

# Low-Rank Phase Retrieval

Namrata Vaswani, Seyedehsara Nayer, and Yonina C. Eldar, *Fellow, IEEE*

**Abstract**—We develop two iterative algorithms for solving the low-rank phase retrieval (LRPR) problem. LRPR refers to recovering a low-rank matrix  $X$  from magnitude-only (phaseless) measurements of random linear projections of its columns. Both methods consist of a spectral initialization step followed by an iterative algorithm to maximize the observed data likelihood. We obtain sample complexity bounds for our proposed initialization approach to provide a good approximation of the true  $X$ . When the rank is low enough, these bounds are significantly lower than what existing single vector phase retrieval algorithms need. Via extensive experiments, we show that the same is also true for the proposed complete algorithms.

**Index Terms**—Phase retrieval, low rank matrix recovery, sparse representations.

## I. INTRODUCTION

IN RECENT years there has been a large amount of work on the phase retrieval (PR) problem and on its generalization. The original PR problem involves recovering a length- $n$  signal  $x$  from the magnitudes of its discrete Fourier transform (DFT) coefficients. Generalized PR replaces the DFT by inner products with any set of measurement vectors,  $\mathbf{a}_i$ . Thus, the goal is to recover  $x$  from  $|\mathbf{a}_i'x|^2$ ,  $i = 1, 2, \dots, m$ . These magnitude-only measurements are referred to as *phaseless measurements*. PR is a classical problem that occurs in many applications such as X-ray crystallography, astronomy, and ptychography because the phase information is either difficult or impossible to obtain [3]. Algorithms for solving it have existed since the work of Gerchberg and Saxton and Fineup [4], [5]. In recent years, there has been much renewed interest in PR, e.g., [3], [6]–[18] and in sparse PR, e.g., [19]–[21].

One popular class of approaches, pioneered in Candès *et al.* [6], [7], solves PR by recovering the rank one matrix  $Z := xx'$  from  $\mathbf{y}_i := |\mathbf{a}_i'x|^2 = \text{trace}(\mathbf{a}_i\mathbf{a}_i'Z)$  via a semi-definite relaxation. This can provably recover  $x$  (up to a global phase uncertainty) using only  $m = cn$  independent identically distributed (iid) Gaussian phaseless

measurements. However because of the ‘lifting’, its computational and storage complexity depends on  $n^2$  instead of on  $n$  (it needs to recover an  $n \times n$  matrix instead of an  $n$ -length vector). Here, and throughout the paper, the letter  $c$  is re-used to denote different numerical constants each time it is used.

In more recent works, non-convex methods, that do not lift the problem to higher dimensions, have been explored along with provable guarantees [9]–[11]. An alternating minimization (AltMin) technique with spectral initialization, AltMinPhase, was developed and analyzed in [9]. The AltMin step of this approach is essentially the same as the old Gerchberg-Saxton algorithm [4]. A gradient descent method with spectral initialization, called Wirtinger Flow (WF), was studied in [10]. In [11], truncated WF (TWF), which introduced a truncation technique to further improve WF performance, was developed. It was shown that TWF recovers  $x$  from only  $cn$  iid Gaussian phaseless measurements, while the number of iterations needed for getting an error of order  $\epsilon$  is  $c \log(1/\epsilon)$  (converges geometrically). AltMinPhase and WF require more measurements,  $cn \log^3 n$  and  $cn \log n$ , respectively. WF also has a slower convergence rate. Two recent modifications of TWF [22], [23] have the same order complexities but improved empirical performance.

*Problem Setting:* In this work, instead of a single vector  $x$ , we consider a set of  $q$  vectors,  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$ , such that the  $n \times q$  matrix,

$$\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q],$$

has rank  $r \ll \min(n, q)$ . For each column  $\mathbf{x}_k$  of  $\mathbf{X}$ , we observe a set of  $m$  measurements of the form

$$\mathbf{y}_{i,k} := |\mathbf{a}_{i,k}'\mathbf{x}_k|^2, \quad i = 1, 2, \dots, m, \quad k = 1, 2, \dots, q. \quad (1)$$

The measurement vectors,  $\mathbf{a}_{i,k}$ , are mutually independent. Our goal is to recover the matrix  $\mathbf{X}$  from these  $m$  phaseless measurements  $\mathbf{y}_{i,k}$ . Since we have magnitude-only measurements of each column  $\mathbf{x}_k$ , we can only hope to recover each column  $\mathbf{x}_k$  up to a global phase ambiguity. We refer to the above problem as *low rank phase retrieval (LRPR)*.

In some applications, the goal may be to only recover the span of the columns of  $\mathbf{X}$ ,  $\text{range}(\mathbf{X})$ . This would be the case, for example, if one is interested in only seeing the principal directions of variation of the dataset, and not in recovering the dataset itself. We refer to this easier problem as *phaseless PCA (principal component analysis)*.

A motivating application for LRPR is dynamic astronomical imaging such as solar imaging where the sun’s surface properties gradually change over time [24]. The changes are usually due to a much smaller number of factors,  $r$ , than the size of the image,  $n$ , or the total number of images,  $q$ . If the images are arranged as 1D vectors  $\mathbf{x}_k$ , then the resulting matrix is approximately low rank. As another potential application, consider a Fourier ptychography imaging system that captures a dynamic scene exhibiting a temporal evolution; this is often the case when observing live biological specimens in vitro. Suppose the scene resolution is  $n$  and the total number of captured frames is  $q$ . If the dynamics is approximated to be linear and slow changing, then the matrix formed by stacking the frames next to each other can be modeled as a rank- $r$  matrix, where  $r \ll \min(q, n)$ . Similar applications involving a sequence of gradually changing images also occur in X-ray and sub-diffraction imaging systems. Moreover, if we are only interested in identifying the principal directions of

Manuscript received July 29, 2016; revised December 24, 2016 and January 31, 2017; accepted January 31, 2017. Date of publication March 21, 2017; date of current version June 5, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Yves Tourneret. This paper was presented in part at the IEEE Statistical Signal Processing Workshop, Palma de Mallorca, Spain, June 26–29, 2016. A short version of this work will be presented in part at the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, March 5–9, 2017. (Corresponding author: Namrata Vaswani.)

N. Vaswani and S. Nayer are with the Iowa State University, Ames, IA 50011 USA (e-mail: namrata@iastate.edu; sarana@iastate.edu).

Y. C. Eldar is with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: yonina@ee.technion.ac.il).

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the authors. This includes additional material not included within the paper itself. This material is 265 KB in size.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2017.2684758

variation of the image sequences, then the problem becomes that of phaseless PCA. PCA is often the first step for classification, clustering, modeling, or other exploratory data analysis.

*Contributions:* This work has two contributions. We propose iterative algorithms for solving the LRPR problem described above. Our solution approach relies on the fact that a rank  $r$  matrix  $\mathbf{X}$  can be expressed (non-uniquely) as  $\mathbf{X} = \mathbf{U}\mathbf{B}$  where  $\mathbf{U}$  is an  $n \times r$  matrix with mutually orthonormal columns. Its first step consists of a spectral initialization step, motivated by TWF, for first initializing  $\mathbf{U}$ , and then, the columns of  $\mathbf{B}$ . The remainder of the algorithm is developed in one of two ways: using a projected gradient descent strategy to modify the TWF iterates (LRPR1); or an AltMin algorithm, motivated by AltMinPhase, that directly exploits the decomposition  $\mathbf{X} = \mathbf{U}\mathbf{B}$  (LRPR2). Via extensive experiments, we demonstrate that both LRPR1 and LRPR2 have better sample complexity than TWF; with LRPR2 being the best. Moreover, when enough measurements are available for TWF to work, we show that the LRPR initialization can also be used to speed up basic TWF for solving LRPR.

Our second, and most important, contribution is a sample complexity bound for the proposed initialization to get within an  $\varepsilon$  ball of the true  $\mathbf{X}$ . Our results show that, if the goal is to only initialize  $\mathbf{U}$  with subspace recovery error below a fixed level, say  $\varepsilon = 1/4$ , then a total of  $mq = cnr^2/\varepsilon^2 = 16cnr^2$  iid Gaussian measurements suffice with high probability (whp). When  $r$  is small,  $nr^2$  is only slightly larger than  $nr$  which is the minimum required by any technique to recover the span of  $\mathbf{U}$ . If the goal is to also initialize the  $\mathbf{x}_k$ 's with normalized error below say  $\varepsilon = 1/4$ , then we need more measurements, but still significantly fewer than TWF. For example, if  $r \leq c \log n$  and  $q \geq cn$ , then, only  $16c\sqrt{n}$  measurements per column are required. We note that our guarantees assume that a different set of measurements is used for initializing  $\mathbf{U}$  and  $\mathbf{B}$  (see Model 3.1).

As seen in many earlier works, e.g., AltMinPhase [9], resampled WF [10, Algorithm 2 and Theorem 5.1] or TWF [11], the sample complexity of the entire algorithm is equal to or smaller than that of the initialization step for a fixed error level<sup>1</sup>. This is why initialization guarantees are important.

Our problem setting assumes a *different (mutually independent) set of measurement vectors is used for imaging each column  $\mathbf{x}_k$* . This is critical for guaranteeing the improved sample complexity of our solution approach over single-vector PR methods because this is what ensures that the  $m$  matrices  $\mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'$  are all mutually independent conditioned on  $\mathbf{X}$ . Hence, we can exploit averaging over  $m$  such matrices when estimating  $\mathbf{U}$ . If  $\mathbf{a}_{i,k} = \mathbf{a}_{i,1}$  (same  $\mathbf{a}_i$ 's are used), then this benefit disappears since only  $m$  of the above matrices are mutually independent. We demonstrate this in Table I (last column) in Sec. V. We discuss the practical implications of our setting in Sec. III-D.

Two other works that also generalize WF [10], but to solve a completely different problem include [25], [26]. These study the problem of recovering a rank  $r$  matrix  $\mathbf{M}$  from measurements of the form  $\mathbf{z}_i = \text{trace}(\mathbf{A}_i' \mathbf{M})$ . This is the low rank matrix sensing problem studied in [27] and a lot of earlier and later works. In our problem, if we use the *same*  $\mathbf{a}_i$ 's for all columns  $\mathbf{x}_k$ , and define  $\mathbf{z}_i := \sum_k (\mathbf{a}_i' \mathbf{x}_k)^2$  and  $\mathbf{M} := \sum_k \mathbf{x}_k \mathbf{x}_k'$ , then we could use the strategy of [25], [26] or, in fact, any low-rank matrix sensing technique, e.g., AltMinSense from [27], to recover  $\mathbf{M}$  from the  $\mathbf{z}_i$ 's; followed by recovering  $\text{range}(\mathbf{U})$  as its column space. However, for the reasons explained above, use of same measurement vectors will not yield any advantage over single vector PR. When using different  $\mathbf{a}_{i,k}$ 's, none of these methods are applicable.

<sup>1</sup>For AltMinPhase, the initialization sample complexity (for achieving a given fixed error) is  $cn \log^3 n$  while it is only  $cn \log n$  per iteration for the rest of the algorithm. For resampled WF, it is  $cn \log^2 n$  for initialization and  $cn \log n$  for the rest of the algorithm, while for TWF, it is  $cn$  both for the initialization and for the complete algorithm.

*Notation:* The notation  $\mathbf{a}_{i,k} \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  means that the vectors  $\mathbf{a}_{i,k}$  are iid real Gaussian vectors with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ ; and  $\mathbf{b}_k \stackrel{\text{indep}}{\sim} \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$  means that the  $\mathbf{b}_k$ 's are mutually independent and  $\mathbf{b}_k$  is generated from  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ . We use  $'$  to denote matrix or vector conjugate transpose, and  $\|\cdot\|_p$  to denote the  $l_p$  norm of a vector or the induced  $l_p$  norm of a matrix. When the subscript  $p$  is missing, i.e., when we just write  $\|\cdot\|$ , it denotes the  $l_2$  norm of a vector or the induced  $l_2$  norm of a matrix. We use  $\mathbf{I}$  to denote the identity matrix. The notation  $\mathbb{1}_\zeta$  is the indicator function for statement  $\zeta$ , i.e.,  $\mathbb{1}_\zeta = 1$  if  $\zeta$  is true and  $\mathbb{1}_\zeta = 0$  otherwise. For a vector  $\mathbf{z}$ ,  $|\mathbf{z}|$ ,  $\sqrt{\mathbf{z}}$  and  $\text{phase}(\mathbf{z})$  compute the *element-wise* magnitude, square-root, and phase of each entry of  $\mathbf{z}$ , and  $\text{diag}(\mathbf{z})$  creates a diagonal matrix with entries from  $\mathbf{z}$ .

*Paper Organization:* In Sec. II, we develop the proposed LRPR initialization approach (LRPR-init). We obtain sample complexity bounds for it in Sec. III. In Sec. IV, we explain how LRPR-init can be used to develop iterative algorithms for LRPR that are either faster than basic TWF (LRPR+TWF) or need a smaller  $m$  to work (LRPR1 and LRPR2). Numerical experiments backing our claims are shown in Sec. V. We prove our results from Sec. III in Sec. VI, and conclude in Sec. VII.

The algorithms proposed in this work are applicable for both real and complex measurements. Experiments are shown for both cases too. Moreover, as shown in our experiments, our algorithms also apply to noisy measurements. However, for simplicity, we state and prove our guarantees only for the real Gaussian measurements' case. Their extension to complex Gaussian measurements is straightforward.

## II. LOW RANK PR (LRPR) INITIALIZATION

Our goal is to recover an  $n \times q$  low rank matrix  $\mathbf{X}$  from phaseless measurements of linear projections of each of its columns, i.e., from  $\mathbf{y}_{i,k} := (\mathbf{a}_{i,k}' \mathbf{x}_k)^2$ ,  $i = 1, 2, \dots, m$ , and  $k = 1, 2, \dots, q$ . In this section, we develop an approach to obtain an initial estimate of  $\mathbf{X}$  that relies on the fact that a rank  $r$  matrix  $\mathbf{X}$  can be expressed (non-uniquely<sup>2</sup>) as  $\mathbf{X} = \mathbf{U}\mathbf{B}$  where  $\mathbf{U}$  is an  $n \times r$  matrix with mutually orthonormal columns and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_q]$  is an  $r \times q$  matrix. The proposed initialization approach first computes an estimate of  $\text{range}(\mathbf{U})$ , i.e., it returns  $\hat{\mathbf{U}}$  that may be very different from  $\mathbf{U}$  in Frobenius norm, but their spans are close, i.e., the subspace error,  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$  is small. Here,

$$\text{SE}(\hat{\mathbf{U}}, \mathbf{U}) := \|(\mathbf{I} - \hat{\mathbf{U}}\hat{\mathbf{U}}')\mathbf{U}\|$$

quantifies the subspace error (principal angle) between the range spaces of two matrices  $\hat{\mathbf{U}}, \mathbf{U}$  with mutually orthonormal columns. Using  $\hat{\mathbf{U}}$ , we find estimates  $\hat{\mathbf{b}}_k$  so that  $\text{dist}(\hat{\mathbf{U}}\hat{\mathbf{b}}_k, \mathbf{x}_k)$  is small. Here,

$$\text{dist}(\mathbf{z}_1, \mathbf{z}_2) := \min_{\phi \in [0, 2\pi]} \|\mathbf{z}_1 - e^{\sqrt{-1}\phi} \mathbf{z}_2\|$$

quantifies the phase-invariant distance<sup>3</sup> between two complex vectors  $\mathbf{z}_1, \mathbf{z}_2$  [10], [11]. The  $\mathbf{b}_k$ 's are initialized by estimating

$$\mathbf{g}_k := \hat{\mathbf{U}}' \mathbf{x}_k = \hat{\mathbf{U}}' \mathbf{U} \mathbf{b}_k$$

for each  $k$ , and setting  $\hat{\mathbf{b}}_k = \hat{\mathbf{g}}_k$ . Because  $\hat{\mathbf{U}}$  can be arbitrarily rotated w.r.t.  $\mathbf{U}$ , this approach may not give accurate estimates of the individual  $\mathbf{b}_k$ 's, i.e.,  $\text{dist}(\mathbf{b}_k, \hat{\mathbf{b}}_k)$  may not be small.

<sup>2</sup>We can rewrite  $\mathbf{X}$  as  $\mathbf{X} = (\mathbf{U}\mathbf{R})(\mathbf{R}'\mathbf{B})$  for any rotation matrix  $\mathbf{R}$ .

<sup>3</sup>When  $\mathbf{z}_1$  and  $\mathbf{z}_2$  are both real, the phase is only  $+1$  or  $-1$ , and so,  $\text{dist}(\mathbf{z}_1, \mathbf{z}_2) = \min(\|\mathbf{z}_1 - \mathbf{z}_2\|, \|\mathbf{z}_1 + \mathbf{z}_2\|)$ .

TABLE I  
REAL GAUSSIAN MEASUREMENT VECTORS: INITIALIZATION ERROR COMPARISONS. LRPR: LRPR-INIT, TWF: TWF-INIT, TWFPROJ: TWFPROJ-INIT

$m/n$	$\Pr(\hat{r} = r)$	LRPR ( $r$ known)	LRPR ( $r$ unknown)	LRPR ( $\hat{r} = 2r$ )	LRPR (noisy meas)	TWFproj	TWF	LRPR-same
$q = 100$								
0.10	0.24	1.23	1.31	1.22	1.37	0.96	1.63	0.99
0.50	0.68	0.33	0.45	0.55	0.52	0.67	1.47	1.03
1.00	1.00	0.17	0.17	0.30	0.19	0.53	1.34	1.04
$q = 1000$								
0.10	1.00	0.39	0.39	0.70	0.47	0.71	1.62	0.98
0.50	1.00	0.10	0.10	0.26	0.12	0.57	1.47	1.02
1.00	1.00	0.05	0.05	0.13	0.06	0.48	1.34	1.04

### A. LRPR-init: Spectral Initialization for LRPR

LRPR-init is a two step approach. We first initialize  $\mathbf{U}$  using a truncated spectral initialization idea [11]. For this, define

$$\mathbf{Y}_{U,0} := \frac{1}{mq} \sum_{i=1}^m \sum_{k=1}^q \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'.$$

Let  $\frac{1}{q} \sum_{k=1}^q \mathbf{x}_k \mathbf{x}_k' \stackrel{\text{EVD}}{=} \mathbf{U} \bar{\Lambda} \mathbf{U}'$  denote the reduced eigenvalue decomposition (EVD) of  $\mathbf{X} \mathbf{X}'/q$ . Thus,  $\mathbf{U}$  is an  $n \times r$  matrix with orthonormal columns and  $\bar{\Lambda}$  is an  $r \times r$  diagonal matrix. It is not hard to see that [10, Lemma A.1],

$$\mathbb{E}[\mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'] = 2\mathbf{x}_k \mathbf{x}_k' + \|\mathbf{x}_k\|^2 \mathbf{I}. \quad (2)$$

and, therefore,

$$\mathbb{E}[\mathbf{Y}_{U,0}] = 2\mathbf{U} \bar{\Lambda} \mathbf{U}' + \text{trace}(\bar{\Lambda}) \mathbf{I}.$$

Clearly, the subspace spanned by the top  $r$  eigenvectors of this matrix is equal to  $\text{range}(\mathbf{U})$  and the gap between its  $r$ -th and  $(r+1)$ -th eigenvalue is  $2\lambda_{\min}(\bar{\Lambda})$ . If  $m$  and  $q$  are large enough, then, one can use an appropriate law of large numbers' result to argue that  $\mathbf{Y}_{U,0}$  will be close to its expected value whp. By the  $\sin \theta$  theorem [28], as long as  $2\lambda_{\min}(\bar{\Lambda})$  is large compared to  $\|\mathbf{Y}_{U,0} - \mathbb{E}[\mathbf{Y}_{U,0}]\|$ , the same will also be true for the span of the top  $r$  eigenvectors of  $\mathbf{Y}_{U,0}$ .

However, as explained in [11], because  $\mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'$  can be written as  $\mathbf{w} \mathbf{w}'$  with  $\mathbf{w}$  a heavy-tailed random vector, more samples will be needed for the law of large numbers to take effect than if  $\mathbf{w}$  were not heavy-tailed. To remedy this situation, we use the truncation idea suggested in [11] and compute  $\hat{\mathbf{U}}$  as the top  $r$  eigenvectors of

$$\mathbf{Y}_U := \frac{1}{mq} \sum_i \sum_k \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}' \mathbb{1}_{\{y_{i,k} \leq 9 \frac{\sum_i y_{i,k}}{m}\}}. \quad (3)$$

The idea of truncation is to average only over those  $(i, k)$ 's for which  $y_{i,k}$  is not too far from its empirical mean.

