Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise

Chenlu Qiu, Namrata Vaswani, Brian Lois, and Leslie Hogben

Abstract

This work studies the recursive robust principal components analysis (PCA) problem. Here, "robust" refers to robustness to both independent and correlated sparse outliers. If the outlier is the signal-of-interest, this problem can be interpreted as one of recursively recovering a time sequence of sparse vectors, S_t, in the presence of large but structured noise, L_t. The structure that we assume on L_t is that L_t is dense and lies in a low dimensional subspace that is either fixed or changes "slowly enough". We do not assume any model on the sequence of sparse vectors. Their support sets and their nonzero element values may be either independent or correlated over time (usually in many applications they are correlated). The only thing required is that there be *some* support change every so often. A key application where this problem occurs is in video surveillance where the goal is to separate a slowly changing background (L_t) from moving foreground objects (S_t) on-the-fly. To solve the above problem, we introduce a novel solution called Recursive Projected CS (ReProCS). Under mild assumptions, we show that, with high probability (w.h.p.), ReProCS can exactly recover the support set of S_t at all times; and the reconstruction errors of both S_t and L_t are upper bounded by a time-invariant and small value at all times.

Keywords: robust PCA, sparse recovery, compressive sensing

This version has the following changes: (a) Algorithm 1 and the proof of the main result, Theorem 4.2, have been reorganized to make them shorter and easier to follow; (b) the undersampled case has been removed.

I. Introduction

Principal Components' Analysis (PCA) is a widely used dimension reduction technique that finds a small number of orthogonal basis vectors, called principal components (PCs), along which most of the variability of the dataset lies. It is well known that PCA is sensitive to outliers. Accurately computing the PCs in the presence of outliers is called robust PCA [3], [4], [5], [6]. Often, for time series data, the PCs space changes gradually over time. Updating it on-the-fly (recursively) in the presence of outliers, as more data comes in is referred to as online or recursive robust PCA [7], [8], [9]. "Outlier" is a loosely defined term that refers to any corruption that is not small compared to the true data vector and that occurs occasionally. As suggested in [10], [5], an outlier can be nicely modeled as a sparse vector whose nonzero values can have any magnitude.

A key application where the robust PCA problem occurs is in video analysis where the goal is to separate a slowly changing background from moving foreground objects [4], [5]. If we stack each frame as a column vector, the background is well modeled as lying in a low dimensional subspace that may gradually change over time, while the moving foreground objects constitute the sparse outliers [10], [5] which change in a correlated fashion over time. Other applications include sensor networks based detection and tracking of abnormal events such as forest fires or oil spills; or online detection of brain activation patterns from functional MRI (fMRI) sequences (the "active" part of the brain can be interpreted as a correlated sparse outlier). Clearly, in all these applications, an online solution is desirable. In this work, we focus on this case, i.e. on *recursive robust PCA that is robust to both independent and correlated sparse outliers*.

The moving objects or the active regions of the brain or the oil spill region may be "outliers" for the PCA problem, but in most cases, these are actually the signals-of-interest whereas the background image is the noise. Also, all the above signals-of-interest are sparse vectors that change in a correlated fashion over time. Thus, this problem can also be interpreted as one of

C. Qiu, N. Vaswani are with the ECE dept at Iowa State University. B. Lois, L. Hogben are with the Mathematics dept. Email: {chenlu,namrata}@iastate.edu.
Some ideas of this paper were presented at Allerton 2010 and 2011 [1], [2].

recursively recovering a time sequence of correlated sparse signals, S_t, from measurements $M_t := S_t + L_t$ that are corrupted by (potentially) large magnitude but dense and structured noise, L_t. The structure that we require is that L_t be dense and lie in a low dimensional subspace that is either fixed or changes "slowly enough" in the sense quantified in Sec III-B. For the fMRI application, one would also like to study the following more general problem: recover S_t from $M_t := AS_t + \tilde{L}_t$ where $\tilde{L}_t = AL_t$ and A is a matrix with more columns than rows (fat matrix). Since L_t is low-dimensional, so is $\tilde{L}_t$. This is done in ongoing work [11].

A. Related Work

There has been a large amount of work on robust PCA, e.g. [4], [5], [6], [3], [12], [13], [14], and recursive robust PCA e.g. [7], [8], [9]. In most of these works, either the locations of the missing/corruped data points are assumed known [7] (not a practical assumption); or they first detect the corrupted data points and then replace their values using nearby values [8]; or weight each data point in proportion to its reliability (thus soft-detecting and down-weighting the likely outliers) [4], [9]; or just remove the entire outlier vector [13], [14]. Detecting or soft-detecting outliers (S_t) as in [8], [4], [9] is easy when the outlier magnitude is large, but not otherwise. When the signal of interest is S_t, the most difficult situation is when nonzero elements of S_t have small magnitude compared to those of L_t and in this case, these approaches do not work.

In a series of recent works [5], [6], a new and elegant solution to robust PCA called Principal Components' Pursuit (PCP) has been proposed, that does not require a two step outlier location detection/correction process and also does not throw out the entire vector. It redefines batch robust PCA as a problem of separating a low rank matrix, $\mathcal{L}_t := [L_1, \dots, L_t]$, from a sparse matrix, $\mathcal{S}_t := [S_1, \dots, S_t]$, using the measurement matrix, $\mathcal{M}_t := [M_1, \dots, M_t] = \mathcal{L}_t + \mathcal{S}_t$. Other recent works that also study batch algorithms for recovering a sparse $\mathcal{S}_t$ and a low-rank $\mathcal{L}_t$ from $\mathcal{M}_t := \mathcal{L}_t + \mathcal{S}_t$ or from undersampled measurements include [15], [16], [17], [18], [19], [20], [21], [22], [23], [24].

It was shown in [5] that one can recover $\mathcal{L}_t$ and $\mathcal{S}_t$ exactly by solving

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_* + \lambda \|\mathcal{S}\|_1 \text{ subject to } \mathcal{L} + \mathcal{S} = \mathcal{M}_t \tag{1}$$

provided that (a) $\mathcal{L}_t$ is dense (its left and right singular vectors satisfy certain conditions); (b) any element of the matrix $\mathcal{S}_t$ is nonzero w.p. ϱ, and zero w.p. $1 - \varrho$, independent of all others (in particular, this means that the support sets of the different S_t's are independent over time); and (c) the rank of $\mathcal{L}_t$ and the support size of $\mathcal{S}_t$ are small enough. Here $\|A\|_*$ is the nuclear norm of A (sum of singular values of A) while $\|A\|_1$ is the ℓ_1 norm of A seen as a long vector.

As explained earlier, a key application where the robust PCA problem occurs is in video layering where the foreground sequence is sparse while the background sequence is approximately low dimensional. In this case, it is fair to assume that the background changes are dense (i.e. $\mathcal{L}_t$ is dense). However, the assumption that the foreground support is independent over time is not a valid one. Foreground objects typically move in a correlated fashion, and may even not move for a few frames. This often results in $\mathcal{S}_t$ being sparse as well as low rank.

B. Motivation for Proposed Algorithm

The question then is, what can we do if $\mathcal{L}_t$ is low rank and dense, but $\mathcal{S}_t$ may be both sparse and low rank? Clearly in this case, without any extra information, in general, it is not possible to separate $\mathcal{S}_t$ and $\mathcal{L}_t$. Suppose that an initial short sequence of L_t's is available. For example, in the video application, it is often realistic to assume that an initial background-only training sequence is available. Can we use this to do anything better?

One possible solution is as follows. We can compute the matrix containing the left singular vectors of the initial short training sequence, $\hat{P}_0$. This can be used to modify PCP as follows. We solve

$$\min_{\mathcal{S}} \|\mathcal{S}\|_1, \text{ subject to } \|(I - \hat{P}_0 \hat{P}_0')(\mathcal{M}_t - \mathcal{S})\|_F \leq \epsilon, \tag{2}$$

where $\|.\|_F$ is the Frobenius norm. This then becomes the standard ℓ_1 minimization solution for a batch sparse recovery problem in noise. As we show later in Lemma 3.2, denseness of $\hat{P}_0$ (ensured by denseness of $\mathcal{L}_t$), ensures that the restricted isometry constant for $(I - \hat{P}_0\hat{P}_0')$ is small and hence $\mathcal{S}_t$ can be recovered accurately by solving (2) as long as the "noise" it sees is small. Here the "noise" is $(I - \hat{P}_0\hat{P}_0')\mathcal{L}_t$. This is small only if span$(\hat{P}_0)$ approximately contains span$(\mathcal{L}_t)$ (in the sense defined in Definition 2.2), i.e. the subspace spanned by the future background frames is an approximate subset of that of the initial training dataset. This is unreasonable to expect in a long sequence. Even though the change of subspace from one time instant to the next is usually "slow" (one way to quantify this is given in Sec III-B), the net change over a long sequence can be significant.

To address this issue, we can replace the above by the following. Using $\hat{P}_0$, we solve (2) for the next set of α frames and use the resulting estimates, $\hat{S}_t$, to get $\hat{L}_t = M_t - \hat{S}_t$. If the subspace changes during this period, because of the slow subspace change assumption (of Sec III-B), the projection of L_t along the newly added directions will be small for the first α frames, thus ensuring that the $\hat{S}_t$'s, and hence the $\hat{L}_t$'s, are accurately estimated for this period. The estimated $\hat{L}_t$'s can be used in a recursive PCA or a projection PCA algorithm to get an updated estimate of span$(\mathcal{L}_t)$ which now includes the span of the newly added directions. Using the new subspace basis estimate, $\hat{P}$, for solving (2) for the next set of α frames, will reduce the "noise" seen by it. Thus a more accurate set of $\hat{S}_t$'s, and hence $\hat{L}_t$'s, can be computed for this period. This, in turn, will help get a more accurate estimate of span$(\mathcal{L}_t)$. For simplicity, and to get a fully recursive solution, one can replace (2) by solving the ℓ_1 problem at each time separately. This is the key idea of our proposed algorithm which we call Recursive Projected Compressive Sensing or ReProCS.

In an earlier conference paper [1], we first introduced the ReProCS idea. It used an algorithm motivated by recursive PCA [7] for updating the subspace estimates on-the-fly. Recursive PCA is a fast algorithm for solving the PCA problem when data comes in sequentially. However, as we explain in Appendix F, it is difficult to obtain performance guarantees for PCA. In this work, we instead use a modification called projection PCA, which can be analyzed more easily. The performance of both approaches in simulation experiments is similar.

C. Our Contributions

If the noise, L_t, lies in a "slowly changing" low dimensional subspace as defined in Sec III-B, under mild assumptions, we show that, w.h.p, ReProCS can exactly recover the support of S_t at all times; and the reconstruction errors of both S_t and L_t are upper bounded by a time invariant and small value at all times. Unlike [6], our result does not assume any model on the sparse vectors, S_t. In particular, it allows the support sets of the S_t's to be either independent, e.g. generated via the model of [5] (resulting in $\mathcal{S}_t$ being full rank w.h.p.), or correlated over time (can result in $\mathcal{S}_t$ being low rank). As explained in Sec IV-D, the only thing that is required is that there be *some* support changes every so often. We should point out that some of the other works that study the batch problem, e.g. [5], [20], also allow $\mathcal{S}_t$ to be low rank.

If L_t is the signal of interest, then ReProCS is a solution to recursive robust PCA in the presence of sparse and possibly correlated outliers. To the best of our knowledge, this is the first rigorous analysis of any recursive (online) robust PCA approach and definitely the first to study recursive (online) robust PCA with correlated outliers. Our results directly apply to the missing data case as well or equivalently the case where the outlier locations are known (see Sec IX-A).

The proof techniques used in our work are very different from those used to obtain performance guarantees in the other recent batch robust PCA works [5], [6], [12], [14], [13], [15], [16], [23], [21], [20], [22], [24]. The works of [14], [13] also study a different case: that where an entire vector is either an outlier or an inlier. Our proof utilizes (a) sparse recovery results [25]; (b) results from matrix perturbation theory that bound the estimation error in computing the eigenvectors of a perturbed Hermitian matrix with respect to eigenvectors of the original Hermitian matrix (sin θ theorem [26]) and that bound the perturbed eigenvalues (Weyl's theorem [27]) and (c) high probability bounds on eigenvalues of sums of independent random matrices (matrix Hoeffding inequality [28]).

A key difference of our approach compared with most existing work analyzing finite sample PCA, e.g. [29] and references therein, is that it needs to provably work in the presence of perturbation/noise that is correlated with L_t. Most existing works, including [29] and the references it discusses, assume that the noise is independent of the data.

When L_t is the signal of interest, the ReProCS approach is related to that of [30], [31], [32] in that all of these first try to nullify the low dimensional signal by projecting the measurement vector into a subspace perpendicular to that of the low dimensional signal, and then solve for the sparse "error" vector (outlier). However, the big difference is that in all of these works the basis for the subspace of the low dimensional signal is *perfectly known.* Our work studies *the case where the subspace is not known*. We have an initial approximate estimate of the subspace, but over time it can change quite significantly. The only thing we require is that the changes per unit time are "slow" in a sense quantified in Sec III-B.

In this work, to keep things simple, we use ℓ_1 minimization done separately for each time instant (also referred to as basis pursuit denoising (BPDN)) [25], [33]. However, this can be replaced by any other sparse recovery algorithm, either recursive or batch, as long as the batch algorithm is applied to α frames at a time (with α selected as explained in Sec IV). Notice that ReProCS allows correlated sparse vectors. If something is known about the correlation model, one could replace BPDN by modified-CS or support-predicted modified-CS [34]. Also, many of the batch CS algorithms from literature could be used.

D. Paper Organization

The rest of the paper is organized as follows. We give the notation and background required for the rest of the paper in Sec II. The problem definition and the model assumptions are given in Sec III. We explain the ReProCS algorithm and give its performance guarantees (Theorem 4.2) in Sec IV. The proof outline and the terms used in the proof are defined in Sec V. The main lemmas needed to prove Theorem 4.2 are given in Sec VI. The proof of Theorem 4.2, which follows easily from the main lemmas is also in section VI. In sections VII and VIII we prove the two main lemmas. A more general subspace change model, ReProCS with deletion, and the extension of our results to the missing data case is discussed in Sec IX. In Sec X-A, we show that our slow subspace change model indeed holds for real videos. In Sec X-B, we explain how one can automatically set parameters for ReProCS in practice. In Sec X-C, we show numerical experiments demonstrating Theorem 4.2, as well as comparisons of ReProCS and practical ReProCS with PCP. Conclusions and future work are given in Sec XI.

II. Notation and Background

A. Notation

For a set $T \subset \{1, 2, \dots, n\}$, we use $|T|$ to denote its cardinality, i.e., the number of elements in T. We use T^c to denote its complement w.r.t. $\{1, 2, \dots n\}$, i.e. $T^c := \{i \in \{1, 2, \dots n\} : i \notin T\}$.

We use the interval notation, $[t_1, t_2]$, to denote the set of all integers between and including t_1 to t_2, i.e. $[t_1, t_2] := \{t_1, t_1 + 1, \dots, t_2\}$. For a vector v, v_i denotes the ith entry of v and v_T denotes a vector consisting of the entries of v indexed by T. We use $\|v\|_p$ to denote the ℓ_p norm of v. The support of v, supp(v), is the set of indices at which v is nonzero, supp$(v) := \{i : v_i \neq 0\}$. We say that v is s-sparse if $|\text{supp}(v)| \leq s$.

For a matrix B, B' denotes its transpose, and $B^\dagger$ its pseudo-inverse. For a matrix with linearly independent columns, $B^\dagger = (B'B)^{-1}B'$. We use $\|B\|_2 := \max_{x \neq 0} \|Bx\|_2 / \|x\|_2$ to denote the induced 2-norm of the matrix. Also, $\|B\|_*$ is the nuclear norm (sum of singular values) and $\|B\|_{\max}$ denotes the maximum over the absolute values of all its entries. We let $\sigma_i(B)$ denotes the ith largest singular value of B. For a Hermitian matrix, B, we use the notation $B \overset{EVD}{=} U \Lambda U'$ to denote the eigenvalue decomposition of B. Here U is an orthonormal matrix and Λ is a diagonal matrix with entries arranged in decreasing order. Also, we use $\lambda_i(B)$ to denote the ith largest eigenvalue of a Hermitian matrix B and we use $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$ denote its maximum and minimum eigenvalues. If B is Hermitian positive semi-definite (p.s.d.), then $\lambda_i(B) = \sigma_i(B)$. For Hermitian matrices B_1 and B_2, the notation $B_1 \preceq B_2$ means that $B_2 - B_1$ is p.s.d. Similarly, $B_1 \succeq B_2$ means that $B_1 - B_2$ is p.s.d.

For a Hermitian matrix B, $\|B\|_2 = \sqrt{\max(\lambda_{\max}^2(B), \lambda_{\min}^2(B))}$ and thus, $\|B\|_2 \leq b$ implies that $-b \leq \lambda_{\min}(B) \leq \lambda_{\max}(B) \leq b$.

We use I to denote an identity matrix of appropriate size. For an index set T and a matrix B, B_T is the sub-matrix of B containing columns with indices in the set T. Notice that $B_T := BI_T$. Given a matrix B of size $m \times n$ and B_2 of size $m \times n_2$, $[B \ B_2]$ constructs a new matrix by concatenating matrices B and B_2 in a horizontal direction.

For a tall matrix P, span(P) denotes the subspace spanned by the column vectors of P.

The notation [.] denotes an empty matrix.

Definition 2.1. *We refer to a tall matrix P as a* basis matrix *if it satisfies $P'P = I$.*

Definition 2.2. *For a basis matrix P and any other matrix B, we say that "span(P) approximately contains span(B)" if $\|(I - PP')B\|_2/\|B\|_2$ is small.*

Definition 2.3. *The s-*restricted isometry constant (RIC) *[30], δ_s, for an $n \times m$ matrix Ψ is the smallest real number satisfying $(1 - \delta_s)\|x\|_2^2 \le \|\Psi_T x\|_2^2 \le (1 + \delta_s)\|x\|_2^2$ for all sets $T \subseteq \{1, 2, \dots n\}$ with $|T| \le s$ and all real vectors x of length $|T|$.*

It is easy to see that $\max_{T:|T|\le s} \|({\Psi_T}'\Psi_T)^{-1}\|_2 \le \frac{1}{1-\delta_s(\Psi)}$ [30].

Definition 2.4. *We give some notation for random variables in this definition.*

1) *We let $\mathbf{E}[Z]$ denote the expectation of a random variable (r.v.) Z and $\mathbf{E}[Z|X]$ denote its conditional expectation given another r.v. X.*
2) *Let $\mathcal{B}$ be a set of values that a r.v. Z can take. We use $\mathcal{B}^e$ to denote the* event *$Z \in \mathcal{B}$, i.e. $\mathcal{B}^e := \{Z \in \mathcal{B}\}$.*
3) *The probability of any event $\mathcal{B}^e$ can be expressed as [35],*

$$\mathbf{P}(\mathcal{B}^e) := \mathbf{E}[\mathbb{I}_{\mathcal{B}}(Z)].$$

where

$$\mathbb{I}_{\mathcal{B}}(Z) := \begin{cases} 1 & \textit{if } Z \in \mathcal{B} \\ 0 & \textit{otherwise} \end{cases}$$

is the indicator function on the set $\mathcal{B}$.

4) *For two events $\mathcal{B}^e, \tilde{\mathcal{B}^e}$, $\mathbf{P}(\mathcal{B}^e|\tilde{\mathcal{B}^e})$ refers to the conditional probability of $\mathcal{B}^e$ given $\tilde{\mathcal{B}^e}$, i.e. $\mathbf{P}(\mathcal{B}^e|\tilde{\mathcal{B}^e}) := \mathbf{P}(\mathcal{B}^e, \tilde{\mathcal{B}^e})/\mathbf{P}(\tilde{\mathcal{B}^e})$.*
5) *For a r.v. X, and a set $\mathcal{B}$ of values that the r.v. Z can take, the notation $\mathbf{P}(\mathcal{B}^e|X)$ is defined as*

$$\mathbf{P}(\mathcal{B}^e|X) := \mathbf{E}[\mathbb{I}_{\mathcal{B}}(Z)|X].$$

Notice that $\mathbf{P}(\mathcal{B}^e|X)$ is a r.v. (it is a function of the r.v. X) that always lies between zero and one.

Finally, RHS refers to the right hand side of an equation or inequality; w.p. means "with probability"; and w.h.p. means "with high probability". Also we use $a \lesssim b$ to indicate (in a non-rigorous sense) that the dominant term in the upper bound on a is b.

B. Compressive Sensing result

The error bound for noisy compressive sensing (CS) based on the RIC is as follows [25].

Theorem 2.5 ([25])**.** *Suppose we observe*

$$y := \Psi x + z$$

where z is the noise. Let $\hat{x}$ be the solution to following problem

$$\min_x \|x\|_1 \textit{ subject to } \|y - \Psi x\|_2 \le \xi \tag{3}$$

Assume that x is s-sparse, $\|z\|_2 \le \xi$, and $\delta_{2s}(\Psi) < b(\sqrt{2} - 1)$ for some $0 \le b < 1$. Then the solution of (3) obeys

$$\|\hat{x} - x\|_2 \le C_1 \xi$$

with $C_1 = \dfrac{4\sqrt{1 + \delta_{2s}(\Psi)}}{1 - (\sqrt{2}+1)\delta_{2s}(\Psi)} \le \dfrac{4\sqrt{1 + b(\sqrt{2} - 1)}}{1 - (\sqrt{2}+1)b(\sqrt{2} - 1)}$.

Remark 2.6. *Notice that if* b *is small enough,* C_1 *is a small constant but* $C_1 > 1$*. For example, if* $\delta_{2s}(\Psi) \leq 0.15$*, then* $C_1 \leq 7$*. If* $C_1\xi > \|x\|_2$*, the normalized reconstruction error bound would be greater than* 1*, making the result useless. Hence, (3) gives a small reconstruction error bound only for the small noise case, i.e., the case where* $\|z\|_2 \leq \xi \ll \|x\|_2$*. In fact this is true for most existing literature on CS and sparse recovery, with the exception of [10], [36], [37] (focus on large but sparse noise) and [5], [6].*

C. Results from linear algebra

Davis and Kahan's $\sin\theta$ theorem [26] studies the rotation of eigenvectors by perturbation.

Theorem 2.7 ($\sin\theta$ theorem [26])**.** *Given two Hermitian matrices* $\mathcal{A}$ *and* $\mathcal{H}$ *satisfying*

$$\mathcal{A} = \begin{bmatrix} E \; E_\perp \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & A_\perp \end{bmatrix} \begin{bmatrix} E' \\ {E_\perp}' \end{bmatrix}, \; \mathcal{H} = \begin{bmatrix} E \; E_\perp \end{bmatrix} \begin{bmatrix} H & B' \\ B & H_\perp \end{bmatrix} \begin{bmatrix} E' \\ {E_\perp}' \end{bmatrix} \tag{4}$$

where $[E \; E_\perp]$ *is an orthonormal matrix. The two ways of representing* $\mathcal{A} + \mathcal{H}$ *are*

$$\mathcal{A} + \mathcal{H} = \begin{bmatrix} E \; E_\perp \end{bmatrix} \begin{bmatrix} A + H & B' \\ B & A_\perp + H_\perp \end{bmatrix} \begin{bmatrix} E' \\ {E_\perp}' \end{bmatrix} = \begin{bmatrix} F \; F_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda_\perp \end{bmatrix} \begin{bmatrix} F' \\ {F_\perp}' \end{bmatrix}$$

where $[F \; F_\perp]$ *is another orthonormal matrix. Let* $\mathcal{R} := (\mathcal{A} + \mathcal{H})E - \mathcal{A}E = \mathcal{H}E$*. If* $\lambda_{\min}(A) > \lambda_{\max}(\Lambda_\perp)$*, then*

$$\|(I - FF')E\|_2 \leq \frac{\|\mathcal{R}\|_2}{\lambda_{\min}(A) - \lambda_{\max}(\Lambda_\perp)}$$

The above result bounds the amount by which the two subspaces span(E) and span(F) differ as a function of the norm of the perturbation $\|\mathcal{R}\|_2$ and of the gap between the minimum eigenvalue of A and the maximum eigenvalue of $\Lambda_\perp$. Next, we state Weyl's theorem which bounds the eigenvalues of a perturbed Hermitian matrix, followed by Ostrowski's theorem.

