# TARM: A TURBO-TYPE ALGORITHM FOR LOW-RANK MATRIX RECOVERY

*Zhipeng Xue*[⋆] *Xiaojun Yuan*[†] *Junjie Ma*[‡]

[⋆] School of Information Science and Technology, ShanghaiTech University
[†] National Laboratory of Science and Technology on Communications, UESTC
[‡] Department of Statistics, Columbia University

**ABSTRACT**

This paper is concerned with the affine rank minimization (ARM) problem for low-rank matrix recovery purposes. Inspired by the recently proposed Turbo-CS algorithm in the field of compressed sensing, we propose a turbo-type algorithm for ARM, termed Turbo-ARM (TARM). For matrix recovery problems with a large class of random measurement matrices, the performance of TARM can be analyzed via the state evolution framework. Our numerical results show that TARM achieves state-of-the-art reconstruction performance, and our results are further confirmed by state evolution analysis.

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

## 1. INTRODUCTION

Low-rank matrices have wide applications in various fields, including collaborative filtering, system identification, and remote sensing. The basic problem of these applications is to recover a matrix from a small number of observations by exploiting the low-rank property [1, 2]. Consider a rank-$r$ matrix $\boldsymbol{X}^* \in \mathbb{R}^{n_1 \times n_2}$ where $r, n_1$, and $n_2$ satisfy $r \ll n_1$ and $r \ll n_2$. We aim to recover $\boldsymbol{X}^*$ from an affine measurement

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}^*) \tag{1}$$

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 it as low-rank matrix recovery; when $\mathcal{A}$ is a selector that outputs a subset of the entries of $\boldsymbol{X}^*$, we refer to the problem as matrix completion. Both of the problems can be done by solving the following affine rank minimization (ARM) problem:

$$\begin{aligned} &\min_{\boldsymbol{X}} \text{rank}(\boldsymbol{X}) \\ &\text{s.t. } \boldsymbol{y} = \mathcal{A}(\boldsymbol{X}). \end{aligned} \tag{2}$$

However, (2) is a nonconvex problem and computationally prohibitive. To handle this, a popular alternative to (2) is the following nuclear norm minimization problem [3]:

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

Recht et al. [3] proved that when the restricted isometry property (RIP) holds for the linear operator $\mathcal{A}$, the solution of the ARM problem is equivalent to the solution of the nuclear norm minimization problem. The nuclear norm minimization problem can be formulated as a semidefinite programing (SDP) problem and various existing convex optimization algorithms can be adopted. An interior point method was proposed in [4], however, the high computational complexity prevents its application for large scale problems. Several low-cost iterative methods were proposed, such as the singular value thresholding (SVT) method [5] and the proximal gradient algorithm [6]. These algorithms involve calculating the full singular values of a large matrix at each iteration which is computational complex, and the parameters involved in these algorithms should be tuned carefully to achieve a good performance.

To reduce the computational complexity of SVD involved in above algorithms, iterative hard thresholding (IHT) algorithms were proposed [7, 8]. These algorithms involve a projection step which projects a matrix into a low-rank space at each iteration. The convergence of these algorithms are very fast when the rank is small and the computational complexity are also very low since only a few singular values need to calculated. In [9, 10, 11], the alternation minimization methods for matrix completion were proposed. The advantage of these methods is that no SVD is required.

All above algorithms only give the convergence guarantee when certain RIP holds, no convergence prediction can be made for a given linear operator. In this paper, we propose a fast-convergence and low-complexity algorithm for low-rank matrix recovery called the turbo-type affine rank minimization (TARM) algorithm. TARM is inspired by the idea of denoising-based Turbo-CS for compressed sensing proposed in [12]. Based on the turbo principle in [12], the parameters in TARM can be determined and a scalar function called state evolution can be derived for TARM when the linear operator is right-orthogonally invariant. The state evolution accurately predicts the per iteration MSE of TARM, as shown in our numerical results. We compare our algorithm to the existing algorithms for low-rank matrix recovery. Numerical results show that our proposed algorithm works much better than all the existing algorithms in terms of both computational complexity and converge rate. TARM can also be extended to

matrix completion problem. Due to page restriction, we will not discuss TARM for matrix completion in this paper.