Next we consider initialization of the  $\mathbf{b}_k$ 's. Define the matrix

$$\mathbf{M}_k := \frac{1}{m} \sum_i \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'. \quad (4)$$

Suppose that  $\hat{\mathbf{U}}$  is independent of the  $\mathbf{M}_k$ 's. Then, from (2), conditioned on  $\hat{\mathbf{U}}$ ,

$$\mathbb{E}[\hat{\mathbf{U}}' \mathbf{M}_k \hat{\mathbf{U}}] = \hat{\mathbf{U}}' (2\mathbf{x}_k \mathbf{x}_k' + \|\mathbf{x}_k\|^2 \mathbf{I}) \hat{\mathbf{U}} = 2\mathbf{g}_k \mathbf{g}_k' + \|\mathbf{x}_k\|^2 \mathbf{I}.$$

The top eigenvector of this expectation is proportional to  $\mathbf{g}_k$  and the gap between its first and second eigenvalues is  $2\|\mathbf{g}_k\|^2 = 2\|\hat{\mathbf{U}}' \mathbf{U} \mathbf{b}_k\|^2$ . Thus, as long as  $\hat{\mathbf{U}}$  is a good estimate of  $\mathbf{U}$  (in terms of SE), the eigen-gap will be close to  $2\|\mathbf{b}_k\|^2$ . Therefore, we can argue that the normalized top eigenvector of  $\hat{\mathbf{U}}' \mathbf{M}_k \hat{\mathbf{U}}$ , denoted  $\hat{\mathbf{v}}_k$ , will be a good estimate of  $\mathbf{v}_k := \mathbf{g}_k / \|\mathbf{g}_k\|$ . Using this idea, we initialize the  $\mathbf{x}_k$ 's as  $\hat{\mathbf{x}}_k = \hat{\mathbf{U}} \hat{\mathbf{v}}_k \hat{\nu}_k$  where  $\hat{\nu}_k = \sqrt{\sum_i y_{i,k} / m}$  is an estimate of  $\nu_k := \|\mathbf{g}_k\|$ . We do not use truncation here because  $\mathbf{g}_k$  is an

---

### Algorithm 1: Low Rank PR Initialization (LRPR-init).

---

Set  $\hat{r} = \arg \max_j (\lambda_j(\mathbf{Y}_U) - \lambda_{j+1}(\mathbf{Y}_U))$  with  $\mathbf{Y}_U$  defined in (3).

- 1) Compute  $\hat{\mathbf{U}}$  as top  $\hat{r}$  eigenvectors of  $\mathbf{Y}_U$ .
- 2) For each  $k = 1, 2, \dots, q$ ,
  - a) compute  $\hat{\mathbf{v}}_k$  as the top eigenvector of  $\hat{\mathbf{U}}' \frac{1}{m} \sum_i \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}' \hat{\mathbf{U}}$ .
  - b) compute  $\hat{\nu}_k := \sqrt{\frac{1}{m} \sum_i y_{i,k}}$ ; set  $\hat{\mathbf{b}}_k = \hat{\mathbf{g}}_k = \hat{\mathbf{v}}_k \hat{\nu}_k$

Output  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{x}}_k := \hat{\mathbf{U}} \hat{\mathbf{b}}_k$  for all  $k = 1, 2, \dots, q$ .

---



---

### Algorithm 2: Projected-TWF initialization (TWFproj-init).

---

- 1) For each  $k = 1, 2, \dots, q$ , set  $\hat{\mathbf{x}}_k^0$  as the top eigenvector of  $\frac{1}{m} \sum_{i=1}^m \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}' \mathbb{1}_{\{y_{i,k} \leq 9 \frac{\sum_i y_{i,k}}{m}\}}$ , scaled by  $\sqrt{\sum_{i=1}^m y_{i,k} / m}$ ; create  $\hat{\mathbf{X}}^{0, TWF}$ .
  - 2) Project  $\hat{\mathbf{X}}^{0, TWF}$  onto the space of rank  $r$  matrices to get  $\hat{\mathbf{X}}^0$ .
- 

$r$  length vector, with  $r \ll n$ , and we need to use many more than  $r$  measurements for accurate recovery.

The complete approach, LRPR-init, is summarized in Algorithm 1. Note that this uses the same set of measurements to recover  $\mathbf{U}$  and  $\mathbf{b}_k$ 's. But, as seen from our numerical experiments, it still works well in practice. For our analysis in Sec. III, we assume that a new set of measurements is available for computing  $\hat{\mathbf{g}}_k$ , and thus  $\mathbf{M}_k$  is independent of  $\hat{\mathbf{U}}$ .

Algorithm 1 also estimates the rank  $r$  automatically by looking for the maximum gap between consecutive eigenvalues of  $\mathbf{Y}_U$ . As we explain in Sec. III-C, under a simple assumption on the eigenvalues of  $\bar{\Lambda}$ , this returns the correct rank whp.

### B. Projected-TWF Initialization

Another way to obtain an initial estimate of the low rank matrix  $\mathbf{X}$  would be to project the matrix formed by the TWF initialization for each column  $\mathbf{x}_k$  onto the space of rank  $r$  matrices. This is summarized in Algorithm 2. However, as we show in Sec. V, Tables I and II, this approach performs much worse than LRPR-init. The reason is that it does not simultaneously exploit averaging of the matrices  $\mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'$  over both  $i$  and  $k$ .

### III. SAMPLE COMPLEXITY BOUNDS FOR LRPR-INIT

In this section, we obtain sample complexity bounds for getting a provably accurate initial estimate of both  $\mathbf{U}$  and of the  $\mathbf{x}_k$ 's whp. For simplicity, our results assume iid real Gaussian measurement vectors,  $\mathbf{a}_{i,k}$ . As will be evident from the proofs, the extension to complex Gaussian vectors is straightforward. In Sec. III-A, we provide a guarantee for the case when  $\mathbf{X}$  is a deterministic unknown matrix with known rank  $r$ . These hold whp over measurement vectors  $\mathbf{a}_{i,k}$ . In

TABLE II  
COMPLEX GAUSSIAN MEASUREMENT VECTORS: INITIALIZATION COMPARISONS. LRPR: LRPR-INIT, TWF: TWF-INIT, TWFPROJ: TWFPROJ-INIT

$m/n$	$\Pr(\hat{r} = r)$	LRPR ( $r$ known)	LRPR ( $r$ unknown)	LRPR ( $\hat{r} = 2r$ )	LRPR (noisy meas)	TWFproj	TWF
$q = 100$							
0.10	0.21	1.29	1.32	1.30	1.37	0.97	1.66
0.50	0.41	0.53	0.67	0.76	0.75	0.76	1.55
1.00	0.97	0.27	0.29	0.47	0.36	0.63	1.45
7.00	1.00	0.04	0.04	0.07	0.05	0.13	0.65
$q = 1000$							
0.10	0.99	0.51	0.51	0.84	0.62	0.75	1.67
0.50	1.00	0.15	0.15	0.38	0.19	0.63	1.54
1.00	1.00	0.08	0.08	0.21	0.10	0.55	1.45
7.00	1.00	0.01	0.01	0.03	0.02	0.11	0.65

**Algorithm 3:** LRPR-init-theoretical: initialization with partitioned measurements.

**Known  $r$ :** Set  $\hat{r} = r$ .

**Unknown  $r$ :** Set  $\hat{r} = \arg \max_j (\lambda_j(\mathbf{Y}_U) - \lambda_{j+1}(\mathbf{Y}_U))$  where  $\mathbf{Y}_U$  is defined in (3).

- 1) Compute  $\hat{\mathbf{U}}$  as top  $\hat{r}$  eigenvectors of  $\mathbf{Y}_U$  defined in (3).
- 2) For each  $k = 1, 2, \dots, q$ ,
  - a) compute  $\hat{\mathbf{v}}_k$  as the top eigenvector of

$$\mathbf{Y}_{b,k} := \hat{\mathbf{U}}' \left( \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} \mathbf{y}_{i,k}^{new} \mathbf{a}_i^{new} \mathbf{a}_i^{new'} \right) \hat{\mathbf{U}}, \quad (5)$$

- b) compute  $\hat{\nu}_k = \sqrt{\frac{1}{\tilde{m}} \sum_i \mathbf{y}_{i,k}^{new}}$ ; set  $\hat{\mathbf{b}}_k = \hat{\mathbf{g}}_k = \hat{\mathbf{v}}_k \hat{\nu}_k$ .

Output  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{x}}_k := \hat{\mathbf{U}} \hat{\mathbf{b}}_k$  for all  $k = 1, 2, \dots, q$ .

Sec. III-B, we give results for the case of  $\mathbf{X}$  being random with known rank  $r$ . These hold whp both over matrices  $\mathbf{X}$  generated from the assumed probability distribution and over measurement vectors  $\mathbf{a}_{i,k}$ . In Sec. III-C, we show how we can extend both sets of results to the unknown rank case.

The proof of our results consists of two parts. We first bound the subspace recovery error  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$ . Next, we use this to bound the error in estimating the  $\mathbf{x}_k$ 's,  $\text{dist}(\hat{\mathbf{x}}_k, \mathbf{x}_k)$ . To do this, we show that, if  $\hat{\mathbf{U}}$  is a given matrix with  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$  small enough, and if the measurement vectors and the measurements that are used to estimate the  $\mathbf{b}_k$ 's are independent of  $\hat{\mathbf{U}}$ , then, whp,  $\text{dist}^2(\hat{\mathbf{x}}_k, \mathbf{x}_k)$  can be shown to be bounded by  $c\varepsilon \|\mathbf{x}_k\|^2$  for any chosen  $\varepsilon$ . To ensure that the independence assumption holds, we use a standard trick developed in many earlier works, e.g., [9]. We analyze a ‘‘partitioned’’ version of Algorithm 1. Denote the total number of measurements by  $m_{\text{tot}}$ . We partition these into two disjoint sets of size  $m$  and  $\tilde{m}$  respectively; we use the first set for estimating  $\mathbf{U}$  and the second set for estimating the  $\mathbf{b}_k$ 's. Denote the first set of measurements and measurement vectors by  $\mathbf{y}_{i,k}$  and  $\mathbf{a}_{i,k}$  respectively. Denote the second set by  $\mathbf{y}_{i,k}^{new}$  and  $\mathbf{a}_i^{new}$  respectively. Since the different  $\mathbf{b}_k$ 's are recovered independently, for the second set, we can use the same measurement vectors,  $\mathbf{a}_i^{new}$ , for all the  $\mathbf{x}_k$ 's. Thus, we have the following setting.

*Model 3.1 (Measurement model):* For each  $\mathbf{x}_k$ ,

- we observe  $\mathbf{y}_{i,k} := (\mathbf{a}_{i,k}' \mathbf{x}_k)^2$  where  $\mathbf{a}_{i,k} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$ , for  $i = 1, 2, \dots, m$ ; and
- we observe  $\mathbf{y}_{i,k}^{new} := (\mathbf{a}_i^{new'} \mathbf{x}_k)^2$  where  $\mathbf{a}_i^{new} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$ , for  $i = 1, 2, \dots, \tilde{m}$ .
- The sets of vectors  $\{\mathbf{a}_i^{new}, i = 1, 2, \dots, \tilde{m}\}$  and  $\{\mathbf{a}_{i,k}, i = 1, 2, \dots, m, k = 1, 2, \dots, q\}$  are mutually independent.

Thus we have a total of  $m_{\text{tot}} = m + \tilde{m}$  measurements per vector  $\mathbf{x}_k$ .

With measurements taken as above, we study Algorithm 3.

### A. Main Results for Deterministic $\mathbf{X}$ - Known Rank Case

Let

$$\frac{1}{q} \mathbf{X} \mathbf{X}' \stackrel{\text{EVD}}{=} \mathbf{U} \bar{\mathbf{\Lambda}} \mathbf{U}', \quad (6)$$

and  $\mathbf{X} = \mathbf{U} \mathbf{B}$ . Thus,  $\bar{\mathbf{\Lambda}} = \frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k'$ . Let  $\bar{\lambda}_{\max}$  and  $\bar{\lambda}_{\min}$  denote the maximum and minimum eigenvalues of  $\bar{\mathbf{\Lambda}}$ . Define

$$\rho := \frac{\max_k \|\mathbf{x}_k\|^2}{\frac{1}{q} \sum_k \|\mathbf{x}_k\|^2}, \text{ and } \kappa := \frac{\bar{\lambda}_{\max}}{\bar{\lambda}_{\min}}. \quad (7)$$

Thus,  $\kappa$  is the condition number of  $\mathbf{X} \mathbf{X}'$ . Using  $\rho$ , we can bound  $\max_k \|\mathbf{b}_k\|^2 = \max_k \|\mathbf{x}_k\|^2$  in terms of  $\bar{\lambda}_{\max}$  as<sup>4</sup>

$$\max_k \|\mathbf{b}_k\|^2 = \max_k \|\mathbf{x}_k\|^2 = \rho \sum_{j=1}^r \lambda_j(\bar{\mathbf{\Lambda}}) \leq r \rho \bar{\lambda}_{\max}.$$

We then have the following result.

*Theorem 3.2 (Deterministic  $\mathbf{X}$ ):* Consider an unknown deterministic rank  $r$  matrix  $\mathbf{X}$ . Assume that the measurements of its columns are generated according to Model 3.1. Consider the output of Algorithm 3 (known  $r$  case). Suppose that  $r \leq cn^{1/5}$ . For an  $\varepsilon < 1$ , if

$$\tilde{m} \geq \frac{c\sqrt{n}}{\varepsilon^2}, \quad m \geq \frac{c\kappa^2 \cdot r^4 \log n (\log \tilde{m})^2}{\varepsilon^2},$$

$$mq \geq \frac{c\rho^2 \kappa^2 \cdot nr^4 (\log \tilde{m})^2}{\varepsilon^2},$$

then, with probability at least  $1 - 4 \exp(-cn) - \frac{32q}{n^4}$ ,

$$1) \quad \text{SE}(\hat{\mathbf{U}}, \mathbf{U}) \leq \frac{c\varepsilon}{r \log \tilde{m}};$$

2) for all  $k = 1, 2, \dots, q$ ,  $\text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)^2 \leq c\varepsilon \|\mathbf{x}_k\|^2$ , and so

$$\text{NormErr}(\mathbf{X}, \hat{\mathbf{X}}) := \frac{\sum_{k=1}^q \text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)^2}{\sum_{k=1}^q \|\mathbf{x}_k\|^2} \leq c\varepsilon.$$

Furthermore, if  $q \leq cn^2$ , then the above event holds with probability at least  $1 - c/n^2$ .

*Proof:* The proof is given in Section VI. ■

Notice that our lower bounds depend on  $\kappa^2$  where  $\kappa$  is the condition number of  $\mathbf{X} \mathbf{X}'$ . This is pretty typical, e.g., it is also the case in [25]–[27] and many other works. It may be possible to remove this dependence by borrowing ideas from [29]. A second point to note is that the probability of the good event depends inversely on  $q$ . This dependence comes from needing to ensure that each of the  $q$  vectors  $\mathbf{x}_k$  are accurately recovered. However, the dependence is pretty

<sup>4</sup>This follows because  $\frac{1}{q} \sum_k \|\mathbf{x}_k\|^2 = \text{trace}(\frac{1}{q} \sum_k \mathbf{x}_k \mathbf{x}_k') = \text{trace}(\frac{1}{q} \mathbf{X} \mathbf{X}') = \text{trace}(\bar{\mathbf{\Lambda}}) = \sum_{j=1}^r \lambda_j(\bar{\mathbf{\Lambda}})$ .

weak: when  $q < cn^2$ , the probability can be further lower bounded by  $1 - c/n^2$ .

For the rest of our discussion, assume that  $\varepsilon$ ,  $\kappa$  and  $\rho$  are fixed. We compare our result with that of TWF initialization [11]. TWF has the best sample complexity,  $m \geq cn$ , for single vector PR. Since Theorem 3.2 provides a guarantee for LRPR-init which exploits the low-rank property of  $\mathbf{X}$ , when  $q/r$  is large, its per column sample complexity is significantly smaller than that of TWF. For example, if  $r = c \log n$  and  $q = cr^4(\log n)^3 = (\log n)^7$ , then it needs  $\tilde{m} = c\sqrt{n}$  and  $m = cn/\log n$  and hence  $m_{\text{tot}} := m + \tilde{m} = cn/\log n$ . When  $q$  is larger, for example,  $q = c\sqrt{n}$ , it only requires  $m_{\text{tot}} = c\sqrt{nr^4} \log^3 n = c\sqrt{n}(\log n)^7$ . For  $q \geq cn$ , just  $m_{\text{tot}} = c\sqrt{n}$  measurements suffice. This is also backed up by our numerical experiments; see Tables I and II. Here we used  $n = 100$  and  $r = 2$ . With as few as  $m = 5\sqrt{n}$  measurements, when  $q = 100$ , the LRPR-init normalized error is 0.33. When  $q = 1000$ , this error is only 0.1.

When the goal is to only recover  $\mathbf{U}$  with subspace error at most  $\varepsilon$  (and not the  $\mathbf{x}_k$ 's), the required lower bounds can be relaxed further. In particular, we have the following corollary.

*Corollary 3.3:* In the setting of Theorem 3.2, if  $\tilde{m} = 0$ ,  $m \geq \frac{c\kappa^2 r^2 \log n}{\varepsilon^2}$  and  $mq \geq \frac{c\rho^2 \kappa^2 nr^2}{\varepsilon^2}$ , then with probability at least  $1 - 2 \exp(-cn) - \frac{2q}{n^4}$ ,  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U}) \leq c\varepsilon$ .

Recall that  $\hat{\mathbf{U}}$  is an  $n \times r$  matrix and hence has  $nr$  unknowns. From Corollary 3.3, for a fixed  $\varepsilon$ ,  $\rho$ , and  $\kappa$ , one needs a total of only  $mq = cnr^2$  measurements to recover  $\mathbf{U}$ . When  $r$  is small, e.g.,  $r = c \log n$ , this is only slightly more than the minimum required which would be  $nr$ .

To recover the  $\mathbf{b}_k$ 's, it follows from Theorem 3.2 that we need an extra set of  $\tilde{m} \geq c\sqrt{n}$  measurements.

- The lower bound  $\tilde{m} \geq c\sqrt{n}$  can be replaced by  $\tilde{m} \geq cn^{1/5}$ , or in fact  $cn^{1/d}$  for any integer  $d \geq 2$ , and our result will not change, except for numerical constants.
- We can even replace  $\tilde{m} \geq c\sqrt{n}$  by  $\tilde{m} \geq cr \log^4 r$ , which is much weaker, but then Theorem 3.2 will hold with probability lower bounded by only  $1 - \frac{8q}{\tilde{m}^8} - 2 \exp(-cn) - \frac{8q}{n^4}$ .

In Theorem 3.2, we also need an extra factor of  $(r \log \tilde{m})^2$  in the lower bounds on  $m$  and  $mq$  as compared to Corollary 3.3. This is needed because our algorithm recovers  $\mathbf{g}_k := \hat{\mathbf{U}}' \mathbf{U} \mathbf{b}_k$  and sets  $\hat{\mathbf{x}}_k = \hat{\mathbf{U}} \hat{\mathbf{g}}_k$ . Thus, for it to give an accurate enough estimate of  $\mathbf{x}_k$ , we need to ensure that  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$  is very small so that  $\|\hat{\mathbf{U}}' \mathbf{U}\|$  is close to one. In particular we need  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U}) \leq \varepsilon/r \log \tilde{m}$ . Guaranteeing this requires a larger lower bound on  $mq$  and  $m$  than just ensuring  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U}) \leq \varepsilon$ .

## B. Main Results for Random $\mathbf{X}$ - Known Rank Case

First consider an independent zero mean Gaussian model on the  $\mathbf{b}_k$ 's.

*Model 3.4:* Assume that  $\mathbf{x}_k = \mathbf{U} \mathbf{b}_k$  with  $\mathbf{b}_k \stackrel{\text{indep}}{\sim} \mathcal{N}(0, \mathbf{\Lambda}_k)$ ,  $\mathbf{\Lambda}_k$  diagonal, and  $\mathbf{b}_k$ 's independent of  $\mathbf{U}$ . The matrix  $\mathbf{U}$  can follow any probability distribution. Define

$$\bar{\mathbf{\Lambda}} := \frac{1}{q} \sum_k \mathbb{E}[\mathbf{b}_k \mathbf{b}_k'] = \frac{1}{q} \sum_k \mathbf{\Lambda}_k,$$

let  $\bar{\lambda}_{\min}$  be its minimum eigenvalue,  $\bar{\lambda}_{\max}$  its maximum eigenvalue, and  $\kappa := \frac{\bar{\lambda}_{\max}}{\bar{\lambda}_{\min}}$  its condition number. Assume also that, for all  $k = 1, 2, \dots, q$ ,

$$\lambda_{k, \max} := \lambda_{\max}(\mathbf{\Lambda}_k) \leq c \bar{\lambda}_{\max}.$$

This is ensured, for example, if  $\max_k \lambda_{k, \max} \leq c \min_k \lambda_{k, \max}$ .