Theorem 2.8 (Weyl [27])**.** *Let* $\mathcal{A}$ *and* $\mathcal{H}$ *be two* $n \times n$ *Hermitian matrices. For each* $i = 1, 2, \ldots, n$ *we have*

$$\lambda_i(\mathcal{A}) + \lambda_{\min}(\mathcal{H}) \leq \lambda_i(\mathcal{A} + \mathcal{H}) \leq \lambda_i(\mathcal{A}) + \lambda_{\max}(\mathcal{H})$$

Theorem 2.9 (Ostrowski [27])**.** *Let* H *and* W *be* $n \times n$ *matrices, with* H *Hermitian and* W *nonsingular. For each* $i = 1, 2 \ldots n$*, there exists a positive real number* θ_i *such that* $\lambda_{\min}(WW') \leq \theta_i \leq \lambda_{\max}(WW')$ *and* $\lambda_i(WHW') = \theta_i \lambda_i(H)$*. Therefore,*

$$\lambda_{\min}(WHW') \geq \lambda_{\min}(WW')\lambda_{\min}(H)$$

The following lemma proves some simple linear algebra facts.

Lemma 2.10. *Suppose that* P*,* $\hat{P}$ *and* Q *are three basis matrices. Also,* P *and* $\hat{P}$ *are of the same size,* $Q'P = 0$ *and* $\|(I - \hat{P}\hat{P}')P\|_2 = \zeta_*$*. Then,*

1) $\|(I - \hat{P}\hat{P}')PP'\|_2 = \|(I - PP')\hat{P}\hat{P}'\|_2 = \|(I - PP')\hat{P}\|_2 = \|(I - \hat{P}\hat{P}')P\|_2 = \zeta_*$
2) $\|PP' - \hat{P}\hat{P}'\|_2 \leq 2\|(I - \hat{P}\hat{P}')P\|_2 = 2\zeta_*$
3) $\|\hat{P}'Q\|_2 \leq \zeta_*$
4) $\sqrt{1 - \zeta_*^2} \leq \sigma_i((I - \hat{P}\hat{P}')Q) \leq 1$

The proof is in the Appendix.

D. High probability tail bounds for sums of independent random matrices

The following lemma follows easily using Definition 2.4. We will use this at various places in the paper.

Lemma 2.11. *Suppose that* $\mathcal{B}$ *is the set of values that the r.v.s* X, Y *can take. Suppose that* $\mathcal{C}$ *is a set of values that the r.v.* X *can take. For a* $0 \leq p \leq 1$*, if* $\mathbf{P}(\mathcal{B}^e|X) \geq p$ *for all* $X \in \mathcal{C}$*, then* $\mathbf{P}(\mathcal{B}^e|\mathcal{C}^e) \geq p$ *as long as* $\mathbf{P}(\mathcal{C}^e) > 0$*.*

The proof is in the Appendix.

The following lemma is an easy consequence of the chain rule of probability applied to a contracting sequence of events.

Lemma 2.12. *For a sequence of events $E_0^e, E_1^e, \dots E_m^e$ that satisfy $E_0^e \supseteq E_1^e \supseteq E_2^e \dots \supseteq E_m^e$, the following holds*

$$\mathbf{P}(E_m^e | E_0^e) = \prod_{k=1}^{m} \mathbf{P}(E_k^e | E_{k-1}^e).$$

Proof: $\mathbf{P}(E_m^e|E_0^e) = \mathbf{P}(E_m^e, E_{m-1}^e, \dots E_0^e|E_0^e) = \prod_{k=1}^m \mathbf{P}(E_k^e|E_{k-1}^e, E_{k-2}^e, \dots E_0^e) = \prod_{k=1}^m \mathbf{P}(E_k^e|E_{k-1}^e)$. ■

Next, we state the matrix Hoeffding inequality [28, Theorem 1.3] which gives tail bounds for sums of independent random matrices.

Theorem 2.13 (Matrix Hoeffding for a zero mean Hermitian matrix [28])**.** *Consider a finite sequence $\{Z_t\}$ of independent, random, Hermitian matrices of size $n \times n$, and let $\{A_t\}$ be a sequence of fixed Hermitian matrices. Assume that each random matrix satisfies (i) $\mathbf{P}(Z_t^2 \preceq A_t^2) = 1$ and (ii) $\mathbf{E}(Z_t) = 0$. Then, for all $\epsilon > 0$,*

$$\mathbf{P}\left(\lambda_{\max}\left(\sum_t Z_t\right) \le \epsilon\right) \ge 1 - n \exp\left(\frac{-\epsilon^2}{8\sigma^2}\right), \ \textit{where } \sigma^2 = \Big\| \sum_t A_t^2 \Big\|_2$$

The following two corollaries of Theorem 2.13 are easy to prove. The proofs are given in the Appendix.

Corollary 2.14 (Matrix Hoeffding conditioned on another random variable for a nonzero mean Hermitian matrix)**.** *Given an α-length sequence $\{Z_t\}$ of random Hermitian matrices of size $n \times n$, a r.v. X, and a set $\mathcal{C}$ of values that X can take. Assume that, for all $X \in \mathcal{C}$, (i) Z_t's are conditionally independent given X; (ii) $\mathbf{P}(b_1 I \preceq Z_t \preceq b_2 I | X) = 1$ and (iii) $b_3 I \preceq \frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X) \preceq b_4 I$. Then for all $\epsilon > 0$,*

$$\mathbf{P}\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_t Z_t\right) \le b_4 + \epsilon \Big| X\right) \ge 1 - n \exp\left(\frac{-\alpha\epsilon^2}{8(b_2 - b_1)^2}\right) \ \textit{for all } X \in \mathcal{C}$$

$$\mathbf{P}\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_t Z_t\right) \ge b_3 - \epsilon \Big| X\right) \ge 1 - n \exp\left(\frac{-\alpha\epsilon^2}{8(b_2 - b_1)^2}\right) \ \textit{for all } X \in \mathcal{C}$$

The proof is in the Appendix.

Corollary 2.15 (Matrix Hoeffding conditioned on another random variable for an arbitrary nonzero mean matrix)**.** *Given an α-length sequence $\{Z_t\}$ of random Hermitian matrices of size $n \times n$, a r.v. X, and a set $\mathcal{C}$ of values that X can take. Assume that, for all $X \in \mathcal{C}$, (i) Z_t's are conditionally independent given X; (ii) $\mathbf{P}(\|Z_t\|_2 \le b_1 | X) = 1$ and (iii) $\|\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X)\|_2 \le b_2$. Then, for all $\epsilon > 0$,*

$$\mathbf{P}\left(\Big\|\frac{1}{\alpha}\sum_t Z_t\Big\|_2 \le b_2 + \epsilon \Big| X\right) \ge 1 - (n_1 + n_2) \exp\left(\frac{-\alpha\epsilon^2}{32 b_1^2}\right) \ \textit{for all } X \in \mathcal{C}$$

The proof is in the Appendix.

III. Problem Definition and Model Assumptions

We give the problem definition below followed by the model and then describe the two key assumptions.

A. Problem Definition

The measurement vector at time t, M_t, is an n dimensional vector which can be decomposed as

$$M_t = L_t + S_t \tag{5}$$

Here S_t is a sparse vector with support set size at most s and minimum magnitude of nonzero values at least $S_{\min}$. L_t is a dense but low dimensional vector, i.e. $L_t = P_{(t)} a_t$ where $P_{(t)}$ is an $n \times r_{(t)}$ basis matrix with $r_{(t)} \ll n$, that changes every so often. $P_{(t)}$ and a_t change according to the model given below. We are given an accurate estimate of the subspace in which the initial t_{train} L_t's lie, i.e. we are given a basis matrix $\hat{P}_0$ so that $\|(I - \hat{P}_0 \hat{P}_0')P_0\|_2$ is small. Here P_0 is a basis matrix for $\text{span}(\mathcal{L}_{t_{\text{train}}})$, i.e. $\text{span}(P_0) = \text{span}(\mathcal{L}_{t_{\text{train}}})$. Also, for the first t_{train} time instants, S_t is either zero or very small. The goal is

1) to estimate both S_t and L_t at each time $t > t_{\text{train}}$, and
2) to estimate span$(\mathcal{L}_t)$ every so often, i.e. compute $\hat{P}_{(t)}$ so that the subspace estimation error, $\text{SE}_{(t)} := \|(I - \hat{P}_{(t)}\hat{P}_{(t)}')P_{(t)}\|_2$ is small.

Notation for S_t. We do not assume anything about S_t except sparsity. Let $T_t := \{i : \ (S_t)_i \neq 0\}$ denote the support of S_t. Define

$$S_{\min} := \min_{t > t_{\text{train}}} \min_{i \in T_t} |(S_t)_i|, \text{ and } s := \max_t |T_t|$$

In words, $S_{\min}$ is a lower bound on the magnitude of a non-zero entry of S_t for all t, and s is an upper bound on the support size of S_t for all t.

Model on L_t. $\{L_t\}$ is a sequence of dense vectors satisfying the following model.

1) L_t lies in a low dimensional subspace that changes every-so-often. Let t_j denote the change times. Then the following holds.
   a) $L_t = P_{(t)}a_t$ with $P_{(t)} = P_j$ for all $t_j \leq t < t_{j+1}$, $j = 0, 1, 2 \cdots J$, i.e. there is a maximum of J subspace change times. We let $t_0 = 0$. We can define $t_{J+1} = \infty$.
   b) P_j is an $n \times r_j$ basis matrix with $r_j \ll n$ and $r_j \ll (t_{j+1} - t_j)$.
   c) At the change times, t_j, P_j changes as $P_j = [P_{j-1} \ P_{j,\text{new}}]$ where $P_{j,\text{new}}$ is a $n \times c_{j,\text{new}}$ basis matrix with $P_{j,\text{new}}' P_{j-1} = 0$. Thus $r_j = r_{j-1} + c_{j,\text{new}}$. This model is illustrated in Fig 1.
   d) There exists a constant c_{mx} such that $0 \leq c_{j,\text{new}} \leq c_{mx}$ for all j.
2) The vector of coefficients, $a_t := P_{(t)}{}'L_t$, is a random variable (r.v.) with the following properties.
   a) The a_t are mutually independent over t.
   b) It is a zero mean bounded r.v., i.e. $\mathbf{E}(a_t) = 0$ and there exists a constant γ_* s.t. $\|a_t\|_\infty \leq \gamma_*$ for all t.
   c) Its covariance matrix $\Lambda_t := \text{Cov}[a_t] = \mathbf{E}(a_t a_t')$ is diagonal with $\lambda^- := \min_t \lambda_{\min}(\Lambda_t) > 0$ and $\lambda^+ := \max_t \lambda_{\max}(\Lambda_t) < \infty$. Thus the condition number of any Λ_t is bounded by $f := \frac{\lambda^+}{\lambda^-}$.
   d) For $t_j \leq t < t_{j+1}$, $a_t = P_j{}'L_t$ is an r_j length vector which can be split as

$$a_t = \begin{bmatrix} a_{t,*} \\ a_{t,\text{new}} \end{bmatrix}$$

where $a_{t,*} := P_{j-1}{}'L_t$ is an r_{j-1} length vector of coefficients for the existing directions and $a_{t,\text{new}} := P_{j,\text{new}}{}'L_t$ is a $c_{j,\text{new}}$ length vector of coefficients for the new directions. Thus, for this interval, L_t can be rewritten as

$$L_t = [P_{j-1} \ P_{j,\text{new}}] \begin{bmatrix} a_{t,*} \\ a_{t,\text{new}} \end{bmatrix} = P_{j-1}a_{t,*} + P_{j,\text{new}}a_{t,\text{new}}$$

Also, Λ_t can be split as $\Lambda_t = \begin{bmatrix} (\Lambda_t)_* & 0 \\ 0 & (\Lambda_t)_{\text{new}} \end{bmatrix}$ where $(\Lambda_t)_* = \text{Cov}[a_{t,*}]$ and $(\Lambda_t)_{\text{new}} = \text{Cov}[a_{t,\text{new}}]$ are diagonal matrices.

3) P_j and a_t change slowly in the sense quantified below in Sec III-B.
4) The L_t's, and hence their subspace basis matrices P_j, are dense as quantified in Sec III-C.

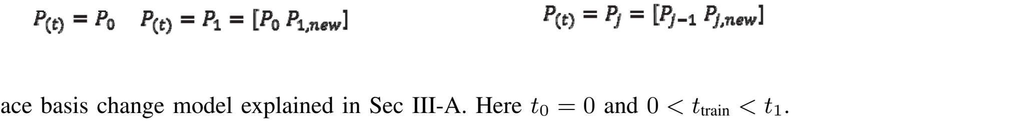

$P_{(t)} = P_0$
$P_{(t)} = P_1 = [P_0 \ P_{1,new}]$
$P_{(t)} = P_j = [P_{j-1} \ P_{j,new}]$

Fig. 1. The subspace basis change model explained in Sec III-A. Here $t_0 = 0$ and $0 < t_{\text{train}} < t_1$.

We discuss the above model assumptions after stating our main result in Sec. IV-C.

In the above model, the rank of P_j will keep increasing whenever $c_{j,\text{new}} > 0$. In other words, the dimension of the subspace in which the current L_t will keep increasing. However, in practical applications, this may not happen. Some directions may also get removed from P_j so that the subspace dimension remains roughly constant with time. This can be modeled as $P_j = [P_{j-1} \ P_{j,\text{new}}] \setminus P_{j,\text{old}}$. As we show in Corollary 9.1 in Sec IX-B, even with this more general model, our proposed algorithm directly applies and its performance guarantees also change only slightly.

We do not use this more general model here to keep the notation simple. Recall that our goal is to estimate span$(\mathcal{L}_t)$. With our current model (no removals), span$(P_{(t)}) =$ span$(\mathcal{L}_t)$ and thus our goal can be equivalently stated as one of computing a basis matrix $\hat{P}_{(t)}$ so that span$(\hat{P}_{(t)}) \approx$ span$(P_{(t)})$.

B. Slow subspace change

By slow subspace change we mean all of the following.

First, the delay between consecutive subspace change times, $t_{j+1} - t_j$, is large enough.

Second, the magnitude of the projection of L_t along the newly added directions, $a_{t,\text{new}}$, is initially small, i.e. $\max_{t_j \le t < t_j+\alpha} \|a_{t,\text{new}}\|_\infty \le \gamma_{\text{new}}$, with $\gamma_{\text{new}} \ll \gamma_*$ and $\gamma_{\text{new}} \ll S_{\min}$, but can increase gradually. We model this as follows. Split the interval $[t_j, t_{j+1} - 1]$ into α length periods. We assume that

$$\max_j \max_{t \in [t_j+(k-1)\alpha, t_j+k\alpha-1]} \|a_{t,\text{new}}\|_\infty \le \gamma_{\text{new},k} := \min(v^{k-1}\gamma_{\text{new}}, \gamma_*)$$

for a $v > 1$ but not too large[1]. This assumption is verified for real video data in Sec. X-A.

Third, the number of newly added directions is small, i.e. $c_{j,\text{new}} \le c_{mx} \ll r_0$. This is also verified in Sec. X-A.

C. Measuring denseness of a matrix and its relation with RIC

For a tall $n \times r$ matrix, B, or for a $n \times 1$ vector, B, we define the the denseness coefficient as follows:

$$\kappa_s(B) := \max_{|T| \le s} \frac{\|I_T{}'B\|_2}{\|B\|_2}.$$

where $\|.\|_2$ is the matrix or vector 2-norm respectively [2]. Clearly, $\kappa_s(B) \le 1$. First consider an n-length vector B. Then κ_s measures the denseness (non-compressibility) of B. A small value indicates that the entries in B are spread out, i.e. it is a dense vector. A large value indicates that it is compressible (approximately or exactly sparse). The worst case (largest possible value) is $\kappa_s(B) = 1$ which indicates that B is an s-sparse vector. The best case is $\kappa_s(B) = \sqrt{s/n}$ and this will occur if each entry of B has the same magnitude. Similarly, for an $n \times r$ matrix B, a small κ_s means that most (or all) of its columns are dense vectors.

Remark 3.1. *The following facts should be noted about $\kappa_s(.)$.*

1) *For an $n \times r$ matrix B, $\kappa_s(B)$ is an increasing function of s.*
2) *For an $n \times r$ basis matrix B, $\kappa_s(B)$ is an increasing function of $r =$ rank(B).*
3) *A loose bound on $\kappa_s(B)$ obtained using triangle inequality is $\kappa_s(B) \le s\kappa_1(B)$.*
4) *For a basis matrix P, $\|P\|_2 = 1$ and hence $\kappa_s(P) = \max_{|T| \le s} \|I_T'P\|_2$ and $\kappa_s(PP') = \kappa_s(P)$. Thus, for any other basis matrix Q for which span$(Q) =$ span(P), $\kappa_s(P) = \kappa_s(Q)$. Thus, $\kappa_s(P)$ is a property of span(P), which is the subspace spanned by the columns of P, and not of the actual entries of P.*

The lemma below relates the denseness coefficient of a basis matrix P to the RIC of $I - PP'$. The proof is in the Appendix.

[1] Small γ_{new} and slowly increasing $\gamma_{\text{new},k}$ is needed for the noise seen by the sparse recovery step to be small. However, if γ_{new} is zero or very small, it will be impossible to estimate the new subspace. This will not happen in our model because $\gamma_{\text{new}} \ge \lambda^- > 0$.

[2] In future work [11] we define $\kappa_s(B) := \max_{|T| \le s} \|I_T{}'Q(B)\|_2$ where $Q(B)$ is a basis matrix for span(B) ie the columns of $Q(B)$ form an orthonormal basis for span(B). With this definition item 4 of Remark 3.1 is immediate. Also, all of our results will still hold with this new definition.

Lemma 3.2. *For an $n \times r$ basis matrix P (i.e P satisfying $P'P = I$),*

$$\delta_s(I - PP') = \kappa_s^2(P).$$

In other words, if P is dense enough (small κ_s), then the RIC of $I - PP'$ is small. Thus, using Theorem 2.5, all s-sparse vectors, S_t can be accurately recovered from $y_t := (I - PP')S_t + \beta_t$ if β_t is small noise.

In this work, we assume an upper bound on $\kappa_s(P_{j-1})$, and a tighter upper bound on $\kappa_s(P_{j,\text{new}})$. Additionally, we also assume denseness of another matrix, $D_{j,\text{new},k}$, whose columns span the currently unestimated part of $\text{span}(P_{j,\text{new}})$ (see Theorem 4.2).

As we explain in Sec IV-E, the denseness coefficient $\kappa_s(B)$ is related to the denseness assumption required by PCP [5].

IV. Recursive Projected CS (ReProCS) and its Performance Guarantees

In Sec IV-B, we explain the ReProCS algorithm and why it works. We give its performance guarantees in Sec. IV-C and discuss the assumptions used by our result in Sec. IV-D. A qualitative discussion w.r.t. the result for PCP is given in Sec IV-E. Practical parameter setting for ReProCS is discussed later in Sec. X-B.

We summarize the Recursive Projected CS (ReProCS) algorithm in Algorithm 2. It uses the following definition.

Definition 4.1. *Define the time interval $\mathcal{I}_{j,k} := [t_j + (k-1)\alpha, t_j + k\alpha - 1]$ for $k = 1, \dots K$ and $\mathcal{I}_{j,K+1} := [t_j + K\alpha, t_{j+1} - 1]$. Here, K is the algorithm parameter in Algorithm 2.*

A. The Projection-PCA algorithm

Given a data matrix $\mathcal{D}$, a basis matrix P and an integer r, projection-PCA (proj-PCA) applies PCA on $\mathcal{D}_{\text{proj}} := (I - PP')\mathcal{D}$, i.e., it computes the top r eigenvectors (the eigenvectors with the largest r eigenvalues) of $\frac{1}{\alpha_{\mathcal{D}}}\mathcal{D}_{\text{proj}}{\mathcal{D}_{\text{proj}}}'$. Here $\alpha_{\mathcal{D}}$ is the number of column vectors in $\mathcal{D}$. This is summarized in Algorithm 1.

If $P = [.]$, then projection-PCA reduces to standard PCA, i.e. it computes the top r eigenvectors of $\frac{1}{\alpha_{\mathcal{D}}}\mathcal{D}\mathcal{D}'$.

We should mention that the idea of projecting perpendicular to a partly estimated subspace has been used in different contexts in past work [38], [14].

Algorithm 1 projection-PCA: $Q \leftarrow \text{proj-PCA}(\mathcal{D}, P, r)$

```
1) Projection: compute $\mathcal{D}_{\text{proj}} \leftarrow (I - PP')\mathcal{D}$
2) PCA: compute $\frac{1}{\alpha_{\mathcal{D}}}\mathcal{D}_{\text{proj}}{\mathcal{D}_{\text{proj}}}' \overset{EVD}{=} \begin{bmatrix} Q \; Q_{\perp} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda_{\perp} \end{bmatrix} \begin{bmatrix} Q' \\ {Q_{\perp}}' \end{bmatrix}$ where $Q$ is an $n \times r$ basis matrix and $\alpha_{\mathcal{D}}$ is the number of columns in $\mathcal{D}$.
```

B. Recursive Projected CS (ReProCS)

The key idea of ReProCS is as follows. Assume that the current basis matrix $P_{(t)}$ has been accurately predicted using past estimates of L_t, i.e. we have $\hat{P}_{(t-1)}$ with $\|(I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)})P_{(t)}\|_2$ small. We project the measurement vector, M_t, into the space perpendicular to $\hat{P}_{(t-1)}$ to get the projected measurement vector $y_t := \Phi_{(t)} M_t$ where $\Phi_{(t)} = I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)}$ (step 1a). Since the $n \times n$ projection matrix, $\Phi_{(t)}$ has rank $n - r_*$ where $r_* = \text{rank}(\hat{P}_{(t-1)})$, therefore y_t has only $n - r_*$ "effective" measurements[3], even though its length is n. Notice that y_t can be rewritten as $y_t = \Phi_{(t)} S_t + \beta_t$ where $\beta_t := \Phi_{(t)} L_t$. Since $\|(I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)})P_{(t)}\|_2$ is small, the projection nullifies most of the contribution of L_t and so the projected noise β_t is small. Recovering the n dimensional sparse vector S_t from y_t now becomes a traditional sparse recovery or CS problem in small noise [33], [39], [40]. We use ℓ_1 minimization to recover it (step 1b). If the current basis matrix $P_{(t)}$, and hence its estimate, $\hat{P}_{(t-1)}$, is dense enough, then, by Lemma 3.2, the RIC of $\Phi_{(t)}$ is small enough. Using Theorem 2.5, this ensures that S_t can be accurately recovered from y_t.

