2000 to 2011, he was a Faculty Member of the ECE Department, University of Illinois at Urbana-Champaign, Champaign, IL, USA. From 2009 to 2014, he was a Principal Researcher and the Research Manager for the Visual Computing Group, Microsoft Research Asia, Beijing, China. From 2014 to 2017, he was a Professor and the Executive Dean with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He has coauthored two textbooks, *An Invitation to 3D Vision* (Springer, 2004) and *Generalized Principal Component Analysis* (Springer, 2016). Prof. Yi was the recipient of the Best Paper Awards from ICCV, ECCV, and ACCV. He was the recipient of the CAREER Award from the NSF and the YIP Award from the ONR. He has been an Associate Editor for IJCV, SIIMS, SIMODS, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and IEEE TRANSACTIONS INFORMATION THEORY. He is a Fellow of the Association for Computing Machinery.