With this model, notice that  $\mathbb{E}[\frac{1}{q} \sum_k \mathbf{x}_k \mathbf{x}_k'] \stackrel{\text{EVD}}{=} \mathbf{U} \bar{\mathbf{\Lambda}} \mathbf{U}'$ .

In using Model 3.4, there are two main changes. The first is that we need to apply a law of large numbers result to show that  $\frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k'$  is close to  $\bar{\mathbf{\Lambda}}$  whp. This will hold only when  $q$  is large enough, and, hence, our result will also need another lower bound on  $q$ . The second change is that we need to replace  $r\rho\bar{\lambda}_{\max}$  by  $r(10 \log n)\bar{\lambda}_{\max}$  in the lower bound on  $mq$ . This is the high probability upper bound on  $\|\mathbf{b}_k\|^2$  under Model 3.4. Moreover, because of these two changes, the probability of the good event reduces slightly.

*Theorem 3.5 (Gaussian model):* In the setting of Theorem 3.2, suppose that the  $\mathbf{x}_k$ 's satisfy Model 3.4. For a  $\varepsilon < 1$ , if

$$\begin{aligned} \tilde{m} &\geq \frac{c\sqrt{n}}{\varepsilon^2}, \quad m \geq \frac{c\kappa^2 r^4 (\log \tilde{m})^2}{\varepsilon^2}, \\ mq &\geq \frac{c\kappa^2 nr^4 (\log \tilde{m})^2 (\log n)^2}{\varepsilon^2}, \quad q \geq \frac{c\kappa^2 r^3 (\log n) (\log \tilde{m})^2}{\varepsilon^2}, \end{aligned}$$

then, the conclusions of Theorem 3.2 hold with probability at least  $1 - 2 \exp(-cn) - \frac{36q}{n^4} - \frac{20}{n^2}$ .

*Proof:* See Section VI. ■

Observe that the lower bound on  $q$  in Theorem 3.5 is not very restrictive. From the lower bounds on  $\tilde{m}$  and  $mq$ ,  $q$  anyway needs to be more than  $cr^4(\log n)^4$  in order to get a lower bound on  $m_{\text{tot}}$  that is smaller than  $cn$  (which is the best lower bound achievable by a single vector PR method).

As will be evident from the proof of Theorem 3.5, any random model that ensures that (a)  $\max_k \|\mathbf{b}_k\|^2$  is bounded whp, and (b)  $\frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k'$  is close to  $\bar{\mathbf{\Lambda}}$  whp will suffice. For example, even if the  $\mathbf{b}_k$ 's in Model 3.4 have nonzero and different means, a similar result can be proved. More generally, as we state below, a sub-Gaussian assumption works as well. The independence assumption on  $\mathbf{b}_k$ 's may also be weakened to any other assumption that ensures that (b) holds, however we do not pursue it here.

*Corollary 3.6 (sub-Gaussian model):* Let  $\mathbf{x}_k = \mathbf{U} \mathbf{b}_k$  with  $\mathbf{b}_k$ 's being independent of  $\mathbf{U}$ . Let  $\bar{\mathbf{\Lambda}} := \mathbb{E}[\frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k']$ , let  $\bar{\lambda}_{\max}$  be its maximum eigenvalue and  $\kappa$  its condition number. Assume that the  $\mathbf{b}_k$ 's are independent sub-Gaussian random vectors with sub-Gaussian norm bounded by  $c\sqrt{\bar{\lambda}_{\max}}$ .

With this model replacing Model 3.4 on  $\mathbf{X}$ , Theorem 3.5 holds with probability at least  $1 - 2 \exp(-cn) - \frac{cq}{n^4} - \frac{c}{n^2}$ .

*Proof:* In the proof of Theorem 3.5, only the proofs of Lemmas 6.8, 6.9 change. ■

## C. Main Results - Unknown Rank Case

We now turn to the setting where the rank  $r$  is unknown and show how Theorem 3.2 can be modified for this setting. Other results are modified similarly.

Consider the rank estimation approach given in Algorithm 3. We have the following corollary.

*Corollary 3.7:* Consider Algorithm 3 (unknown  $r$  case). Assume the setting of Theorem 3.2 with  $\varepsilon \leq 0.001$ . If, in addition,  $\kappa \leq 10$  and if  $\bar{\mathbf{\Lambda}}$  is such that  $\lambda_j - \lambda_{j+1} \leq 0.9\bar{\lambda}_{\min}$ , then, with the probability given in Theorem 3.2,

- 1)  $\hat{r} = r$ , and,
- 2) all conclusions of Theorem 3.2 hold.

Another way to correctly estimate  $r$  is via thresholding.

*Corollary 3.8:* Consider Algorithm 3 with rank estimated as follows. Set  $\hat{r}$  as the smallest index  $j$  for which  $\lambda_j(\mathbf{Y}_U) - \lambda_n(\mathbf{Y}_U) \geq 0.25\bar{\lambda}_{\min}$ . Assume the setting of Theorem 3.2 with  $\varepsilon \leq 0.001$ . Then, if  $\kappa < 124$ , then, with the probability given in Theorem 3.2,

- 1)  $\hat{r} = r$ , and,
- 2) all conclusions of Theorem 3.2 hold.

The rank estimation approach of Algorithm 3 does not require knowledge of any model parameters. Hence it is easily applicable for real data (even without training samples being available). However, it works only when consecutive eigenvalues of  $\bar{\mathbf{\Lambda}}$  (consecutive nonzero

---

**Algorithm 4:** LRPR+TWF (TWF initialized using LRPR-init).
 

---

- 1) Initialize  $\hat{\mathbf{X}}^0$  using Algorithm 1
- 2) For each  $t \geq 0$ , do:
  - for each  $k, k = 1, 2, \dots, q$ , update

$$\hat{\mathbf{x}}_k^{t+1} = \hat{\mathbf{x}}_k^t - \frac{\mu}{m} \sum_{i=1}^m \frac{\mathbf{y}_{i,k} - |\mathbf{a}_{i,k}^T \hat{\mathbf{x}}_k^t|^2}{\mathbf{a}_{i,k}^T \hat{\mathbf{x}}_k^t} \mathbf{a}_{i,k} \mathbb{1}_{\mathcal{E}_1^i \cup \mathcal{E}_2^i} \quad (8)$$

where the events  $\mathcal{E}_1^i, \mathcal{E}_2^i$  are defined in [11, eq. 28].

---

singular values of  $\mathbf{X}$ ) are not too far apart. On the other hand, the thresholding based approach of Corollary 3.8 does not require any extra assumptions beyond those in Theorem 3.2 and  $\kappa < 124$ . However it necessitates knowledge of  $\bar{\lambda}_{\min}$ .

#### D. Using Different Measurement Vectors for Each $\mathbf{x}_k$

Our problem setting requires that we use a different set of  $m$  measurement vectors  $\mathbf{a}_{i,k}$  for each column  $\mathbf{x}_k$  in order to estimate  $\mathbf{U}$ . Thus, any application where our algorithms are used needs to apply a total of  $m_{\text{tot}}q$  measurement vectors (often, masks), and also needs to store a total of  $m_{\text{tot}}q$  length- $n$  measurement vectors. However, observe that, because we use different measurement vectors and exploit the low-rank property of the matrix  $\mathbf{X}$ , if  $r \leq c \log n$ , we only need  $m_{\text{tot}}q = cn \text{polylog}(n)$  measurement vectors. Here  $\text{polylog}(n)$  refers to a polynomial in  $(\log n)$ .

On the other hand, if we used the same measurement vectors for each column  $\mathbf{x}_k$ , we would require only  $m$  measurement vectors. But we would need  $m \geq cn$  such vectors. Since  $cn \text{polylog}(n)$  is only a little larger than  $cn$ , our setting is not much more difficult to implement in practice than the same measurement vectors' setting.

An advantage of our setting is as follows. In practice, the region being imaged changes continuously over time. Thus, using our approach, one can just acquire a total of  $m_{\text{mat}}$  independent measurements by imaging the region of interest for a certain period of time. The value of  $q$  (and hence of  $m = m_{\text{mat}}/q$ ) may be decided later depending on the desired tradeoff between temporal resolution and accuracy per pixel<sup>5</sup>. If the changes are gradual, then one can use a smaller value of  $q$ , but gain in accuracy with a larger  $m$ .

#### IV. LOW RANK PR (LRPR) - COMPLETE ALGORITHM

So far we developed an initialization procedure that directly exploited the low-rank property of  $\mathbf{X}$ . Here, we explain three possible ways to develop a complete LRPR algorithm. The first, given next, uses LRPR-init to only speed up TWF.

##### A. LRPR+TWF: Speeding up TWF

Consider the LRPR problem and an application where acquiring measurements is not expensive, but computational power is. In this case, we can use LRPR-init (Algorithm 1) to jointly initialize all columns of the matrix  $\mathbf{X}$ , followed by using the best existing vector PR algorithm such as TWF [11] for recovering each column separately. TWF implements truncated gradient descent for maximizing data likelihood under Poisson measurement noise. We summarize TWF initialized with LRPR-init (LRPR+TWF) in Algorithm 4. LRPR+TWF still requires  $m_{\text{tot}} \geq cn$  measurements per column, but, as we explain next, it needs  $c(\log n - 12 \log \log n)$  fewer iterations to converge than the original TWF (basic TWF, Algorithm 4 initialized using Algorithm 7). To see this, consider the result of

<sup>5</sup>If the measurements are masked-Fourier, then this can be done with the constraint that  $m$  is an integer multiple of  $n$ .

---

**Algorithm 5:** LRPR1: LRPR via projected gradient descent.
 

---

- 1) Initialize  $\hat{\mathbf{X}}^0$  using Algorithm 1 (LRPR-init).
  - 2) For each  $t \geq 0$ , do
    - a) for each  $k, k = 1, 2, \dots, q$ , compute  $\hat{\mathbf{x}}_k^{t+1}$  using (8) defined in Algorithm 4. Call the resulting matrix  $\hat{\mathbf{X}}^{t+1, \text{TWF}}$ ;
    - b) project  $\hat{\mathbf{X}}^{t+1, \text{TWF}}$  on the space of rank  $r$  matrices to get  $\hat{\mathbf{X}}^{t+1}$ .
- 

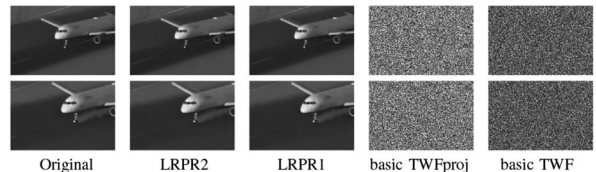


Fig. 1. First column: frame 1 and 104, of the original plane video. Next three columns: frames recovered using the various methods from  $m = 3n$  phaseless masked Fourier (CDP model) measurements.

Theorem 3.2. Another way to interpret this is as follows. Suppose we are given  $m = cn$  and  $\tilde{m} = cn$ . Then, it is clear that this result holds for any  $\varepsilon$  satisfying

$$\varepsilon^2 \geq c \max \left( \frac{1}{\sqrt{n}}, \frac{\kappa^2 r^4 (\log n)^3}{n}, \frac{\rho^2 \kappa^2 r^4 (\log n)^2}{q} \right).$$

If  $q \geq c\sqrt{n}$ , then this means that  $\varepsilon = c \frac{\rho \kappa r^2 (\log n)}{n^{1/4}}$  works. Combining this with [11, Theorem 1], we have the following corollary.

*Corollary 4.1 (LRPR+TWF):* Consider Algorithm 4. If  $m = cn$ ,  $\tilde{m} = cn$ ,  $r \leq c \log n$ , and  $q \geq c\sqrt{n}$  (but  $q \leq cn^2$ ), then, there exists universal constants  $b_1 < 1$  and  $c_1$ , such that, with probability, at least  $1 - c/n^2$ ,

$$\begin{aligned} \text{NormErr}(\mathbf{X}, \hat{\mathbf{X}}^t) &\leq (1 - b_1)^t \text{NormErr}(\mathbf{X}, \hat{\mathbf{X}}^0) \\ &\leq (1 - b_1)^t c_1 \frac{\rho \kappa (\log n)^3}{n^{1/4}}. \end{aligned}$$

From Corollary 4.1, to reduce the final error,  $\text{NormErr}(\mathbf{X}, \hat{\mathbf{X}}^T)$ , below a given tolerance,  $\varepsilon_{\text{fin}}$ , Algorithm 4 needs a total of  $T$  iterations, with  $T$  satisfying

$$T \geq \frac{-\log \varepsilon_{\text{fin}}}{-\log(1 - b_1)} - \frac{0.25(\log n - 12 \log \log n - \log(c_1 \rho \kappa))}{-\log(1 - b_1)}.$$

On the other hand, basic TWF (Algorithm 4 initialized with TWF initialization, Algorithm 7) needs  $T \geq \frac{-\log \varepsilon_{\text{fin}}}{-\log(1 - b_1)}$ . Thus, using LRPR-init to initialize TWF reduces the number of iterations needed for TWF to converge by  $c(\log n - 12 \log \log n)$ . However, LRPR-init is also roughly  $r$  times more expensive than TWF-init. As seen from Fig. 2(a), when  $r$  is small, the reduction in number of iterations still results in lower total time taken by LRPR+TWF as compared to basic TWF.

##### B. LRPR1: Low Rank PR via Projected Gradient Descent

The simplest way to develop a complete algorithm that exploits the low rank property of  $\mathbf{X}$  is to use a projected gradient descent approach to modify TWF. This projects the TWF output at each iteration onto the space of rank  $r$  matrices. We summarize the complete LRPR1 approach (projected-TWF initialized with LRPR-init) in Algorithm 5. When  $m$  is small, this results in significantly improved

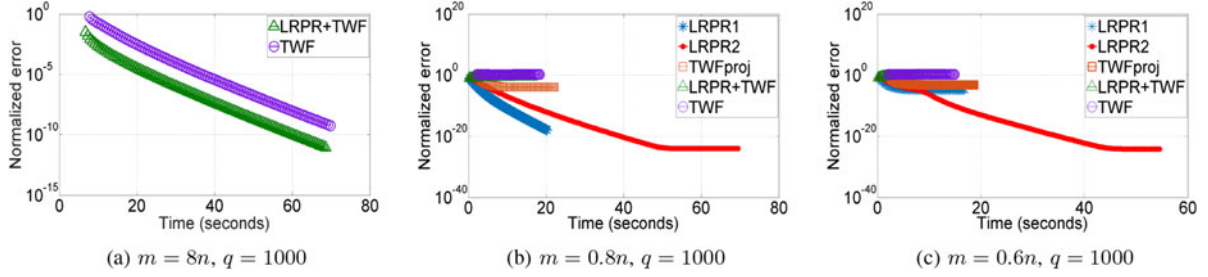


Fig. 2. Plot of reconstruction error, NormErr, as a function of the computation time taken. We obtain each plot as follows. For each iteration  $t = 0, 1, 2, \dots, 100$ , we plot the error at iteration  $t$  against the time taken until iteration  $t$ . This is why, for algorithms with lower per iteration cost, the plot ends earlier, e.g., in (b), LRPR1 took only 20 seconds to complete 100 iterations and hence its plot ends at that time.

performance over TWF because it exploits the low-rank structure of the matrix  $X$  at each step. For an example, see Fig. 2(b).

### C. LRPR2: Low Rank PR via Alternating Minimization

The third and most powerful approach is to modify the entire algorithm to directly exploit the low-rank property of the matrix  $X$ , i.e., to use its decomposition as  $X = UB$ . This idea can be used to modify TWF or AltMinPhase (Gerchberg-Saxton algorithm) or, in fact, many of the other PR methods from literature, e.g., [12], [13]. As noted by an anonymous reviewer, the last two are significantly faster than Gerchberg-Saxton. TWF is truncated gradient descent to minimize the negative data likelihood under a Poisson noise assumption, where as AltMinPhase is an AltMin approach to minimize the squared loss function (data likelihood under iid Gaussian noise). For noise-free measurements, this distinction is immaterial, and all methods apply.

Modifying TWF for the set of variables  $U, B$  needs to be done with care, and needs to include a step that ensures that one of  $\|U\|$  or  $\|B\|$  does not keep increasing. An early attempt along these lines is given in [1].

Modifying the AltMin strategy is simpler and we explain it here. Let  $\mathbf{y}_k := [\mathbf{y}_{1,k}, \mathbf{y}_{2,k}, \dots, \mathbf{y}_{m,k}]'$  and  $\mathbf{A}_k := [\mathbf{a}_{1,k}, \mathbf{a}_{2,k}, \dots, \mathbf{a}_{m,k}]$ . Then  $\sqrt{\mathbf{y}_k} = |\mathbf{A}_k' \mathbf{x}_k|$ . Suppose that the phase information were available, i.e., suppose that we had access to a diagonal matrix  $\hat{C}_k$  so that  $\hat{C}_k \sqrt{\mathbf{y}_k} = \mathbf{A}_k' \mathbf{x}_k$ . Then recovering  $X$  from these linear measurements would be an example of a low-rank matrix recovery problem. This itself can be solved by minimizing over  $U$  and  $B$  alternatively as in [27]. With  $B$  fixed, this is a least squares (LS) recovery problem for  $U$  and vice versa. With estimates of  $U$  and  $B$ , we can estimate the phase matrix  $\hat{C}_k$  as the  $\hat{C}_k = \text{diag}(\text{phase}(\mathbf{A}_k' \hat{U} \hat{\mathbf{b}}_k))$ . The proposed complete algorithm, LRPR2, summarized in Algorithm 6, alternates between these three steps. The per iteration cost of the AltMin approach is larger than that of TWFproj iterates and hence LRPR2 is often slower than LRPR1, e.g., see Fig. 2(b). However, from numerical experiments, LRPR2 needs the smallest value of  $m$  to converge as seen, for example, in Fig. 2(c).

We show the power of both LRPR1 and LRPR2 for recovering a real video from coded diffraction pattern (CDP) measurements in Fig. 1. As can be seen, with as few as  $m = 3n$  CDP measurements, both these methods significantly outperform basic TWFproj (Algorithm 5 initialized using Algorithm 2) and basic TWF (Algorithm 4 initialized using Algorithm 7). This experiment is inspired by an analogous experiment for recovering a regular camera image from CDP measurements reported in [11, Fig. 2]. While this is not a real practical application since the video used is a regular camera video of a moving airplane, this example illustrates two points: (i) many real image sequences are indeed approximately low-rank; and (ii) our algorithm has significant advantage over single vector PR methods for jointly recovering this approxi-

---

#### Algorithm 6: LRPR2: LRPR via Alternating Minimization.

---

- 1) Let  $\hat{U}$  and  $\hat{\mathbf{b}}_k$  denote the output of Algorithm 1.
  - 2) For  $t = 1$  to  $T$ , repeat the following three steps:
    - a)  $\hat{C}_k \leftarrow \text{diag}(\text{phase}(\mathbf{A}_k' \hat{U} \hat{\mathbf{b}}_k))$ , for  $k = 1, 2, \dots, q$
    - b)  $\hat{U} \leftarrow \arg \min_{\hat{U}} \sum_k \|\hat{C}_k \sqrt{\mathbf{y}_k} - \mathbf{A}_k' \hat{U} \hat{\mathbf{b}}_k\|^2$
    - c)  $\hat{\mathbf{b}}_k \leftarrow \arg \min_{\hat{\mathbf{b}}_k} \|\hat{C}_k \sqrt{\mathbf{y}_k} - \mathbf{A}_k' \hat{U} \hat{\mathbf{b}}_k\|^2$ , for  $k = 1, 2, \dots, q$
  - 3) Output  $\hat{U}$  and  $\hat{\mathbf{x}}_k = \hat{U} \hat{\mathbf{b}}_k$  for all  $k = 1, 2, \dots, q$ .
- Steps 2 and 3 involve solving a Least Squares (LS) problem which can be solved in closed form as follows.
- Step 2: Let  $\hat{U}_{vec}$  be the columnwise vectorized version of  $\hat{U}$ . Compute  $\hat{U}_{vec} = (\sum_k \mathbf{M}_k' \mathbf{M}_k)^{-1} \sum_k (\mathbf{M}_k' \hat{C}_k \sqrt{\mathbf{y}_k})$  where  $\mathbf{M}_k := [\mathbf{A}_k'(\hat{\mathbf{b}}_k)_1, \mathbf{A}_k'(\hat{\mathbf{b}}_k)_2, \dots, \mathbf{A}_k'(\hat{\mathbf{b}}_k)_r]$ . Reshape  $\hat{U}_{vec}$  to get  $\hat{U}$ . For large sized problems, conjugate gradient for LS (CGLS) is a faster approach to solve the LS problem since it does not require matrix inversion.
  - Step 3:  $\hat{\mathbf{b}}_k = (\mathbf{M}' \mathbf{M})^{-1} \mathbf{M}' \hat{C}_k \sqrt{\mathbf{y}_k}$  where  $\mathbf{M} = \mathbf{A}_k' \hat{U}$ .
- 

---

#### Algorithm 7: TWF initialization (TWF-init).

---

For each  $k = 1, 2, \dots, q$ , set  $\hat{\mathbf{x}}_k^0$  as the top eigenvector of  $\sum_{i=1}^m \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}' \mathbb{1}_{\{y_{i,k} \leq 9 \frac{\sum_i y_{i,k}}{m}\}}$  scaled by  $\sqrt{\sum_{i=1}^m \mathbf{y}_{i,k} / m}$ .