[3] i.e. some r_* entries of y_t are linear combinations of the other $n - r_*$ entries

Algorithm 2 Recursive Projected CS (ReProCS)

```
Parameters: algorithm parameters: $\xi$, $\omega$, $\alpha$, $K$, model parameters: $t_j$, $r_0$, $c_{j,\text{new}}$
(set as in Theorem 4.2 or as in Sec X-B when the model is not known)
Input: $M_t$, Output: $\hat{S}_t$, $\hat{L}_t$, $\hat{P}_{(t)}$
Initialization: Compute $\hat{P}_0 \leftarrow$ proj-PCA($[L_1, L_2, \cdots, L_{t_{\text{train}}}], [.], r_0$) and set $\hat{P}_{(t)} \leftarrow \hat{P}_0$.
Let $j \leftarrow 1$, $k \leftarrow 1$.
For $t > t_{\text{train}}$, do the following:
  1) Estimate $T_t$ and $S_t$ via Projected CS:
      a) Nullify most of $L_t$: compute $\Phi_{(t)} \leftarrow I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)}$, compute $y_t \leftarrow \Phi_{(t)} M_t$
      b) Sparse Recovery: compute $\hat{S}_{t,\text{cs}}$ as the solution of $\min_x \|x\|_1 \ s.t. \ \|y_t - \Phi_{(t)} x\|_2 \le \xi$
      c) Support Estimate: compute $\hat{T}_t = \{i : \ |(\hat{S}_{t,\text{cs}})_i| > \omega\}$
      d) LS Estimate of $S_t$: compute $(\hat{S}_t)_{\hat{T}_t} = ((\Phi_t)_{\hat{T}_t})^\dagger y_t, \ (\hat{S}_t)_{\hat{T}_t^c} = 0$
  2) Estimate $L_t$: $\hat{L}_t = M_t - \hat{S}_t$.
  3) Update $\hat{P}_{(t)}$: K Projection PCA steps.
      a) If $t = t_j + k\alpha - 1$,
          i) $\hat{P}_{j,new,k} \leftarrow$ proj-PCA$\left(\left[\hat{L}_{t_j+(k-1)\alpha}, \dots, \hat{L}_{t_j+k\alpha-1}\right], \hat{P}_{j-1}, c_{j,\text{new}}\right)$.
          ii) set $\hat{P}_{(t)} \leftarrow [\hat{P}_{j-1} \ \hat{P}_{j,\text{new},k}]$; increment $k \leftarrow k+1$.
          Else
          i) set $\hat{P}_{(t)} \leftarrow \hat{P}_{(t-1)}$.
      b) If $t = t_j + K\alpha - 1$, then set $\hat{P}_j \leftarrow [\hat{P}_{j-1} \ \hat{P}_{j,\text{new},K}]$. Increment $j \leftarrow j+1$. Reset $k \leftarrow 1$.
  4) Increment $t \leftarrow t+1$ and go to step 1.
```

By thresholding on the recovered S_t, one gets an estimate of its support (step 1c). By computing a least squares (LS) estimate of S_t on the estimated support and setting it to zero everywhere else (step 1d), we can get a more accurate final estimate, $\hat{S}_t$, as first suggested in [41]. This $\hat{S}_t$ is used to estimate L_t as $\hat{L}_t = M_t - \hat{S}_t$. As we explain in the proof of Lemma 8.1, if the support estimation threshold, ω, is chosen appropriately, we can get exact support recovery, i.e. $\hat{T}_t = T_t$. In this case, the error $e_t := \hat{S}_t - S_t = L_t - \hat{L}_t$ has the following simple expression:

$$e_t = I_{T_t} {(\Phi_{(t)})_{T_t}}^\dagger \beta_t = I_{T_t} [(\Phi_{(t)})'_{T_t} (\Phi_{(t)})_{T_t}]^{-1} {I_{T_t}}' \Phi_{(t)} L_t \tag{6}$$

The second equality follows because ${(\Phi_{(t)})_T}' \Phi_{(t)} = (\Phi_{(t)} I_T)' \Phi_{(t)} = {I_T}' \Phi_{(t)}$ for any set T. Consider a $t \in \mathcal{I}_{j,1}$. At this time, L_t satisfies $L_t = P_{j-1} a_{t,*} + P_{j,\text{new}} a_{t,\text{new}}$, $P_{(t)} = P_j = [P_{j-1}, P_{j,\text{new}}]$, $\hat{P}_{(t-1)} = \hat{P}_{j-1}$ and so $\Phi_{(t)} = \Phi_{j,0} := I - \hat{P}_{j-1}\hat{P}'_{j-1}$. Let $\Phi_{j,k} := I - \hat{P}_{j-1}\hat{P}'_{j-1} - \hat{P}_{j,\text{new},k}\hat{P}'_{j,\text{new},k}$ (with $\hat{P}_{j,\text{new},0} = [.]$), $\zeta_{j,k} := \|\Phi_{j,k} P_{j,\text{new}}\|_2$, $\kappa_{s,k} := \max_j \kappa_s(\Phi_{j,k} P_{j,\text{new}})$, $\phi_k := \max_j \max_{|T| \le s} \|[(\Phi_{j,k})'_T (\Phi_{j,k})_T]^{-1}\|_2$, $r_* := r_0 + (j-1)c_{mx}$, and $c := c_{mx}$. We assume that the delay between change times is large enough so that by $t = t_j$, $\hat{P}_{(t-1)} = \hat{P}_{j-1}$ is an accurate enough estimate of P_{j-1}, i.e. $\|\Phi_{j,0} P_{j-1}\|_2 \le r_* \zeta$ for a ζ small enough. Using $\|{I_{T_t}}' \Phi_{j,0} P_{j-1}\|_2 \le \|\Phi_{j,0} P_{j-1}\|_2 \le r_* \zeta$, $\|{I_{T_t}}' \Phi_{j,0} P_{\text{new}}\|_2 \le \kappa_{s,0} \|\Phi_{j,0} P_{j,\text{new}}\|_2$ and $\zeta_{j,0} = \|\Phi_{j,0} P_{\text{new}}\|_2 \le 1$, we get that $\|e_t\|_2 \le \phi_0 r_* \zeta \sqrt{r_*} \gamma_* + \phi_0 \kappa_{s,0} \sqrt{c} \gamma_{\text{new}}$. The denseness assumption on P_{j-1}; $\|\Phi_{j,0} P_{j-1}\|_2 \le r_* \zeta$ and $\phi_0 \le 1/(1 - \delta_s(\Phi_{j,0}))$ ensure that ϕ_0 is only slightly more than one (see Lemma 8.3). If $\sqrt{\zeta} < 1/\gamma_*$, the first term in the bound on $\|e_t\|_2$ is of the order of $\sqrt{\zeta}$ and hence negligible. The denseness assumption on $\Phi_{j,0} P_{j,\text{new}}$, whose columns span the currently unestimated part of span($P_{j,\text{new}}$), ensures that $\kappa_{s,0}$ is significantly less than one. As a result, $\phi_0 \kappa_{s,0} < 1$ and so the error $\|e_t\|_2$ is of the order of $\sqrt{c}\gamma_{\text{new}}$. Since $\gamma_{\text{new}} \ll S_{\min}$ and c is assumed to be small, thus, $\|e_t\|_2 = \|S_t - \hat{S}_t\|_2$ is small compared with $\|S_t\|_2$, i.e. S_t is recovered accurately. With each projection PCA step, as we explain below, the error e_t becomes even smaller.

Since $\hat{L}_t = M_t - \hat{S}_t$ (step 2), e_t also satisfies $e_t = L_t - \hat{L}_t$. Thus, a small e_t means that L_t is also recovered accurately. The

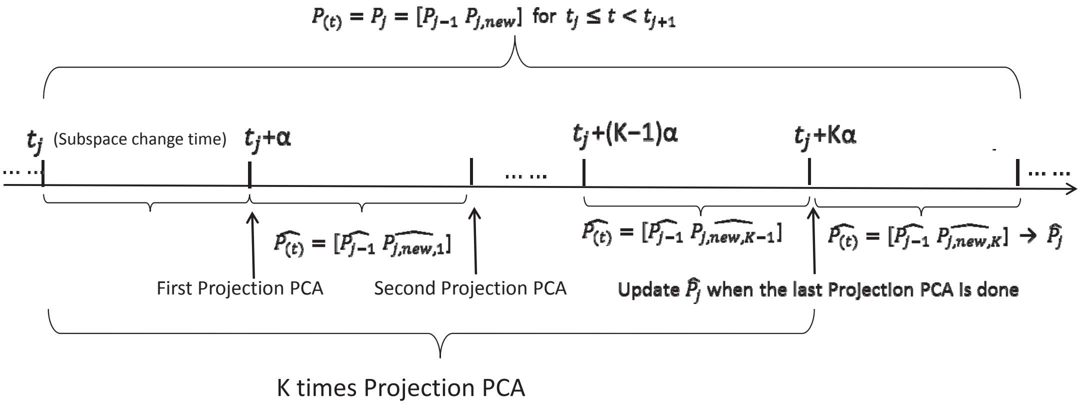

$P_{(t)} = P_j = [P_{j-1}\ P_{j,new}]$ for $t_j \le t < t_{j+1}$
t_j (Subspace change time)
$t_j+\alpha$
$t_j+(K-1)\alpha$
$t_j+K\alpha$
$\widehat{P_{(t)}} = [\widehat{P_{j-1}}\ \widehat{P_{j,new,1}}]$
$\widehat{P_{(t)}} = [\widehat{P_{j-1}}\ \widehat{P_{j,new,K-1}}]$
$\widehat{P_{(t)}} = [\widehat{P_{j-1}}\ \widehat{P_{j,new,K}}] \rightarrow \hat{P}_j$
First Projection PCA
Second Projection PCA
Update $\hat{P}_j$ when the last Projection PCA is done
K times Projection PCA

Fig. 2. The K projection PCA steps.

estimated $\hat{L}_t$'s are used to obtain new estimates of $P_{j,\text{new}}$ every α frames for a total of $K\alpha$ frames via a modification of the standard PCA procedure, which we call projection PCA (step 3). We illustrate the projection PCA algorithm in Fig 2. In the first projection PCA step, we get the first estimate of $P_{j,\text{new}}$, $\hat{P}_{j,\text{new},1}$. For the next α frame interval, $\hat{P}_{(t-1)} = [\hat{P}_{j-1}, \hat{P}_{j,\text{new},1}]$ and so $\Phi_{(t)} = \Phi_{j,1}$. Using this in the projected CS step reduces the projection noise, β_t, and hence the reconstruction error, e_t, for this interval, as long as $\gamma_{\text{new},k}$ increases slowly enough Smaller e_t makes the perturbation seen by the second projection PCA step even smaller, thus resulting in an improved second estimate $\hat{P}_{j,\text{new},2}$. Within K updates (K chosen as given in Theorem 4.2), under mild assumptions, it can be shown that both $||e_t||_2$ and the subspace error drop down to a constant times $\sqrt{\zeta}$. At this time, we update $\hat{P}_j$ as $\hat{P}_j = [\hat{P}_{j-1}, \hat{P}_{j,\text{new},K}]$.

The reason we need the projection PCA algorithm as given in step 3 of Algorithm 2 is because the error $e_t = \hat{L}_t - L_t = S_t - \hat{S}_t$ is correlated with L_t. We explain this point in detail in Appendix F. We also discuss there some other alternatives.

C. Performance Guarantees

We state the main result here first and then discuss it in the next two subsections. The proof outline is given in Sec V and the actual proof is given in Sec VI.

Theorem 4.2. *Consider Algorithm 2. Let $c := c_{mx}$ and $r := r_0 + (J-1)c$. Assume that L_t obeys the model given in Sec. III-A and there are a total of J change times. Assume also that the initial subspace estimate is accurate enough, i.e. $\|(I - \hat{P}_0 \hat{P}_0')P_0\| \le r_0 \zeta$, for a ζ that satisfies*

$$\zeta \le \min\left(\frac{10^{-4}}{r^2}, \frac{1.5 \times 10^{-4}}{r^2 f}, \frac{1}{r^3 \gamma_*^2}\right) \text{ where } f := \frac{\lambda^+}{\lambda^-}$$

If the following conditions hold:

1) *the algorithm parameters are set as $\xi = \xi_0(\zeta)$, $7\rho\xi \le \omega \le S_{\min} - 7\rho\xi$, $K = K(\zeta)$, $\alpha \ge \alpha_{add}(\zeta)$, where $\xi_0(\zeta), \rho, K(\zeta), \alpha_{add}(\zeta)$ are defined in Definition 5.1.*
2) *P_{j-1}, $P_{j,new}$, $D_{j,new,k} := (I - \hat{P}_{j-1}\hat{P}_{j-1}' - \hat{P}_{j,new,k}\hat{P}_{j,new,k}')P_{j,new}$ and $Q_{j,new,k} := (I - P_{j,new}P_{j,new}{}')\hat{P}_{j,new,k}$ have dense enough columns, i.e.*

$$\kappa_{2s}(P_{J-1}) \le \kappa_{2s,*}^+ := 0.3$$
$$\kappa_{2s}(P_{j,new}) \le \kappa_{2s,new}^+ := 0.15$$
$$\kappa_s(D_{j,new,k}) \le \kappa_s^+ := 0.15$$
$$\kappa_{2s}(Q_{j,new,k}) \le \tilde{\kappa}_{2s}^+ := 0.15$$

for all $j = 1, \dots, J-1$ and $k = 0, \dots, K$, with $\hat{P}_{j,new,0} = [.]$ (empty matrix).

3) *for a given value of $S_{\min}$, the subspace change is slow enough, i.e.*

$$\begin{aligned}
&\min_j (t_{j+1} - t_j) > K\alpha, \\
&\max_j \max_{t_j+(k-1)\alpha \le t < t_j+k\alpha} \|a_{t,new}\|_\infty \le \gamma_{new,k} := \min(1.2^{k-1}\gamma_{new}, \gamma_*), \ \textit{for all } k = 1, 2, \dots K, \\
&14\rho\xi_0(\zeta) \le S_{\min},
\end{aligned}$$

4) *the condition number of the covariance matrix of $a_{t,new}$ averaged over $t \in \mathcal{I}_{j,k}$, is bounded, i.e.*

$$g_{j,k} \le g^+ = \sqrt{2}$$

where $g_{j,k}$ is defined in Definition 5.1 and g^+ is defined in Definition 5.8.

then, with probability at least $(1 - n^{-10})$, at all times, t, all of the following hold:

1) *at all times, t,*

$$\hat{T}_t = T_t \ \ \textit{and}$$

$$\|e_t\|_2 = \|L_t - \hat{L}_t\|_2 = \|\hat{S}_t - S_t\|_2 \le 0.18\sqrt{c}\gamma_{new} + 1.2\sqrt{\zeta}(\sqrt{r} + 0.06\sqrt{c}).$$

2) *the subspace error $SE_{(t)} := \|(I - \hat{P}_{(t)}\hat{P}_{(t)}')P_{(t)}\|_2$ satisfies*

$$\begin{aligned}
SE_{(t)} &\le \begin{cases} (r_0 + (j-1)c)\zeta + 0.4c\zeta + 0.6^{k-1} & \textit{if } \ t \in \mathcal{I}_{j,k}, \ k = 1, 2 \dots K \\ (r_0 + jc)\zeta & \textit{if } \ t \in \mathcal{I}_{j,K+1} \end{cases} \\
&\le \begin{cases} 10^{-2}\sqrt{\zeta} + 0.6^{k-1} & \textit{if } \ t \in \mathcal{I}_{j,k}, \ k = 1, 2 \dots K \\ 10^{-2}\sqrt{\zeta} & \textit{if } \ t \in \mathcal{I}_{j,K+1} \end{cases}
\end{aligned}$$

3) *the error $e_t = \hat{S}_t - S_t = L_t - \hat{L}_t$ satisfies the following at various times*

$$\begin{aligned}
\|e_t\|_2 &\le \begin{cases} 0.18\sqrt{c}0.72^{k-1}\gamma_{new} + 1.2(\sqrt{r} + 0.06\sqrt{c})(r_0 + (j-1)c)\zeta\gamma_* & \textit{if } \ t \in \mathcal{I}_{j,k}, \ k = 1, 2 \dots K \\ 1.2(r_0 + jc)\zeta\sqrt{r}\gamma_* & \textit{if } \ t \in \mathcal{I}_{j,K+1} \end{cases} \\
&\le \begin{cases} 0.18\sqrt{c}0.72^{k-1}\gamma_{new} + 1.2(\sqrt{r} + 0.06\sqrt{c})\sqrt{\zeta} & \textit{if } \ t \in \mathcal{I}_{j,k}, \ k = 1, 2 \dots K \\ 1.2\sqrt{r}\sqrt{\zeta} & \textit{if } \ t \in \mathcal{I}_{j,K+1} \end{cases}
\end{aligned}$$

This result says the following. Consider Algorithm 2. Assume that the initial subspace error is small enough. If (a) the algorithm parameters are set appropriately; (b) the matrices defining the previous subspace, the newly added subspace, and the currently unestimated part of the newly added subspace are dense enough; (c) the subspace change is slow enough; and (d) the condition number of the average covariance matrix of $a_{t,\text{new}}$ is small enough, then, w.h.p., we will get exact support recovery at all times. Moreover, the sparse recovery error will always be bounded by $0.18\sqrt{c}\gamma_{\text{new}}$ plus a constant times $\sqrt{\zeta}$. Since ζ is very small, $\gamma_{\text{new}} \ll S_{\min}$, and c is also small, the normalized reconstruction error for recovering S_t will be small at all times.

In the second conclusion, we bound the subspace estimation error, $\text{SE}_{(t)}$. When a subspace change occurs, this error is initially bounded by one. The above result shows that, w.h.p., with each projection PCA step, this error decays exponentially and falls below $0.01\sqrt{\zeta}$ within K projection PCA steps. The third conclusion shows that, with each projection PCA step, w.h.p., the sparse recovery error as well as the error in recovering L_t also decay in a similar fashion.

D. Discussion

First consider the choices of α and of K. Notice that $K = K(\zeta)$ is larger if ζ is smaller. Also, α_{add} is inversely proportional to ζ. Thus, if we want to achieve a smaller lowest error level, ζ, we need to compute projection PCA over larger durations α and we need more number of projection PCA steps K.

Now consider the assumptions made on the model. We assume slow subspace change, i.e. the delay between change times is large enough, $\|a_{t,\text{new}}\|_\infty$ is initially below γ_{new} and increases gradually, and $14\rho\xi_0 \le S_{\min}$ which holds if c_{mx} and γ_{new} are small enough. Small c_{mx}, small initial $a_{t,\text{new}}$ (i.e. small γ_{new}) and its gradual increase are verified for real video data in Sec.

X-A. As explained there, one cannot estimate the delay between change times with just one video sequence of a particular type (need an ensemble) and hence the first assumption cannot be verified.

We also assume that condition number of the average covariance matrix of $a_{t,\text{new}}$, is not too large. This is an assumption made for simplicity. It can be removed if the newly added eigenvalues can be separated into clusters so that the condition number of each cluster is small (even though the overall condition number is large). This latter assumption is usually true for real data. Under this assumption, we can use the cluster projection PCA approach described in [42] for ReProCS with deletion. The idea is to use projection PCA to first only recover the eigenvectors corresponding to the cluster with the largest eigenvalues; then project perpendicular to these and $\hat{P}_{j-1}$ to recover the eigenvectors for the next cluster and so on.

Other than these, we assume the independence of a_t's over time. This is done so that we can use the matrix Hoeffding inequality [28, Theorem 1.3] to obtain high probability bounds on the terms in the subspace error bound. In simulations, and in experiments with real data, we are able to also deal with correlated a_t's. In future work, it should be possible to replace independence by a milder assumption, e.g. a random walk model on the a_t's. In that case, at $t_j + k\alpha - 1$, one would compute the eigenvectors of $(1/\alpha)\sum_{t\in\mathcal{I}_{j,k}} \Phi_{j,0}(\hat{L}_t - \hat{L}_{t-1})(\hat{L}_t - \hat{L}_{t-1})'\Phi_{j,0}'$. Moreover, one may need to use the matrix Azuma inequality [28, Theorem 7.1] instead of Hoeffding to bound the terms in the subspace error bound.

Finally, we assume denseness of P_{j-1} and $P_{j,\text{new}}$ as well as of $D_{j,\text{new},k}$ and $Q_{j,\text{new},k}$ in condition 2. As we explain in Sec IV-E, denseness of P_{j-1} and $P_{j,\text{new}}$ is a subset of the assumptions made in earlier works [5]. It is valid for the video application because typically the changes of the background sequence are global, e.g. due to illumination variation affecting the entire image or due to textural changes such as water motion or tree leaves' motion etc. Thus, most columns of the matrix $\mathcal{L}_t$ are dense and consequently the same is true for any basis matrix for span$(\mathcal{L}_t)$. Now consider denseness of $D_{j,\text{new},k}$ whose columns span the currently unestimated part of the newly added subspace. Our proof actually only needs $\|{I_{T_t}}'D_{j,\text{new},k}\|_2/\|D_{j,\text{new},k}\|_2$ to be small at every projection PCA time, $t = t_j + k\alpha - 1$. We attempted to verify this in simulations done with a dense P_j and $P_{j,\text{new}}$. Except for the case of exactly constant support of S_t, in all other cases (including the case of very gradual support change, e.g. the models considered in Sec X-C), this ratio was small for most projection PCA times. We also saw that even if at a few projection PCA times, this ratio was close to one, that just meant that, at those times, the subspace error remained roughly equal to that at the previous time. As a result, a larger K was required for the subspace error to become small enough. It did not mean that the algorithm became unstable. It should be possible to use a similar idea to modify our result as well. An analogous discussion applies also to $Q_{j,\text{new},k}$. In fact denseness of $Q_{j,\text{new},k}$ is not essential, it is possible to prove a slightly more complicated version of Theorem 4.2 without assuming denseness of $Q_{j,\text{new},k}$.

E. Discussion w.r.t. the PCP result

First of all, as mentioned earlier, we solve a recursive version of the robust PCA problem where as PCP in [5] solves a batch one. Also, the proof techniques used are very different. Hence it is impossible to do a direct comparison of the two results. Also, the PCP algorithm assumes no model knowledge, whereas our algorithm does assume knowledge of model parameters. Of course, in Sec X-B, we have explained how to set the parameters in practice when the model is not known.

The first key difference between our result and that of PCP [5] is as follows. The result of [5] assumes that any element of the $n \times t$ matrix $\mathcal{S}_t$ is nonzero w.p. ϱ, and zero w.p. $1 - \varrho$, independent of all others (in particular, this means that the support sets of the different S_t's are independent over time). This assumption ensures that w.h.p. $\mathcal{S}_t$ is sparse but full rank and hence ensures that it can be separated from $\mathcal{L}_t$ which is low rank but dense. As explained earlier, the assumption of independent support sets of S_t is not valid for real video data where the foreground objects usually move in a highly correlated fashion over time. On the other hand, our result for ReProCS does not put any such assumption on the support sets of the S_t's. In simulations, we show examples where the support is generated in a highly correlated fashion thus resulting in a low rank and sparse $\mathcal{S}_t$ and ReProCS is still able to recover S_t accurately. The reason it can do this is because it assumes accurate knowledge of the subspace spanned by the first few columns of $\mathcal{L}_t$ and it assumes slow subspace change. However, ReProCS does need denseness of $D_{j,\text{new},k}$ and in simulations, we observe that this reduces when the support of S_t changes very infrequently.

Next let us compare the denseness assumptions. Let $\mathcal{L}_t = U\Sigma V'$ be its SVD. Then, for $t \in [t_j, t_{j+1}-1]$, $U = [P_{j-1}, P_{j,\text{new}}]$ and the a_t's are the various columns of the matrix $\Sigma V'$. Thus $V = [a_1, a_2 \dots a_t]'\Sigma^{-1}$. PCP [5] assumes denseness of U and of V: it requires $\kappa_1(U) \leq \sqrt{\mu r/n}$ and $\kappa_1(V) \leq \sqrt{\mu r/n}$ for a constant $\mu \geq 1$. Moreover, it also requires $\|UV'\|_{\max} \leq \sqrt{\mu r}/n$. Here $\|B\|_{\max} := \max_{i,j} |(B)_{i,j}|$. This last assumption is a particularly strong one. On the other hand, our denseness assumptions are on P_{j-1} and $P_{j,\text{new}}$ which are sub-matrices of U, and on $D_{j,\text{new},k}$ whose columns span the currently unestimated part of span$(P_{j,\text{new}})$. We do not need denseness of V and we do need a bound on $\|UV'\|_{\max}$. However, an additional assumption that we do need is the independence of a_t's over time. We have discussed above in Sec IV-D how this can possibly be relaxed.

V. Definitions and Proof Outline for Theorem 4.2

A. Definitions

A few quantities are already defined in the model (Sec III-A), Definition 4.1, Algorithm 2, and Theorem 4.2. Here we define more quantities needed for the proofs.

Definition 5.1. *We define here the parameters used in Theorem 4.2.*

1) *Define* $K(\zeta) := \left\lceil \frac{\log(0.6c\zeta)}{\log 0.6} \right\rceil$
2) *Define* $\xi_0(\zeta) := \sqrt{c}\gamma_{new} + \sqrt{\zeta}(\sqrt{r} + \sqrt{c})$
3) *Define* $\rho := \max_t\{\kappa_1(\hat{S}_{t,cs} - S_t)\}$. *Notice that* $\rho \leq 1$.
4) *Define*

$$\alpha_{add} := \left\lceil (\log 6KJ + 11\log n)\frac{8 \cdot 24^2}{\zeta^2(\lambda^-)^2} \max\left(\min(1.2^{4K}\gamma_{new}^4, \gamma_*^4), \frac{16}{c^2}, 4(0.186\gamma_{new}^2 + 0.0034\gamma_{new} + 2.3)^2\right)\right\rceil$$

In words, α_{add} *is the smallest value of the number of data points,* α, *needed for one projection PCA step to ensure that Theorem 4.2 holds w.p. at least* $(1 - n^{-10})$.

5) *Define the condition number of* $Cov(a_{t,new})$ *averaged over* $t \in \mathcal{I}_{j,k}$ *as*

$$g_{j,k} := \frac{{\lambda_{j,new,k}}^+}{{\lambda_{j,new,k}}^-} \text{ where}$$

$${\lambda_{j,new,k}}^+ := \lambda_{\max}\left(\frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}}(\Lambda_t)_{new}\right), \quad {\lambda_{j,new,k}}^- := \lambda_{\min}\left(\frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}}(\Lambda_t)_{new}\right).$$

Notice that $\lambda^- \leq {\lambda_{j,new,k}}^- \leq {\lambda_{j,new,k}}^+ \leq \lambda^+$ *and thus* $g_{j,k} \leq f = \lambda^+/\lambda^-$. *Recall that* $\Lambda_t = Cov[a_t] = \mathbf{E}(a_t a_t{}')$, $(\Lambda_t)_{new} = \mathbf{E}(a_{t,new}a'_{t,new})$, $\lambda^- = \min_t \lambda_{\min}(\Lambda_t)$ *and* $\lambda^+ = \max_t \lambda_{\max}(\Lambda_t)$.