## 2. THE TARM ALGORITHM

The diagram of TARM is illustrated in Fig. 1. There are two concatenated modules in TARM. Module A estimates the low-rank matrix $\boldsymbol{X}^*$ via a linear estimator $\mathcal{E}(\cdot)$ based on the linear observation $\boldsymbol{y}$ and the input $\boldsymbol{X}$. Then the function $\mathcal{E}^{ext}(\cdot)$, which linearly combines $\mathcal{E}(\boldsymbol{X})$ and $\boldsymbol{X}$, is employed to decorrelate the input and output estimation errors of Module A. The superscript "ext" stands for extrinsic, since the output of $\mathcal{E}^{ext}(\cdot)$ is usually referred to as an extrinsic message. In Module B, the output $\boldsymbol{R}$ of Module A is passed to a denoiser $\mathcal{D}(\cdot)$ which suppresses the estimation error by exploiting the low-rank structure of $\boldsymbol{X}^*$. The denoised output $\boldsymbol{Z}$ is then passed to a function $\mathcal{D}^{ext}(\cdot)$ which linearly combines $\boldsymbol{Z}$ and $\boldsymbol{R}$ for input output errors decorrelation. The two modules are executed iteratively to refine the estimation. By comparison, we see that the diagram of TARM in Fig. 1 is a matrix version of the D-Turbo-CS algorithm in Fig. 2 of [12].

The details of TARM are presented in Algorithm 1. We shall note that there are various choices of $\mathcal{D}(\cdot)$ in the literature, such as the best rank-$r$ approximation [13] and the singular value thresholding [14]. In this paper, due to its outstanding performance, we focus on the best rank-$r$ approximation defined by

$$\mathcal{D}(\boldsymbol{R}) = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{4}$$

where $\sigma_i, \boldsymbol{u}_i$, and $\boldsymbol{v}_i$ are respectively the $i$-th singular value and the corresponding left and right singular vectors of the input $\boldsymbol{R}$.

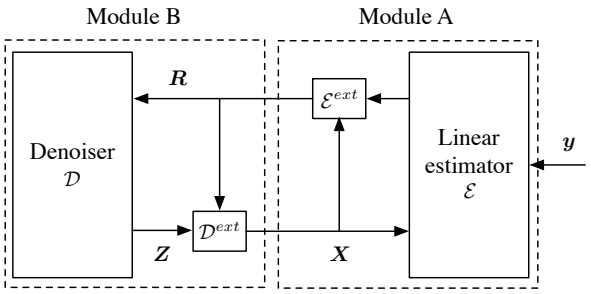


**Fig. 1**: The diagram of the TARM algorithm.

## 3. LOW-RANK MATRIX RECOVERY

We now consider low-rank matrix recovery with linear operator $\mathcal{A}$ being right-orthogonally invariant. Denote the vector form of $\boldsymbol{X}$ by $\boldsymbol{x} = \text{vec}(\boldsymbol{X}) = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_n^T]^T$, where $\boldsymbol{x}_i$ is the $i$th column of $\boldsymbol{X}$. The linear operator $\mathcal{A}$ can be generally expressed as $\mathcal{A}(\boldsymbol{X}) = \boldsymbol{A}\text{vec}(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a matrix representation of $\mathcal{A}$. Consider a linear operator $\mathcal{A}$ with matrix form $\boldsymbol{A}$ and SVD form $\boldsymbol{A} = \boldsymbol{U}_A \boldsymbol{\Sigma}_A \boldsymbol{V}_A^T$, where $\boldsymbol{U}_A$ and $\boldsymbol{V}_A$ are orthogonal matrices and $\boldsymbol{\Sigma}_{\boldsymbol{A}}$ is a diagonal matrix, if $\boldsymbol{V}_A$ is a Haar distributed random matrix independent of $\boldsymbol{\Sigma}_A$, we say that $\mathcal{A}$ is a right-orthogonally invariant linear (ROIL) operator. We focus on two types of ROIL operators in this paper: 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. The linear operator $\mathcal{A}$ is normalized such that the length of each row vector of $\boldsymbol{A}$ is 1.