---

mately low-rank video. For a detailed explanation of this and some more such experiments, please see Supplementary Material and <http://www.ece.iastate.edu/namrata/LRPR/>.

## V. NUMERICAL EXPERIMENTS

We discuss here the results of three sets of experiments. All experiments were done on a single laptop which had these specifications: Intel(R) CPU E3-1240 v5 3.50 GHz, Installed memory: 32 GB, System type is 64 bit.

*Experiment 1:* The first experiment shows the power of the proposed initialization approach, LRPR-init (Algorithm 1), by comparing its initialization error with that of TWF initialization (TWF-init, Algorithm 7) and of TWFproj-init (Algorithm 2). TWF-init does not use knowledge of rank, TWFproj-init assumes  $r$  is known, while LRPR-init estimates the rank automatically as explained earlier. For a fair comparison with TWFproj-init, we also show the error of LRPR-init with  $\hat{r} = r$ . Data was generated as follows. The matrix  $U$  is obtained by orthonormalizing an  $n \times r$  matrix with iid Gaussian entries;  $\mathbf{b}_k$ 's were generated as being iid uniformly distributed between  $-1$  and  $1$ ; and we set  $\mathbf{x}_k = U \mathbf{b}_k$ . Measurements were generating using (1).

We used  $n = 100$ ,  $r = 2$ ,  $\mathbf{a}_{i,k} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$  (Table I) and  $\mathbf{a}_{i,k} \stackrel{\text{iid}}{\sim} \mathcal{CN}(0, \mathbf{I})$  (Table II) and varied  $m$  and  $q$ . Here  $\mathcal{CN}$  refers to a circu-

larly symmetric complex Gaussian distribution. We show 100-time Monte Carlo averaged errors. The averaging is only over the measurement vectors. As can be seen from both tables, LRPR-init significantly outperforms TWF-init and TWFproj-init when  $m$  is small. The reason is that LRPR-init estimates  $\mathbf{X} = \mathbf{U}\mathbf{B}$  by first estimating  $\text{range}(\mathbf{U})$  as the top  $r$  eigenvectors of  $\mathbf{Y}_U$ ; and  $\mathbf{Y}_U$  averages the nearly  $mq$  mutually independent matrices  $\mathbf{M}_{i,k} := \mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'$ . This is possible to do because  $\mathbb{E}[\mathbf{y}_{i,k} \mathbf{a}_{i,k} \mathbf{a}_{i,k}'] = 2\mathbf{U}\mathbf{b}_k \mathbf{b}_k' \mathbf{U}' + \|\mathbf{b}_k\|^2 \mathbf{I}$ . Thus even though the expected values are different for different  $k$ , all have span of top  $r$  eigenvectors equal to  $\text{range}(\mathbf{U})$ . In TWF-init, the averaging over measurements of different columns is not exploited at all, and thus, unsurprisingly, it has the worst performance. In TWFproj-init, averaging over both  $k$  and  $i$  is exploited, but not simultaneously - the first step is TWF-init which averages only over  $i$ . The projection step can be interpreted as averaging over the  $q$  rank-one matrices  $\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k'$  (where  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{\text{TWF,init}}$ ), followed by computing its top  $r$  eigenvectors and projecting  $\hat{\mathbf{X}}^{\text{TWF,init}}$  onto their subspace. In situations where the TWF-init error itself is very large, the second step does not help much.

When the product  $mq$  is large, the rank  $\hat{r}$  is correctly estimated by LRPR-init (Algorithm 1) either always or most of the time. We display a Monte Carlo estimate of the probability of  $\hat{r} = r$  in the 2nd column. In these cases, LRPR with  $\hat{r}$  known versus  $\hat{r}$  estimated both have similar errors (3rd and 4th columns). Inspired by a reviewer's concern, we also evaluate LRPR-init with  $\hat{r}$  deliberately set to a wrong value  $2r$  in the 5th column. As can be seen, the error degradation is gradual even with a wrong rank estimate.

Finally, Table I shows errors of LRPR-*Same* in the last column. This refers to LRPR operating on measurements of the form  $\mathbf{y}_{i,k} := (\mathbf{a}_i' \mathbf{x}_k)^2$ . Because it uses the same  $\mathbf{a}_i$ 's for all columns  $\mathbf{x}_k$ , there are only  $m$  (and not  $mq$ ) mutually independent matrices to average over. Hence its errors are almost as large as those of TWF.

*Experiment 2:* We evaluated LRPR-init (Algorithm 1) in the noisy measurements case. We generated  $\mathbf{y}_{i,k} = (\mathbf{a}_{i,k}' \mathbf{x}_k)^2 + \mathbf{w}_{i,k}$  where  $\mathbf{w}_{i,k}$  were iid uniform between  $-1$  and  $1$  (so that the noise variance was  $0.33$ ). The Monte Carlo estimate of  $\mathbb{E}[\|\mathbf{a}_{i,k}' \mathbf{x}_k\|^2]$  was  $0.67$  leading to a signal-to-noise ratio of  $2$ . The results are shown in the 6th columns of Tables I and II. Observe that the error of LRPR-init even with noisy measurements is smaller than the errors of TWFproj-init and TWF-init with noise-free measurements.

*Experiment 3:* For various values of  $m$ , we evaluate the speed of convergence of the five complete algorithms - basic TWF (Algorithm 4 initialized with Algorithm 7), basic TWFproj (Algorithm 2 initialized with Algorithm 5), LRPR+TWF (Algorithm 4), LRPR1 (Algorithm 5), and LRPR2 (Algorithm 6). We define "converges" as NormErr below  $10^{-10}$ . We generated data as in the noise-free complex Gaussian case described above with  $n = 100$ ,  $r = 2$  and  $q = 1000$ .

In Fig. 2(a), we compare the speed of error decay of TWF when initialized with either TWF-init (TWF) or with the proposed initialization, LRPR-init (LRPR+TWF). We used  $m = 8n$  (large enough  $m$  for TWF iterations to converge). For  $t = 0, 1, 2, \dots, 100$ , we plot the error at the end of iteration  $t$  on the y-axis and the time taken till the end of iteration  $t$  on the x-axis ( $t = 0$  corresponds to initialization). As can be seen, LRPR-init takes longer to finish than TWF-init (the first 'triangle' is to the right of the first circle). However, because LRPR-init results in much lower initialization error, LRPR+TWF needs much fewer iterations to "converge", and, so the total time taken by it to "converge" is also smaller.

If  $m$  is reduced to  $m = 0.8n$  measurements, as can be seen from Fig. 2(b), neither of TWF or LRPR+TWF converge. Basic TWFproj also does not converge and this is because its initialization error is larger (for reasons explained earlier). However, both LRPR1 and LRPR2 converge. It is also apparent that LRPR1 is significantly faster than LRPR2. This is because its per iteration cost is lower.

If  $m$  is reduced further to  $m = 0.6n$  (Fig. 2(c)), then LRPR1 does not converge whereas LRPR2 still does. This is because LRPR2 iterates directly exploit the split-up  $\mathbf{X} = \mathbf{U}\mathbf{B}$  whereas LRPR1 iterates first implement a TWF iteration and then project the resulting matrix onto the space of rank  $r$  matrices.

## VI. PROOFS OF THEOREMS 3.2 AND 3.5

The approach for proving both Theorems 3.2 and 3.5 is similar. In Sec. VI-A, we summarize the two results that will be used in our proof - the Davis-Kahan  $\sin \theta$  theorem [28] and a simple modification of Theorem 5.39 of Vershynin [30]. The  $\sin \theta$  theorem bounds the subspace error between the principal subspaces of a given Hermitian matrix and its perturbed version. The Vershynin result is a probabilistic concentration bound for the empirical covariance matrix of independent sub-Gaussian random vectors. This will be used to bound the terms from the bound obtained by applying the  $\sin \theta$  theorem. In Sec. VI-B, we bound  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$  both under a deterministic and a random assumption on  $\mathbf{X}$ . In Sec. VI-C, we use this to bound  $\text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ , again under both the deterministic and random settings. In Sec. VI-D, we combine these results to prove Theorem 3.2. In Sec. VI-E, we prove Theorem 3.5. Finally, we prove the two corollaries for the unknown rank case - Corollaries 3.7 and 3.8 - in Sec. VI-F.

The derivations in this section use many useful results about sub-Gaussian and sub-exponential r.v.'s and the  $\epsilon$ -net taken from [30]. These are summarized in Appendix A. The lemmas that are not proved here are proved in Appendix B.

### A. Davis-Kahan $\sin \theta$ Theorem and Vershynin's Result

We first state a simple corollary of the Davis-Kahan  $\sin \theta$  theorem [28, Sec. 2] that follows from it using Weyl's inequality (see [31], [32] for a proof).

*Theorem 6.1* (*sin  $\theta$  theorem* [28]): Consider a Hermitian matrix  $\mathbf{D}$  and its perturbed version  $\hat{\mathbf{D}}$ . Define  $\mathbf{H} := \hat{\mathbf{D}} - \mathbf{D}$ . Let  $\mathbf{E}$  be the matrix of top  $r$  eigenvectors of  $\mathbf{D}$ , and let  $\mathbf{F}$  be the matrix of top  $r$  eigenvectors<sup>6</sup> of  $\hat{\mathbf{D}}$ . If  $\lambda_r(\mathbf{D}) - \lambda_{r+1}(\mathbf{D}) - \|\mathbf{H}\| > 0$ , then

$$\text{SE}(\mathbf{F}, \mathbf{E}) := \|(\mathbf{I} - \mathbf{F}\mathbf{F}')\mathbf{E}\| \leq \frac{\|\mathbf{H}\|}{\lambda_r(\mathbf{D}) - \lambda_{r+1}(\mathbf{D}) - \|\mathbf{H}\|}.$$

In Sec. VI-B, we will use the above result with  $\hat{\mathbf{D}} = \mathbf{Y}_U$  and  $\mathbf{D}$  being the expected value of a matrix that is close to it. In Sec. VI-C, we will use it similarly for  $\mathbf{Y}_{b,k}$ .

Theorem 6.2 below is a simple generalization of Theorem 5.39 of [30].

*Theorem 6.2:* Suppose that  $\mathbf{w}_j$ ,  $j = 1, 2, \dots, N$ , are  $n$ -length independent, sub-Gaussian random vectors with sub-Gaussian norms bounded by  $K$ .

- 1) For an  $\epsilon < 1$  and a given vector  $\mathbf{z}$ , with probability (w.p.)  $\geq 1 - 2 \exp(-c\epsilon^2 N)$ ,

$$\left| \mathbf{z}' \left( \frac{1}{N} \sum_j (\mathbf{w}_j \mathbf{w}_j' - \mathbb{E}[\mathbf{w}_j \mathbf{w}_j']) \right) \mathbf{z} \right| \leq 4\epsilon K^2 \|\mathbf{z}\|^2.$$

- 2) For an  $\epsilon < 1$ , w.p.  $\geq 1 - 2 \exp(n \log 9 - c\epsilon^2 N)$ ,

$$\left\| \frac{1}{N} \sum_j (\mathbf{w}_j \mathbf{w}_j' - \mathbb{E}[\mathbf{w}_j \mathbf{w}_j']) \right\| \leq 4\epsilon K^2.$$

*Proof:* The proof follows that of Theorem 5.39 in [30]. It is given in the Supplementary Material.  $\blacksquare$

<sup>6</sup>More generally,  $\mathbf{E}$  and  $\mathbf{F}$  can be any matrices whose columns span the space of top  $r$  eigenvectors of  $\mathbf{D}$  and  $\hat{\mathbf{D}}$  respectively.



1) *Proving Theorems 3.2 and 3.5 simultaneously:* Define the following (trivial) model.

*Model 6.3:* The matrix  $\mathbf{X}$  is a deterministic unknown.

This just makes it simpler to simultaneously obtain subspace error bounds under the assumptions of both Theorems 3.2 and 3.5. Notice that, if we write  $\mathbf{x}_k = \mathbf{U}\mathbf{b}_k$ , then the definitions of  $\bar{\Lambda}$ ,  $\rho$  and  $\kappa$  given in (6) and (7) in Sec. III-A imply that, under Model 6.3,  $\bar{\Lambda} = \frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k'$ ,  $\kappa$  is its condition number, and  $\max_k \|\mathbf{b}_k\|^2 = \max_k \|\mathbf{x}_k\|^2 \leq r\rho\bar{\lambda}_{\max}$ .

## B. Bounding $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$

Recall that  $\hat{\mathbf{U}}$  is the matrix of top  $r$  eigenvectors of  $\mathbf{Y}_U$ . To bound  $\text{SE}(\hat{\mathbf{U}}, \mathbf{U})$  using Theorem 6.1, we define a matrix  $\Sigma^-$  such that (i)  $\mathbf{U}$  is the matrix of its top  $r$  eigenvectors; and (ii) there is a significant nonzero gap between its  $r$ -th and  $(r+1)$ -th eigenvalues. More specifically, we let  $\Sigma^- = c_1 \mathbf{U} \bar{\Lambda} \mathbf{U}' + c_2 \mathbf{I}$  where  $c_1$  and  $c_2$  are positive constants that are defined later. Clearly,  $\lambda_r(\Sigma^-) - \lambda_{r+1}(\Sigma^-) = c_1 \bar{\lambda}_{\min}$ . By Theorem 6.1, if  $\lambda_r(\Sigma^-) - \lambda_{r+1}(\Sigma^-) > \|\mathbf{Y}_U - \Sigma^-\|$ , then,

$$\text{SE}(\hat{\mathbf{U}}, \mathbf{U}) \leq \frac{\|\mathbf{Y}_U - \Sigma^-\|}{c_1 \bar{\lambda}_{\min} - \|\mathbf{Y}_U - \Sigma^-\|}. \quad (9)$$

Thus, all we need now is to specify  $\Sigma^-$  and find a high probability upper bound on  $\|\mathbf{Y}_U - \Sigma^-\|$ .

To this end, as also done in [11, Appendix C], we first lower and upper bound  $\mathbf{Y}_U$  in order to replace  $\frac{1}{m} \sum_i \mathbf{y}_{i,k}$  in its indicator function expression by a constant. Recall that  $\mathbf{Y}_U$  is defined in (3) and that  $\frac{1}{m} \sum_i \mathbf{y}_{i,k} = \mathbf{x}_k' (\frac{1}{m} \sum_i \mathbf{a}_{i,k} \mathbf{a}_{i,k}') \mathbf{x}_k$ . By Fact A.3, item 3, in Appendix A,  $\mathbf{a}_{i,k}$ 's are sub-Gaussian with sub-Gaussian norm bounded by  $c$ . Thus, using the first part of Theorem 6.2, conditioned on  $\mathbf{x}_k$ ,  $|\frac{1}{m} \sum_i \mathbf{y}_{i,k} - \|\mathbf{x}_k\|^2| \leq \epsilon_1 \|\mathbf{x}_k\|^2$  w.p.  $\geq 1 - 2 \exp(-c\epsilon_1^2 m)$ . The constant multiplying  $\epsilon_1$  is moved into the  $c$  in the probability. This bound holds for all  $k = 1, 2, \dots, q$  w.p.  $\geq 1 - 2q \exp(-c\epsilon_1^2 m)$ . This implies that, with the same probability, conditioned on  $\mathbf{X}$ ,  $\mathbf{Y}^- \preceq \mathbf{Y}_U \preceq \mathbf{Y}^+$ , where

$$\mathbf{Y}^- := \frac{1}{mq} \sum_{i=1}^m \sum_{k=1}^q \mathbf{w}_{i,k}^- \mathbf{w}_{i,k}'^-, \quad \mathbf{Y}^+ := \frac{1}{mq} \sum_{i=1}^m \sum_{k=1}^q \mathbf{w}_{i,k}^+ \mathbf{w}_{i,k}'^+,$$

$$\mathbf{w}_{i,k}^- := \left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right) \mathbf{a}_{i,k} \mathbb{1}_{\left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right)^2 \leq 9(1-\epsilon_1)} \|\mathbf{x}_k\|, \quad \text{and}$$

$$\mathbf{w}_{i,k}^+ := \left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right) \mathbf{a}_{i,k} \mathbb{1}_{\left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right)^2 \leq 9(1+\epsilon_1)} \|\mathbf{x}_k\|.$$

Notice that  $\mathbf{w}_{i,k}^+$  is  $\mathbf{w}_{i,k}^-$  with  $9(1-\epsilon_1)$  replaced by  $9(1+\epsilon_1)$  in the indicator function. The following claim is immediate.

*Lemma 6.4:* Conditioned on  $\mathbf{X}$ , w.p.  $\geq 1 - 2q \exp(-c\epsilon_1^2 m)$ ,  $\|\mathbf{Y}_U - \mathbf{Y}^-\| \leq \|\mathbf{Y}^+ - \mathbf{Y}^-\|$ .

Define

$$\Sigma^- := \mathbb{E}[\mathbf{Y}^-] \quad \text{and} \quad \Sigma^+ := \mathbb{E}[\mathbf{Y}^+].$$

We obtain expressions for these in the next lemma.

*Lemma 6.5:* Let  $\xi \sim \mathcal{N}(0, 1)$ . Define

$$\beta_1^- = \beta_1^-(\epsilon_1) := \mathbb{E}[(\xi^4 - \xi^2) \mathbb{1}_{\xi^2 \leq 9(1-\epsilon_1)}],$$

$$\beta_2^- = \beta_2^-(\epsilon_1) := \mathbb{E}[\xi^2 \mathbb{1}_{\xi^2 \leq 9(1-\epsilon_1)}].$$

Under both Models 6.3 and 3.4,

$$\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] = \beta_1^- \mathbf{U} \left( \frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k' \right) \mathbf{U}' + \beta_2^- \left( \frac{1}{q} \sum_k \|\mathbf{b}_k\|^2 \right) \mathbf{I},$$

$$\text{and } \Sigma^- = \beta_1^- \mathbf{U} \bar{\Lambda} \mathbf{U}' + \beta_2^- \text{trace}(\bar{\Lambda}) \mathbf{I}.$$

The matrices  $\mathbb{E}[\mathbf{Y}^+ | \mathbf{X}]$  and  $\Sigma^+$  have similar expressions where we replace  $\beta_i^-$  by  $\beta_i^+$ ,  $i = 1, 2$ . For defining  $\beta_i^+$ , replace  $9(1-\epsilon_1)$  in the indicator function by  $9(1+\epsilon_1)$ .

By the triangle inequality and Lemma 6.4, conditioned on  $\mathbf{X}$ , w.p.  $\geq 1 - 2q \exp(-c\epsilon_1^2 m)$ ,

$$\begin{aligned} \|\mathbf{Y}_U - \Sigma^-\| &\leq \|\mathbf{Y}_U - \mathbf{Y}^-\| + \|\mathbf{Y}^- - \Sigma^-\| \\ &\leq \|\mathbf{Y}^+ - \mathbf{Y}^-\| + \|\mathbf{Y}^- - \Sigma^-\| \\ &\leq 2\|\mathbf{Y}^- - \Sigma^-\| + \|\mathbf{Y}^+ - \Sigma^+\| + \|\Sigma^+ - \Sigma^-\|. \end{aligned}$$

To bound  $\|\mathbf{Y}^- - \Sigma^-\|$ , we first bound  $\|\mathbf{Y}^- - \mathbb{E}[\mathbf{Y}^- | \mathbf{X}]\|$  using the second claim of Theorem 6.2. We bound  $\|\mathbf{Y}^+ - \Sigma^+\|$  similarly.