Definition 5.2. *We define the noise seen by the sparse recovery step at time* t *as*

$$\beta_t := \|(I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)})L_t\|_2.$$

Also define the reconstruction error of S_t *as*

$$e_t := \hat{S}_t - S_t.$$

Here $\hat{S}_t$ *is the final estimate of* S_t *after the LS step in Algorithm 2. Notice that* e_t *also satisfies* $e_t = L_t - \hat{L}_t$.

Definition 5.3. *We define the subspace estimation errors as follows. Recall that* $\hat{P}_{j,new,0} = [.]$ *(empty matrix).*

$$\begin{aligned} SE_{(t)} &:= \|(I - \hat{P}_{(t)}\hat{P}'_{(t)})P_{(t)}\|_2, \\ \zeta_{j,*} &:= \|(I - \hat{P}_{j-1}\hat{P}'_{j-1})P_{j-1}\|_2 \\ \zeta_{j,k} &:= \|(I - \hat{P}_{j-1}\hat{P}'_{j-1} - \hat{P}_{j,new,k}\hat{P}'_{j,new,k})P_{j,new}\|_2 \end{aligned}$$

Remark 5.4. *Recall from the model given in Sec III-A and from Algorithm 2 that*

1) $\hat{P}_{j,new,k}$ *is orthogonal to* $\hat{P}_{j-1}$, *i.e.* $\hat{P}'_{j,new,k}\hat{P}_{j-1} = 0$
2) $\hat{P}_{j-1} := [\hat{P}_0, \hat{P}_{1,new,K}, \dots \hat{P}_{j-1,new,K}]$ *and* $P_{j-1} := [P_0, P_{1,new}, \dots P_{j-1,new}]$

3) *for* $t \in \mathcal{I}_{j,k+1}$, $\hat{P}_{(t)} = [\hat{P}_{j-1}, \hat{P}_{j,new,k}]$ *and* $P_{(t)} = P_j = [P_{j-1}, P_{j,new}]$.
4) $\Phi_{(t)} := I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)}$

From Definition 5.3 and the above, it is easy to see that

1) $\zeta_{j,*} \leq \zeta_{1,*} + \sum_{j'=1}^{j-1} \zeta_{j',K}$
2) $SE_{(t)} \leq \zeta_{j,*} + \zeta_{j,k} \leq \zeta_{1,*} + \sum_{j'=1}^{j-1} \zeta_{j',K} + \zeta_{j,k}$ *for* $t \in \mathcal{I}_{j,k+1}$.

Definition 5.5. *Define the following*

1) $\Phi_{j,k}$, $\Phi_{j,0}$ *and* ϕ_k
   a) $\Phi_{j,k} := I - \hat{P}_{j-1}\hat{P}'_{j-1} - \hat{P}_{j,new,k}\hat{P}'_{j,new,k}$ *is the CS matrix for* $t \in \mathcal{I}_{j,k+1}$, *i.e.* $\Phi_{(t)} = \Phi_{j,k}$ *for this duration.*
   b) $\Phi_{j,0} := I - \hat{P}_{j-1}\hat{P}'_{j-1}$ *is the CS matrix for* $t \in \mathcal{I}_{j,1}$, *i.e.* $\Phi_{(t)} = \Phi_{j,0}$ *for this duration.* $\Phi_{j,0}$ *is also the matrix used in all of the projection PCA steps for* $t \in [t_j, t_{j+1} - 1]$.
   c) $\phi_k := \max_j \max_{T:|T|\leq s} \|((\Phi_{j,k})_T{}'(\Phi_{j,k})_T)^{-1}\|_2$. *It is easy to see that* $\phi_k \leq \frac{1}{1-\max_j \delta_s(\Phi_{j,k})}$ *[30].*
2) $D_{j,new,k}$, $D_{j,new}$ *and* $D_{j,*}$
   a) $D_{j,new,k} := \Phi_{j,k}P_{j,new}$. *span*$(D_{j,new,k})$ *is the unestimated part of the newly added subspace for any* $t \in \mathcal{I}_{j,k+1}$.
   b) $D_{j,new} := D_{j,new,0} = \Phi_{j,0}P_{j,new}$. *span*$(D_{j,new})$ *is interpreted similarly for any* $t \in \mathcal{I}_{j,1}$.
   c) $D_{j,*,k} := \Phi_{j,k}P_{j-1}$. *span*$(D_{j,*,k})$ *is the unestimated part of the existing subspace for any* $t \in \mathcal{I}_{j,k}$
   d) $D_{j,*} := D_{j,*,0} = \Phi_{j,0}P_{j-1}$. *span*$(D_{j,*,k})$ *is interpreted similarly for any* $t \in \mathcal{I}_{j,1}$
   e) *Notice that* $\zeta_{j,0} = \|D_{j,new}\|_2$, $\zeta_{j,k} = \|D_{j,new,k}\|_2$, $\zeta_{j,*} = \|D_{j,*}\|_2$. *Also, clearly,* $\|D_{j,*,k}\|_2 \leq \zeta_{j,*}$.

Definition 5.6.

1) *Let* $D_{j,new} \overset{QR}{=} E_{j,new}R_{j,new}$ *denote its QR decomposition. Here* $E_{j,new}$ *is a basis matrix and* $R_{j,new}$ *is upper triangular.*
2) *Let* $E_{j,new,\perp}$ *be a basis matrix for the orthogonal complement of span*$(E_{j,new}) =$ *span*$(D_{j,new})$. *To be precise,* $E_{j,new,\perp}$ *is a* $n \times (n - c_{j,new})$ *basis matrix that satisfies* $E'_{j,new,\perp}E_{j,new} = 0$.
3) *Using* $E_{j,new}$ *and* $E_{j,new,\perp}$, *define* $A_{j,k}$, $A_{j,k,\perp}$, $H_{j,k}$, $H_{j,k,\perp}$ *and* $B_{j,k}$ *as*
$$A_{j,k} := \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new}{}'\Phi_{j,0}L_tL_t{}'\Phi_{j,0}E_{j,new}$$
$$A_{j,k,\perp} := \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new,\perp}{}'\Phi_{j,0}L_tL_t{}'\Phi_{j,0}E_{j,new,\perp}$$
$$H_{j,k} := \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new}{}'\Phi_{j,0}(e_te_t{}' - L_te_t{}' - e_tL_t{}')\Phi_{j,0}E_{j,new}$$
$$H_{j,k,\perp} := \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new,\perp}{}'\Phi_{j,0}(e_te_t{}' - L_te_t{}' - e_tL_t{}')\Phi_{j,0}E_{j,new,\perp}$$
$$B_{j,k} := \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new,\perp}{}'\Phi_{j,0}\hat{L}_t\hat{L}'_t\Phi_{j,0}E_{j,new} = \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} E_{j,new,\perp}{}'\Phi_{j,0}(L_t - e_t)(L_t{}' - e_t{}')\Phi_{j,0}E_{j,new}$$
4) *Define*
$$\mathcal{A}_{j,k} := \begin{bmatrix} E_{j,new} \; E_{j,new,\perp} \end{bmatrix} \begin{bmatrix} A_{j,k} & 0 \\ 0 & A_{j,k,\perp} \end{bmatrix} \begin{bmatrix} E_{j,new}{}' \\ E_{j,new,\perp}{}' \end{bmatrix}$$
$$\mathcal{H}_{j,k} := \begin{bmatrix} E_{j,new} \; E_{j,new,\perp} \end{bmatrix} \begin{bmatrix} H_{j,k} & B_{j,k}{}' \\ B_{j,k} & H_{j,k,\perp} \end{bmatrix} \begin{bmatrix} E_{j,new}{}' \\ E_{j,new,\perp}{}' \end{bmatrix}$$
5) *From the above, it is easy to see that*
$$\mathcal{A}_{j,k} + \mathcal{H}_{j,k} = \frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} \Phi_{j,0}\hat{L}_t\hat{L}'_t\Phi_{j,0}.$$
6) *Recall from Algorithm 2 that* $\mathcal{A}_{j,k} + \mathcal{H}_{j,k} \overset{EVD}{=} \begin{bmatrix} \hat{P}_{j,new,k} \; \hat{P}_{j,new,k,\perp} \end{bmatrix} \begin{bmatrix} \Lambda_k & 0 \\ 0 & \Lambda_{k,\perp} \end{bmatrix} \begin{bmatrix} \hat{P}'_{j,new,k} \\ \hat{P}'_{j,new,k,\perp} \end{bmatrix}$ *is the EVD of* $\mathcal{A}_{j,k} + \mathcal{H}_{j,k}$. *Here* $\hat{P}_{j,new,k}$ *is a* $n \times c_{j,new}$ *basis matrix.*

7) *Using the above, $\mathcal{A}_{j,k} + \mathcal{H}_{j,k}$ can be decomposed in two ways as follows:*

$$\begin{aligned}
\mathcal{A}_{j,k} + \mathcal{H}_{j,k} &= \begin{bmatrix} \hat{P}_{j,new,k} \, \hat{P}_{j,new,k,\perp} \end{bmatrix} \begin{bmatrix} \Lambda_k & 0 \\ 0 & \Lambda_{k,\perp} \end{bmatrix} \begin{bmatrix} \hat{P}'_{j,new,k} \\ \hat{P}'_{j,new,k,\perp} \end{bmatrix} \\
&= \begin{bmatrix} E_{j,new} \, E_{j,new,\perp} \end{bmatrix} \begin{bmatrix} A_{j,k} + H_{j,k} & B'_{j,k} \\ B_{j,k} & A_{j,k,\perp} + H_{j,k,\perp} \end{bmatrix} \begin{bmatrix} {E_{j,new}}' \\ {E_{j,new,\perp}}' \end{bmatrix}
\end{aligned}$$

Remark 5.7. *Thus, from the above definition, $\mathcal{H}_{j,k} = \frac{1}{\alpha}[\Phi_0 \sum_t (-L_t e_t' - e_t L_t' + e_t e_t')\Phi_0 + F + F']$ where $F := E_{new,\perp} E'_{new,\perp} \Phi_0 \sum_t L_t L_t' \Phi_0 E_{new} E'_{new} = E_{new,\perp} E'_{new,\perp} (D_{*,k-1} a_{t,*})(D_{*,k-1} a_{t,*} + D_{new,k-1} a_{t,new})' E_{new} E'_{new}$. Since $\mathbf{E}[a_{t,*} a'_{t,new}] = 0$, $\|\frac{1}{\alpha} F\|_2 \lesssim r^2 \zeta^2 \lambda^+$ w.h.p. Recall $\lesssim$ means (in an informal sense) that the RHS contains the dominant terms in the bound.*

Definition 5.8. *In the sequel, we let*

1) $r := r_0 + (J-1) c_{mx}$ *and* $c := c_{mx} = \max_j c_{j,new}$,
2) $\kappa_{s,*} := \max_j \kappa_s(P_{j-1})$, $\kappa_{s,new} := \max_j \kappa_s(P_{j,new})$, $\kappa_{s,k} := \max_j \kappa_s(D_{j,new,k})$, $\tilde{\kappa}_{s,k} := \max_j \kappa_s((I - P_{j,new} {P_{j,new}}') \hat{P}_{j,new,k})$, $g_k := \max_j g_{j,k}$,
3) $\kappa^+_{2s,*} := 0.3$, $\kappa^+_{2s,new} := 0.15$, $\kappa^+_s := 0.15$, $\tilde{\kappa}^+_{2s} := 0.15$ *and* $g^+ := \sqrt{2}$ *are the upper bounds assumed in Theorem 4.2 on* $\max_j \kappa_{2s}(P_j)$, $\max_j \kappa_{2s}(P_{j,new})$, $\max_j \max_k \kappa_s(D_{j,new,k})$, $\max_j \kappa_{2s}(Q_{j,new,k})$ *and* $\max_j \max_k g_{j,k}$ *respectively.*
4) $\phi^+ := 1.1735$ *We see later that this is an upperbound on ϕ_k under the assumptions of Theorem 4.2.*
5) $\gamma_{new,k} := \min(1.2^{k-1} \gamma_{new}, \gamma_*)$
   (recall that the theorem assumes $\max_j \max_{t \in \mathcal{I}_{j,k}} \|a_{t,new}\|_\infty \leq \gamma_{new,k}$)
6) $P_{j,*} := P_{j-1}$ *and* $\hat{P}_{j,*} := \hat{P}_{j-1}$ *(see Remark 5.9).*

Remark 5.9. *Notice that the subscript j always appears as the first subscript, while k is the last one. At many places in this paper, we remove the subscript j for simplicity. Whenever there is only one subscript, it refers to the value of k, e.g., Φ_0 refers to $\Phi_{j,0}$, $\hat{P}_{new,k}$ refers to $\hat{P}_{j,new,k}$. Also, $P_* := P_{j-1}$ and $\hat{P}_* := \hat{P}_{j-1}$.*

Definition 5.10. *Define the following:*

1) $\zeta_*^+ := r\zeta$ *(We note that $\zeta_*^+ = (r_0 + (j-1)c)\zeta$ will also work.)*
2) *Define the sequence $\{{\zeta_k}^+\}_{k=0,1,2,\cdots K}$ recursively as follows*

$$\begin{aligned}
\zeta_0^+ &:= 1 \\
\zeta_k^+ &:= \frac{b + 0.125c\zeta}{1 - (\zeta_*^+)^2 - (\zeta_*^+)^2 f - 0.125c\zeta - b} \quad \text{for } k \geq 1,
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
b &:= C\kappa_s^+ g^+ \zeta_{k-1}^+ + \tilde{C}(\kappa_s^+)^2 g^+ (\zeta_{k-1}^+)^2 + C' f (\zeta_*^+)^2 \\
C &:= \frac{2\kappa_s^+ \phi^+}{\sqrt{1 - (\zeta_*^+)^2}} + \phi^+, \\
C' &:= (\phi^+)^2 + \frac{2\phi^+}{\sqrt{1 - (\zeta_*^+)^2}} + 1 + \phi^+ + \frac{\kappa_s^+ \phi^+}{\sqrt{1 - (\zeta_*^+)^2}} + \frac{\kappa_s^+ (\phi^+)^2}{\sqrt{1 - (\zeta_*^+)^2}}, \\
\tilde{C} &:= (\phi^+)^2 + \frac{\kappa_s^+ (\phi^+)^2}{\sqrt{1 - (\zeta_*^+)^2}}.
\end{aligned}$$

As we will see, ζ_*^+ and ζ_k^+ are the high probability upper bounds on $\zeta_{j,*}$ and $\zeta_{j,k}$ (defined in Definition 5.3) under the assumptions of Theorem 4.2.

Definition 5.11. *Define the random variable $X_{j,k} := \{a_1, a_2, \cdots, a_{t_j + k\alpha - 1}\}$.*

Recall that the a_t's are mutually independent over t.

Definition 5.12. *Define the set $\check{\Gamma}_{j,k}$ as follows:*

$$\check{\Gamma}_{j,k} := \{X_{j,k} : \zeta_{j,k} \leq \zeta_k^+ \text{ and } \hat{T}_t = T_t \text{ and } e_t \text{ satisfies } (6) \text{ for all } t \in \mathcal{I}_{j,k}\}$$
$$\check{\Gamma}_{j,K+1} := \{X_{j+1,0} : \hat{T}_t = T_t \text{ and } e_t \text{ satisfies } (6) \text{ for all } t \in \mathcal{I}_{j,K+1}\}$$

Definition 5.13. *Recursively define the sets $\Gamma_{j,k}$ as follows:*

$$\Gamma_{1,0} := \{X_{1,0} : \zeta_{1,*} \leq r\zeta \text{ and } \hat{T}_t = T_t \text{ and } e_t \text{ satisfies } (6) \text{ for all } t \in [t_{\text{train}+1} : t_1 - 1]\}$$
$$\Gamma_{j,k} := \Gamma_{j,k-1} \cap \check{\Gamma}_{j,k} \quad k = 1, 2, \dots, K, j = 1, 2, \dots, J$$
$$\Gamma_{j+1,0} := \Gamma_{j,K} \cap \check{\Gamma}_{j,K+1} \quad j = 1, 2, \dots, J$$

B. Proof Outline for Theorem 4.2

The proof of Theorem 4.2 essentially follows from two main lemmas, 6.1 and 6.2. Lemma 6.1 gives an exponentially decaying upper bound on ζ_k^+ defined in Definition 5.10. ζ_k^+ will be shown to be a high probability upper bound for ζ_k under the assumptions of the Theorem. Lemma 6.2 says that conditioned on $X_{j,k-1} \in \Gamma_{j,k-1}$, $X_{j,k}$ will be in $\Gamma_{j,k}$ w.h.p.. In words this says that if, during the time interval $\mathcal{I}_{j,k-1}$, the algorithm has worked well (recovered the support of S_t exactly and recovered the background subspace with subspace recovery error below $\zeta_{k-1}^+ + \zeta_*^+$), then it will also work well in $\mathcal{I}_{j,k}$ w.h.p.. The proof of Lemma 6.2 requires two lemmas: one for the projected CS step and one for the projection PCA step of the algorithm. These are lemmas 8.1 and 8.2. The proof Lemma 8.1 follows using Lemmas 6.1, 3.2, 2.10, the CS error bound (Theorem 2.5), and some straightforward steps. The proof of Lemma 8.2 is longer and uses a lemma based on the $\sin\theta$ and Weyl theorems (Theorems 2.7 and 2.8) to get a bound on ζ_k. From here we use the matrix Hoeffding inequalities (Corollaries 2.14 and 2.15) to bound each of the terms in the bound on ζ_k to finally show that, conditioned on $\Gamma_{j,k-1}^e$ $\zeta_k \leq \zeta_k^+$ w.h.p.. These are Lemmas 8.6 and 8.7.

VI. Main Lemmas and Proof of Theorem 4.2

Recall that when there is only one subscript, it refers to the value of k (i.e. $\zeta_k = \zeta_{j,k}$).

Lemma 6.1 (Exponential decay of ζ_k^+)**.** *Assume that the bounds on ζ from Theorem 4.2 hold. Define the sequence ζ_k^+ as in Definition 5.10. Then*

1) $\zeta_0^+ = 1$ *and* $\zeta_k^+ \leq 0.6^k + 0.4c\zeta$ *for all* $k = 1, 2, \dots, K$,
2) *the denominator of* ζ_k^+ *is positive for all* $k = 1, 2, \dots, K$.

We will prove this lemma in Section VII.

Lemma 6.2. *Assume that all the conditions of Theorem 4.2 hold. Also assume that* $\mathbf{P}(\Gamma_{j,k-1}^e) > 0$. *Then*

$$\mathbf{P}(\Gamma_{j,k}^e | \Gamma_{j,k-1}^e) \geq p_k(\alpha, \zeta) \geq p_K(\alpha, \zeta) \quad \text{for all } k = 1, 2, \dots, K,$$

where $p_k(\alpha, \zeta)$ is defined in equation (9).

Remark 6.3. *Under the assumptions of Theorem 4.2, it is easy to see that the following holds. For any $k = 1, 2 \dots K$, $\Gamma_{j,k}^e$ implies that $\zeta_{j,*} \leq \zeta_*^+$*

From the definition of $\Gamma_{j,k}^e$, $\zeta_{j',K} \leq \zeta_K^+$ for all $j' \leq j-1$. By Lemma 6.1 and the definition of K in Definition 5.1, $\zeta_K^+ \leq 0.6^K + 0.4c\zeta \leq c\zeta$ for all $j' \leq j-1$. Using Remark 5.4, $\zeta_{j,*} \leq \zeta_1^* + \sum_{j'=1}^{j-1} \zeta_{j',K} \leq r_0\zeta + (j-1)c\zeta \leq \zeta_*^+$.

Proof of Theorem 4.2:

The theorem is a direct consequence of Lemmas 6.1, 6.2, and Lemma 2.12.

Notice that $\Gamma_{j,0}^e \supseteq \Gamma_{j,1}^e \supseteq \cdots \supseteq \Gamma_{j,K,0}^e \supseteq \Gamma_{j+1,0}^e$. Thus, by Lemma 2.12, $\mathbf{P}(\Gamma_{j+1,0}^e | \Gamma_{j,0}^e) = \mathbf{P}(\Gamma_{j+1,0}^e | \Gamma_{j,K}^e) \prod_{k=1}^{K} \mathbf{P}(\Gamma_{j,k}^e | \Gamma_{j,k-1}^e)$ and $\mathbf{P}(\Gamma_{J+1,0} | \Gamma_{1,0}) = \prod_{j=1}^{J} \mathbf{P}(\Gamma_{j+1,0}^e | \Gamma_{j,0}^e)$.

Using Lemmas 6.2, and the fact that $p_k(\alpha,\zeta) \geq p_K(\alpha,\zeta)$ (see their respective definitions in Lemma 8.7 and equation (9)), we get $\mathbf{P}(\Gamma^e_{J+1,0}|\Gamma_{1,0,0}) \geq p_K(\alpha,\zeta)^{KJ}$. Also, $\mathbf{P}(\Gamma^e_{1,0}) = 1$. This follows by the assumption on $\hat{P}_0$ and Lemma 8.1. Thus, $\mathbf{P}(\Gamma^e_{J+1,0}) \geq p_K(\alpha,\zeta)^{KJ}$.

Using the definition of α_{add} we get that $\mathbf{P}(\Gamma^e_{J+1,0}) \geq p_K(\alpha,\zeta)^{KJ} \geq 1 - n^{-10}$ whenever $\alpha \geq \alpha_{\text{add}}$.

The event $\Gamma^e_{J+1,0}$ implies that $\hat{T}_t = T_t$ and e_t satisfies (6) for all $t < t_{J+1}$. Using Remarks 5.4 and 6.3, $\Gamma^e_{J+1,0}$ implies that all the bounds on the subspace error hold. Using these, $\|a_{t,\text{new}}\|_2 \leq \sqrt{c}\gamma_{\text{new},k}$, and $\|a_t\|_2 \leq \sqrt{r}\gamma_*$, $\Gamma^e_{J+1,0}$ implies that all the bounds on $\|e_t\|_2$ hold (the bounds are obtained in Lemma 8.1).

Thus, all conclusions of the the result hold w.p. at least $1 - n^{-10}$. ■

VII. Proof of Lemma 6.1

Proof: First recall the definition of ζ_k^+ (Definition 5.10). Recall from Definition 5.8 that $\kappa_s^+ := 0.15$, $\phi^+ := 1.1735$, and $g^+ := \sqrt{2}$. So we can make these substitutions directly. Notice that ζ_k^+ is an increasing function of ζ_*^+, ζ, c, and f. Therefore we can use upper bounds on each of these quantities to get an upper bound on ζ_k^+. From the definition of ζ in Theorem 4.2 and $\zeta_*^+ := r\zeta$ we get

- $\zeta_*^+ \leq 10^{-4}$
- $\zeta_*^+ f \leq 1.5 \times 10^{-4}$
- $c\zeta \leq 10^{-4}$
- $\dfrac{\zeta_*^+}{c\zeta} = \dfrac{r\zeta}{c\zeta} = \dfrac{r}{c} \leq r$ (We assume that $c \geq 1$. Recall that c is the upper bound on the number of new directions added)
- $\zeta_*^+ fr = r^2 f\zeta \leq 1.5 \times 10^{-4}$

First we prove by induction that $\zeta_k^+ \leq \zeta_{k-1}^+ \leq 0.6$ for all $k \geq 1$. Notice that $\zeta_0^+ = 1$ by definition.

- Base case ($k = 1$): Using the above bounds we get that $\zeta_1^+ < 0.5985 < 1 = \zeta_0^+$.
- For the induction step, assume that $\zeta_{k-1}^+ \leq \zeta_{k-2}^+$. Then because ζ_k^+ is increasing in ζ_{k-1}^+ we get that $\zeta_k^+ = f_{inc}(\zeta_{k-1}^+) \leq f_{inc}(\zeta_{k-2}^+) = \zeta_{k-1}^+$.