**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: $\boldsymbol{R}^{(t)} = \boldsymbol{X}^{(t-1)} + \mu_t \mathcal{A}^T(\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}^{(t-1)}))$ // Module A
4: $\boldsymbol{Z}^{(t)} = \mathcal{D}(\boldsymbol{R}^{(t)})$ // Denoiser $\mathcal{D}$
5: $\boldsymbol{X}^{(t)} = c_t(\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)})$ // $\mathcal{D}^{ext}$
6: **end while**
**Output:** $\boldsymbol{Z}^{(t)}$

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

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

Similar assumptions have been introduced in the design of Turbo-CS in [15] (see also [16]). Later these assumptions were rigorously analysed in [17, 18] using the conditioning technique [19], based on which the state evolution was established to characterized the behavior of the Turbo-CS algorithm. However, the analysis in [17] is focused on the case that the denoiser $\mathcal{D}(\cdot)$ is separable, i.e., the function $\mathcal{D}(\cdot)$ is individually applied to each element of the input, while the denoisers involved here (such as the best-rank-$r$ appriximation in (6)) are non-separable. Therefore, the technique in [17] and [19] cannot be applied here. In this paper, we introduce these assumptions without rigorous proof. The recent work [20] on state evolution of AMP for non-seperable denoisers may shed some light on a possible rigorous justification of our conjecture. We will give simulations to verify our assumptions.

### 3.1. Evaluation of $\{\mu_t\}, \{\alpha_t\}$, and $\{c_t\}$

We first note that when we set $c_t = 1$ and $\alpha_t = 0$ for any $t$, TARM reduces to the SVP and NIHT algorithms (with different choices of $\mu_t$). However, in TARM, we follow the turbo principles [12, 15] to determine these parameters. That is, $\{\mu_t\}, \{c_t\}$, and $\{\alpha_t\}$ need to satisfy the following conditions:

- Condition 1: $\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{X}^{(t-1)} - \boldsymbol{X}^* \right\rangle = 0$,
- Condition 2: $\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{X}^{(t)} - \boldsymbol{X}^* \right\rangle = 0$,
- Condition 3: $\|\boldsymbol{X}^{(t)} - \boldsymbol{X}^*\|_F^2$ is minimized.

Condition 1 requires that the input and output estimation errors of Module A are uncorrelated. Similarly, Condition 2 requires that the input and output estimation errors of Module B are uncorrelated. Condition 3 requires that the mean square error (MSE) of the output estimation error of Module B is minimized.

We next approximately calculate these parameters satisfy these conditions under large system limit. We start with $\mu_t$. From Condition 1, we have

$$\mu_t = \frac{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}^*\|_F^2}{\|\mathcal{A}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}^*)\|_2^2} \tag{5a}$$

$$= \frac{1}{\tilde{\boldsymbol{x}}^T \boldsymbol{V}_A \boldsymbol{\Sigma}_A^T \boldsymbol{\Sigma}_A \boldsymbol{V}_A^T \tilde{\boldsymbol{x}}} \tag{5b}$$

$$= \frac{1}{\boldsymbol{v}_A^T \boldsymbol{\Sigma}_A^T \boldsymbol{\Sigma}_A \boldsymbol{v}_A} \tag{5c}$$

$$= \frac{n}{m} \tag{5d}$$

where (5a) follows from Condition 1, (5b) follows by utilizing the matrix form of $\mathcal{A}$, $\tilde{\boldsymbol{x}} = \frac{\mathrm{vec}(\boldsymbol{X}^{(t-1)} - \boldsymbol{X}^*)}{\|\boldsymbol{X}^{(t-1)} - \boldsymbol{X}^*\|_F}$, (5c) follows by letting $\boldsymbol{V}_A^T \tilde{\boldsymbol{x}} = \boldsymbol{v}_A$, and (5d) follows from the facts that (i) $\boldsymbol{v}_A$ is a unit vector uniformly distributed over the sphere (since $\|\tilde{\boldsymbol{x}}\|_2 = 1$ and $\boldsymbol{V}_A$ is haar distributed), (ii) $\boldsymbol{V}_A$ is independent of $\boldsymbol{\Sigma}_A$, and (iii) $\|\boldsymbol{\Sigma}_A\|_F^2 = m$.