*Lemma 6.6:* Conditioned on  $\mathbf{X}$ , w.p.  $\geq 1 - 2 \exp(n \log 9 - \epsilon_2^2 m q)$ ,

$$\|\mathbf{Y}^- - \mathbb{E}[\mathbf{Y}^- | \mathbf{X}]\| \leq \epsilon_2 \max_k \|\mathbf{b}_k\|^2.$$

The same bound holds with the same probability for  $\|\mathbf{Y}^+ - \mathbb{E}[\mathbf{Y}^+ | \mathbf{X}]\|$ .

*Remark 6.7:* Since the claim of Lemma 6.6 holds with the same probability lower bound for all  $\mathbf{X}$ , it also holds with the same probability lower bound if we average over  $\mathbf{X}$ . The same is true for Lemma 6.4.

The next lemma bounds  $\max_k \|\mathbf{b}_k\|^2 = \max_k \|\mathbf{x}_k\|^2$ .

*Lemma 6.8:* Under Model 6.3,  $\max_k \|\mathbf{b}_k\|^2 \leq r\rho\bar{\lambda}_{\max}$ . Under Model 3.4, w.p.  $\geq 1 - 2q/n^4$ ,  $\max_k \|\mathbf{b}_k\|^2 \leq r(10 \log n) \bar{\lambda}_{\max}$ .

Under Model 6.3,  $\Sigma^- = \mathbb{E}[\mathbf{Y}^- | \mathbf{X}]$  and so  $\|\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] - \Sigma^-\| = 0$ . Under Model 3.4, we use the second claim of Theorem 6.2, to bound  $\|\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] - \Sigma^-\|$  as follows.

*Lemma 6.9:* Under Model 6.3,  $\|\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] - \Sigma^-\| = 0$ . Under Model 3.4, w.p.  $\geq 1 - 2 \exp(r \log 9 - c\epsilon_3^2 q) - 18 \exp(-c\epsilon_3^2 \frac{q}{r})$ ,

$$\|\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] - \Sigma^-\| \leq \epsilon_3 \bar{\lambda}_{\max}.$$

Combining Lemmas 6.6, 6.8 and 6.9, we can bound  $\|\mathbf{Y}^- - \Sigma^-\|$  under both models. We get the same bound on  $\|\mathbf{Y}^+ - \Sigma^+\|$  as well.

Finally, we bound  $\|\Sigma^+ - \Sigma^-\|$  using the fact that, for all  $\xi > \xi_0$ ,  $\xi^d e^{-\frac{\xi^2}{4}} \leq b$  for any  $d > 1$ .

*Lemma 6.10:* If  $\epsilon_1 \leq 1/9$ , then

$$\begin{aligned} \|\Sigma^+ - \Sigma^-\| &\leq [(\beta_1^+(\epsilon_1) - \beta_1^-(\epsilon_1)) + (\beta_2^+(\epsilon_1) - \beta_2^-(\epsilon_1))] r \bar{\lambda}_{\max} \\ &\leq 30r\epsilon_1 \bar{\lambda}_{\max} \end{aligned}$$

Combining the above bounds, we conclude the following.

*Corollary 6.11:* Let

$$p_{U,1} := 2q \exp(-c\epsilon_1^2 m) + 4 \exp(n \log 9 - c\epsilon_2^2 m q),$$

$$p_{U,2} := p_{U,1} + \frac{4q}{n^4} + 4 \exp(r \log 9 - c\epsilon_3^2 q) + 36 \exp(-c\epsilon_3^2 \frac{q}{r}),$$

$$\epsilon_{U,1} := 3r\epsilon_2\rho + 30r\epsilon_1,$$

$$\epsilon_{U,2} := 3r\epsilon_2(10 \log n) + 30r\epsilon_1 + 3\epsilon_3. \quad (10)$$

Then,

- 1) under Model 6.3, w.p.  $\geq 1 - p_{U,1}$ ,  $\|\mathbf{Y}_U - \Sigma^-\| \leq \epsilon_{U,1} \bar{\lambda}_{\max}$ ; and
- 2) under Model 3.4, w.p.  $\geq 1 - p_{U,2}$ ,  $\|\mathbf{Y}_U - \Sigma^-\| \leq \epsilon_{U,2} \bar{\lambda}_{\max}$ .

Finally, to bound the subspace error of  $\hat{U}$ , we also require a lower bound on  $\beta_1^-$ . This follows easily using the fact that, for all  $\xi > \xi_0$ ,  $\xi^d e^{-\frac{\xi^2}{4}} \leq b$  for any  $d > 1$ .

*Lemma 6.12:* If  $\epsilon_1 \leq 1/9$ , then

$$\beta_1^-(\epsilon_1) = 2 - \mathbb{E}[(\xi^4 - \xi^2)\mathbb{1}_{\xi^2 \geq (9-9\epsilon_1)}] \geq 0.5.$$

Applying the sin  $\theta$  theorem, Theorem 6.1, and the last two claims above, we get the following result.

*Corollary 6.13 (SE( $\hat{U}$ ,  $U$ ) bound):* Let Model 1 be Model 6.3 and Model 2 be Model 3.4. If  $\kappa\epsilon_{U,d} < 1/16$  then, under Model  $d$ , w.p.  $\geq 1 - p_{U,d}$ ,

$$\delta_U := \text{SE}(\hat{U}, U) \leq 2.3\kappa\epsilon_{U,d}.$$

*Proof:* Using Theorem 6.1,

$$\text{SE}(\hat{U}, U) \leq \frac{\|\mathbf{Y}_U - \Sigma^-\|}{\beta_1^- \bar{\lambda}_{\min} - \|\mathbf{Y}_U - \Sigma^-\|}.$$

Since  $\epsilon_1 \leq \epsilon_{U,d} \leq \kappa\epsilon_{U,d} \leq 1/16$ , using Lemma 6.12,  $\beta_1^- \geq 0.5$ . Thus, under Model  $d$ , w.p.  $\geq 1 - p_{U,d}$ ,

$$\delta_U := \text{SE}(\hat{U}, U) \leq \frac{\frac{\kappa\epsilon_{U,d}}{0.5}}{1 - \frac{\kappa\epsilon_{U,d}}{0.5}} < \frac{8}{7} \frac{1}{0.5} \kappa\epsilon_{U,d} < 2.3\kappa\epsilon_{U,d},$$

completing the proof.  $\blacksquare$

### C. Bounding $\text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)$

Recall that  $\delta_U := \text{SE}(\hat{U}, U)$  was bounded above. Here, we bound  $\text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ 's in terms of  $\delta_U$  and other quantities.

*Remark 6.14:* For notational simplicity, we let  $\mathbf{x} = \mathbf{x}_k$ ,  $\mathbf{b} := \mathbf{b}_k$ ,  $\mathbf{y}_i := \mathbf{y}_{i,k}^{new}$ ,  $\mathbf{a}_i := \mathbf{a}_i^{new}$ ,  $\mathbf{Y}_b := \mathbf{Y}_{b,k}$  defined in (5) in Algorithm 3. Since the different  $\mathbf{b}_k$ 's are recovered separately, but using the same technique, this notation does not cause any confusion at most places. Where it does, we clarify.

In this section, we state all results conditioned on  $\mathbf{x}$  and  $\hat{U}$ . Under this conditioning, in all our claims, the probability of the desired event is lower bounded by a value that does not depend on  $\mathbf{x}$  or  $\hat{U}$ . Thus, the same probability lower bound holds even when we average over  $\mathbf{x}$  and  $\hat{U}$  (holds unconditionally).

Notice that  $\mathbf{x} = \mathbf{U}\mathbf{b}$  can be rewritten as

$$\mathbf{x} = \hat{U}\mathbf{g} + \mathbf{e}, \text{ where}$$

$$\mathbf{g} := \hat{U}'\mathbf{x} = \hat{U}'\mathbf{U}\mathbf{b}, \mathbf{e} := (\mathbf{I} - \hat{U}\hat{U}')\mathbf{x}.$$

We can further split  $\mathbf{g}$  as  $\mathbf{g} = \nu\hat{\mathbf{v}}$  where  $\nu = \|\mathbf{g}\|$  and  $\hat{\mathbf{v}} = \mathbf{g}/\nu$ . Recall that we estimate  $\hat{\mathbf{x}}$  as

$$\hat{\mathbf{x}} = \hat{U}\hat{\mathbf{g}} = \hat{U}\hat{\mathbf{v}}\hat{\nu},$$

where  $\hat{\mathbf{v}}$  is the top eigenvector of  $\mathbf{Y}_b$  and  $\hat{\nu} = \sqrt{\frac{1}{\tilde{m}} \sum_i \mathbf{y}_i}$ . The following fact is immediate.

*Fact 6.15:* The vector  $\mathbf{e}$  and the scalar  $\nu := \|\mathbf{g}\|$  satisfy  $\|\mathbf{e}\| \leq \delta_U \|\mathbf{b}\|$ , and  $7(1 - \delta_U)\|\mathbf{b}\| \leq \nu \leq \|\mathbf{b}\|$ .

Using the definition of  $\text{dist}$ , it is easy to see that<sup>8</sup>

$$\text{dist}(\mathbf{x}_1, \mathbf{x}_2) \leq \text{dist}(\mathbf{x}_1, \mathbf{x}_3) + \|\mathbf{x}_3 - \mathbf{x}_2\|. \quad (11)$$

From (11) and Fact 6.15,

- $\text{dist}(\mathbf{x}, \hat{\mathbf{x}}) \leq \delta_U \|\mathbf{b}\| + \text{dist}(\mathbf{g}, \hat{\mathbf{g}})$ ;
- $\text{dist}(\mathbf{g}, \hat{\mathbf{g}}) \leq \|\mathbf{b}\| \text{dist}(\mathbf{v}, \hat{\mathbf{v}}) + |\nu - \hat{\nu}|$ ;

<sup>7</sup>using  $\nu = \|\hat{U}\mathbf{g}\| \geq \|\mathbf{x}\| - \|\mathbf{e}\| \geq (1 - \delta_U)\|\mathbf{b}\|$ ,

<sup>8</sup> $\text{dist}(\mathbf{x}_1, \mathbf{x}_2) = \min_{\phi \in [0, 2\pi]} \|\mathbf{x}_1 - e^{j\phi}(\mathbf{x}_2 + \mathbf{x}_3 - \mathbf{x}_3)\| \leq \min_{\phi \in [0, 2\pi]} (\|\mathbf{x}_1 - e^{j\phi}\mathbf{x}_3\| + \|\mathbf{x}_2 - \mathbf{x}_3\|) = \text{dist}(\mathbf{x}_1, \mathbf{x}_3) + \|\mathbf{x}_2 - \mathbf{x}_3\|.$

- $\text{dist}(\mathbf{v}, \hat{\mathbf{v}})^2 = 2(1 - |\hat{\mathbf{v}}'\mathbf{v}|)$ , and  $\text{SE}(\hat{\mathbf{v}}, \mathbf{v}) = \|\mathbf{v} - \hat{\mathbf{v}}\mathbf{v}'\| \geq 1 - |\hat{\mathbf{v}}'\mathbf{v}|$ . Thus,

$$\text{dist}(\mathbf{x}, \hat{\mathbf{x}}) \leq \delta_U \|\mathbf{b}\| + \sqrt{2\text{SE}(\hat{\mathbf{v}}, \mathbf{v})} \|\mathbf{b}\| + |\nu - \hat{\nu}| \quad (12)$$

We now need to bound  $\text{SE}(\hat{\mathbf{v}}, \mathbf{v})$  and  $|\nu - \hat{\nu}|$ . Using the first claim of Theorem 6.2, we can bound the latter as follows.

*Lemma 6.16 ( $|\hat{\nu} - \nu|$  bound):* Conditioned on  $\mathbf{x}$  and  $\hat{U}$ , w.p.  $\geq 1 - 2\exp(-c\epsilon_4^2\tilde{m})$ ,  $|\hat{\nu} - \nu| \leq \epsilon_4 \|\mathbf{b}\| + \delta_U \|\mathbf{b}\|$ . This holds for all  $\nu_k$ 's,  $k = 1, 2, \dots, q$ , w.p.  $\geq 1 - 2q\exp(-c\epsilon_4^2\tilde{m})$ .

To bound  $\text{SE}(\hat{\mathbf{v}}, \mathbf{v})$ , we use Theorem 6.1 with  $\hat{D} = \mathbf{Y}_b$ . To define a matrix  $\mathbf{D}$ , whose top eigenvector is  $\mathbf{v}$ , let

$$\tilde{\mathbf{a}}_i := \hat{U}'\mathbf{a}_i, \quad (13)$$

$$\mathbf{Y}_g := \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} (\tilde{\mathbf{a}}_i'\mathbf{g})^2 \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'. \quad (14)$$

It is easy to see that, conditioned on  $\hat{U}$ ,  $\tilde{\mathbf{a}}_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$ . By letting  $\mathbf{g}$  be the first column of  $\mathbf{I}$  and using rotation invariance of  $\tilde{\mathbf{a}}_i$  [10, Lemma A.1],

$$\mathbb{E}[\mathbf{Y}_g | \mathbf{x}, \hat{U}] = 2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I}.$$

Clearly, the top eigenvector of this matrix is proportional to  $\mathbf{g}$  and the desired eigen-gap is  $2\|\mathbf{g}\|^2 \geq 2(1 - \delta_U)^2 \|\mathbf{b}\|^2$ . So, we can use  $\mathbf{D} = 2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I}$  for applying Theorem 6.1. It remains to bound  $\|\mathbf{Y}_b - (2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I})\|$ .

Recall that  $\mathbf{Y}_b = \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} (\mathbf{a}_i'\mathbf{x})^2 \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'$ . Thus,

$$\begin{aligned} & \|\mathbf{Y}_b - (2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I})\| \\ & \leq \|\mathbf{Y}_g - (2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I})\| + \|\mathbf{Y}_{e,1}\| + 2\|\mathbf{Y}_{e,2}\|, \end{aligned} \quad (15)$$

where

$$\mathbf{Y}_{e,1} := \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} \mathbf{e}'\mathbf{a}_i \mathbf{a}_i' \mathbf{e} \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i', \text{ and}$$

$$\mathbf{Y}_{e,2} := \frac{1}{\tilde{m}} \sum_{i=1}^{\tilde{m}} \mathbf{e}'\mathbf{a}_i \tilde{\mathbf{a}}_i' \mathbf{g} \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'.$$

To bound  $\|\mathbf{Y}_g - (2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I})\|$ , we use following modification of Theorem 4.1 of [9].

*Lemma 6.17:* [modified version of Theorem 4.1 of [9]] Let  $\mathbf{Y}_g$  be as defined in (14). Recall that  $\mathbf{g} := \hat{U}'\mathbf{U}\mathbf{b}$  is an  $r$ -length vector and, conditioned on  $\hat{U}$ ,  $\tilde{\mathbf{a}}_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$ . If  $\tilde{m} > cr \log^4 r / \epsilon_5^2$ , then, conditioned on  $\mathbf{x}$  and  $\hat{U}$ , w.p.  $\geq 1 - 4/\tilde{m}^8$ ,

$$\begin{aligned} & \|\mathbf{Y}_g - (2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I})\| \\ & \leq \left( \sqrt{\frac{8000r \log^4 \tilde{m}}{\tilde{m}}} + \frac{\sqrt{2}}{\tilde{m}^4} \right) \|\mathbf{g}\|^2 \leq 2\epsilon_5 \|\mathbf{g}\|^2. \end{aligned}$$

This holds for all  $\mathbf{g}_k$ 's, for  $k = 1, 2, \dots, q$ , w.p.  $\geq 1 - 4q/\tilde{m}^8$ .

Next, consider  $\|\mathbf{Y}_{e,1}\|$ . In the argument below, everything is conditioned on  $\hat{U}$  and  $\mathbf{x}$ . Suppose that  $\tilde{m} > r$ . Using Cauchy-Schwartz for matrices, Theorem 8.4, with  $\tilde{\mathbf{X}}_i = \mathbf{e}'\mathbf{a}_i \mathbf{a}_i' \mathbf{e}$  and  $\tilde{\mathbf{Y}}_i = \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'$ , and

simplifying the resulting bounds, we get

$$\begin{aligned} \|\mathbf{Y}_{e,1}\|^2 &\leq \left\| \frac{1}{\tilde{m}} \sum_i e' \mathbf{a}_i \mathbf{a}_i' e e' \mathbf{a}_i \mathbf{a}_i' e \right\| \cdot \left\| \frac{1}{\tilde{m}} \sum_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i' \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i' \right\| \\ &\leq \max_i \mathbf{a}_i' e e' \mathbf{a}_i \left\| \frac{1}{\tilde{m}} \sum_i e' \mathbf{a}_i \mathbf{a}_i' e \right\| \\ &\quad \times \max_i \tilde{\mathbf{a}}_i' \tilde{\mathbf{a}}_i \left\| \frac{1}{\tilde{m}} \sum_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i' \right\| \\ &\leq (80r \log \tilde{m}) \|\mathbf{b}\|^2 (1 + \epsilon_4) \|e\|^2 (20r \log \tilde{m}) (1 + \epsilon_4) \\ &\leq 1600 (r \log \tilde{m})^2 \delta_U^2 (1 + \epsilon_4)^2 \|\mathbf{b}\|^4, \end{aligned}$$

w.p.  $\geq 1 - 4/\tilde{m}^8 - 2 \exp(-c\epsilon_4^2 \tilde{m}) - 2 \exp(r \log 9 - c\epsilon_4^2 \tilde{m})$ .

The first inequality is by Cauchy-Schwartz (Theorem 8.4). The second one pulls out the scalar  $\max_i \mathbf{a}_i' e e' \mathbf{a}_i$  from the first summation and the scalar  $\max_i \tilde{\mathbf{a}}_i' \tilde{\mathbf{a}}_i$  from the second summation. The third inequality relied on the following arguments to bound the four terms. It used (i) Theorem 6.2, part 1 with  $\mathbf{z} = e$  to bound  $\|e'(\frac{1}{\tilde{m}} \sum_i \mathbf{a}_i \mathbf{a}_i')e\|$ ; (ii) Theorem 6.2, part 2 for bounding  $\frac{1}{\tilde{m}} \sum_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'$ ; and (iii) Fact A.3, item 4 in Appendix A to bound  $\max_i \|\tilde{\mathbf{a}}_i\|^2$  by  $20r \log \tilde{m}$  w.p.  $\geq 1 - 2/\tilde{m}^8$  (since  $\tilde{m} > r$ ). This was possible because conditioned on  $\hat{\mathbf{U}}$ ,  $\tilde{\mathbf{a}}_i \sim \mathcal{N}(0, \mathbf{I})$ . (iv) Finally, it used the triangle inequality, Fact A.3, item 4 and the fact that  $\tilde{\mathbf{a}}_i = \hat{\mathbf{U}}' \mathbf{a}_i \sim \mathcal{N}(0, \mathbf{I})$  and  $\mathbf{U}' \mathbf{a}_i \sim \mathcal{N}(0, \mathbf{I})$  to bound

$$\begin{aligned} \max_i |\mathbf{a}_i' e| &\leq \max_i \|\mathbf{a}_i' \mathbf{U}\| \|\mathbf{b}\| + \max_i \|\mathbf{a}_i' \hat{\mathbf{U}}\| \|\hat{\mathbf{U}}' \mathbf{U}\| \|\mathbf{b}\| \\ &\leq 2\sqrt{20r \log \tilde{m}} \|\mathbf{b}\|, \end{aligned}$$

w.p.  $\geq 1 - 2/\tilde{m}^8$ . The fourth inequality in the bound on  $\|\mathbf{Y}_{e,1}\|^2$  follows using  $\|e\| \leq \delta_U \|\mathbf{b}\|$ .

The value of  $\|\mathbf{Y}_{e,2}\|$  can be bounded in a similar fashion:

$$\begin{aligned} \|\mathbf{Y}_{e,2}\|^2 &\leq \frac{1}{\tilde{m}} \sum_i e' \mathbf{a}_i \tilde{\mathbf{a}}_i' \mathbf{g} \mathbf{g}' \tilde{\mathbf{a}}_i \mathbf{a}_i' e \left\| \frac{1}{\tilde{m}} \sum_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i' \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i' \right\| \\ &\leq 400 (r \log \tilde{m})^2 \delta_U^2 (1 + \epsilon_4)^2 \|\mathbf{b}\|^4, \end{aligned}$$

with the same probability. The main difference here is that we bound  $\max_i (\tilde{\mathbf{a}}_i' \mathbf{g})^2$  instead of  $\max_i (\mathbf{a}_i' e)^2$ . To do this, we use  $\max_i (\tilde{\mathbf{a}}_i' \mathbf{g})^2 \leq \max_i \|\tilde{\mathbf{a}}_i\|^2 \|\mathbf{g}\|^2 \leq 20r (\log \tilde{m}) \|\mathbf{b}\|^2$  w.p.  $\geq 1 - 2/\tilde{m}^8$ . Thus, we have the following lemma.