1) To prove the first claim, first rewrite ζ_k^+ as

$$\zeta_k^+ = \zeta_{k-1}^+ \frac{C\kappa_s^+ g^+ + \tilde{C}(\kappa_s^+)^2 g^+(\zeta_{k-1}^+)}{1 - (\zeta_*^+)^2 - (\zeta_*^+)^2 f - 0.125c\zeta - b} + c\zeta \frac{C(\zeta_*^+ f)\frac{(\zeta_*^+)}{c\zeta} + .125}{1 - (\zeta_*^+)^2 - (\zeta_*^+)^2 f - 0.125c\zeta - b}$$

Where $C, \tilde{C}$, and b are as in Definition 5.10. Using the above bounds including $\zeta_{k-1}^+ \leq .6$ we get that

$$\zeta_k^+ \leq \zeta_{k-1}^+(0.6) + c\zeta(0.16) = \zeta_0^+(0.6)^k + \sum_{i=0}^{k-1}(0.6)^k(0.16)c\zeta \leq \zeta_0^+(0.6)^k + \sum_{i=0}^{\infty}(0.6)^k(0.16)c\zeta \leq 0.6^k + 0.4c\zeta$$

2) To see that the denominator is positive, observe that the denominator is decreasing in all of its arguments: $\zeta_*^+, \zeta_*^+ f, c\zeta$, and b. Using the same upper bounds as before, we get that the denominator is greater than or equal to $0.78 > 0$.

■

VIII. Proof of Lemma 6.2

The proof of Lemma 6.2 follows from two lemmas. The first is the final conclusion for the projected CS step for $t \in \mathcal{I}_{j,k}$. The second is the final conclusion for one projection PCA (i.e.) for $t \in \mathcal{I}_{j,k}$. We will state the two lemmas first and then proceed to prove them in order.

Lemma 8.1 (Projected Compressed Sensing Lemma)**.** *Assume that all conditions of Theorem 4.2 hold.*

1) *For all $t \in \mathcal{I}_{j,k}$, for any $k = 1, 2, \ldots K$, if $X_{j,k-1} \in \Gamma_{j,k-1}$,*
   a) *the projection noise β_t satisfies $\|\beta_t\|_2 \leq \zeta_{k-1}^+\sqrt{c}\gamma_{new,k} + \zeta_*^+\sqrt{r}\gamma_* \leq \sqrt{c}0.72^{k-1}\gamma_{new} + 1.06\sqrt{\zeta} \leq \xi_0$.*
   b) *the CS error satisfies $\|\hat{S}_{t,cs} - S_t\|_2 \leq 7\xi_0$.*
   c) $\hat{T}_t = T_t$

d) e_t *satisfies* (6) *and* $\|e_t\|_2 \leq \phi^+[\kappa_s^+ \zeta_{k-1}^+ \sqrt{c}\gamma_{new,k} + \zeta_*^+ \sqrt{r}\gamma_*] \leq 0.18 \cdot 0.72^{k-1}\sqrt{c}\gamma_{new} + 1.17 \cdot 1.06\sqrt{\zeta}$. *Recall that* (6) *is*

$$I_{T_t}(\Phi_{(t)})_{T_t}{}^\dagger \beta_t = I_{T_t}[(\Phi_{(t)})'_{T_t}(\Phi_{(t)})_{T_t}]^{-1} I_{T_t}{}'\Phi_{(t)} L_t$$

2) *For all* $k = 1, 2, \ldots K$, $\mathbf{P}(\hat{T}_t = T_t$ *and* e_t *satisfies (6) for all* $t \in \mathcal{I}_{j,k} | \Gamma_{j,k-1}^e) = 1$.

Lemma 8.2 (Subspace Recovery Lemma)**.** *Assume that all the conditions of Theorem 4.2 hold. Let* $\zeta_*^+ = r\zeta$. *Then, for all* $k = 1, 2, \ldots K$,

$$\mathbf{P}(\zeta_k \leq \zeta_k^+ | \Gamma_{j,k-1}^e) \geq p_k(\alpha, \zeta)$$

where ζ_k^+ *is defined in Definition 5.10 and* $p_k(\alpha, \zeta)$ *is defined in* (9).

Proof of Lemma 6.2: Observe that $\mathbf{P}(\Gamma_{j,k}|\Gamma_{j,k-1}) = \mathbf{P}(\check{\Gamma}_{j,k}|\Gamma_{j,k-1})$. The lemma then follows by combining Lemma 8.2 and item 2 of Lemma 8.1. ■

A. Proof of Lemma 8.1

In order to prove Lemma 8.1 we first need a bound on the RIC of the compressed sensing matrix Φ_k.

Lemma 8.3 (Bounding the RIC of Φ_k)**.** *Recall that* $\zeta_* := \|(I - \hat{P}_*\hat{P}_*')P_*\|_2$. *The following hold.*

1) *Suppose that a basis matrix* P *can be split as* $P = [P_1, P_2]$ *where* P_1 *and* P_2 *are also basis matrices. Then* $\kappa_s^2(P) = \max_{T:|T|\leq s} \|I_T' P\|_2^2 \leq \kappa_s^2(P_1) + \kappa_s^2(P_2)$.
2) $\kappa_s^2(\hat{P}_*) \leq \kappa_{s,*}^2 + 2\zeta_*$
3) $\kappa_s(\hat{P}_{new,k}) \leq \kappa_{s,new} + \tilde{\kappa}_{s,k}\zeta_k + \zeta_*$
4) $\delta_s(\Phi_0) = \kappa_s^2(\hat{P}_*) \leq \kappa_{s,*}^2 + 2\zeta_*$
5) $\delta_s(\Phi_k) = \kappa_s^2([\hat{P}_* \ \hat{P}_{new,k}]) \leq \kappa_s^2(\hat{P}_*) + \kappa_s^2(\hat{P}_{new,k}) \leq \kappa_{s,*}^2 + 2\zeta_* + (\kappa_{s,new} + \tilde{\kappa}_{s,k}\zeta_k + \zeta_*)^2$ *for* $k \geq 1$

Proof:

1) Since P is a basis matrix, $\kappa_s^2(P) = \max_{|T|\leq s} \|I_T{}'P\|_2^2$. Also, $\|I_T{}'P\|_2^2 = \|I_T{}'[P_1, P_2][P_1, P_2]'I_T\|_2 = \|I_T{}'(P_1P_1' + P_2P_2')I_T\|_2 \leq \|I_T{}'P_1P_1'I_T\|_2 + \|I_T{}'P_2P_2'I_T\|_2$. Thus, the inequality follows.
2) For any set T with $|T| \leq s$, $\|I_T{}'\hat{P}_*\|_2^2 = \|I_T{}'\hat{P}_*\hat{P}_*'I_T\|_2 = \|I_T{}'(\hat{P}_*\hat{P}_*' - P_*P_*{}' + P_*P_*{}')I_T\|_2 \leq \|I_T{}'(\hat{P}_*\hat{P}_*' - P_*P_*{}')I_T\|_2 + \|I_T{}'P_*P_*{}'I_T\|_2 \leq 2\zeta_* + \kappa_{s,*}^2$. The last inequality follows using Lemma 2.10 with $P = P_*$ and $\hat{P} = \hat{P}_*$.
3) By Lemma 2.10 with $P = P_*$, $\hat{P} = \hat{P}_*$ and $Q = P_{\text{new}}$, $\|P_{\text{new}}{}'\hat{P}_*\|_2 \leq \zeta_*$. By Lemma 2.10 with $P = P_{\text{new}}$ and $\hat{P} = \hat{P}_{\text{new},k}$, $\|(I - P_{\text{new}}P_{\text{new}}')\hat{P}_{\text{new},k}\|_2 = \|(I - \hat{P}_{\text{new},k}\hat{P}_{\text{new},k}')P_{\text{new}}\|_2$. For any set T with $|T| \leq s$, $\|I_T{}'\hat{P}_{\text{new},k}\|_2 \leq \|I_T{}'(I - P_{\text{new}}P_{\text{new}}')\hat{P}_{\text{new},k}\|_2 + \|I_T{}'P_{\text{new}}P_{\text{new}}'\hat{P}_{\text{new},k}\|_2 \leq \tilde{\kappa}_{s,k}\|(I - P_{\text{new}}P_{\text{new}}{}')\hat{P}_{\text{new},k}\|_2 + \|I_T{}'P_{\text{new}}\|_2 = \tilde{\kappa}_{s,k}\|(I - \hat{P}_{\text{new},k}\hat{P}_{\text{new},k}')P_{\text{new}}\|_2 + \|I_T{}'P_{\text{new}}\|_2 \leq \tilde{\kappa}_{s,k}\|D_{\text{new},k}\|_2 + \tilde{\kappa}_{s,k}\|\hat{P}_*\hat{P}_*'P_{\text{new}}\|_2 + \|I_T{}'P_{\text{new}}\|_2 \leq \tilde{\kappa}_{s,k}\zeta_k + \tilde{\kappa}_{s,k}\zeta_* + \kappa_{s,\text{new}} \leq \tilde{\kappa}_{s,k}\zeta_k + \zeta_* + \kappa_{s,\text{new}}$. Taking $\max$ over $|T| \leq s$ the claim follows.
4) This follows using Lemma 3.2 and the second claim of this lemma.
5) This follows using Lemma 3.2 and the first three claims of this lemma.

■

Corollary 8.4. *If the conditions of Theorem 4.2 are satisfied, and* $X_{j,k-1} \in \Gamma_{j,k-1}$, *then*

1) $\delta_s(\Phi_0) \leq \delta_{2s}(\Phi_0) \leq {\kappa_{2s,*}^+}^2 + 2\zeta_*^+ < 0.1 < 0.1479$
2) $\delta_s(\Phi_{k-1}) \leq \delta_{2s}(\Phi_{k-1}) \leq {\kappa_{2s,*}^+}^2 + 2\zeta_*^+ + (\kappa_{2s,new}^+ + \tilde{\kappa}_{2s,k-1}^+\zeta_{k-1}^+ + \zeta_*^+)^2 < 0.1479$
3) $\phi_{k-1} \leq \frac{1}{1-\delta_s(\Phi_{k-1})} < \phi^+$

Proof: This follows using Lemma 8.3, the definition of $\Gamma_{j,k-1}$, and the bound on ζ_{k-1}^+ from Lemma 6.1. ■

The following are striaghtforward bounds that will be useful for the proof of Lemma 8.1 and later.

Fact 8.5. *Under the assumptions of Theorem 4.2:*

1) $\zeta\gamma_* \leq \frac{\sqrt{\zeta}}{(r_0+(J-1)c)^{3/2}} \leq \sqrt{\zeta}$
2) $\zeta_*^+ \leq \frac{10^{-4}}{(r_0+(J-1)c)} \leq 10^{-4}$
3) $\zeta_*^+\gamma_*^2 \leq \frac{1}{(r_0+(J-1)c)^2} \leq 1$
4) $\zeta_*^+\gamma_* \leq \frac{\sqrt{\zeta}}{\sqrt{r_0+(J-1)c}} \leq \sqrt{\zeta}$
5) $\zeta_*^+ f \leq \frac{1.5\times 10^{-4}}{r_0+(J-1)c} \leq 1.5 \times 10^{-4}$
6) $\zeta_{k-1}^+ \leq 0.6^{k-1} + 0.4c\zeta$ *(from Lemma 6.1)*
7) $\zeta_{k-1}^+\gamma_{\textit{new},k} \leq (0.6 \cdot 1.2)^{k-1}\gamma_{\textit{new}} + 0.4c\zeta\gamma_* \leq 0.72^{k-1}\gamma_{\textit{new}} + \frac{0.4\sqrt{\zeta}}{\sqrt{r_0+(J-1)c}} \leq 0.72^{k-1}\gamma_{\textit{new}} + 0.4\sqrt{\zeta}$
8) $\zeta_{k-1}^+\gamma_{\textit{new},k}^2 \leq (0.6 \cdot 1.2^2)^{k-1}\gamma_{\textit{new}}^2 + 0.4c\zeta\gamma_*^2 \leq 0.864^{k-1}\gamma_{\textit{new}}^2 + \frac{0.4}{(r_0+(J-1)c)^2} \leq 0.864^{k-1}\gamma_{\textit{new}}^2 + 0.4$

Proof of Lemma 8.1: Recall that $X_{j,k-1} \in \Gamma_{j,k-1}$ implies that $\zeta_* \leq \zeta_*^+$ and $\zeta_{k-1} \leq \zeta_{k-1}^+$.

1) a) For $t \in \mathcal{I}_{j,k}$, $\beta_t := (I - \hat{P}_{(t-1)}\hat{P}'_{(t-1)})L_t = D_{*,k-1}a_{t,*} + D_{\text{new},k-1}a_{t,\text{new}}$. Thus, using Fact 8.5

$$\begin{aligned}\|\beta_t\|_2 &\leq \zeta_*\sqrt{r}\gamma_* + \zeta_{k-1}\sqrt{c}\gamma_{\text{new},k} \\ &\leq \sqrt{\zeta}\sqrt{r} + (0.72^{k-1}\gamma_{\text{new}} + .4\sqrt{\zeta})\sqrt{c} \\ &= \sqrt{c}0.72^{k-1}\gamma_{\text{new}} + \sqrt{\zeta}(\sqrt{r} + 0.4\sqrt{c}) \leq \xi_0.\end{aligned}$$

b) By Corollary 8.4, $\delta_{2s}(\Phi_{k-1}) < 0.15 < \sqrt{2} - 1$. Given $|T_t| \leq s$, $\|\beta_t\|_2 \leq \xi_0 = \xi$, by Theorem 2.5, the CS error satisfies

$$\|\hat{S}_{t,\text{cs}} - S_t\|_2 \leq \frac{4\sqrt{1 + \delta_{2s}(\Phi_{k-1})}}{1 - (\sqrt{2}+1)\delta_{2s}(\Phi_{k-1})}\xi_0 < 7\xi_0.$$

c) Using the above and the definition of ρ, $\|\hat{S}_{t,\text{cs}} - S_t\|_\infty \leq 7\rho\xi_0$. Since $\min_{i\in T_t}|(S_t)_i| \geq S_{\min}$ and $(S_t)_{T_t^c} = 0$, $\min_{i\in T_t}|(\hat{S}_{t,cs})_i| \geq S_{\min} - 7\rho\xi_0$ and $\min_{i\in T_t^c}|(\hat{S}_{t,\text{cs}})_i| \leq 7\rho\xi_0$. If $\omega < S_{\min} - 7\rho\xi_0$, then $\hat{T}_t \supseteq T_t$. On the other hand, if $\omega > 7\rho\xi_0$, then $\hat{T}_t \subseteq T_t$. Since $S_{\min} > 14\rho\xi_0$ (condition 3 of the theorem) and ω satisfies $7\rho\xi_0 \leq \omega \leq S_{\min} - 7\rho\xi_0$ (condition 1 of the theorem), then the support of S_t is exactly recovered, i.e. $\hat{T}_t = T_t$.

d) Given $\hat{T}_t = T_t$, the LS estimate of S_t satisfies $(\hat{S}_t)_{T_t} = [(\Phi_{k-1})_{T_t}]^\dagger y_t = [(\Phi_{k-1})_{T_t}]^\dagger(\Phi_{k-1}S_t + \Phi_{k-1}L_t)$ and $(\hat{S}_t)_{T_t^c} = 0$ for $t \in \mathcal{I}_{j,k}$. Also, $(\Phi_{k-1})_{T_t}{}'\Phi_{k-1} = I_{T_t}{}'\Phi_{k-1}$ (this follows since $(\Phi_{k-1})_{T_t} = \Phi_{k-1}I_{T_t}$ and $\Phi'_{k-1}\Phi_{k-1} = \Phi_{k-1}$). Using this, the LS error $e_t := \hat{S}_t - S_t$ satisfies (6). Thus, using Fact 8.5 and condition 2 of the theorem,

$$\begin{aligned}\|e_t\|_2 &\leq \phi^+(\zeta_*^+\sqrt{r}\gamma_* + \kappa_{s,k-1}\zeta_{k-1}^+\sqrt{c}\gamma_{\text{new},k}) \\ &\leq 1.2\left(\sqrt{r}\sqrt{\zeta} + \sqrt{c}0.15(0.72)^{k-1} + \sqrt{c}0.06\sqrt{\zeta}\right) \\ &= 0.18\sqrt{c}0.72^{k-1}\gamma_{\text{new}} + 1.2\sqrt{\zeta}(\sqrt{r} + 0.06\sqrt{c}).\end{aligned}$$

2) The second claim is just a restatement of the first.

■

B. Proof of Lemma 8.2

The proof of Lemma 8.2 will use the next two lemmas (8.6, and 8.7).

Lemma 8.6. *If* $\lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2 > 0$*, then*

$$\zeta_k \leq \frac{\|\mathcal{R}_k\|_2}{\lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2} \leq \frac{\|\mathcal{H}_k\|_2}{\lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2} \tag{8}$$

where $\mathcal{R}_k := \mathcal{H}_k E_{\textit{new}}$ *and* A_k*,* $A_{k,\perp}$*,* $\mathcal{H}_k$ *are defined in Definition 5.6.*

Proof: Since $\lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2 > 0$, so $\lambda_{\min}(A_k) > \|A_{k,\perp}\|_2$. Since A_k is of size $c_{\text{new}} \times c_{\text{new}}$ and $\lambda_{\min}(A_k) > \|A_{k,\perp}\|_2$, $\lambda_{c_{\text{new}}+1}(\mathcal{A}_k) = \|A_{k,\perp}\|_2$. By definition of EVD, and since Λ_k is a $c_{\text{new}} \times c_{\text{new}}$ matrix, $\lambda_{\max}(\Lambda_{k,\perp}) = \lambda_{c_{\text{new}}+1}(\mathcal{A}_k + \mathcal{H}_k)$. By Weyl's theorem (Theorem 2.8), $\lambda_{c_{\text{new}}+1}(\mathcal{A}_k + \mathcal{H}_k) \leq \lambda_{c_{\text{new}}+1}(\mathcal{A}_k) + \|\mathcal{H}_k\|_2 = \|A_{k,\perp}\|_2 + \|\mathcal{H}_k\|_2$. Therefore, $\lambda_{\max}(\Lambda_{k,\perp}) \leq$

$\|A_{k,\perp}\|_2 + \|\mathcal{H}_k\|_2$ and hence $\lambda_{\min}(A_k) - \lambda_{\max}(\Lambda_{k,\perp}) \geq \lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2 > 0$. Apply the $\sin\theta$ theorem (Theorem 2.7) with $\lambda_{\min}(A_k) - \lambda_{\max}(\Lambda_{k,\perp}) > 0$, we get

$$\|(I - \hat{P}_{\text{new},k}\hat{P}'_{\text{new},k})E_{\text{new}}\|_2 \leq \frac{\|\mathcal{R}_k\|_2}{\lambda_{\min}(A_k) - \lambda_{\max}(\Lambda_{k,\perp})} \leq \frac{\|\mathcal{H}_k\|_2}{\lambda_{\min}(A_k) - \|A_{k,\perp}\|_2 - \|\mathcal{H}_k\|_2}$$

Since $\zeta_k = \|(I - \hat{P}_{\text{new},k}\hat{P}'_{\text{new},k})D_{\text{new}}\|_2 = \|(I - \hat{P}_{\text{new},k}\hat{P}'_{\text{new},k})E_{\text{new}}R_{\text{new}}\|_2 \leq \|(I - \hat{P}_{\text{new},k}\hat{P}'_{\text{new},k})E_{\text{new}}\|_2$, the result follows. The last inequality follows because $\|R_{\text{new}}\|_2 = \|E'_{\text{new}}D_{\text{new}}\|_2 \leq 1$. ■

Lemma 8.7 (High probability bounds for each of the terms in the ζ_k bound (8))**.** *Assume the conditions of Theorem 4.2 hold. Also assume that* $\mathbf{P}(\Gamma^e_{j,k-1}) > 0$ *for all* $1 \leq k \leq K+1$. *Then, for all* $1 \leq k \leq K$

1) $\mathbf{P}\left(\lambda_{\min}(A_k) \geq \lambda^-_{new,k}\left(1 - (\zeta_*^+)^2 - \frac{c\zeta}{12}\right) \middle| \Gamma^e_{j,k-1}\right) > 1 - p_{a,k}(\alpha,\zeta)$ *where*
$$p_{a,k}(\alpha,\zeta) := c\exp\left(\frac{-\alpha\zeta^2(\lambda^-)^2}{8\cdot 24^2\cdot\min(1.2^{4k}\gamma_{new}^4, \gamma_*^4)}\right) + c\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2\cdot 4^2}\right)$$
2) $\mathbf{P}\left(\lambda_{\max}(A_{k,\perp}) \leq \lambda^-_{new,k}\left((\zeta_*^+)^2 f + \frac{c\zeta}{24}\right) \middle| \Gamma^e_{j,k-1}\right) > 1 - p_b(\alpha,\zeta)$ *where*
$$p_b(\alpha,\zeta) := (n-c)\exp\left(\frac{-\alpha c^2\zeta(\lambda^-)^2}{8\cdot 24^2}\right)$$
3) $\mathbf{P}\left(\|\mathcal{H}_k\|_2 \leq \lambda^-_{new,k}(b + 0.125c\zeta) \middle| \Gamma^e_{j,k-1}\right) \geq 1 - p_c(\alpha,\zeta)$ *where b is as defined in Definition 5.10 and*
$$\begin{aligned} p_c(\alpha,\zeta) :=& n\exp\left(\frac{-\alpha\zeta^2(\lambda^-)^2}{8\cdot 24^2(0.0324\gamma_{new}^2 + 0.0072\gamma_{new} + 0.0004)^2}\right) + \\ & n\exp\left(\frac{-\alpha\zeta^2(\lambda^-)^2}{32\cdot 24^2(0.06\gamma_{new}^2 + 0.0006\gamma_{new} + 0.4)^2}\right) + \\ & n\exp\left(\frac{-\alpha\zeta^2(\lambda^-)^2\epsilon^2}{32\cdot 24^2(0.186\gamma_{new}^2 + 0.00034\gamma_{new} + 2.3)^2}\right). \end{aligned}$$

Proof of Lemma 8.2: Lemma 8.2 now follows by combining Lemmas 8.6 and 8.7 and defining

$$p_k(\alpha,\zeta) := 1 - p_{a,k}(\alpha,\zeta) - p_b(\alpha,\zeta) - p_c(\alpha,\zeta). \tag{9}$$

■

As above, we will start with some simple facts that will be used to prove Lemma 8.7.

For convenience, we will use $\frac{1}{\alpha}\sum_t$ to denote $\frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}}$

Fact 8.8. *Under the assumptions of Theorem 4.2 the following are true.*

1) *The matrices D_{new}, R_{new}, E_{new}, $D_*, D_{new,k-1}$, Φ_{k-1} are functions of the r.v. $X_{j,k-1}$. All terms that we bound for the first two claims of the lemma are of the form $\frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} Z_t$ where $Z_t = f_1(X_{j,k-1})Y_t f_2(X_{j,k-1})$, Y_t is a sub-matrix of $a_t a_t'$ and $f_1(.)$ and $f_2(.)$ are functions of $X_{j,k-1}$.*
2) *$X_{j,k-1}$ is independent of any a_t for $t \in \mathcal{I}_{j,k}$, and hence the same is true for the matrices D_{new}, R_{new}, E_{new}, $D_*, D_{new,k-1}$, Φ_{k-1}. Also, a_t's for different $t \in \mathcal{I}_{j,k}$ are mutually independent. Thus, conditioned on $X_{j,k-1}$, the Z_t's defined above are mutually independent.*
3) *All the terms that we bound for the third claim contain e_t. Using the second claim of Lemma 8.1, conditioned on $X_{j,k-1}$, e_t satisfies (6) w.p. one whenever $X_{j,k-1} \in \Gamma_{j,k-1}$. Conditioned on $X_{j,k-1}$, all these terms are also of the form $\frac{1}{\alpha}\sum_{t\in\mathcal{I}_{j,k}} Z_t$ with Z_t as defined above, whenever $X_{j,k-1} \in \Gamma_{j,k-1}$. Thus, conditioned on $X_{j,k-1}$, the Z_t's for these terms are mutually independent, whenever $X_{j,k-1} \in \Gamma_{j,k-1}$.*
4) *It is easy to see that $\|\Phi_{k-1}P_*\|_2 \leq \zeta_*$, $\zeta_0 = \|D_{new}\|_2 \leq 1$, $\Phi_0 D_{new} = \Phi_0' D_{new} = D_{new}$, $\|R_{new}\| \leq 1$, $\|(R_{new})^{-1}\| \leq 1/\sqrt{1-\zeta_*^2}$, $E_{new,\perp}{}' D_{new} = 0$, and $\|E_{new}{}'\Phi_0 e_t\| = \|(R'_{new})^{-1}D'_{new}\Phi_0 e_t\| = \|(R_{new})^{-1}D'_{new}e_t\| \leq \|(R'_{new})^{-1}D'_{new}I_{T_t}\|\|e_t\| \leq \frac{\kappa_s^+}{\sqrt{1-\zeta_*^2}}\|e_t\|$. The bounds on $\|R_{new}\|$ and $\|(R_{new})^{-1}\|$ follow using Lemma 2.10 and the fact that $\sigma_i(R_{new}) = \sigma_i(D_{new})$.*
5) *$X_{j,k-1} \in \Gamma_{j,k-1}$ implies that*

a) $\zeta_* \leq \zeta_*^+$ *(see Remark 6.3)*

b) $\zeta_{k-1} \leq \zeta_{k-1}^+ \leq 0.6^{k-1} + 0.4c\zeta$ *(This follows by the definition of* $\Gamma_{j,k-1}$ *and Lemma 6.1.)*

6) *Item 5 implies that*

a) $\lambda_{\min}(R_{new} R_{new}{}') \geq 1 - (\zeta_*^+)^2$*. This follows from Lemma 2.10 and the fact that* $\sigma_{\min}(R_{new}) = \sigma_{\min}(D_{new})$*.*

b) $\|I_{T_t}{}' \Phi_{k-1} P_*\|_2 \leq \|\Phi_{k-1} P_*\|_2 \leq \zeta_* \leq \zeta_*^+$, $\|I_{T_t}{}' D_{new,k-1}\|_2 \leq \kappa_{s,k-1}\zeta_{k-1} \leq \kappa_s^+ \zeta_{k-1}^+$.