We next consider the approximation of $\alpha_t$. We first note that

$$\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{X}^{(t)} - \boldsymbol{X}^* \right\rangle \tag{6a}$$

$$= \left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, c_t(\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}) - \boldsymbol{X}^* \right\rangle \tag{6b}$$

$$\approx c_t \left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)} \right\rangle \tag{6c}$$

where (6a) follows by substituting $\boldsymbol{X}^{(t)}$ in line 5 of Algorithm 1, and (6b) follows from Assumption 2 that $\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{X}^* \right\rangle \approx 0$. Combining (6) and Condition 2, we have

$$\alpha_t = \frac{\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{Z}^{(t)} \right\rangle}{\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \boldsymbol{R}^{(t)} \right\rangle} \tag{7a}$$

$$= \frac{\left\langle \boldsymbol{R}^{(t)} - \boldsymbol{X}^*, \mathcal{D}(\boldsymbol{R}^{(t)}) \right\rangle}{n v_t} \tag{7b}$$

$$\approx \frac{1}{n} \sum_{i,j} \frac{\partial \mathcal{D}(\boldsymbol{R}^{(t)})}{\partial R_{i,j}^{(t)}} \tag{7c}$$

$$= \frac{1}{n} \mathrm{div}(\mathcal{D}(\boldsymbol{R}^{(t)})) \tag{7d}$$

where (7b) follows from $\boldsymbol{Z}^{(t)} = \mathcal{D}(\boldsymbol{R}^{(t)})$ and Assumption 2 that the elements of $\boldsymbol{R}^{(t)} - \boldsymbol{X}^*$ are i.i.d. Gaussian with variance $v_t$, the approximation in (7c) follows from Stein's lemma [21] since the entries of $\boldsymbol{R}^{(t)} - \boldsymbol{X}^*$ are i.i.d. Gaussian distributed, and (7d) is from the definition of the divergence.

Finally, we consider $c_t$. We first note

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2 = \|\boldsymbol{X}^{(t)} - \boldsymbol{X}^*\|_F^2 + \|\boldsymbol{X}^* - \boldsymbol{R}^{(t)}\|_F^2 + 2\left\langle \boldsymbol{X}^{(t)} - \boldsymbol{X}^*, \boldsymbol{X}^* - \boldsymbol{R}^{(t)} \right\rangle \tag{8a}$$

$$= \|\boldsymbol{X}^{(t)} - \boldsymbol{X}^*\|_F^2 + \|\boldsymbol{X}^* - \boldsymbol{R}^{(t)}\|_F^2 \tag{8b}$$

where (8b) follows from Condition 2. Recall that in the $t$-th iteration $\boldsymbol{R}^{(t)}$ is a function of $\mu_t$ but not of $\alpha_t$ and $c_t$. Thus, minimizing $\|\boldsymbol{X}^{(t)} - \boldsymbol{X}^*\|_F^2$ is equivalent to minimizing $\|\boldsymbol{X}^{(t)} - \boldsymbol{R}^{(t)}\|_F^2$. For any given $\alpha_t$, the optimal $c_t$ is given by

$$c_t = \frac{\left\langle \boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}, \boldsymbol{R}^{(t)} \right\rangle}{\|\boldsymbol{Z}^{(t)} - \alpha_t \boldsymbol{R}^{(t)}\|_F^2}. \tag{9}$$

### 3.2. State Evolution of TARM

Based on Assumptions 1 and 2, we derive the mean square error (MSE) transfer functions of the two modules in TARM separately to accurately characterize the per iteration MSE performance of the TARM algorithm.

When $\mathcal{A}$ is a ROIL operator and the empirical distribution of eigenvalue $\theta$ of $\frac{1}{n_2}(\boldsymbol{X}^*)^T \boldsymbol{X}^*$ converges almost surely to the density function $p(\theta)$ as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$, $\frac{r}{n_2} \to \lambda$, the state evolution of TARM is given by

$$v_t = f(\tau_t) \tag{10a}$$

$$\tau_{t+1} = g(v_t) \tag{10b}$$

where $\tau_t$ and $v_t$ are respectively the input noise level of Module A and Module B. For partial orthogonal ROIL operator, $f(\tau) = \left(\frac{1}{\delta} - 1\right)\tau$; for Gaussian ROIL operator, $f(\tau) = \frac{1}{\delta}\tau$; and $g(v)$ is given by

$$g(v) = \frac{v - \lambda\left(1 + \frac{1}{\rho}\right)v - \lambda v^2 \Delta}{\frac{v - \lambda\left(1+\frac{1}{\rho}\right)v - \lambda v^2 \Delta}{1 + \lambda\left(1+\frac{1}{\rho}\right)v + \lambda v^2 \Delta}\alpha^2 + (1-\alpha)^2} - v \tag{11}$$