**Lemma 6.18:** Suppose that  $\tilde{m} > r$ . Conditioned on  $\mathbf{x}$  and  $\hat{\mathbf{U}}$ , w.p.  $\geq 1 - 6/\tilde{m}^8 - 4 \exp(-c\epsilon_4^2 \tilde{m}) - 2 \exp(r \log 9 - c\epsilon_4^2 \tilde{m})$ ,

$$\begin{aligned} \|\mathbf{Y}_{e,1}\| &\leq 40 (r \log \tilde{m}) \delta_U (1 + \epsilon_4) \|\mathbf{b}\|^2, \\ \|\mathbf{Y}_{e,2}\| &\leq 20 (r \log \tilde{m}) \delta_U (1 + \epsilon_4) \|\mathbf{b}\|^2. \end{aligned}$$

This holds for all  $\mathbf{g}_k$ 's for  $k = 1, 2, \dots, q$ , w.p.  $\geq 1 - 6q/\tilde{m}^8 - 4q \exp(-c\epsilon_4^2 \tilde{m}) - 2 \exp(r \log 9 - c\epsilon_4^2 \tilde{m})$ .

Let

$$p_g := \frac{10q}{\tilde{m}^8} + 4q \exp(-c\epsilon_4^2 \tilde{m}) + 2 \exp(r \log 9 - c\epsilon_4^2 \tilde{m}). \quad (16)$$

Using (15) and the bounds from Lemmas 6.17 and 6.18, we conclude the following. If  $\tilde{m} \geq \frac{1}{\epsilon_5} cr \log^4 r$ , then, conditioned on  $\mathbf{X}$  and  $\hat{\mathbf{U}}$ , w.p.  $\geq 1 - p_g$ , for all  $k = 1, 2, \dots, q$ ,

$$\|\mathbf{Y}_{b,k} - (2\mathbf{g}_k \mathbf{g}_k' + \|\mathbf{g}_k\|^2 \mathbf{I})\| \leq (2\epsilon_5 + 80r \log \tilde{m} (1 + \epsilon_4) \delta_U) \|\mathbf{b}_k\|^2.$$

Using Theorem 6.1 and Fact 6.15, with the same probability,

$$\begin{aligned} \text{SE}(\hat{\mathbf{v}}_k, \mathbf{v}_k) &\leq \frac{\|\mathbf{Y}_{b,k} - (2\mathbf{g}_k \mathbf{g}_k' + \|\mathbf{g}_k\|^2 \mathbf{I})\|}{2\|\mathbf{g}_k\|^2 - \|\mathbf{Y}_{b,k} - (2\mathbf{g}_k \mathbf{g}_k' + \|\mathbf{g}_k\|^2 \mathbf{I})\|} \\ &\leq \frac{(2\epsilon_5 + 80r \log \tilde{m} (1 + \epsilon_4) \delta_U)}{(2 - 2\delta_U - (2\epsilon_5 + 80r \log \tilde{m} (1 + \epsilon_4) \delta_U))}. \end{aligned}$$

If the numerator is smaller than  $1 - 2\delta_U$ , then

$$\text{SE}(\hat{\mathbf{v}}_k, \mathbf{v}_k) \leq (2\epsilon_5 + 80r \log \tilde{m} (1 + \epsilon_4) \delta_U).$$

As before, we can average over  $\mathbf{X}$  and  $\hat{\mathbf{U}}$  and still get all the events above to hold with the same probability. Using the above bound, (12), and Lemma 6.16, we get the following.

**Corollary 6.19 (dist( $\hat{\mathbf{x}}_k, \mathbf{x}_k$ ) bound):** If  $\tilde{m} \geq \frac{1}{\epsilon_5} cr \log^4 r$ , w.p.  $\geq 1 - p_g - 2q \exp(-c\epsilon_4^2 \tilde{m})$ ,  $\text{dist}(\hat{\mathbf{x}}_k, \mathbf{x}_k)$  is upper bounded by

$$(\epsilon_4 + 2\delta_U + \sqrt{2(2\epsilon_5 + 80r \log \tilde{m} (1 + \epsilon_4) \delta_U)}) \|\mathbf{b}_k\|,$$

for all  $k = 1, 2, \dots, q$ , if  $2\epsilon_5 + r \log \tilde{m} (1 + \epsilon_4) \delta_U < 1 - 2\delta_U$ .

#### D. Proof of Theorem 3.2

Combining Corollaries 6.13 and 6.19, we can conclude the following. If  $\tilde{m} \geq cr \log^4 r / \epsilon_5^2$ , then, w.p.  $\geq 1 - p_{U,1} - p_g - 2q \exp(-c\epsilon_4^2 \tilde{m})$ ,  $\text{dist}(\hat{\mathbf{x}}_k, \mathbf{x}_k)$  is bounded by

$$\left( \epsilon_4 + 4.6\kappa\epsilon_{U,1} + \sqrt{2(2\epsilon_5 + 184r \log \tilde{m} (1 + \epsilon_4) \kappa\epsilon_{U,1})} \right) \|\mathbf{b}_k\|$$

for all  $k = 1, 2, \dots, q$ , as long as  $\kappa\epsilon_{U,1} \leq 1/16$  (this automatically implies  $\epsilon_1 < 1/9$ ) and  $(2\epsilon_5 + 184r \log \tilde{m} (1 + \epsilon_4) \kappa\epsilon_{U,1}) \leq 1 - 4.6\kappa\epsilon_{U,1}$ .

Recall that  $\epsilon_{U,1} = r\rho\epsilon_2 + r\epsilon_1$ . Thus to get  $\text{dist}(\hat{\mathbf{x}}_k, \mathbf{x}_k)$  below  $c\sqrt{\epsilon} \|\mathbf{b}_k\|$ , for an  $\epsilon < 1$ , we set

$$\epsilon_1 = \frac{\epsilon}{150\kappa r^2 \log \tilde{m}}, \quad \epsilon_2 = \frac{\epsilon}{15\kappa \rho^2 \log \tilde{m}}, \quad \epsilon_4 = \sqrt{\epsilon}, \quad \epsilon_5 = \epsilon/5.$$

With these choices notice that, if  $\tilde{m} \geq 3$  and  $r \geq 3$ , then  $\kappa\epsilon_{U,1} = \kappa(3r\rho\epsilon_2 + 30r\epsilon_1) = 2\epsilon/(5r \log \tilde{m}) < 1/16$ ; and  $2\epsilon_5 + r \log \tilde{m} (1 + \epsilon_4) \kappa\epsilon_{U,1} \leq 1 - 2/16$ .

Using the expressions for  $p_{U,1}$  and  $p_g$ , to get the probability of the desired event below  $1 - 2 \exp(-cn) - 32q/n^4$ , we need

$$\begin{aligned} \frac{\epsilon^2}{c\kappa^2 r^4 \log^2 \tilde{m}} m &\geq 4 \log n, \quad \frac{\epsilon^2}{c\kappa^2 \rho^2 r^4 \log^2 \tilde{m}} m q \geq cn, \\ \tilde{m} &\geq c\sqrt{n}, \quad \tilde{m} \geq \frac{cr \log^4 r}{\epsilon^2}, \\ \epsilon \tilde{m} &\geq 4 \log n, \quad \tilde{m} \geq \frac{(4 \log n + (\log 9)r)}{\epsilon}. \end{aligned}$$

We obtained these bounds by taking each probability term and finding a lower bound on  $\tilde{m}$ ,  $m$  or  $m q$  to get it below either  $cq/n^4$  or  $2 \exp(-cn)$ . Theorem 3.2 assumes  $r \leq n^{1/5}$ . Thus,  $r \log^4 r \leq c\sqrt{n}$ ,  $r \leq c\sqrt{n}$ . For large  $n$ ,  $\log n \leq c\sqrt{n}$ .

#### E. Proof of Theorem 3.5

We use the same approach as above. With Model 3.4, both  $\epsilon_U$  and  $p_U$  are larger. We have  $\epsilon_U = \epsilon_{U,2}$  which is equal to  $3\epsilon_3$  plus  $\epsilon_{U,1}$  with  $\rho$  replaced by  $10 \log n$ . Also,  $p_U = p_{U,2} = p_{U,1} + 2q/n^4 + 2 \exp(r \log 9 - c\epsilon_3^2 q) + 18 \exp(-c\epsilon_3^2 q)$ .

Thus, two things change in the conditions required to get the error below  $c\sqrt{\epsilon}$  w.p.  $\geq 1 - 2 \exp(-cn) - 16q/n^4 - 2q/n^4 - 20/n^4$ . First,  $\rho^2$  is replaced by  $(10 \log n)^2$  in the lower bound on  $m q$ . Second, we set  $\epsilon_3 = \frac{\epsilon}{\kappa r \log \tilde{m}}$ . With this, to get  $2 \exp(r \log 9 - c\epsilon_3^2 q) < 2/n^4$

and  $18 \exp(-ce_3^2 \frac{q}{r}) < 18/n^4$ , we need the assumed lower bound on  $q$ .

### F. Proof of Corollary 3.7 and Corollary 3.8

From the proof given in the previous subsections, we need  $\hat{r} = r$  in order to apply the  $\sin \theta$  theorem (Theorem 6.1) to bound  $\text{SE}(\hat{U}, U)$  in Corollary 6.13.

Using Corollary 6.11, w.p.  $\geq 1 - p_{U,1}$ ,

$$\begin{aligned} \|\mathbf{Y}_U - \Sigma^-\| &\leq \epsilon_{U,1} \bar{\lambda}_{\max} = \kappa \epsilon_{U,1} \bar{\lambda}_{\min}, \text{ where} \\ \Sigma^- &:= \beta_1^-(\epsilon_1) \mathbf{U} \bar{\Lambda} \mathbf{U}' + \beta_2^-(\epsilon_1) \text{trace}(\bar{\Lambda}) \mathbf{I}, \end{aligned} \quad (17)$$

and  $\beta_1^-(\epsilon_1), \beta_2^-(\epsilon_1)$  are defined in Lemma 6.5. Let  $\beta_1^- := \beta_1^-(\epsilon_1)$ . Suppose that  $\varepsilon \leq 0.001$ . From Sec. VI-D,  $\epsilon_1 \leq \epsilon_{U,1} \leq \varepsilon \leq 0.001$ . Using  $\kappa \leq 10, 2\kappa \epsilon_{U,1} \leq 0.02$ .

Thus, from (17), Weyl's inequality [33], and  $\bar{\lambda}_j - \bar{\lambda}_{j+1} \leq 0.9\bar{\lambda}_{\min}$ , we conclude the following: for a  $j < r$  and a  $j' > r$ , w.p.  $\geq 1 - p_{U,1}$ ,

$$\begin{aligned} \lambda_r(\mathbf{Y}_U) - \lambda_{r+1}(\mathbf{Y}_U) &\geq (\beta_1^- - 2\kappa \epsilon_{U,1}) \bar{\lambda}_{\min} \\ &\geq (\beta_1^- - 0.02) \bar{\lambda}_{\min}, \\ \lambda_j(\mathbf{Y}_U) - \lambda_{j+1}(\mathbf{Y}_U) &\leq \lambda_j(\Sigma^-) - \lambda_{j+1}(\Sigma^-) + 2\kappa \epsilon_{U,1} \bar{\lambda}_{\min} \\ &= \beta_1^- (\bar{\lambda}_j - \bar{\lambda}_{j+1}) + 2\kappa \epsilon_{U,1} \bar{\lambda}_{\min}, \\ &\leq (0.9\beta_1^- + 0.02) \bar{\lambda}_{\min}, \text{ and} \\ \lambda_{j'}(\mathbf{Y}_U) - \lambda_{j'+1}(\mathbf{Y}_U) &\leq \lambda_{j'}(\Sigma^-) - \lambda_{j'+1}(\Sigma^-) + 2\kappa \epsilon_{U,1} \bar{\lambda}_{\min} \\ &= 0 + 2\kappa \epsilon_{U,1} \bar{\lambda}_{\min} \leq 0.02 \bar{\lambda}_{\min}. \end{aligned}$$

By Lemma 6.12,  $\beta_1^- \geq 0.5$ . Using this, we conclude that

$$\begin{aligned} \lambda_r(\mathbf{Y}_U) - \lambda_{r+1}(\mathbf{Y}_U) &\geq (\beta_1^- - 0.02) \bar{\lambda}_{\min} \\ &> (0.9\beta_1^- + 0.02) \bar{\lambda}_{\min} \\ &\geq \lambda_j(\mathbf{Y}_U) - \lambda_{j+1}(\mathbf{Y}_U), \text{ and} \\ \lambda_r(\mathbf{Y}_U) - \lambda_{r+1}(\mathbf{Y}_U) &\geq (\beta_1^- - 0.02) \bar{\lambda}_{\min} \\ &> 0.02 \bar{\lambda}_{\min} \geq \lambda_{j'}(\mathbf{Y}_U) - \lambda_{j'+1}(\mathbf{Y}_U) \end{aligned}$$

for a  $j < r$  and a  $j' > r$ . The second row used  $0.1\beta_1^- > 0.04$  and the last row used  $\beta_1^- - 0.02 > 0.02$ .

Thus, if  $\kappa \leq 10$ , and  $\bar{\lambda}_j - \bar{\lambda}_{j+1} \leq 0.9\bar{\lambda}_{\min}$ , then, under the assumptions of Theorem 3.2, w.p.  $\geq 1 - p_{U,1}$ ,  $\lambda_j(\mathbf{Y}_U) - \lambda_{j+1}(\mathbf{Y}_U)$  is largest for  $j = r$ , i.e.,  $\hat{r} = r$ . Using this and then proceeding exactly as before, we obtain Corollary 3.7.

To get Corollary 3.8, use (17), Weyl's inequality [33], and  $\kappa \leq 124$ , to argue that, w.p.  $\geq 1 - p_{U,1}$ , for any  $j > r$ ,

$$\begin{aligned} \lambda_r(\mathbf{Y}_U) - \lambda_n(\mathbf{Y}_U) &\geq (\beta_1^- - 0.002\kappa) \bar{\lambda}_{\min} \geq (0.5 - 0.246) \bar{\lambda}_{\min}, \\ \lambda_j(\mathbf{Y}_U) - \lambda_n(\mathbf{Y}_U) &\leq 0.002\kappa \bar{\lambda}_{\min} < 0.25 \bar{\lambda}_{\min}. \end{aligned}$$

Thus, w.p.  $\geq 1 - p_{U,1}$ ,  $j = r$  is the smallest index for which  $\lambda_r(\mathbf{Y}_U) - \lambda_n(\mathbf{Y}_U) \geq 0.25 \bar{\lambda}_{\min}$  and hence the rank estimation approach of Corollary 3.8 returns  $\hat{r} = r$ .

## VII. CONCLUSIONS AND FUTURE WORK

We presented two iterative phase retrieval algorithms – LRPR1 and LRPR2 – for recovering a set of  $q$  unknown vectors lying in a low ( $r$ ) dimensional subspace of  $\mathbb{R}^n$  from their phaseless measurements. Both methods are initialized by a two step spectral initialization procedure, called LRPR-init, that first estimates the subspace

from which all the vectors are generated, and then estimates the projection, of each vector, into the estimated subspace. The rest of LRPR1 involves projected truncated gradient descent. The remainder of LRPR2 involves alternating minimization to update the estimates of  $U, B$ , and the unknown phase of  $(\mathbf{a}_{i,k}, \mathbf{x}_k)$  for each  $i, k$ .

We obtained sample complexity bounds for LRPR-init and argued that, when  $q/r$  is large, these are much smaller than those for TWF or any other single-vector PR method. Via extensive experiments, we also showed that the same is true for both the complete algorithms – LRPR1 and LRPR2. Between the two, LRPR2 has better performance, but also higher per iteration computational cost, than LRPR1.

In future work we will analyze the complete LRPR2 algorithm. This should replace the dependence of sample complexity on  $1/\varepsilon^2$  by a dependence on  $-\log \varepsilon$ .

## APPENDIX A PRELIMINARIES

As explained in [30],  $\epsilon$ -nets are a convenient means to discretize compact metric spaces. The following definition is [30, Definition 5.1] for the unit sphere.

*Definition A.1 ( $\epsilon$ -net and covering number of the unit sphere in  $\mathbb{R}^n$ ):* For an  $\epsilon > 0$ , a subset  $\mathcal{N}_\epsilon$  of the unit sphere in  $\mathbb{R}^n$  is called an  $\epsilon$ -net if, for every vector  $\mathbf{x}$  on the unit sphere, there exists a vector  $\mathbf{y} \in \mathcal{N}_\epsilon$  such that  $\|\mathbf{y} - \mathbf{x}\| \leq \epsilon$ .

The covering number of the unit sphere in  $\mathbb{R}^n$ , is the size of the smallest  $\epsilon$ -net,  $\mathcal{N}_\epsilon$ , on it.

*Fact A.2 (Facts about  $\epsilon$ -nets).*

- 1) By Lemma 5.2 of [30], the covering number of the unit sphere in  $\mathbb{R}^n$  is upper bounded by  $(1 + \frac{2}{\epsilon})^n$ .
- 2) By Lemma 5.4 of [30], for a symmetric matrix,  $\mathbf{W}$ ,  $\|\mathbf{W}\| = \max_{\mathbf{x}: \|\mathbf{x}\|=1} \|\mathbf{x}' \mathbf{W} \mathbf{x}\| \leq \frac{1}{1-2\epsilon} \max_{\mathbf{x} \in \mathcal{N}_\epsilon} \|\mathbf{x}' \mathbf{W} \mathbf{x}\|$ .

*Fact A.3 (Facts about sub-Gaussian random vectors):*

- 1) If  $\mathbf{x}$  is a sub-Gaussian random vector with sub-Gaussian norm  $K$ , then for any vector  $\mathbf{z}$ , (i)  $\mathbf{x}' \mathbf{z}$  is sub-Gaussian with sub-Gaussian norm bounded by  $K\|\mathbf{z}\|$ ; (ii)  $(\mathbf{x}' \mathbf{z})^2$  is sub-exponential with sub-exponential norm bounded by  $2K^2\|\mathbf{z}\|^2$ ; and (iii)  $(\mathbf{x}' \mathbf{z})^2 - \mathbb{E}[(\mathbf{x}' \mathbf{z})^2]$  is centered (zero-mean), sub-exponential with sub-exponential norm bounded by  $4K^2\|\mathbf{z}\|^2$ . This follows from the definition of a sub-Gaussian random vector; Lemma 5.14 and Remark 5.18 of [30].
- 2) By [30, Corollary 5.17], if  $x_i, i = 1, 2, \dots, N$ , are a set of independent, centered, sub-exponential r.v.'s with sub-exponential norm bounded by  $K_\epsilon$ , then, for an  $\varepsilon < 1$ ,

$$\Pr \left( \left| \sum_{i=1}^N x_i \right| > \varepsilon K_\epsilon N \right) \leq 2 \exp(-c\varepsilon^2 N).$$

- 3) If  $\mathbf{x} \sim \mathcal{N}(0, \bar{\Lambda})$  with  $\bar{\Lambda}$  diagonal, then  $\mathbf{x}$  is sub-Gaussian with  $\|\mathbf{x}\|_{\varphi_2} \leq c\sqrt{\bar{\lambda}_{\max}}$ . Moreover, if  $\mathbf{y} = x_1 \mathbf{x}$  where  $x_1$  is a zero mean bounded r.v. with bound  $M$ , then  $\|\mathbf{y}\|_{\varphi_2} \leq cM\sqrt{\bar{\lambda}_{\max}}$ .
- 4) If  $\mathbf{x}_i \sim \mathcal{N}(0, \bar{\Lambda})$ , for  $i = 1, 2, \dots, N$ , are  $n$ -length random vectors and  $\bar{\Lambda}$  is diagonal, then

$$\begin{aligned} \Pr \left( \max_{i=1,2,\dots,N} \|\mathbf{x}_i\|^2 \leq \bar{\lambda}_{\max} \cdot n \cdot 2\nu \right) \\ \geq 1 - 2nN \exp(-\nu), \text{ for } \nu > 1. \end{aligned}$$

This is a direct consequence of eq. 5.5 of [30] which says that if  $x \sim \mathcal{N}(0, 1)$ , then  $\Pr(|x| > t) \leq 2 \exp(-t^2/2)$  for a  $t > 1$ . Using this along with the union bound first for bounding  $\|\mathbf{x}_i\|^2 = \sum_{j=1}^n (x_{ij})^2$  for a given  $i$  and then for bounding its max over  $i$  gives the above result.