7) *By Weyl's theorem (Theorem 2.8), for a sequence of matrices* B_t*,* $\lambda_{\min}(\sum_t B_t) \geq \sum_t \lambda_{\min}(B_t)$ *and* $\lambda_{\max}(\sum_t B_t) \leq \sum_t \lambda_{\max}(B_t)$*.*

Proof of Lemma 8.7:

Consider $A_k := \frac{1}{\alpha}\sum_t E_{\text{new}}{}' \Phi_0 L_t L_t{}' \Phi_0 E_{\text{new}}$. Notice that $E_{\text{new}}{}' \Phi_0 L_t = R_{\text{new}} a_{t,\text{new}} + E_{\text{new}}{}' D_* a_{t,*}$. Let $Z_t = R_{\text{new}} a_{t,\text{new}} a_{t,\text{new}}{}' R_{\text{new}}{}'$ and let $Y_t = R_{\text{new}} a_{t,\text{new}} a_{t,*}{}' D_*{}' E_{\text{new}}{}' + E_{\text{new}}{}' D_* a_{t,*} a_{t,\text{new}}{}' R_{\text{new}}{}'$, then

$$A_k \succeq \frac{1}{\alpha}\sum_t Z_t + \frac{1}{\alpha}\sum_t Y_t \tag{10}$$

Consider $\sum_t Z_t = \sum_t R_{\text{new}} a_{t,\text{new}} a_{t,\text{new}}{}' R_{\text{new}}'$.

1) Using item 2 of Fact 8.8, the Z_t's are conditionally independent given $X_{j,k-1}$.

2) Using item 2, Ostrowoski's theorem (Theorem 2.9), and item 6, for all $X_{j,k-1} \in \Gamma_{j,k-1}$, $\lambda_{\min}\left(\mathbf{E}(\frac{1}{\alpha}\sum_t Z_t | X_{j,k-1})\right) = \lambda_{\min}\left(R_{\text{new}} \frac{1}{\alpha}\sum_t \mathbf{E}(a_{t,\text{new}} a_{t,\text{new}}{}') R_{\text{new}}{}'\right) \geq \lambda_{\min}\left(R_{\text{new}} R_{\text{new}}{}'\right) \lambda_{\min}\left(\frac{1}{\alpha}\sum_t \mathbf{E}(a_{t,\text{new}} a_{t,\text{new}}{}')\right) \geq (1 - (\zeta_*^+)^2)\lambda_{\text{new},k}^-$.

3) Finally, using items 4 and the bound on $\|a_t\|_\infty$ from the model, conditioned on $X_{j,k-1}$, $0 \preceq Z_t \preceq c\gamma_{\text{new},k}^2 I \leq c\max\left((1.2)^{2k}\gamma_{\text{new}}^2, \gamma_*^2\right) I$ holds w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$.

Thus, applying Corollary 2.14 with $\epsilon = \frac{c\zeta\lambda^-}{24}$, we get

$$\mathbf{P}\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_t Z_t\right) \geq (1 - (\zeta_*^+)^2)\lambda_{\text{new},k}^- - \frac{c\zeta\lambda^-}{24}\middle| X_{j,k-1}\right) \geq 1 - c\exp\left(\frac{-\alpha\zeta^2(\lambda^-)^2}{8\cdot 24^2 \cdot \min(1.2^{4k}\gamma_{\text{new}}^4, \gamma_*^4)}\right) \tag{11}$$

for all $X_{j,k-1} \in \Gamma_{j,k-1}$.

Consider $Y_t = R_{\text{new}} a_{t,\text{new}} a_{t,*}{}' D_*{}' E_{\text{new}} + E_{\text{new}}{}' D_* a_{t,*} a_{t,\text{new}}{}' R_{\text{new}}{}'$.

1) Using item 2, the Y_t's are conditionally independent given $X_{j,k-1}$.

2) Using item 2 and the fact that $a_{t,\text{new}}$ and $a_{t,*}$ are mutually uncorrelated, $\mathbf{E}\left(\frac{1}{\alpha}\sum_t Y_t | X_{j,k-1}\right) = 0$ for all $X_{j,k-1} \in \Gamma_{j,k-1}$.

3) Using the bound on $\|a_t\|_\infty$, items 4, 6, and Fact 8.5, conditioned on $X_{j,k-1}$, $\|Y_t\| \leq 2\sqrt{cr}\zeta_*^+ \gamma_* \gamma_{\text{new},k} \leq 2\sqrt{cr}\zeta_*^+ \gamma_*^2 \leq 2$ holds w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$.

Thus, under the same conditioning, $-bI \preceq Y_t \preceq bI$ with $b = 2$ w.p. one.

Thus, applying Corollary 2.14 with $\epsilon = \frac{c\zeta\lambda^-}{24}$, we get

$$\mathbf{P}\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_t Y_t\right) \geq \frac{-c\zeta\lambda^-}{24}\middle| X_{j,k-1}\right) \geq 1 - c\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2 \cdot (2b)^2}\right) \text{ for all } X_{j,k-1} \in \Gamma_{j,k-1} \tag{12}$$

Combining (10), (11) and (12) and using the union bound, $\mathbf{P}(\lambda_{\min}(A_k) \geq \lambda_{\text{new},k}^-(1 - (\zeta_*^+)^2) - \frac{c\zeta\lambda^-}{12} | X_{j,k-1}) \geq 1 - p_a(\alpha, \zeta)$ for all $X_{j,k-1} \in \Gamma_{j,k-1}$. The first claim of the lemma follows by using $\lambda_{\text{new},k}^- \geq \lambda^-$ and then applying Lemma 2.11 with $X \equiv X_{j,k-1}$ and $\mathcal{C} \equiv \Gamma_{j,k-1}$.

Now consider $A_{k,\perp} := \frac{1}{\alpha}\sum_t E_{\text{new},\perp}{}' \Phi_0 L_t L_t{}' \Phi_0 E_{\text{new},\perp}$. Using item 4, $E_{\text{new},\perp}{}' \Phi_0 L_t = E_{\text{new},\perp}{}' D_* a_{t,*}$. Thus, $A_{k,\perp} = \frac{1}{\alpha}\sum_t Z_t$ with $Z_t = E_{\text{new},\perp}{}' D_* a_{t,*} a_{t,*}{}' D_*{}' E_{\text{new},\perp}$ which is of size $(n-c) \times (n-c)$. Using the same ideas as above we can show that $0 \preceq Z_t \preceq r(\zeta_*^+)^2\gamma_*^2 I \preceq \zeta I$ and $\mathbf{E}\left(\frac{1}{\alpha}\sum_t Z_t | X_{j,k-1}\right) \preceq (\zeta_*^+)^2\lambda^+ I$. Thus by Corollary 2.14 with $\epsilon = \frac{c\zeta\lambda^-}{24}$ and Lemma 2.11 the second claim follows.

Using the expression for $\mathcal{H}_k$ given in Definition 5.6, it is easy to see that

$$\|\mathcal{H}_k\|_2 \leq \max\{\|H_k\|_2, \|H_{k,\perp}\|_2\} + \|B_k\|_2 \leq \left\|\frac{1}{\alpha}\sum_t e_t e_t{}'\right\|_2 + \max(\|T2\|_2, \|T4\|_2) + \|B_k\|_2 \tag{13}$$

where $T2 := \frac{1}{\alpha}\sum_t {E_{\text{new}}}'\Phi_0(L_t{e_t}' + e_t{L_t}')\Phi_0 E_{\text{new}}$ and $T4 := \frac{1}{\alpha}\sum_t {E_{\text{new},\perp}}'\Phi_0(L_t{e_t}' + {e_t}'L_t)\Phi_0 E_{\text{new},\perp}$. The second inequality follows by using the facts that (i) $H_k = T1 - T2$ where $T1 := \frac{1}{\alpha}\sum_t {E_{\text{new}}}'\Phi_0 e_t{e_t}'\Phi_0 E_{\text{new}}$, (ii) $H_{k,\perp} = T3 - T4$ where $T3 := \frac{1}{\alpha}\sum_t {E_{\text{new},\perp}}'\Phi_0 e_t{e_t}'\Phi_0 E_{\text{new},\perp}$, and (iii) $\max(\|T1\|_2, \|T3\|_2) \leq \|\frac{1}{\alpha}\sum_t e_t{e_t}'\|_2$. Next, we obtain high probability bounds on each of the terms on the RHS of (13) using the Hoeffding corollaries.

Consider $\|\frac{1}{\alpha}\sum_t e_t{e_t}'\|_2$. Let $Z_t = e_t{e_t}'$.

1) Using item 2, conditioned on $X_{j,k-1}$, the various Z_t's in the summation are independent, for all $X_{j,k-1} \in \Gamma_{j,k-1}$.
2) Using item 6, and the bound on $\|a_t\|_\infty$, conditioned on $X_{j,k-1}$, $0 \preceq Z_t \preceq b_1 I$ w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Here $b_1 := (\kappa_s^+\zeta_{k-1}^+\phi^+\sqrt{c}\gamma_{\text{new},k} + \zeta_*^+\phi^+\sqrt{r}\gamma_*)^2$.
3) Also using item 6, and the bound on $\|a_t\|_\infty$, $0 \preceq \frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X_{j,k-1}) \preceq b_2 I$, with $b_2 := (\kappa_s^+)^2(\zeta_{k-1}^+)^2(\phi^+)^2\lambda_{\text{new},k}^+ + (\zeta_*^+)^2(\phi^+)^2\lambda^+$ for all $X_{j,k-1} \in \Gamma_{j,k-1}$.

Thus, applying Corollary 2.14 with $\epsilon = \frac{c\zeta\lambda^-}{24}$,

$$\mathbf{P}\left(\Big\|\frac{1}{\alpha}\sum_t e_t{e_t}'\Big\|_2 \leq b_2 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \geq 1 - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2 b_1^2}\right) \text{ for all } X_{j,k-1} \in \Gamma_{j,k-1} \tag{14}$$

Consider $T2$. Let $Z_t := {E_{\text{new}}}'\Phi_0(L_t{e_t}' + e_t{L_t}')\Phi_0 E_{\text{new}}$ which is of size $c \times c$. Then $T2 = \frac{1}{\alpha}\sum_t Z_t$.

1) Using item 2, conditioned on $X_{j,k-1}$, the various Z_t's used in the summation are mutually independent, for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Using item 4, ${E_{\text{new}}}'\Phi_0 L_t = R_{\text{new}}a_{t,\text{new}} + {E_{\text{new}}}'D_* a_{t,*}$ and ${E_{\text{new}}}'\Phi_0 e_t = ({R_{\text{new}}}')^{-1}{D_{\text{new}}}'e_t$.
2) Thus, using items 4, 6, and the bound on $\|a_t\|_\infty$, it follows that conditioned on $X_{j,k-1}$, $\|Z_t\|_2 \leq 2\tilde{b}_3 \leq 2b_3$ w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Here, $\tilde{b}_3 := \frac{\kappa_s^+}{\sqrt{1-(\zeta_*^+)^2}}\phi^+(\kappa_s^+\zeta_{k-1}^+\sqrt{c}\gamma_{\text{new},k} + \sqrt{r}\zeta_*^+\gamma_*)(\sqrt{c}\gamma_{\text{new},k} + \sqrt{r}\zeta_*^+\gamma_*)$ and $b_3 := \frac{1}{\sqrt{1-(\zeta_*^+)^2}}(\phi^+ c{\kappa_s^+}^2\zeta_{k-1}^+\gamma_{\text{new},k}^2 + \phi^+\sqrt{rc}{\kappa_s^+}^2\zeta_{k-1}^+\zeta_*^+\gamma_{\text{new},k}\gamma_* + \phi^+\sqrt{rc}\kappa_s^+\zeta_*^+\gamma_*\gamma_{\text{new},k} + \phi^+ r{\zeta_*^+}^2\gamma_*^2)$.
3) Also, $\|\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X_{j,k-1})\|_2 \leq 2\tilde{b}_4 \leq 2b_4$ where $\tilde{b}_4 := \frac{\kappa_s^+}{\sqrt{1-(\zeta_*^+)^2}}\phi^+\kappa_s^+\zeta_{k-1}^+\lambda_{\text{new},k}^+ + \frac{\kappa_s^+}{\sqrt{1-(\zeta_*^+)^2}}\phi^+(\zeta_*^+)^2\lambda^+$ and $b_4 := \frac{\kappa_s^+}{\sqrt{1-(\zeta_*^+)^2}}\phi^+\kappa_s^+\zeta_{k-1}^+\lambda_{\text{new},k}^+ + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}\phi^+(\zeta_*^+)^2\lambda^+$.

Thus, applying Corollary 2.15 with $\epsilon = \frac{c\zeta\lambda^-}{24}$,

$$\mathbf{P}\left(\|T2\|_2 \leq 2b_4 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \geq 1 - c\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2\cdot 4b_3^2}\right) \text{ for all } X_{j,k-1} \in \Gamma_{j,k-1}$$

Consider $T4$. Let $Z_t := {E_{\text{new},\perp}}'\Phi_0(L_t{e_t}' + e_t{L_t}')\Phi_0 E_{\text{new},\perp}$ which is of size $(n-c)\times(n-c)$. Then $T4 = \frac{1}{\alpha}\sum_t Z_t$.

1) Using item 2, conditioned on $X_{j,k-1}$, the various Z_t's used in the summation are mutually independent, for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Using item 4, ${E_{\text{new},\perp}}'\Phi_0 L_t = {E_{\text{new},\perp}}'D_* a_{t,*}$.
2) Thus, conditioned on $X_{j,k-1}$, $\|Z_t\|_2 \leq 2b_5$ w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Here $b_5 := \phi^+ r(\zeta_*^+)^2\gamma_*^2 + \phi^+\sqrt{rc}\kappa_s^+\zeta_*^+\zeta_{k-1}^+\gamma_*\gamma_{\text{new},k}$ This follows using items 6 and the bound on $\|a_t\|_\infty$.
3) Also, $\|\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X_{j,k-1})\|_2 \leq 2b_6,\ b_6 := \phi^+(\zeta_*^+)^2\lambda^+$.

Applying Corollary 2.15 with $\epsilon = \frac{c\zeta\lambda^-}{24}$,

$$\mathbf{P}\left(\|T4\|_2 \leq 2b_6 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \geq 1 - (n-c)\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2\cdot 4b_5^2}\right) \text{ for all } X_{j,k-1} \in \Gamma_{j,k-1}$$

Consider $\max(\|T2\|_2, \|T4\|_2)$. Since $b_3 > b_5$ (follows because $\zeta_{k-1}^+ \leq 1$) and $b_4 > b_6$, so $2b_6 + \frac{c\zeta\lambda^-}{24} < 2b_4 + \frac{c\zeta\lambda^-}{24}$ and $1 - (n-c)\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2\cdot 4b_5^2}\right) > 1 - (n-c)\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2\cdot 4b_3^2}\right)$. Therefore, for all $X_{j,k-1} \in \Gamma_{j,k-1}$, $\mathbf{P}\left(\|T4\|_2 \leq 2b_4 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \geq 1 - (n-c)\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2\cdot 4b_3^2}\right)$.

By the union bound, for all $X_{j,k-1} \in \Gamma_{j,k-1}$,

$$\mathbf{P}\left(\max(\|T2\|_2, \|T4\|_2) \leq 2b_4 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \geq 1 - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2\cdot 4b_3^2}\right) \tag{15}$$

Consider $\|B_k\|_2$. Let $Z_t := {E_{\text{new},\perp}}'\Phi_0(L_t - e_t)({L_t}' - {e_t}')\Phi_0 E_{\text{new}}$ which is of size $(n-c)\times c$. Then $B_k = \frac{1}{\alpha}\sum_t Z_t$. Using item 4, ${E_{\text{new},\perp}}'\Phi_0(L_t - e_t) = {E_{\text{new},\perp}}'(D_* a_{t,*} - \Phi_0 e_t)$, ${E_{\text{new}}}'\Phi_0(L_t - e_t) = R_{\text{new}}a_{t,\text{new}} + {E_{\text{new}}}'D_* a_{t,*} + (R'_{\text{new}})^{-1}D'_{\text{new}}e_t$. Also,

$\|Z_t\|_2 \le b_7$ w.p. one for all $X_{j,k-1} \in \Gamma_{j,k-1}$ and $\|\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t | X_{j,k-1})\|_2 \le b_8$ for all $X_{j,k-1} \in \Gamma_{j,k-1}$. Here

$$\begin{aligned} b_7 :=& (\sqrt{r}\zeta_*^+(1+\phi^+)\gamma_* + (\kappa_s^+)\zeta_{k-1}^+\phi^+\sqrt{c}\gamma_{\text{new},k}) \cdot \\ & \left(\sqrt{c}\gamma_{\text{new},k} + \sqrt{r}\zeta_*^+\left(1 + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}\kappa_s^+\phi^+\right)\gamma_* + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}{\kappa_s^+}^2\zeta_{k-1}^+\phi^+\sqrt{c}\gamma_{\text{new},k}\right) \end{aligned}$$

and

$$\begin{aligned} b_8 := & \left(\kappa_s^+\zeta_{k-1}^+\phi^+ + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}(\kappa_s^+)^3(\zeta_{k-1}^+)^2(\phi^+)^2\right)\lambda_{\text{new},k}^+ + \\ & (\zeta_*^+)^2\left(1 + \phi^+ + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}\kappa_s^+\phi^+ + \frac{1}{\sqrt{1-(\zeta_*^+)^2}}\kappa_s^+(\phi^+)^2\right)\lambda^+ \end{aligned}$$

Thus, applying Corollary 2.15 with $\epsilon = \frac{c\zeta\lambda^-}{24}$,

$$\mathbf{P}\left(\|B_k\|_2 \le b_8 + \frac{c\zeta\lambda^-}{24}\Big|X_{j,k-1}\right) \ge 1 - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2 b_7^2}\right) \text{ for all } X_{j,k-1} \in \Gamma_{j,k-1} \tag{16}$$

Using (13), (14), (15) and (16) and the union bound, for any $X_{j,k-1} \in \Gamma_{j,k-1}$,

$$\begin{aligned} & \mathbf{P}\left(\|\mathcal{H}_k\|_2 \le b_9 + \frac{c\zeta\lambda^-}{8}\Big|X_{j,k-1}\right) \ge \\ & 1 - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{8\cdot 24^2 b_1^2}\right) - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2}{32\cdot 24^2\cdot 4b_3^2}\right) - n\exp\left(\frac{-\alpha c^2\zeta^2(\lambda^-)^2\epsilon^2}{32\cdot 24^2 b_7^2}\right) \end{aligned}$$

where

$$\begin{aligned} b_9 :=& b_2 + 2b_4 + b_8 \\ =& \left(\left(\frac{2(\kappa_s^+)^2\phi^+}{\sqrt{1-(\zeta_*^+)^2}} + \kappa_s^+\phi^+\right)\zeta_{k-1}^+ + \left((\kappa_s^+)^2(\phi^+)^2 + \frac{(\kappa_s^+)^3(\phi^+)^2}{\sqrt{1-(\zeta_*^+)^2}}\right)(\zeta_{k-1}^+)^2\right)\lambda_{\text{new},k}^+ + \\ & \left((\phi^+)^2 + \frac{2\phi^+}{\sqrt{1-(\zeta_*^+)^2}} + 1 + \phi^+ + \frac{\kappa_s^+\phi^+}{\sqrt{1-(\zeta_*^+)^2}} + \frac{\kappa_s^+(\phi^+)^2}{\sqrt{1-(\zeta_*^+)^2}}\right)(\zeta_*^+)^2\lambda^+ \end{aligned}$$

Using $\lambda_{\text{new},k}^- \ge \lambda^-$ and $f := \lambda^+/\lambda^-$, $b_9 + \frac{c\zeta\lambda^-}{8} \le \lambda_{\text{new},k}^-(b + 0.125c\zeta)$. Using Fact 8.5 and substituting $\kappa_s^+ = 0.15$, $\phi^+ = 1.2$, one can upper bound b_1, b_3 and b_7 and show that the above probability is lower bounded by $1 - p_c(\alpha, \zeta)$. Finally, applying Lemma 2.11, the third claim of the lemma follows.

■

IX. Extensions

In Sec IX-A, we show how ReProCS and its performance guarantees can be extended to the missing data (measurements) case. In Sec IX-B, we introduce a more general subspace change model and give the performance guarantees for it. In Sec IX-C, we describe the key idea of ReProCS with deletion which will improve performance for data satisfying the more general model of Sec IX-B.

A. Extension to Missing Data Case

Consider the following problem. The goal is to recover a sequence of L_t's that lie a slowly changing low dimensional subspace from measurements M_t, when some of the measurements may be missing. To be precise, let T_t denote the set of missing measurements at time t. Then, M_t is a sparse vector with support T_t^c, i.e.

$$(M_t)_{T_t^c} = (L_t)_{T_t^c},\ (M_t)_{T_t} = 0$$

and L_t follows the model of Sec III-A. We do not assume any model on T_t, except an upper bound on its size. In particular, it could be correlated over time. This problem is related to the low rank matrix completion problem [43], [44].

As explained in earlier work [5], any solution for low dimensional signal recovery in the presence of sparse outliers can be converted into a solution for low rank matrix completion by re-formulating the problem as $M_t = L_t + S_t$ where $(S_t)_{T_t^c} = 0$ and on T_t, S_t can take on any value. Since we let $(M_t)_{T_t} = 0$, this means that we are letting $(S_t)_{T_t} = -(L_t)_{T_t}$. Notice that this missing data problem is actually easier because T_t is perfectly known.

To recover L_t from missing data, we can use the ReProCS algorithm of Algorithm 2 with the following simple change: remove steps 1b and 1c and replace them by $\hat{T}_t = T_t$.

The performance guarantees given in Theorem 4.2 also apply to this case with the following simple changes. In condition 1, the parameters ξ and ω are not needed; and in condition 3, one can remove the lower bound on $S_{\min}$. Also, the conclusions for recovering L_t, i.e. the bound on $\|L_t - \hat{L}_t\|_2$ are the only ones relevant in that case.

B. More General Model for L_t

The model given in Sec III-A only allows for new directions to get added to P_{j-1}, but not for any directions to get removed. This means that the rank of the subspace in which L_t lies keeps increasing. However, in practice, this may not happen. Consider the following more general model. Assume the model of Sec III-A with the following changes.

1) We assume that $P_j = [P_{j-1} \ P_{j,\text{new}}] \setminus P_{j,\text{old}}$ where $P_{j,\text{old}}$ contains $c_{j,\text{old}}$ columns of P_{j-1}. Thus $r_j = r_{j-1} + c_{j,\text{new}} - c_{j,\text{old}}$.
2) Assume that there exists a constant c_{mx} such that $0 \leq c_{j,\text{new}} \leq c_{mx}$, and $\sum_{i=1}^{j} (c_{i,\text{new}} - c_{i,\text{old}}) \leq c_{mx}$. This ensures that the subspace rank in any period, $r_j := \text{rank}(P_j) = r_0 + \sum_{i=1}^{j} (c_{i,\text{new}} - c_{i,\text{old}}) \leq r_0 + c_{mx} := r_{mx}$.