where

$$\alpha = \left|1 - \frac{1}{\rho}\right| \lambda - \frac{1}{\rho}\lambda^2 + 2\min\left(1, \frac{1}{\rho}\right)\lambda \tag{12}$$

$$\Delta = \frac{1}{r}\sum_{i=1}^{r} \frac{1}{\theta} \xrightarrow{\text{a.s.}} \int_0^\infty \frac{1}{\theta} p(\theta) d\theta \tag{13}$$

as $n_1, n_2, r \to \infty$ with $\frac{n_1}{n_2} \to \rho$, $\frac{r}{n_2} \to \lambda$. The constant $\Delta$ is difficult to obtain since $p(\theta)$ is usually unknown. We give an upper bound that does not depend on $\Delta$:

$$g(v) \le \bar{g}(v) = \frac{v - \lambda(1 + \frac{1}{\rho})v}{(1-\alpha)^2} - v. \tag{14}$$

We will show in the numerical results that the upper bound is tight enough and is good for performance prediction.

## 4. NUMERICAL RESULTS

For the case of partial orthogonal ROIL operator, we generate a partial orthogonal ROIL operator with the matrix form $\boldsymbol{A} = \boldsymbol{SW}$, where $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is a random perturbation matrix and $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is a DCT matrix and for the case of a Gaussian ROIL operator, i.e. $\boldsymbol{A}$ is an i.i.d. Gaussian random matrix of size $m \times n$ with the elements drawn from $\mathcal{N}(0, \frac{1}{n})$. The rank-$r$ matrix $\boldsymbol{X}^* \in \mathbb{R}^{n_1\times n_2}$ is generated by the multiplication of two i.i.d. Gaussian matrices of size $n_1 \times r$ and $r \times n_2$. Note that, although we only consider the noiseless cases in simulation, TARM can be applied to noisy cases.

### 4.1. State evolution

To show how accurate the state evolution of TARM is, we plot the simulation and state evolutions curves of TARM in Fig. 2. From these two figures, we see that the state evolution of TARM is accurate when the dimension is large enough for both partial orthogonal and Gaussian ROIL operator. We also note that the upper bound given by (14) is accurate enough for prediction even the singular value distribution of $\boldsymbol{X}^*$ is not given.

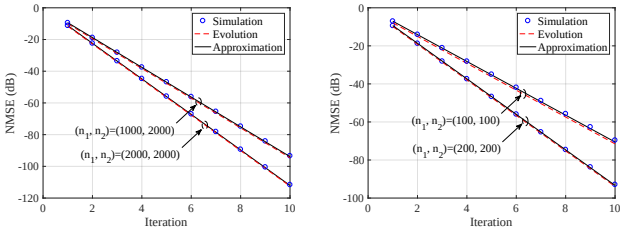


**Fig. 2**: Left: State evolution of TARM for partial orthogonal ROIL operator. $r = 40, m/n = 0.35$, and the dimensions of $\boldsymbol{X}^*$ are shown in the plot. Right: State evolution of TARM for Gaussian ROIL operator. $r = 4, m/n = 0.35$, the dimensions of $\boldsymbol{X}^*$ are shown in the plot.

### 4.2. Performance comparison

We compare TARM with the existing state-of-the-art algorithms for low-rank matrix recovery including singular value projection (SVP) [7], normalized iterative hard thresholding [8], and Riemannian gradient descent (RGrad) [22]. All algorithms are run under the same settings. The per iteration normalized mean square error of each algorithm is plotted in Fig. 3. From these figures, we see that TARM converges much faster than NIHT and RGrad for both Gaussian ROIL operators and partial orthogonal ROIL operators. Note that all the listed algorithms have similar per iteration running time and thus we have the same results when comparing by recovery time.