5) Using [30, Lemma 5.5], if  $\mathbf{x}_i$ 's are sub-Gaussian random vectors with sub-Gaussian norm bounded by  $K$ , then the following generalization of the above fact holds:  $\Pr(\max_{i=1,2,\dots,N} \|\mathbf{x}_i\|^2 \leq K^2 \cdot n \cdot 2\nu) \geq 1 - CnN \exp(-c\nu)$ .

The following is an easy corollary of Cauchy-Schwartz for sums of products of vectors.

*Theorem A.4 (Cauchy-Schwartz for sums of matrices):* For matrices  $\mathbf{X}_t$  and  $\mathbf{Y}_t$ ,  $\|\frac{1}{\alpha} \sum_{t=1}^{\alpha} \tilde{\mathbf{X}}_t \tilde{\mathbf{Y}}_t'\|^2 \leq \lambda_{\max}(\frac{1}{\alpha} \sum_{t=1}^{\alpha} \tilde{\mathbf{X}}_t \tilde{\mathbf{X}}_t') \lambda_{\max}(\frac{1}{\alpha} \sum_{t=1}^{\alpha} \tilde{\mathbf{Y}}_t \tilde{\mathbf{Y}}_t')$ .

## APPENDIX B

### PROOFS OF LEMMAS FROM SECTION VI

We prove the lemmas that were not proved in Section VI.

Define

$$\mathbf{w}_{i,k} := \left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right) \mathbf{a}_{i,k} \mathbb{1}_{\left( \mathbf{a}_{i,k}' \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right)^2 \leq 9(1-\epsilon_1)} \|\mathbf{x}_k\|.$$

Then  $\mathbf{Y}^- = \frac{1}{mq} \sum_k \sum_i \mathbf{w}_{i,k} \mathbf{w}_{i,k}'$ .

*Proof of Lemma 6.5:* Since  $\mathbf{a}_{i,k}$  is rotationally symmetric, to compute  $\mathbb{E}[\mathbf{w}_{i,k} \mathbf{w}_{i,k}' | \mathbf{X}]$  easily, we can let  $\frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}$  be the first column of the identity matrix. With this,  $\mathbf{w}_{i,k} = (\mathbf{a}_{i,k})_1 \mathbf{a}_{i,k} \mathbb{1}_{(\mathbf{a}_{i,k})_1^2 \leq 9(1-\epsilon_1)} \|\mathbf{x}_k\|$ . Thus, using the argument of [11, Appendix C],  $\mathbb{E}[\mathbf{w}_{i,k} \mathbf{w}_{i,k}' | \mathbf{X}] = (\beta_1^- \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}' + \beta_2^- \mathbf{I}) \|\mathbf{x}_k\|^2 = (\beta_1^- \mathbf{x}_k \mathbf{x}_k' + \beta_2^- \|\mathbf{x}_k\|^2 \mathbf{I})$ . Using this with  $\mathbf{x}_k = \mathbf{U} \mathbf{b}_k$ ,  $\bar{\mathbf{\Lambda}} = \frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k'$ , and  $\frac{1}{q} \sum_k \|\mathbf{b}_k\|^2 = \text{trace}(\bar{\mathbf{\Lambda}})$ , both claims follow under Model 6.3. To get  $\Sigma^-$  under Model 3.4, using linearity of expectation and of trace,  $\text{trace}(\bar{\mathbf{\Lambda}}) = \mathbb{E}[\frac{1}{q} \sum_k \|\mathbf{b}_k\|^2]$ . ■

*Proof of Lemma 6.6:* The proof relies on the following.

- Let  $D = \max_k \|\mathbf{b}_k\|^2 = \max_k \|\mathbf{x}_k\|^2$ .
- We first argue that, conditioned on  $\mathbf{X}$ , each  $\mathbf{w}_{i,k}$  is sub-Gaussian with sub-Gaussian norm bounded by  $c\|\mathbf{x}_k\| \leq c\sqrt{D}$ . To show this easily, we use the strategy of [11, Appendix C]. Since  $\mathbf{a}_{i,k}$  is rotationally symmetric, without loss of generality, suppose that  $\frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}$  is the first column of the identity matrix. Then,  $\mathbf{w}_{i,k} = \mathbf{a}_{i,k} (\mathbf{a}_{i,k})_1 \mathbb{1}_{(\mathbf{a}_{i,k})_1^2 \leq 9(1-\epsilon_1)} \|\mathbf{x}_k\|$ . With this simplification,  $\mathbf{w}_{i,k}$  is of the form  $\mathbf{x} x_1$  where  $x_1$  is a bounded r.v. with bound  $\sqrt{9(1-\epsilon_1)} \|\mathbf{x}_k\|$  and  $\mathbf{x}$  is Gaussian with zero mean and covariance matrix  $\mathbf{I}$ . Thus, using Fact A.3, item 3, it is sub-Gaussian with sub-Gaussian norm bounded by  $c\|\mathbf{x}_k\| \leq c\sqrt{D}$ .
- Conditioned on  $\mathbf{X}$ , all the  $\mathbf{w}_{i,k}$ 's are mutually independent. There are  $N = mq$  of them.

Thus, all the  $mq$   $\mathbf{w}_{i,k}$ 's are mutually independent sub-Gaussian random vectors with sub-Gaussian norm bounded by  $c\sqrt{D}$ . So, we can apply the second claim of Theorem 6.2 with  $\mathbf{w}_j$  replaced by  $\mathbf{w}_{i,k}$  and summed over the  $N = mq$  vectors,  $\mathbf{w}_{i,k}$ , to show that  $\|\mathbf{Y}^- - \mathbb{E}[\mathbf{Y}^- | \mathbf{X}]\| \leq \epsilon_2 D$  w.p.  $\geq 1 - 2 \exp(n \log 9 - c\epsilon_2^2 mq)$ . ■

*Proof of Lemma 6.8:* Let  $D = \max_k \|\mathbf{b}_k\|^2$ . Under Model 6.3,  $D \leq r\rho\bar{\lambda}_{\max}$  by definition. Under Model 3.4, we use Fact A.3, item 4 with  $n \equiv r$ ,  $N \equiv q$ ,  $\nu \equiv 5 \log n$  to get  $D \leq 10 \log n$  w.p.  $\geq 1 - 2rqn^{-5} \geq 1 - 2qn^{-4}$  since  $r \leq n$ . ■

*Proof of Lemma 6.9:*  $\|\mathbb{E}[\mathbf{Y}^- | \mathbf{X}] - \Sigma^-\|$  is bounded by

$$\beta_1^- \left\| \frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k' - \bar{\mathbf{\Lambda}} \right\| + \beta_2^- \left\| \frac{1}{q} \sum_k \|\mathbf{b}_k\|^2 - \text{trace}(\bar{\mathbf{\Lambda}}) \right\|$$

Clearly,  $\beta_1^- \leq 2$  and  $\beta_2^- \leq 1$ . By Fact A.3, item 3,  $\mathbf{b}_k$  is sub-Gaussian with  $\|\mathbf{b}_k\|_{\varphi_2} \leq c\sqrt{\lambda_{k,\max}}$ . By model assumption,  $\lambda_{k,\max} \leq c\bar{\lambda}_{\max}$ . Thus,  $\|\mathbf{b}_k\|_{\varphi_2} \leq c\sqrt{\bar{\lambda}_{\max}}$ . Apply the second claim of Theo-

rem 6.2 with  $N \equiv q$ ,  $n \equiv r$  and  $K \equiv c\sqrt{\bar{\lambda}_{\max}}$  to get  $\|\frac{1}{q} \sum_k \mathbf{b}_k \mathbf{b}_k' - \bar{\mathbf{\Lambda}}\| \leq \epsilon_3 \bar{\lambda}_{\max}$  w.p.  $\geq 1 - 2 \exp(r \log 9 - c\epsilon_3^2 q)$ . The constant next to  $\epsilon_3 \bar{\lambda}_{\max}$  has been absorbed into the  $c$  in the probability.

For the second term, apply Theorem 6.2 with  $N \equiv rq$ ,  $n \equiv 1$  and  $K \equiv c\sqrt{\bar{\lambda}_{\max}}$  to get  $|\frac{1}{rq} \sum_k \|\mathbf{b}_k\|^2 - \frac{\text{trace}(\bar{\mathbf{\Lambda}})}{r}| \leq \epsilon \bar{\lambda}_{\max}$  w.p.  $\geq 1 - 2 \exp(\log 9 - c\epsilon^2 rq)$ . Thus,  $|\frac{1}{q} \sum_k \|\mathbf{b}_k\|^2 - \text{trace}(\bar{\mathbf{\Lambda}})| \leq r\epsilon \bar{\lambda}_{\max}$  with the same probability. Use  $\epsilon = \epsilon_3/r$  to conclude that  $|\frac{1}{q} \sum_k \|\mathbf{b}_k\|^2 - \text{trace}(\bar{\mathbf{\Lambda}})| \leq \epsilon_3 \bar{\lambda}_{\max}$  w.p.  $\geq 1 - 18 \exp(-c\epsilon_3^2 \frac{q}{r})$ . ■

*Proof of Lemma 6.10:* Using  $\xi \exp(-\xi^2/2) < 1$  for all  $\xi^2 > 8$ ,

$$\begin{aligned} \beta_2^+(\epsilon_1) - \beta_2^-(\epsilon_1) &= \mathbb{E}[\xi^2 \mathbb{1}_{(9-9\epsilon_1) \leq \xi^2 \leq (9+9\epsilon_1)}] \\ &= 2 \int_{\sqrt{9-9\epsilon_1}}^{\sqrt{9+9\epsilon_1}} \xi^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} d\xi \\ &\leq 2 \frac{1}{\sqrt{2\pi}} \int_{\sqrt{9-9\epsilon_1}}^{\sqrt{9+9\epsilon_1}} \xi d\xi = \frac{1}{\sqrt{2\pi}} (18\epsilon_1) \end{aligned}$$

Similarly, using  $\xi^3 \exp(-\xi^2/2) < 3.1$  for all  $\xi^2 > 8$ ,

$$\beta_1^+(\epsilon_1) - \beta_1^-(\epsilon_1) \leq 3.1 \cdot \frac{1}{\sqrt{2\pi}} 18\epsilon_1 < 22.4\epsilon_1$$

*Proof of Lemma 6.12:* Using  $(\xi^4 - \xi^2) \exp(-\xi^2/4) < 7.58$  for all  $\xi^2 > 8$ , Fact A.3, item 4 and  $\epsilon_1 \leq 1/9$ ,

$$\begin{aligned} \beta_1^- &= \mathbb{E}[(\xi^4 - \xi^2)] - \mathbb{E}[(\xi^4 - \xi^2) \mathbb{1}_{\xi^2 \geq (9-9\epsilon_1)}] \\ &\geq 2 - 7.58 \cdot 2 \int_{\sqrt{9-9\epsilon_1}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} d\xi \\ &= 2 - 7.58 \cdot \sqrt{2} \Pr(x^2 \geq (9-9\epsilon_1)/2) \\ &\geq 2 - 7.58 \cdot \sqrt{2} \exp(-(9-9\epsilon_1)/4) \geq 0.5 \end{aligned}$$

where  $x \sim \mathcal{N}(0,1)$ . ■

*Proof of Lemma 6.16:* By triangle inequality,  $|\hat{\nu} - \nu| \leq |\hat{\nu} - \|\mathbf{x}\| + \|\mathbf{x}\| - \nu|$ . By Fact 6.15,  $\|\mathbf{x}\| - \nu \leq \delta_U \|\mathbf{b}\|$ . We can bound  $|\hat{\nu} - \|\mathbf{x}\||$  by applying Theorem 6.2, part 1. Recall that  $\hat{\nu}^2 = \frac{1}{m} \sum_i \mathbf{y}_i = \frac{1}{m} \sum_i (\mathbf{a}_i' \mathbf{x})^2$  and so, w.p.  $\geq 1 - 2 \exp(-c\epsilon_4^2 \tilde{m})$ ,

$$|\hat{\nu}^2 - \|\mathbf{x}\|^2| = |\mathbf{x}'(\frac{1}{m} \sum_i \mathbf{a}_i \mathbf{a}_i' - \mathbf{I})\mathbf{x}| \leq \epsilon_4 \|\mathbf{x}\|^2$$

Since  $\epsilon_4 < 1$ , this implies that  $(1 - \epsilon_4)\|\mathbf{x}\| \leq \hat{\nu} \leq (1 + \epsilon_4)\|\mathbf{x}\|$  and so  $|\hat{\nu} - \|\mathbf{x}\|| \leq \epsilon_4 \|\mathbf{x}\|$ . Hence, w.p.  $\geq 1 - 2 \exp(-c\epsilon_4^2 \tilde{m})$ ,  $|\hat{\nu} - \nu| \leq (\epsilon_4 + \delta_U)\|\mathbf{x}\|$ . ■

*Proof of Lemma 6.17:* The proof is a simplified and clearer version of the proof of Theorem 4.1 of [9]. The complete proof is given in the Supplementary Document. We give the key ideas here. Let  $\mathbb{E}[\cdot]$  denote expectation conditioned on  $\mathbf{x}, \hat{\mathbf{U}}$ . Recall that  $\mathbb{E}[\mathbf{Y}_g] = 2\mathbf{g}\mathbf{g}' + \|\mathbf{g}\|^2 \mathbf{I}$  and thus we need to bound  $\|\mathbf{Y}_g - \mathbb{E}[\mathbf{Y}_g]\|$ . (1) Due to rotational symmetry of  $\tilde{\mathbf{a}}_i$ 's we can let  $\mathbf{g}$  be the first column of  $\mathbf{I}$ . This gives a simpler expression for  $\mathbf{Y}_g$ . (2) Truncate  $\tilde{\mathbf{a}}_i$ 's as follows: for each  $j = 1, 2, \dots, r$ , let  $(\tilde{\mathbf{a}}_i^{\text{trunc}})_j = (\tilde{\mathbf{a}}_i)_j$  if  $((\tilde{\mathbf{a}}_i)_j)^2 \leq 20 \log m$ ,  $(\tilde{\mathbf{a}}_i^{\text{trunc}})_j = 0$  otherwise. Define  $\mathbf{Y}_g^{\text{trunc}}$  using  $\tilde{\mathbf{a}}_i^{\text{trunc}}$ 's. (3) Apply Theorem 1.4 of [34] (matrix Bernstein) to bound  $\|\mathbf{Y}_g^{\text{trunc}} - \mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]\|$  w.p.  $\geq 1 - 2/\tilde{m}^8$ . (4) By definition,  $\|\mathbf{Y}_g - \mathbf{Y}_g^{\text{trunc}}\| = 0$  w.p.  $\geq 1 - 2/\tilde{m}^8$ . (5) Finally bound  $\|\mathbb{E}[\mathbf{Y}_g] - \mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]\|$  by  $4/\tilde{m}^{4.5}$ . This is easy because both  $\mathbb{E}[\mathbf{Y}_g]$  and  $\mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]$  are diagonal (the latter is diagonal because the truncation ensures that entries of  $\tilde{\mathbf{a}}_i^{\text{trunc}}$  are also mutually independent and

zero mean). Bound the diagonal entries using the following trick: for  $\xi$  large, e.g. for  $\xi > 10$ ,  $\xi^4 e^{-\xi^2/4} < 1$  and so  $\xi^4 e^{-\xi^2/2} < e^{-\xi^2/4}$ ; similarly,  $\xi_1^2 e^{-\xi_1^2/4} \xi_2^2 e^{-\xi_2^2/4} < 1$  for  $\xi_1, \xi_2$  large. ■

### APPENDIX C SUPPLEMENTARY DOCUMENT

*Proof of Theorem 6.2:* The proof strategy is similar to that of Theorem 5.39 of [30]. By Fact A.3, item 1, for each  $j$ , the r.v.s  $w_j'z$  are sub-Gaussian with sub-Gaussian norm bounded by  $K\|z\|$ ;  $(w_j'z)^2$  are sub-exponential with sub-exponential norm bounded by  $2K^2\|z\|^2$ ; and  $(w_j'z)^2 - \mathbb{E}[(w_j'z)^2] = z'(w_j w_j')z - z'(\mathbb{E}[w_j w_j'])z$  are centered sub-exponential with sub-exponential norm bounded by  $4K^2\|z\|^2$ . Also, for different  $j$ 's, these are clearly mutually independent. Thus, by applying Fact A.3, item 2 (Corollary 5.17 of [30]) we get the first part.

To prove the second part, let  $\mathcal{N}_{1/4}$  denote a  $1/4$ -th net on the unit sphere in  $\mathbb{R}^n$ . Let  $\mathbf{W} := \frac{1}{N} \sum_{j=1}^N (w_j w_j' - \mathbb{E}[w_j w_j'])$ . Then by Fact A.2 (Lemma 5.4 of [30])

$$\|\mathbf{W}\| \leq 2 \max_{z \in \mathcal{N}_{1/4}} |z' \mathbf{W} z| \quad (18)$$

Since  $\mathcal{N}_{1/4}$  is a finite set of vectors, all we need to do now is to bound  $|z' \mathbf{W} z|$  for a given vector  $z$  followed by applying the union bound to bound its maximum over all  $z \in \mathcal{N}_{1/4}$ . The former has already been done in the first part. By Fact A.2 (Lemma 5.2 of [30]), the cardinality of  $\mathcal{N}_{1/4}$  is at most  $9^n$ . Thus, using the first part,  $\Pr(\max_{z \in \mathcal{N}_{1/4}} |z' \mathbf{W} z| \geq \frac{4\epsilon K^2}{2}) \leq 9^n \cdot 2 \exp(-c\frac{\epsilon^2}{4}N) = 2 \exp(n \log 9 - c\epsilon^2 N)$ . By (18), we get the result. ■

*Complete Proof of Lemma 6.17:* The proof is a simplified and clearer version of the proof of Theorem 4.1 of [9]. The few differences are as follows: we truncate differently (in a simpler fashion); and we use different constants to get a higher probability of the good event.

Without loss of generality, assume that  $\mathbf{g}$  is unit norm. Recall that  $\mathbf{Y}_g := \frac{1}{m} \sum_i (\tilde{\mathbf{a}}_i' \mathbf{g})^2 \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'$  with  $\tilde{\mathbf{a}}_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I})$ . Since  $\tilde{\mathbf{a}}_i$ 's are rotationally symmetric, without loss of generality, we can assume that  $\mathbf{g}$  is the first column of identity matrix. Then,

$$\mathbf{Y}_g = \frac{1}{m} \sum_i (\tilde{\mathbf{a}}_i)_1^2 \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i'$$

We use Theorem 1.5 of [34] to prove the result. To use this result, we need to first truncate  $\tilde{\mathbf{a}}_i$ 's. In particular we need to definitely truncate  $(\tilde{\mathbf{a}}_i)_1$  and  $\|\tilde{\mathbf{a}}_i\|^2$ . However truncating all entries of  $\tilde{\mathbf{a}}_i$  results in a simpler proof and hence we use this approach. For  $j = 1, 2, \dots, r$ , define

$$\begin{aligned} (\tilde{\mathbf{a}}_i^{\text{trunc}})_j &= (\tilde{\mathbf{a}}_i)_j \text{ if } ((\tilde{\mathbf{a}}_i)_j)^2 \leq 20 \log m, \\ (\tilde{\mathbf{a}}_i^{\text{trunc}})_j &= 0 \text{ otherwise} \end{aligned}$$

Define

$$\mathbf{Y}_g^{\text{trunc}} := \frac{1}{\tilde{m}} \sum_i (\tilde{\mathbf{a}}_i^{\text{trunc}})_1^2 \tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}$$

By Fact A.3, item 4,  $\tilde{\mathbf{a}}_i^{\text{trunc}} = \tilde{\mathbf{a}}_i$  w.p.  $\geq 1 - 2r/\tilde{m}^{10} \geq 1 - 2/\tilde{m}^9$  since  $\tilde{m} \geq r$ . This holds for all  $i = 1, 2, \dots, \tilde{m}$ , w.p.  $\geq 1 - 2/\tilde{m}^8$ . Thus,  $\|\mathbf{Y}_g - \mathbf{Y}_g^{\text{trunc}}\| = 0$  w.p.  $\geq 1 - 2/\tilde{m}^8$ .