Even for this more general model, the ReProCS algorithm of Algorithm 2 works. The only difference is that $\hat{P}_j$ is now actually an estimate of $[P_0, P_{1,\text{new}}, \ldots P_{j,\text{new}}]$ and not of just P_j. However, since P_j is a sub-matrix of this bigger matrix, thus, span$(\hat{P}_j)$ still approximately contains span(P_j) in the sense defined in Definition 2.2. Similary at any time t, $\hat{P}_{(t)}$ is an estimate of the span of a bigger matrix than $P_{(t)}$. However, span$(\hat{P}_{(t)})$ still approximately contains span$(P_{(t)})$.

For the above model, the performance guarantees also remain almost the same. The only change is that we need to bound $\kappa_s([P_0, P_{1,\text{new}}, \ldots P_{J-1,\text{new}}])$ instead of $\kappa_s(P_{J-1})$. The following corollary can be stated.

Corollary 9.1. *Consider Algorithm 2. Assume that L_t obeys the model given above and there are a total of J change times. The result of Theorem 4.2 holds with the following change. We also need $\kappa_s([P_0, P_{1,new}, \ldots P_{J-1,new}]) \leq 0.3$.*

C. ReProCS with deletion

Consider the more general subspace change model given in Sec IX-B above. While ReProCS applies directly for this model as well, one can further improve its performance by also including a deletion step that we briefly explain here.

A limitation of ReProCS is that at time t, it obtains an estimate of the subspace spanned by all the columns of $\mathcal{L}_t$ and projects perpendicular to this subspace in order to approximately nullify L_t before solving the ℓ_1 problem to recover S_t. However, according to the more general model, the current L_t only lies in span(P_j) which is a smaller subspace than span$(\mathcal{L}_t)$ = span$([P_0, P_{1,\text{new}}, \ldots P_{j,\text{new}}])$. In other words, $\text{rank}(P_j)$ is smaller than $\text{rank}([P_0, P_{1,\text{new}}, \ldots P_{j,\text{new}}])$, and so the same is true for the denseness coefficients. Thus, if we can get an estimate of only span(P_j), the RIC of $\Phi_{j,k}$ will be smaller, thus making the sparse recovery more accurate. This, in turn, will mean that the subspace error will also be smaller.

One simple way to estimate only span(P_j) is to delete directions as follows. Before the next change time, but after the current $P_{j,\text{new}}$ has been accurately estimated, we do a standard PCA step. To be precise, at $t = t_j + K\alpha + \alpha_{\text{del}} - 1$, we compute the EVD of $\frac{1}{\alpha_{\text{del}}} \sum_{t=t_j+K\alpha}^{t_j+K\alpha+\alpha_{\text{del}}-1} \hat{L}_t \hat{L}_t'$ and retain the $r_j = r_{j-1} + c_{j,\text{new}} - c_{j,\text{old}}$ eigenvectors with the largest eigenvalues. Use these as the new estimate, $\hat{P}_{(t)}$. It can be shown that as long as f, the maximum condition number of Cov(L_t) for any t, is small enough, doing this will give an accurate estimate, $\hat{P}_{(t)}$, of the current $P_{(t)} = P_j$. Because of this deletion, we will be able to relax the denseness requirement significantly. We will only need $\kappa_s(P_j) < 0.3$ instead of $\kappa_s([P_0, P_{1,\text{new}}, \ldots P_{j,\text{new}}]) < 0.3$.

In [42], we introduce a generalization of this simple deletion strategy that removes the bound on f, but instead only requires that the eigenvalues of the average of $\text{Cov}(L_t)$ be sufficiently clustered. Suppose that there are θ clusters. It achieves this by replacing the simple PCA step described above by θ iterations of projection PCA for clusters.

X. Model Verification, Practical Parameter Setting and Simulation Experiments

We first discuss model verification for real data in Sec X-A. In Sec X-B, we discuss how to set parameters for ReProCS in practice. We describe simulation experiments in Sec X-C.

A. *Model Verification for real data*

We experimented with two background image sequence datasets. The first was a video of lake water motion. The second was a video of window curtains moving due to the wind. The curtain sequence is available at http://home.engineering.iastate.edu/~chenlu/ReProCS/Fig2.mp4. For this sequence, the image size was $n = 5120$ and the number of images, $t_{\max} = 1755$. The lake sequence is available at http://home.engineering.iastate.edu/~chenlu/ReProCS/ReProCS.htm (sequence 3). For this sequence, $n = 6480$ and the number of images, $t_{\max} = 1500$. Any given background image sequence will never be exactly low rank, but only approximately so. Let the data matrix with its empirical mean subtracted be $\mathcal{L}_{full}$. Thus $\mathcal{L}_{full}$ is a $n \times t_{\max}$ matrix. We first "low-rankified" this dataset by computing the EVD of $(1/t_{\max})\mathcal{L}_{full}\mathcal{L}_{full}'$; retaining the 90% eigenvectors' set (i.e. sorting eigenvalues in non-increasing order and retaining all eigenvectors until the sum of the corresponding eigenvalues exceeded 90% of the sum of all eigenvalues); and projecting the dataset into this subspace. To be precise, we computed P_{full} as the matrix containing these eigenvectors and we computed the low-rank matrix $\mathcal{L} = P_{full}P_{full}'\mathcal{L}_{full}$. Thus $\mathcal{L}$ is a $n \times t_{\max}$ matrix with $\text{rank}(\mathcal{L}) < \min(n, t_{\max})$. The curtains dataset is of size 5120×1755, but 90% of the energy is contained in only 34 directions, i.e. $\text{rank}(\mathcal{L}) = 34$. The lake dataset is of size 6480×1500 but 90% of the energy is contained in only 14 directions, i.e. $\text{rank}(\mathcal{L}) = 14$. This indicates that both datasets are indeed approximately low rank.

In practical data, the subspace does not just change as simply as in the model given in Sec. III-A. There are also rotations of the new and existing eigen-directions at each time which have not been modeled there. Moreover, with just one training sequence of a given type, it is not possible to compute $\text{Cov}(L_t)$ at each time t. Thus it is not possible to compute the delay between subspace change times. The only thing we can do is to assume that there may be a change every d frames, and that during these d frames the data is stationary and ergodic, and then estimate $\text{Cov}(L_t)$ for this period using a time average. We proceeded as follows. We took the first set of d frames, $\mathcal{L}_{1:d} := [L_1, L_2 \dots L_d]$, estimated its covariance matrix as $(1/d)\mathcal{L}_{1:d}\mathcal{L}_{1:d}'$ and computed P_0 as the 99.99% eigenvectors' set. Also, we stored the lowest retained eigenvalue and called it λ^-. It is assumed that all directions with eigenvalues below λ^- are due to noise. Next, we picked the next set of d frames, $\mathcal{L}_{d+1:2d} := [L_{d+1}, L_{d+2}, \dots L_{2d}]$; projected them perpendicular to P_0, i.e. computed $\mathcal{L}_{1,p} = (I - P_0P_0')\mathcal{L}_{d+1:2d}$; and computed $P_{1,\text{new}}$ as the eigenvectors of $(1/d)\mathcal{L}_{1,p}\mathcal{L}_{1,p}'$ with eigenvalues equal to or above λ^-. Then, $P_1 = [P_0, P_{1,\text{new}}]$. For the third set of d frames, we repeated the above procedure, but with P_0 replaced by P_1 and obtained P_2. A similar approach was repeated for each batch.

We used $d = 150$ for both the datasets. In each case, we computed $r_0 := \text{rank}(P_0)$, and $c_{mx} := \max_j \text{rank}(P_{j,\text{new}})$. For each batch of d frames, we also computed $a_{t,\text{new}} := P_{j,\text{new}}'L_t$, $a_{t,*} := P_{j-1}'L_t$ and $\gamma_* := \max_t \|a_t\|_\infty$. We got $c_{mx} = 3$ and $r_0 = 8$ for the lake sequence and $c_{mx} = 5$ and $r_0 = 29$ for the curtain sequence. Thus the ratio c_{mx}/r_0 is sufficiently small in both cases. In Fig 3, we plot $\|a_{t,\text{new}}\|_\infty/\gamma_*$ for one 150-frame period of the curtain sequence and for three 150-frame change periods of the lake sequence. If we take $\alpha = 40$, we observe that $\gamma_{\text{new}} := \max_j \max_{t_j \le t < t_j+\alpha} \|a_{t,\text{new}}\|_\infty = 0.125\gamma_*$ for the curtain sequence and $\gamma_{\text{new}} = 0.06\gamma_*$ for the lake sequence, i.e. the projection along the new directions is small for the initial α frames. Also, clearly, it increases slowly. In fact $\|a_{t,\text{new}}\|_\infty \le \max(v^{k-1}\gamma_{\text{new}}, \gamma_*)$ for all $t \in \mathcal{I}_{j,k}$ also holds with $v = 1.5$ for the curtain sequence and $v = 1.8$ for the lake sequence.

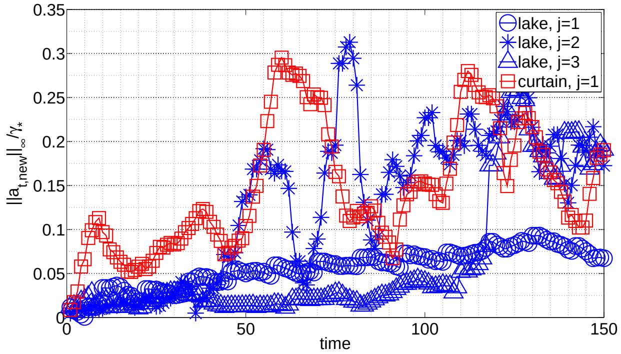


Fig. 3. Verification of slow subspace change. The figure is discussed in Sec X-A.

B. Practical Parameter Setting for ReProCS

The ReProCS algorithm given in Algorithm 2 uses knowledge of $t_j, r_0, c_{j,\text{new}}$ from the model and it has four parameters ξ, ω, α, K that can be set in terms of the model parameters as given in Theorem 4.2. However, it is unreasonable to expect that, in practice, the model parameters are known. We provide here reasonable heuristics for setting both the model and the algorithm parameters automatically. For a vector v, we define the 99%-energy set of v as $T_{0.99}(v) := \{i : |v_i| \geq v_{0.99}\}$ where the threshold $v_{0.99}$ is the largest value of $|v_i|$ so that $\|v_{T_{0.99}}\|_2^2 \geq 0.99\|v\|_2^2$. It is computed by sorting $|v_i|$ in non-increasing order of magnitude. One keeps adding elements to $T_{0.99}$ until $\|v_{T_{0.99}}\|_2^2 \geq 0.99\|v\|_2^2$.

The complete algorithm is summarized in Algorithm 3. We pick $\alpha = 100$ arbitrarily. We let $\xi = \xi_t$ and $\omega = \omega_t$ vary with time. Recall that ξ_t is the upper bound on $\|\beta_t\|_2$. We do not know β_t. All we have is an estimate of β_t from $t-1$, $\hat{\beta}_{t-1} = (I - \hat{P}_{(t-1)}\hat{P}_{(t-1)}')\hat{L}_{t-1}$. We used a value a little larger than $\|\hat{\beta}_{t-1}\|_2$ for ξ_t: we let $\xi_t = 2\|\hat{\beta}_{t-1}\|_2$. The parameter ω_t is the support estimation threshold. One reasonable way to pick this is to use a percentage energy threshold of $\hat{S}_{t,\text{cs}}$ [45]. In this work, we used $\omega_t = 0.5(\hat{S}_{t,\text{cs}})_{0.99}$.

Let $\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_{t_\text{train}}$ denote the eigenvalues of $\frac{1}{t_\text{train}}\sum_{t=1}^{t_\text{train}} L_t {L_t}'$. We estimate r_0 and λ^- as

$$\hat{r}_0 = \max_{i=1,2,\cdots,t_\text{train}-1}\left(\frac{\hat{\lambda}_i - \hat{\lambda}_{i+1}}{\hat{\lambda}_i}\right), \ \hat{\lambda}^- = \hat{\lambda}_{\hat{r}_0} \tag{17}$$

This heuristic relies on the fact that the maximum normalized difference between consecutive eigenvalues is from λ^- to zero.

We split projection PCA into two phases: “detect” and “estimate”. In the “detect” phase, we estimate the change time t_j and the number of new added directions $c_{j,\text{new}}$ as follows. We keep doing projection PCA every α frames and looking for eigenvalues above $\hat{\lambda}^-$. If there are any eigenvalues above $\hat{\lambda}^-$, we let $\hat{t}_j = t - \alpha + 1$ and we let $\hat{c}_{j,\text{new}}$ be the number of these eigenvalues. Also, we increment j and we reset k to one. At this time, the algorithm enters the “estimate” phase. In this phase, we keep doing projection PCA every α frames until the stopping criterion given in step 3(a)iiB of Algorithm 3 is satisfied (this estimates K). The idea is to stop when k exceeds K_min and $\hat{P}_{j,\text{new},k}' P_{j,\text{new}}$ is approximately equal to $\hat{P}_{j,\text{new},k-1}' P_{j,\text{new}}$ three times in a row; or when $k = K_\text{max}$. We pick $K_\text{min} = 5, K_\text{max} = 20$ arbitrarily. When the stopping criterion is satisfied, we let $K_j = k$ and $\hat{P}_j = [\hat{P}_{j-1}, \hat{P}_{j,\text{new},K_j}]$, and the algorithm enters the “detect” phase.

C. Simulation Experiments

1) Data Generation: The simulated data is generated as follows. The measurement matrix $\mathcal{M}_t := [M_1, M_2, \cdots, M_t]$ is of size 2048×5200. It can be decomposed as a sparse matrix $\mathcal{S}_t := [S_1, S_2, \cdots, S_t]$ plus a low rank matrix $\mathcal{L}_t := [L_1, L_2, \cdots, L_t]$.

The sparse matrix $\mathcal{S}_t := [S_1, S_2, \cdots, S_t]$ is generated as follows.

1) For $1 \leq t \leq t_\text{train} = 200$, $S_t = 0$.
2) For $t_\text{train} < t \leq 5200$, S_t has s nonzero elements. The initial support $T_0 = \{1, 2, \ldots s\}$. Every Δ time instants we increment the support indices by 1. For example, for $t \in [t_\text{train}+1, t_\text{train}+\Delta-1]$, $T_t = T_0$, for $t \in [t_\text{train}+\Delta, t_\text{train}+2\Delta-1]$.

Algorithm 3 ReProCS(practical)

```
*Input:* $M_t$, *Output:* $\hat{S}_t$, $\hat{L}_t$, $\hat{P}_{(t)}$.
Initialization: Given training sequence $[L_1, L_2, \cdots, L_{t_{train}}]$, compute the EVD of $\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}} L_t {L_t}' \overset{EVD}{=} E\Lambda E'$ and then estimate $\hat{r}_0$ and $\hat{\lambda}^-$ using (17). Let $\hat{P}_0$ retain the eigenvectors with the $\hat{r}_0$ largest eigenvalues.
At $t = t_{\text{train}}$, let $\hat{P}_{(t)} \leftarrow \hat{P}_0$. Let $j \leftarrow 0$, $k \leftarrow 1$, $\hat{t}_j = t_{\text{train}} + 1$ and $flag \leftarrow detect$. For $t > t_{\text{train}}$, do the following:
  1) Do step 1) of Algorithm 2 but with $\xi$ and $\omega$ replaced by $\xi_t$ and $\omega_t$ computed as explained in Sec X-B.
  2) Do step 2) of Algorithm 2.
  3) Projection PCA: Update $\hat{P}_{(t)}$ as follows.
    a) If $t = \hat{t}_j + k\alpha - 1$, compute EVD of $\frac{1}{\alpha}\sum_{t=\hat{t}_j+(k-1)\alpha}^{\hat{t}_j+k\alpha-1}(I - \hat{P}_{j-1}\hat{P}_{j-1}')\hat{L}_t\hat{L}_t'(I - \hat{P}_{j-1}\hat{P}_{j-1}')$
      i) If $flag = detect$,
        A) If no eigenvalues are above $\hat{\lambda}^-$, then $\hat{P}_{(t)} \leftarrow \hat{P}_{(t-1)}$. Increment $k \leftarrow k + 1$.
        B) If there are eigenvalues above $\hat{\lambda}^-$, then $\hat{t}_j \leftarrow t - \alpha + 1$, $j \leftarrow j + 1$, $k \leftarrow 1$, $flag \leftarrow estimate$.
      ii) Else if $flag = estimate$,
        A) Let $\hat{P}_{j,\text{new},k}$ retain the eigenvectors with eigenvalues above $\hat{\lambda}^-$, $\hat{P}_{(t)} \leftarrow [\hat{P}_{j-1}\ \hat{P}_{j,\text{new},k}]$ and $k \leftarrow k + 1$.
        B) If if $k \geq K_{\min}$ and $\frac{\| \sum_{t-\alpha+1}^{t}(\hat{P}_{j,\text{new},i-1}\hat{P}_{j,\text{new},i-1}' - \hat{P}_{j,\text{new},i}\hat{P}_{j,\text{new},i}')L_t\|_2}{\| \sum_{t-\alpha+1}^{t}\hat{P}_{j,\text{new},i-1}\hat{P}_{j,\text{new},i-1}'L_t\|_2} < 0.01$ for $i = k-2, k-1, k$; or $k = K_{\max}$, then $\hat{K}_j \leftarrow k$, $\hat{P}_j \leftarrow [\hat{P}_{j-1}\ \hat{P}_{j,\text{new},\hat{K}_j}]$ and reset $flag \leftarrow detect$.
    Else ($t \neq \hat{t}_j + k\alpha - 1$) set $\hat{P}_{(t)} \leftarrow \hat{P}_{(t-1)}$.
  4) Increment $t \leftarrow t + 1$ and go to step 1.
```

$T_t = \{2, 3, \dots s + 1\}$ and so on. Thus, the support set changes in a highly correlated fashion over time and this results in the matrix $\mathcal{S}_t$ being low rank. The larger the value of Δ, the smaller will be the rank of $\mathcal{S}_t$ (for $t > t_{\text{train}} + \Delta$).

3) The signs of the nonzero elements of S_t are ± 1 with equal probability and the magnitudes are uniformly distributed between 2 and 3. Thus, $S_{\min} = 2$.

The low rank matrix $\mathcal{L}_t := [L_1, L_2, \cdots, L_t]$ where $L_t := P_{(t)}a_t$ is generated as follows:

1) There are a total of $J = 2$ subspace change times, $t_1 = 301$ and $t_2 = 2501$. Let U be an $2048 \times (r_0 + c_{1,\text{new}} + c_{2,\text{new}})$ orthonormalized random Gaussian matrix.
   a) For $1 \leq t \leq t_1 - 1$, $P_{(t)} = P_0$ has rank r_0 with $P_0 = U_{[1,2,\cdots,r_0]}$.
   b) For $t_1 \leq t \leq t_2 - 1$, $P_{(t)} = P_1 = [P_0\ P_{1,\text{new}}]$ has rank $r_1 = r_0 + c_{1,\text{new}}$ with $P_{1,\text{new}} = U_{[r_0+1,\cdots,r_0+c_{1,\text{new}}]}$.
   c) For $t \geq t_2$, $P_{(t)} = P_2 = [P_1\ P_{2,\text{new}}]$ has rank $r_2 = r_1 + c_{2,\text{new}}$ with $P_{2,\text{new}} = U_{[r_0+c_{1,\text{new}}+1,\cdots,r_0+c_{1,\text{new}}+c_{2,\text{new}}]}$.
2) a_t is independent over t. The various $(a_t)_i$'s are also mutually independent for different i.
   a) For $1 \leq t < t_1$, we let $(a_t)_i$ be uniformly distributed between $-\gamma_{i,t}$ and $\gamma_{i,t}$, where

$$\gamma_{i,t} = \begin{cases} 400 & \text{if } i = 1, 2, \cdots, r_0/4, \forall t, \\ 30 & \text{if } i = r_0/4 + 1, r_0/4 + 2, \cdots, r_0/2, \forall t. \\ 2 & \text{if } i = r_0/2 + 1, r_0/2 + 2, \cdots, 3r_0/4, \forall t. \\ 1 & \text{if } i = 3r_0/4 + 1, 3r_0/4 + 2, \cdots, r_0, \forall t. \end{cases} \tag{18}$$

   b) For $t_1 \leq t < t_2$, $a_{t,*}$ is an r_0 length vector, $a_{t,\text{new}}$ is a $c_{1,\text{new}}$ length vector and $L_t := P_{(t)}a_t = P_1 a_t = P_0 a_{t,*} + P_{1,\text{new}}a_{t,\text{new}}$. $(a_{t,*})_i$ is uniformly distributed between $-\gamma_{i,t}$ and $\gamma_{i,t}$ and $a_{t,\text{new}}$ is uniformly distributed between $-\gamma_{r_1,t}$ and $\gamma_{r_1,t}$, where

$$\gamma_{r_1,t} = \begin{cases} 1.1^{k-1} & \text{if } t_1 + (k-1)\alpha \leq t \leq t_1 + k\alpha - 1, k = 1, 2, 3, 4, \\ 1.1^{4-1} = 1.331 & \text{if } t \geq t_1 + 4\alpha. \end{cases} \tag{19}$$

c) For $t \geq t_2$, $a_{t,*}$ is an $r_1 = r_0 + c_{1,\text{new}}$ length vector, $a_{t,\text{new}}$ is a $c_{2,\text{new}}$ length vector and $L_t := P_{(t)} a_t = P_2 a_t = [P_0 P_{1,\text{new}}] a_{t,*} + P_{2,\text{new}} a_{t,\text{new}}$. Also, $(a_{t,*})_i$ is uniformly distributed between $-\gamma_{i,t}$ and $\gamma_{i,t}$ for $i = 1, 2, \cdots, r_0$ and is uniformly distributed between $-\gamma_{r_1,t}$ and $\gamma_{r_1,t}$ for $i = r_0 + 1, \ldots r_1$. $a_{t,\text{new}}$ is uniformly distributed between $-\gamma_{r_2,t}$ and $\gamma_{r_2,t}$, where

$$\gamma_{r_2,t} = \begin{cases} 1.1^{k-1} & \text{if } t_2 + (k-1)\alpha \leq t \leq t_2 + k\alpha - 1, k = 1, 2, \cdots, 7, \\ 1.1^{7-1} = 1.7716 & \text{if } t \geq t_2 + 7\alpha. \end{cases} \tag{20}$$

Thus for the above model, $\gamma_* = 400$, $\gamma_{\text{new}} = 1$, $\lambda^+ = 53333$, $\lambda^- = 0.3333$ and $f := \frac{\lambda^+}{\lambda^-} = 1.6 \times 10^5$. Also, $S_{\min} = 2$.

We used $\mathcal{L}_{t_{\text{train}}} + \mathcal{N}_{t_{\text{train}}}$ as the training sequence to estimate $\hat{P}_0$. Here $\mathcal{N}_{t_{\text{train}}} = [N_1, N_2, \cdots, N_{t_{\text{train}}}]$ is i.i.d. random noise with each $(N_t)_i$ uniformly distributed between -10^{-3} and 10^{-3}. This is done to ensure that $\text{span}(\hat{P}_0) \neq \text{span}(P_0)$ but only approximates it.

2) Results: For Fig. 4 and Fig. 5, we used $s = 20$, $r_0 = 36$ and $c_{1,\text{new}} = c_{2,\text{new}} = 1$. We let $\Delta = 10$ for Fig. 4 and $\Delta = 50$ for Fig. 5. Because of the correlated support change, the $2048 \times t$ sparse matrix $\mathcal{S}_t = [S_1, S_2, \cdots, S_t]$ is rank deficient in either case, e.g. for Fig. 4, $\mathcal{S}_t$ has rank $29, 39, 49, 259$ at $t = 300, 400, 500, 2600$; for Fig. 5, $\mathcal{S}_t$ has rank $21, 23, 25, 67$ at $t = 300, 400, 500, 2600$. We plot the subspace error $\text{SE}_{(t)}$ and the normalized error for S_t, $\frac{\|\hat{S}_t - S_t\|_2}{\|S_t\|_2}$ averaged over 100 Monte Carlo simulations. We also plot the ratio $\frac{\|{I_{T_t}}' D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$ at the projection PCA times. This serves as a proxy for $\kappa_s(D_{j,\text{new},k})$ (which has exponential computational complexity). In fact, in our proofs, we only need this ratio to be small at every $t = t_j + k\alpha - 1$.