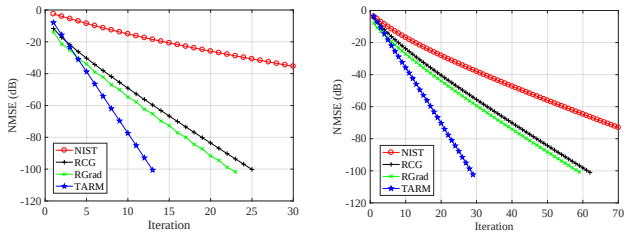


**Fig. 3**: Comparison of algorithms for low-rank matrix recovery. Left: $\mathcal{A}$ is a partial orthogonal ROIL operator with $n_1 = n_2 = 1000$, $r = 50$, and $m/n = 0.39$. Right: $\mathcal{A}$ is a random Gaussian ROIL operator with $n_1 = n_2 = 80$, $r = 10$, and $\frac{m}{r(n_1+n_2-r)} = 3$.

### 4.3. Phase transition

We consider an algorithm to be successful in recovering the low-rank matrix $\boldsymbol{X}^*$ when the following conditions are satisfied: 1) the normalized mean square error $\frac{\|\boldsymbol{X}^{(t)}-\boldsymbol{X}^*\|_F^2}{\|\boldsymbol{X}^*\|_F^2} \leq 10^{-6}$; 2) the iteration time $t < 1000$. The dimension of the manifold of $n_1 \times n_2$ matrices of rank $r$ is $r(n_1+n_2-r)$ [23]. So the minimal number of measurement is $r(n_1 + n_2 - r)$. Thus, the upper bound for successful recovery is $r_{max} \leq \frac{n_1+n_2-\sqrt{(n_1+n_2)^2-4m}}{2}$. From Fig. 4, we see that the phase transition of TARM is closest to the upper bound and considerably higher than the phase transitions of NIHT and RGrad.

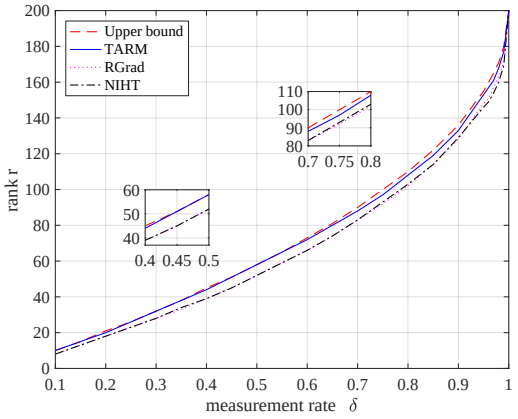


**Fig. 4**: Phase transition of TARM for low-rank matrix recovery with partial orthogonal ROIL operator. $n_1 = n_2 = 200$. For each algorithm, the region below the phase transition curve succeed to the recovery of $\boldsymbol{X}^*$.

## 5. CONCLUSIONS

In this paper, we proposed the TARM algorithm for affine rank minimization problem. TARM shows competitive performance compared with the existing methods in both convergence rate and phase transition.

## 6. REFERENCES


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

[2] 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, Mar. 2016.

[3] 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.

[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, Nov. 2009.

[5] 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.

[6] 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. 615-640, pp. 15, Mar. 2010.

[7] P. Jain, R. Meka, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," in *Proceeding of the Neural Information Processing Systems Conference*, Hyatt Regency, Vancouver CANADA, pp. 937–945, Dec. 2010.

[8] 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.

[9] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2980–2998, May 2010.

[10] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*. Palo Alto, CA, pp. 665–674, June 2013.

[11] D. Gamarnik and S. Misra, "A note on alternating minimization algorithm for the matrix completion problem," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1340–1343, June 2016.

[12] Z. Xue, J. Ma, and X. Yuan, "Denoising-based turbo compressed sensing," *IEEE Access*, 2017.

[13] 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.

[14] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717, Apr. 2009.

[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] J. Ma and L. Ping, "Orthogonal amp," *IEEE Access*, vol. 5, pp. 2020–2033, Jan. 2017.

[17] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, pp. 1588–1592, Aug. 2017.

[18] K. Takeuchi, "Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements," *arXiv preprint arXiv:1701.05284*, 2017.

[19] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 764–785, Jan. 2011.

[20] R. Berthier, A. Montanari, and P.-M. Nguyen, "State evolution for approximate message passing with non-separable functions," *arXiv preprint arXiv:1708.03950*, 2017.

[21] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. STAT.*, pp. 1135–1151, 1981.

[22] 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.

[23] B. Vandereycken, "Low-rank matrix completion by riemannian optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, June 2013.