To apply Theorem 1.5 of [34] (matrix Bernstein), define matrix  $X_i := (\tilde{\mathbf{a}}_i^{\text{trunc}})_1^2 \tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}$ . Clearly  $\|X_i\| \leq 20 \log \tilde{m} \cdot (r20 \log \tilde{m}) = 400r \log^2 \tilde{m} := R$ . Also  $\|\mathbb{E}[X_i^2]\| \leq (\tilde{\mathbf{a}}_i^{\text{trunc}})_1^4 \|\tilde{\mathbf{a}}_i^{\text{trunc}}\|^2 \|\mathbb{E}[\tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}]\| \leq 8000r \log^3 \tilde{m} \|\mathbb{E}[\tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}]\| \leq 8000r \log^3 \tilde{m}$ . Here we used  $\|\mathbb{E}[\tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}]\| \leq 1$ . This is

true because: (a) even with the truncation, the different components of  $\tilde{\mathbf{a}}_i^{\text{trunc}}$  remain independent and zero mean and so  $\mathbb{E}[(\tilde{\mathbf{a}}_i^{\text{trunc}})_{j1} (\tilde{\mathbf{a}}_i^{\text{trunc}})_{j2}] = 0$  for  $j1 \neq j2$ ; thus,  $\mathbb{E}[\tilde{\mathbf{a}}_i^{\text{trunc}} \tilde{\mathbf{a}}_i^{\text{trunc}'}]$  is diagonal; and (b) it is easy to see that  $\mathbb{E}[(\tilde{\mathbf{a}}_i^{\text{trunc}})_j^2] \leq \mathbb{E}[(\tilde{\mathbf{a}}_i)_j^2] = 1$ .

Thus, we can apply the theorem with  $R = 400r \log^2 \tilde{m}$  and  $\sigma^2 = \|\sum_i \mathbb{E}[X_i^2]\| \leq 2\tilde{m}R \log \tilde{m}$ . Picking  $\nu = \sqrt{400 \cdot 20 \frac{r \log^4 \tilde{m}}{\tilde{m}}}$ , we get

$$\begin{aligned} \|\mathbf{Y}_g^{\text{trunc}} - \mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]\| &\geq \nu \\ \text{w.p.} &\leq 2r \exp\left(-\frac{\tilde{m}^2 \nu^2}{\tilde{m}R \log \tilde{m} + \tilde{m}R\nu/3}\right) \\ &\leq 2r \exp\left(-\frac{\tilde{m} \nu^2}{2R \log \tilde{m}}\right) \\ &\leq 2 \exp\left(\log r - \frac{\tilde{m}400 \cdot 20r \log^4 \tilde{m}}{\tilde{m}2 \cdot 400r \log^3 \tilde{m}}\right) \\ &= 2 \exp(\log r - 10 \log \tilde{m}) \leq 2/\tilde{m}^9 \end{aligned}$$

This follows since  $\tilde{m} > cr \log^4 r$  (and so  $\nu < 1$  and  $\tilde{m} > r$ )  
Moreover, w.p.  $\geq 1 - 2/\tilde{m}^8$ ,

$$\|\mathbf{Y}_g^{\text{trunc}} - \mathbf{Y}_g\| = 0$$

Thus, w.p.  $\geq 1 - 4/\tilde{m}^8$ ,

$$\|\mathbf{Y}_g - \mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]\| \leq \nu = \sqrt{8000 \frac{r \log^4 \tilde{m}}{\tilde{m}}}$$

Now we only need to bound  $\|\mathbb{E}[\mathbf{Y}_g^{\text{trunc}}] - \mathbb{E}[\mathbf{Y}_g]\|$ . This is easy and uses the following facts. (a) clearly,  $\mathbb{E}[\mathbf{Y}_g]$  is diagonal; (b)  $\mathbb{E}[\mathbf{Y}_g^{\text{trunc}}]$  is also diagonal since with our truncation the different components of  $\tilde{\mathbf{a}}_i^{\text{trunc}}$  remain independent and zero mean; and (c) thus we only need to bound the diagonal entries of  $\mathbb{E}[\mathbf{Y}_g^{\text{trunc}}] - \mathbb{E}[\mathbf{Y}_g]$ . Consider the  $(1, 1)$ -th entry. We bound this by using the fact that for  $\xi > 10$ ,  $\xi^4 e^{-\xi^2/4} < 1$ . If  $\tilde{m} > 3$ ,  $20 \log \tilde{m} > 18 > 10$  and hence this bound holds over the entire region of integration.

$$\begin{aligned} \mathbb{E}[(\tilde{\mathbf{a}}_i)_1^4 - (\tilde{\mathbf{a}}_i^{\text{trunc}})_1^4] &= 2 \int_{\sqrt{20 \log \tilde{m}}}^{\infty} \xi^4 \frac{e^{-\xi^2/2}}{\sqrt{2\pi}} d\xi \\ &\leq 2 \int_{\sqrt{20 \log \tilde{m}}}^{\infty} \frac{e^{-\xi^2/4}}{\sqrt{2\pi} \cdot 2} \sqrt{2} d\xi \\ &= \sqrt{2} \Pr(|x| > \sqrt{9 \log \tilde{m}/2}) \leq \sqrt{2} \frac{2}{\tilde{m}^{4.5}} \end{aligned}$$

if  $\tilde{m} > 3$ . Here  $x$  is a standard Gaussian r.v.. The last inequality used Fact A.3, item 4.

Next consider the  $(j, j)$ -th entry for  $j > 1$ . This can be bounded using a similar trick.

$$\begin{aligned} &\mathbb{E}[(\tilde{\mathbf{a}}_i)_1^2 (\tilde{\mathbf{a}}_i)_j^2 - (\tilde{\mathbf{a}}_i^{\text{trunc}})_1^2 (\tilde{\mathbf{a}}_i^{\text{trunc}})_j^2] \\ &\leq 2 \int_{\xi_1^2 \geq 20 \log \tilde{m}} \xi_1^2 \xi_2^2 \exp(-(\xi_1^2 + \xi_2^2)/2) \frac{1}{2\pi} d\xi_1 d\xi_2 \\ &= 2 \int_{\xi_1^2 \geq 20 \log \tilde{m}} \xi_1^2 \exp(-\xi_1^2/2) \frac{1}{\sqrt{2\pi}} d\xi_1 \\ &\leq 2 \int_{\xi_1^2 \geq 20 \log \tilde{m}} \exp(-\xi_1^2/4) \frac{\sqrt{2}}{\sqrt{2\pi}2} d\xi_1 \leq \frac{2\sqrt{2}}{\tilde{m}^{4.5}} \end{aligned}$$

Thus, w.p.  $\geq 1 - 4/\tilde{m}^8$ ,

$$\|\mathbf{Y}_g - \mathbb{E}[\mathbf{Y}_g]\| \leq \sqrt{8000 \frac{r \log^4 \tilde{m}}{\tilde{m}} + \frac{4}{\tilde{m}^{4.5}}}.$$

If  $\tilde{m} > cr \log^4 r / \epsilon_5^2$ , then the above bound is below  $2\epsilon_5$ . To see this notice that for  $\tilde{m} = cr \log^4 r / \epsilon_5^2$ ,  $\frac{\log^4 \tilde{m}}{\tilde{m}} \leq c\epsilon_5^2/r$ ; and  $(\log^4 \tilde{m})/\tilde{m}$  is an increasing function (for  $\tilde{m}$  large). ■

## APPENDIX D

### EXPERIMENT DETAILS FOR FIG. 1

We used real videos that are approximately low rank and CDP measurements of their images. Each image (arranged as a 1D vector) corresponds to one  $\mathbf{x}_k$  and hence the entire video corresponds to the matrix  $\mathbf{X}$ . We show results on a moving mouse video and on a moving airplane video (shown in Fig. 1). We show two results with “low-rankified videos”<sup>9</sup> and one result with the original airplane video. The airplane images were of size  $n_1 \times n_2$  with  $n_1 = 240$ ,  $n_2 = 320$ ; the mouse images had  $n_1 = 180$ ,  $n_2 = 319$ . Thus,  $n = n_1 n_2 = 76800$  and  $n = 57420$  respectively. Mouse video had  $q = 90$  frames and airplane one had  $q = 105$  frames.

The CDP measurement model can be understood as follows [11]. First, note that it allows  $m$  to only be an integer multiple of  $n$ ; so let  $m = nL$  for an integer  $L$ . Let  $\mathbf{y}_k$  denote the vector containing all measurements of  $\mathbf{x}_k$ . Then  $\mathbf{y}_k = |\mathbf{A}_k' \mathbf{x}_k|^2$  where  $\mathbf{A}_k = [(\mathbf{F}\mathbf{M}_{k,1})', (\mathbf{F}\mathbf{M}_{k,2})', \dots, (\mathbf{F}\mathbf{M}_{k,L})']$ ; each  $\mathbf{M}_{k,l}$  is a diagonal  $n \times n$  mask matrix with diagonal entries chosen uniformly at random from the set  $\{1, -1, \sqrt{-1}, -\sqrt{-1}\}$ , and  $\mathbf{F} = \mathbf{F}_{1D, n_1} \otimes \mathbf{F}_{1D, n_2}$  where  $\mathbf{F}_{1D, n}$  is the  $n$ -point discrete Fourier transform (DFT) matrix and  $\otimes$  denotes Kronecker product. Thus,  $(\mathbf{F}\mathbf{x}_k)$  is the vectorized version of the 2D-DFT of the image corresponding to  $\mathbf{x}_k$ .

In this experiment,  $n$  and  $m$  are very large and hence the memory complexity is very large. Thus, the algorithm cannot be implemented using matrix-vector multiplies. However, since the measurements are masked-Fourier, we can implement its “operator” version as was also done in the TWF code [11]. All matrix-vector multiplies are replaced by “operators” that use 2D fast Fourier transform (2D-FFT) or 2D-inverse-FFT (2D-IFFT) functions, preceded or followed by applying the measurement masks. This is a much faster and memory efficient implementation. Only the masks need to be stored. The EVD in the initialization step is implemented by a block-power method that uses 2D-FFT. The LS step is implemented using the operator-version of conjugate gradient LS (CGLS) taken from <http://web.stanford.edu/group/SOL/software/cgls/>. TWFproj and LRPR1 are implemented similarly.

For this experiment, we used 50 outer loop iterations in each algorithm. Also, 50 iterations of the block-power method were used. For LRPR2, 3 iterations of CGLS were used. We display the NormErr for LRPR2, LRPR1, TWF (TWF-init+TWF) and TWFproj (TWFproj-init+TWFproj) in Table III. Execution times are again shown in parentheses. Three frames of the results corresponding to the last row of this table are shown in Fig. 1 in Sec. II. As can be seen, LRPR2 has the smallest error in all cases. LRPR2 is also the slowest; it is at least  $r$  times slower than TWF. But, TWF and TWFproj do not work when  $m = nL$  is small: notice that the error is much more than one even for  $L = 3$ . LRPR1 is slower than TWF and TWFproj but is much faster than LRPR2. Notice also that, when LRPR2 error is more than 0.1, LRPR1 error is not too much larger than that of LRPR2; in the regime when LRPR2 error is below 0.001, LRPR1 error is 100–1000 times larger. Thus, if just a good approximate solution is needed, LRPR1 offers a better compromise between speed and

<sup>9</sup>The original video data matrix  $\mathbf{X}_{orig}$  was made exactly low rank by projecting it onto the space of rank- $r$  matrices where  $r$  was chosen to retain 90% of the singular values’ energy.

TABLE III  
RESULTS FOR VIDEOS WITH CDP MEASUREMENTS: THE TABLE IS DISPLAYED AS NORMERR (TIME IN SECONDS)

LRPR2	LRPR1	TWF	TWFproj
Mouse, Low-rankified video, $r = 15, L = 1$			
0.52 (981)	0.65 (548)	NaN (54)	NaN (389)
Mouse, Low-rankified video, $r = 15, L = 2$			
8.0e-04 (18776)	0.07 (905)	2.2 (103)	13 (394)
Plane, Low-rankified video, $r = 6, L = 2$			
7.8e-10 (1036)	6.9e-07 (574)	2.2 (137)	14 (327)
Plane, Original video, $r = 6, L = 2$			
0.579 (1042)	0.583 (567)	2.2 (134)	14 (339)
Plane, Original video, $r = 25, L = 3$			
0.146 (13472)	0.150 (3451)	2.0 (207)	14 (950)

We use the symbol “NaN” to indicate that the TWF or TWFproj code failed. This happens for the  $L = 1$  case (since  $m = n$  is too few measurements for TWF).

performance with fewer measurements. If a very accurate solution is needed but speed is not a concern, LRPR2 is a better idea.

## REFERENCES

- [1] S. Nayer, N. Vaswani, and Y. C. Eldar, “Low rank matrix recovery from column-wise phaseless measurements,” in *Proc. IEEE Statist. Signal Process. Workshop*, Jun. 2016, pp. 1–5.
- [2] S. Nayer, N. Vaswani, and Y. C. Eldar, “Low rank phase retrieval,” in *Proc. 42nd IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017.
- [3] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, “Phase retrieval with application to optical imaging,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, May 2015.
- [4] R. W. Gerchberg and W. O. Saxton, “A practical algorithm for the determination of phase from image and diffraction plane pictures,” *Optik*, vol. 35, pp. 237–246, 1972.
- [5] J. R. Fienup, “Phase retrieval algorithms: A comparison,” *Appl. Opt.*, vol. 21, pp. 2758–2769, 1982.
- [6] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, “Phase retrieval via matrix completion,” *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 199–225, 2013.
- [7] E. J. Candes, T. Strohmer, and V. Voroninski, “Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming,” *Commun. Pure Appl. Math.*, vol. 66, pp. 1241–1274, 2013.
- [8] Y. C. Eldar and S. Mendelson, “Phase retrieval: Stability and recovery guarantees,” *Appl. Comput. Harmon. Anal.*, vol. 36, no. 3, pp. 473–94, May 2014.
- [9] P. Netrapalli, P. Jain, and S. Sanghavi, “Phase retrieval using alternating minimization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2796–2804.
- [10] E. J. Candes, X. Li, and M. Soltanolkotabi, “Phase retrieval via wirtinger flow: Theory and algorithms,” *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1985–2007, Apr. 2015.
- [11] Y. Chen and E. Candes, “Solving random quadratic systems of equations is nearly as easy as solving linear systems,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 739–747.
- [12] H. H. Bauschke, P. L. Combettes, and D. R. Luke, “Phase retrieval, error reduction algorithm, and fienuip variants: A view from convex optimization,” *JOSA A*, vol. 19, no. 7, pp. 1334–1345, 2002.
- [13] H. H. Bauschke, P. L. Combettes, and D. R. Luke, “A new generation of iterative transform algorithms for phase contrast tomography,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 4, pp. iv:89–iv:92.
- [14] S. Mukherjee and C. S. Seelamantula, “An iterative algorithm for phase retrieval with sparsity constraints: Application to frequency domain optical coherence tomography,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 553–556.
- [15] I. Waldspurger, A. dAspremont, and S. Mallat, “Phase recovery, maxcut and complex semidefinite programming,” *Math. Program.*, vol. 149, nos. 1/2, pp. 47–81, 2015.
- [16] F. Fogel, I. Waldspurger, and A. dAspremont, “Phase retrieval for imaging problems,” *Math. Program. Comput.*, vol. 8, pp. 1–25, 2013.
- [17] E. Chouzenoux, J. Pesquet, and A. Repetti, “A block coordinate variable metric forward-backward algorithm,” 2014, [Online]. Available: [www.optimization-online.org/DB\\_HTML/2013/12/4178.html](http://www.optimization-online.org/DB_HTML/2013/12/4178.html)

- [18] A. Repetti, E. Chouzenoux, and J. Pesquet, "A nonconvex regularized approach for phase retrieval," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 1753–1757.
- [19] K. Jaganathan, S. Oymak, and B. Hassibi, "Recovery of sparse 1-D signals from the magnitudes of their Fourier transform," in *Proc. IEEE Int. Symp. Inf. Theory*, 2012, pp. 1473–1477.
- [20] Y. Shechtman, A. Beck, and Y. C. Eldar, "Gespars: Efficient phase retrieval of sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, Feb. 2014.
- [21] A. Szameit *et al.*, "Sparsity-based single-shot subwavelength coherent diffractive imaging," *Nature Mater.*, vol. 11, pp. 455–9, Apr. 2012.
- [22] H. Zhang, Y. Zhou, Y. Liang, and Y. Chi, "Reshaped Wirtinger flow and incremental algorithm for solving quadratic system of equations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2622–2630.
- [23] G. Wang, G. B. Giannakis, and J. Chen, "Scalable solvers of random quadratic equations via stochastic truncated amplitude flow," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1961–1974, Apr. 2017.
- [24] M. Butala, R. Frazin, Y. Chen, and F. Kamalabadi, "A Monte Carlo technique for large-scale dynamic tomography," in *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Process.*, 2007, pp. III-1217–III-1220.
- [25] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, "Low-rank solutions of linear matrix equations via procrustes flow," in *Proc. 33rd Int. Conf. Machine Learn.*, 2016, pp. 964–973.
- [26] Q. Zheng and J. Lafferty, "A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 109–117.
- [27] P. Netrapalli, P. Jain, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, 2013, pp. 665–674.
- [28] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. III," *SIAM J. Numer. Anal.*, vol. 7, pp. 1–46, Mar. 1970.
- [29] M. Hardt and M. Wootters, "Fast matrix completion without the condition number," in *Proc. COLT*, 2014, pp. 638–678.
- [30] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in *Compressed Sensing, Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012, ch. 5, pp. 210–268.
- [31] N. Vaswani and H. Guo, "Correlated-PCA: Principal components' analysis when data and noise are correlated," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1768–1776.
- [32] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007–5039, Aug. 2014.
- [33] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [34] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Found. Comput. Math.*, vol. 12, no. 4, pp. 389–434, 2012.



**Namrata Vaswani** received the B.Tech. degree from the Indian Institute of Technology (IIT-Delhi), New Delhi, India, in 1999, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2004, both in electrical engineering. From 2004 to 2005, she was a Postdoc and a Research Scientist at Georgia Tech. Since Fall 2005, she has been with Iowa State University, where she is currently a Professor of electrical and computer engineering. She also holds a courtesy appointment in mathematics.

Her research interests include the intersection of data science and machine learning for high-dimensional problems, signal and information processing, and computer vision and bio-imaging. Her most recent work has been on provably correct and practically useful algorithms for on-line dynamic robust PCA and for recursive sparse recovery. She has served one term as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2009–2012). She received the Harpole-Pentair Assistant Professorship at Iowa State (2008–09), the Iowa State Early Career Engineering Faculty Research Award (2014), and the IEEE Signal Processing Society Best Paper Award (2014) for her Modified-CS paper (IEEE TRANSACTIONS ON SIGNAL PROCESSING 2010).



**Seyedehsara Nayer** is currently working toward the Ph.D. degree at Iowa State University, Ames, IA, USA, under the supervision of Prof. Namrata Vaswani. Her research interests are around various aspects of information science and focuses on high-dimensional structured estimation (e.g., compressed sensing, matrix completion, phase retrieval), convex and nonconvex optimization, and statistical inference and learning. She was previously working at Signal Processing Research Laboratory, Sharif University of Technology.



**Yonina C. Eldar** (S'98–M'02–SM'07–F'12) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering both from Tel-Aviv University, Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2002. From January 2002 to July 2002, she was a Postdoctoral Fellow with the Digital Signal Processing Group, MIT. She is currently a Professor in the Department of Electrical Engineering, Technion—Israel

Institute of Technology, Haifa, Israel, where she holds the Edwards Chair in engineering. She is also a Research Affiliate with the Research Laboratory of Electronics, MIT and was a Visiting Professor at Stanford University, Stanford, CA, USA. She is the author of the book: *Sampling Theory: Beyond Bandlimited Systems* (Cambridge University Press) and the coauthor of the books: *Compressed Sensing* (Cambridge University Press) and *Convex Optimization Methods in Signal Processing and Communications* (Cambridge University Press). Her research interests include the broad areas of statistical signal processing, sampling theory and compressed sensing, optimization methods, and their applications to biology and optics. She was a Horev Fellow of the Leaders in Science and Technology program at the Technion and an Alon Fellow. She is a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is the Editor-in-Chief of *Foundations and Trends in Signal Processing*, serves on several IEEE committees. She served as an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal of Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the *SIAM Journal on Imaging Sciences*. She received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award (2013), the IEEE/AESS Fred Nathanson Memorial Radar Award (2014), and the IEEE Kiyo Tomiyasu Award (2016). She received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technions Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues and was selected as one of the 50 most influential women in Israel.