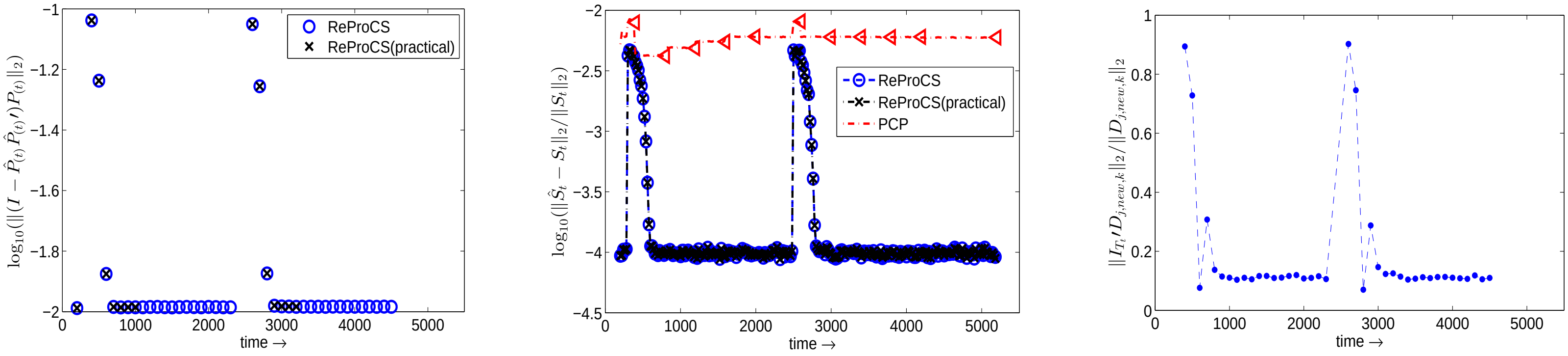
We compared against PCP [5]. At every $t = t_j + 4k\alpha$, we solved (1) with $\lambda = 1/\sqrt{\max(n,t)}$ to recover $\mathcal{S}_t$ and $\mathcal{L}_t$. We used the estimates of S_t for the last 4α frames as the final estimates of $\hat{S}_t$. So, the $\hat{S}_t$ for $t = t_j + 1, \ldots t_j + 4\alpha$ is obtained from PCP done at $t = t_j + 4\alpha$, the $\hat{S}_t$ for $t = t_j + 4\alpha + 1, \ldots t_j + 8\alpha$ is obtained from PCP done at $t = t_j + 8\alpha$ and so on.

As can be seen from Fig. 4, the subspace error $\text{SE}_{(t)}$ of ReProCS decreased exponentially and stabilized after about 4 projection PCA update steps. The averaged normalized error for S_t followed a similar trend. ReProCS(practical) performed similar to ReProCS but stabilized in about 6 projection PCA update steps. In Fig. 5 where $\Delta = 50$, the subspace error $\text{SE}_{(t)}$ also decreased but the decrease was a bit slower as compared to Fig. 4 where $\Delta = 10$. Also, the ratio $\frac{\|{I_{T_t}}' D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$ was now larger. Because of the correlated support change, the error of PCP was larger in both cases. The difference in performance between ReProCS and PCP is larger when $\Delta = 50$.

For Fig. 6, we increased s to 100 and we used $\Delta = 10$. A larger s results in a larger $\frac{\|{I_{T_t}}' D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$ (and larger $\kappa_s(D_{j,\text{new},k})$). Thus, the rate of decrease of $\text{SE}_{(t)}$ is smaller than that for the previous two figures. The error of S_t followed a similar trend.

Finally, if we set $\Delta = \infty$, the ratio $\frac{\|{I_{T_t}}' D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$ was 1 always. As a result, the subspace error and hence the reconstruction error of ReProCS did not decrease from its initial value at the subspace change time. For ReProCS, the average error $\frac{1}{5200} \sum_{t=201}^{5200} \frac{\|\hat{S}_t - S_t\|_2}{\|S_t\|_2} = 8.4 \times 10^{-3}$. The error of PCP was also very high: $\frac{1}{5200} \sum_{t=201}^{5200} \frac{\|\hat{S}_t - S_t\|_2}{\|S_t\|_2} = 0.43$.

We also did one experiment in which we generated T_t of size $s = 100$ uniformly at random from all possible s-size subsets of $\{1, 2, \ldots n\}$. T_t at different times t was also generated independently. In this case, the reconstruction error of ReProCS is $\frac{1}{5000} \sum_{t=201}^{5200} \frac{\|\hat{S}_t - S_t\|_2}{\|S_t\|_2} = 2.8472 \times 10^{-4}$. The error for PCP was 3.5×10^{-3} which is also quite small.

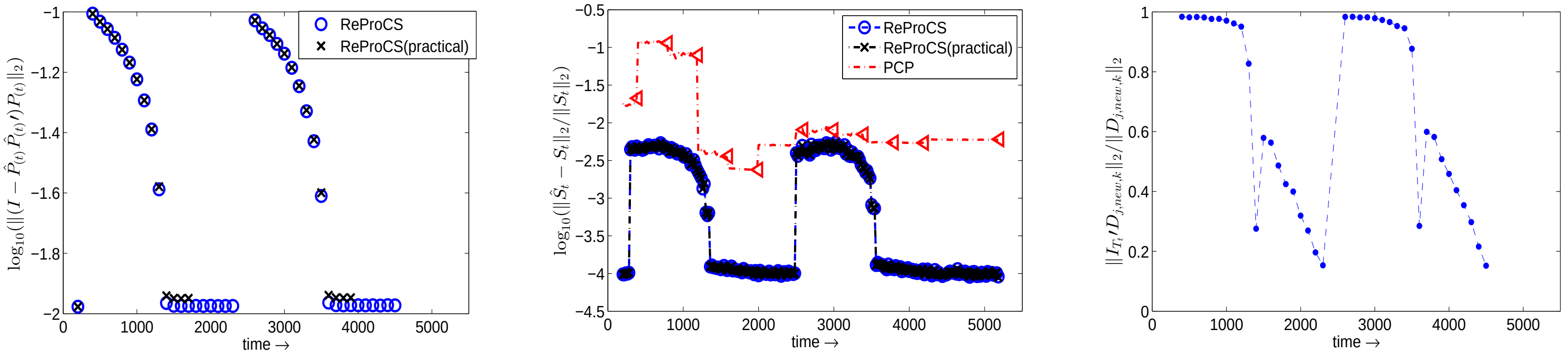


(a) subspace error, $\text{SE}_{(t)}$

(b) recon error of S_t

(c) plot of $\frac{\|I_{T_t}{}'D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$

Fig. 4. $r_0 = 36$, $s = \max_t |T_t| = 20$ and $\Delta = 10$. The times at which PCP is done are marked by red triangles in (b).

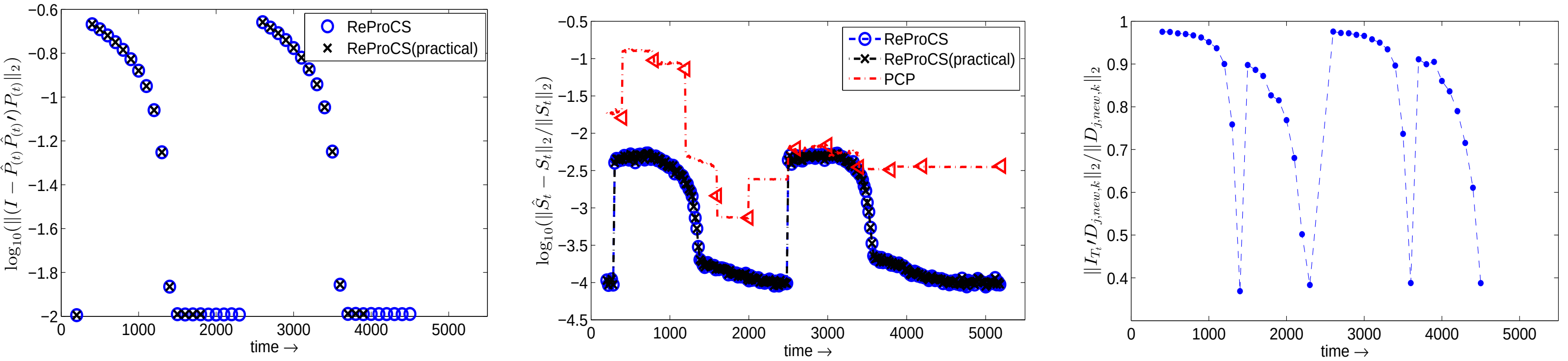


(a) subspace error, $\text{SE}_{(t)}$

(b) recon error of S_t

(c) plot of $\frac{\|I_{T_t}{}'D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$

Fig. 5. $r_0 = 36$, $s = \max_t |T_t| = 20$ and $\Delta = 50$. The times at which PCP is done are marked by red triangles in (b).



(a) subspace error, $\text{SE}_{(t)}$

(b) recon error of S_t

(c) plot of $\frac{\|I_{T_t}{}'D_{j,\text{new},k}\|_2}{\|D_{j,\text{new},k}\|_2}$

Fig. 6. $r_0 = 36$, $s = \max_t |T_t| = 100$ and $\Delta = 10$. The times at which PCP is done are marked by red triangles in (b)

XI. Conclusions and Future Work

In this work, we studied the recursive (online) robust PCA problem, which can also be interpreted as a problem of recursive sparse recovery in the presence of large but structured noise (noise lying in a "slowly changing" low dimensional subspace at all times). We introduced a novel solution approach called Recursive Projected CS or ReProCS that is able to recover both the sparse component, S_t, and low dimensional component, L_t, even when the support set of S_t changes in a correlated fashion over time. Under mild assumptions, we showed that, w.h.p., ReProCS can exactly recover the support set of S_t at all times;

and the reconstruction errors of both S_t and L_t are upper bounded by a time-invariant and small value at all times. We also showed how the algorithm and its guarantees extend to the undersampled measurements' case.

In ongoing work [42], we are developing and analyzing the ReProCS with deletion approach for the more general model given in Sec IX-B. Open questions to be addressed in future work include theoretically studying (a) the correlated a_t's case; (b) the connection between the denseness of $D_{j,\text{new},k}$ and the support change of S_t (that we observe in simulations); and (c) bounding the sparse recovery error even when the support set is not exactly recovered. In the current work, we needed $\hat{T}_t = T_t$ for $t < t_j + k\alpha - 1$ to ensure that the e_t's for $t \in \mathcal{I}_{j,k}$ are conditionally independent given $X_{j,k-1} = [a_1, a_2, \dots a_{t_j+(k-1)\alpha-1}]$.

Appendix

A. *Proof of Lemma 2.10*

Proof: Because P, Q and $\hat{P}$ are basis matrix, $P'P = I$, $Q'Q = I$ and $\hat{P}'\hat{P} = I$.

1) Using $P'P = I$ and $\|M\|_2^2 = \|MM'\|_2$, $\|(I - \hat{P}\hat{P}')PP'\|_2 = \|(I - \hat{P}\hat{P}')P\|_2$. Similarly, $\|(I - PP')\hat{P}\hat{P}'\|_2 = \|(I - PP')\hat{P}\|_2$. Let $D_1 = (I - \hat{P}\hat{P}')PP'$ and let $D_2 = (I - PP')\hat{P}\hat{P}'$. Notice that $\|D_1\|_2 = \sqrt{\lambda_{\max}(D_1'D_1)} = \sqrt{\|D_1'D_1\|_2}$ and $\|D_2\|_2 = \sqrt{\lambda_{\max}(D_2'D_2)} = \sqrt{\|D_2'D_2\|_2}$. So, in order to show $\|D_1\|_2 = \|D_2\|_2$, it suffices to show that $\|D_1'D_1\|_2 = \|D_2'D_2\|_2$. Let $P'\hat{P} \overset{SVD}{=} U\Sigma V'$. Then, $D_1'D_1 = P(I - P'\hat{P}\hat{P}'P)P' = PU(I - \Sigma^2)U'P'$ and $D_2'D_2 = \hat{P}(I - \hat{P}'PP'\hat{P})\hat{P}' = \hat{P}V(I - \Sigma^2)V'\hat{P}'$ are the compact SVD's of $D_1'D_1$ and $D_2'D_2$ respectively. Therefore, $\|D_1'D_1\| = \|D_2'D_2\|_2 = \|I - \Sigma^2\|_2$ and hence $\|(I - \hat{P}\hat{P}')PP'\|_2 = \|(I - PP')\hat{P}\hat{P}'\|_2$.
2) $\|PP' - \hat{P}\hat{P}'\|_2 = \|PP - \hat{P}\hat{P}'PP' + \hat{P}\hat{P}'PP' - \hat{P}\hat{P}'\|_2 \leq \|(I - \hat{P}\hat{P}')PP'\|_2 + \|(I - PP')\hat{P}\hat{P}'\|_2 = 2\zeta_*$.
3) Since $Q'P = 0$, then $\|Q'\hat{P}\|_2 = \|Q'(I - PP')\hat{P}\|_2 \leq \|(I - PP')\hat{P}\|_2 = \zeta_*$.
4) Let $M = (I - \hat{P}\hat{P}')Q)$. Then $M'M = Q'(I - \hat{P}\hat{P}')Q$ and so $\sigma_i((I - \hat{P}\hat{P}')Q) = \sqrt{\lambda_i(Q'(I - \hat{P}\hat{P}')Q)}$. Clearly, $\lambda_{\max}(Q'(I - \hat{P}\hat{P}')Q) \leq 1$. By Weyl's Theorem, $\lambda_{\min}(Q'(I - \hat{P}\hat{P}')Q) \geq 1 - \lambda_{\max}(Q'\hat{P}\hat{P}'Q) = 1 - \|Q'\hat{P}\|_2^2 \geq 1 - \zeta_*^2$. Therefore, $\sqrt{1 - \zeta_*^2} \leq \sigma_i((I - \hat{P}\hat{P}')Q) \leq 1$.

■

B. *Proof of Lemma 2.11*

Proof: It is easy to see that $\mathbf{P}(\mathcal{B}^e, \mathcal{C}^e) = \mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)\mathbb{I}_{\mathcal{C}}(X)]$. If $\mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)|X] \geq p$ for all $X \in \mathcal{C}$, this means that $\mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)|X]\mathbb{I}_{\mathcal{C}}(X) \geq p\mathbb{I}_{\mathcal{C}}(X)$. This, in turn, implies that

$$\mathbf{P}(\mathcal{B}^e, \mathcal{C}^e) = \mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)\mathbb{I}_{\mathcal{C}}(X)] = \mathbf{E}[\mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)|X]\mathbb{I}_{\mathcal{C}}(X)] \geq p\mathbf{E}[\mathbb{I}_{\mathcal{C}}(X)].$$

Recall from Definition 2.4 that $\mathbf{P}(\mathcal{B}^e|X) = \mathbf{E}[\mathbb{I}_{\mathcal{B}}(X,Y)|X]$ and $\mathbf{P}(\mathcal{C}^e) = \mathbf{E}[\mathbb{I}_{\mathcal{C}}(X)]$. Thus, we conclude that if $\mathbf{P}(\mathcal{B}^e|X) \geq p$ for all $X \in \mathcal{C}$, then $\mathbf{P}(\mathcal{B}^e, \mathcal{C}^e) \geq p\mathbf{P}(\mathcal{C}^e)$. Using the definition of $\mathbf{P}(\mathcal{B}^e|\mathcal{C}^e)$, the claim follows. ■

C. *Proof of Corollary 2.14*

Proof:

1) Since, for any $X \in \mathcal{C}$, conditioned on X, the Z_t's are independent, the same is also true for $Z_t - g(X)$ for any function of X. Let $Y_t := Z_t - \mathbf{E}(Z_t|X)$. Thus, for any $X \in \mathcal{C}$, conditioned on X, the Y_t's are independent. Also, clearly $\mathbf{E}(Y_t|X) = 0$. Since for all $X \in \mathcal{C}$, $\mathbf{P}(b_1 I \preceq Z_t \preceq b_2 I|X) = 1$ and since $\lambda_{\max}(.)$ is a convex function, and $\lambda_{\min}(.)$ is a concave function, of a Hermitian matrix, thus $b_1 I \preceq \mathbf{E}(Z_t|X) \preceq b_2 I$ w.p. one for all $X \in \mathcal{C}$. Therefore, $\mathbf{P}(Y_t^2 \preceq (b_2 - b_1)^2 I|X) = 1$ for all $X \in \mathcal{C}$. Thus, for Theorem 2.13, $\sigma^2 = \|\sum_t (b_2 - b_1)^2 I\|_2 = \alpha(b_2 - b_1)^2$. For any $X \in \mathcal{C}$, applying Theorem 2.13 for $\{Y_t\}$'s conditioned on X, we get that, for any $\epsilon > 0$,

$$\mathbf{P}\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_t Y_t\right) \leq \epsilon \Big| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2 - b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

By Weyl's theorem, $\lambda_{\max}(\frac{1}{\alpha}\sum_t Y_t) = \lambda_{\max}(\frac{1}{\alpha}\sum_t(Z_t - \mathbf{E}(Z_t|X)) \geq \lambda_{\max}(\frac{1}{\alpha}\sum_t Z_t) + \lambda_{\min}(\frac{1}{\alpha}\sum_t -\mathbf{E}(Z_t|X))$. Since $\lambda_{\min}(\frac{1}{\alpha}\sum_t -\mathbf{E}(Z_t|X)) = -\lambda_{\max}(\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X)) \geq -b_4$, thus $\lambda_{\max}(\frac{1}{\alpha}\sum_t Y_t) \geq \lambda_{\max}(\frac{1}{\alpha}\sum_t Z_t) - b_4$. Therefore,

$$\mathbf{P}\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_t Z_t\right) \leq b_4 + \epsilon \Big| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

2) Let $Y_t = \mathbf{E}(Z_t|X) - Z_t$. As before, $\mathbf{E}(Y_t|X) = 0$ and conditioned on any $X \in \mathcal{C}$, the Y_t's are independent and $\mathbf{P}(Y_t^2 \preceq (b_2-b_1)^2 I|X) = 1$. As before, applying Theorem 2.13, we get that for any $\epsilon > 0$,

$$\mathbf{P}\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_t Y_t\right) \leq \epsilon \Big| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

By Weyl's theorem, $\lambda_{\max}(\frac{1}{\alpha}\sum_t Y_t) = \lambda_{\max}(\frac{1}{\alpha}\sum_t(\mathbf{E}(Z_t|X) - Z_t)) \geq \lambda_{\min}(\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X)) + \lambda_{\max}(\frac{1}{\alpha}\sum_t -Z_t) = \lambda_{\min}(\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X)) - \lambda_{\min}(\frac{1}{\alpha}\sum_t Z_t) \geq b_3 - \lambda_{\min}(\frac{1}{\alpha}\sum_t Z_t)$ Therefore, for any $\epsilon > 0$,

$$\mathbf{P}\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_t Z_t\right) \geq b_3 - \epsilon \Big| X\right) \geq 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

■

D. *Proof of Corollary 2.15*

Proof: Define the dilation of an $n_1 \times n_2$ matrix M as dilation$(M) := \begin{bmatrix} 0 & M' \\ M & 0 \end{bmatrix}$. Notice that this is an $(n_1+n_2)\times(n_1+n_2)$ Hermitian matrix [28]. As shown in [28, equation 2.12],

$$\lambda_{\max}(\text{dilation}(M)) = \|\text{dilation}(M)\|_2 = \|M\|_2 \tag{21}$$

Thus, the corollary assumptions imply that $\mathbf{P}(\|\text{dilation}(Z_t)\|_2 \leq b_1|X) = 1$ for all $X \in \mathcal{C}$. Thus, $\mathbf{P}(-b_1 I \preceq \text{dilation}(Z_t) \preceq b_1 I|X) = 1$ for all $X \in \mathcal{C}$. Using (21), the corollary assumptions also imply that $\frac{1}{\alpha}\sum_t \mathbf{E}(\text{dilation}(Z_t)|X) = \text{dilation}(\frac{1}{\alpha}\sum_t \mathbf{E}(Z_t|X)) \preceq b_2 I$ for all $X \in \mathcal{C}$. Finally, Z_t's conditionally independent given X, for any $X \in \mathcal{C}$, implies that the same thing also holds for dilation(Z_t)'s. Thus, applying Corollary 2.14 for the sequence $\{\text{dilation}(Z_t)\}$, we get that,

$$\mathbf{P}\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_t \text{dilation}(Z_t)\right) \leq b_2 + \epsilon \Big| X\right) \geq 1 - (n_1+n_2)\exp\left(\frac{-\alpha\epsilon^2}{32b_1^2}\right) \text{ for all } X \in \mathcal{C}$$

Using (21), $\lambda_{\max}(\frac{1}{\alpha}\sum_t \text{dilation}(Z_t)) = \lambda_{\max}(\text{dilation}(\frac{1}{\alpha}\sum_t Z_t)) = \|\frac{1}{\alpha}\sum_t Z_t\|_2$ and this gives the final result. ■

E. *Proof of Lemma 3.2*

Proof: Let $A = I - PP'$. By definition, $\delta_s(A) := \max\{\max_{|T|\leq s}(\lambda_{\max}(A_T'A_T) - 1), \max_{|T|\leq s}(1 - \lambda_{\min}(A_T'A_T)))\}$. Notice that $A_T'A_T = I - I_T'PP'I_T$. Since $I_T'PP'I_T$ is p.s.d., by Weyl's theorem, $\lambda_{\max}(A_T'A_T) \leq 1$. Since $\lambda_{\max}(A_T'A_T) - 1 \leq 0$ while $1 - \lambda_{\min}(A_T'A_T) \geq 0$, thus,

$$\delta_s(I - PP') = \max_{|T|\leq s}\left(1 - \lambda_{\min}(I - I_T'PP'I_T)\right) \tag{22}$$

By Definition, $\kappa_s(P) = \max_{|T|\leq s}\frac{\|I_T'P\|_2}{\|P\|_2} = \max_{|T|\leq s}\|I_T'P\|_2$. Notice that $\|I_T'P\|_2^2 = \lambda_{\max}(I_T'PP'I_T) = 1 - \lambda_{\min}(I - I_T'PP'I_T)$ [4], and so

$$\kappa_s^2(P) = \max_{|T|\leq s}\left(1 - \lambda_{\min}(I - I_T'PP'I_T)\right) \tag{23}$$

From (22) and (23), we get $\delta_s(I - PP') = \kappa_s^2(P)$. ■

F. *The need for projection PCA (as in step 3 of Algorithm 2)*

In this discussion, we remove the subscript j. Also, let $P_* := P_{j-1}$, $\hat{P}_* := \hat{P}_{j-1}$, $r_* = \text{rank}(P_*)$.

[4] This follows because $B = I_T'PP'I_T$ is a Hermitian matrix. Let $B = U\Sigma U'$ be its EVD. Since $UU' = I$, $\lambda_{\min}(I - B) = \lambda_{\min}(U(I-\Sigma)U') = \lambda_{\min}(I - \Sigma) = 1 - \lambda_{\max}(\Sigma) = 1 - \lambda_{\max}(B)$.

1) Why projection PCA and not simple PCA: Consider $t = t_j + k\alpha - 1$ when the k^{th} projection PCA or PCA is done. Since the error $e_t = L_t - \hat{L}_t$ is correlated with L_t, the dominant terms in the perturbation matrix seen by PCA are $(1/(t_j + k\alpha)) \sum_{t=1}^{t_j+k\alpha-1} L_t e_t'$ and its transpose, while for projection PCA, they are $(1/\alpha)\Phi_0 \sum_{t\in\mathcal{I}_{j,k}} L_t e_t' \Phi_0$ and its transpose[5]. The magnitude of L_t can be quite large. The magnitude of e_t is smaller than a constant times that of L_t. The constant is less than one but, at $t = t_j + \alpha - 1$, it is not negligible. Thus, the norm of the perturbation seen by PCA at this time may not be small. As a result, the bound on the subspace error, $\text{SE}_{(t)}$, obtained by applying the $\sin\theta$ theorem may be more than one (and hence meaningless since by definition $\text{SE}_{(t)} \leq 1$). For projection PCA, because of Φ_0, the perturbation is much smaller.

Let $\text{SE}_k := \text{SE}_{(t_j+k\alpha)} = \text{SE}_{(t)}$ denote the subspace error for $t \in \mathcal{I}_{j,k}$. In quantitative terms, for PCA, we can show that $\text{SE}_1 \lesssim \check{C}\kappa_s^+ g^+ + \check{C}' f \zeta_*^+$ for constants $\check{C}, \check{C}'$ that are more than one but not too large. Here g^+ is the upper bound on $g_{j,k}$ (condition number of averaged $\text{Cov}(a_{t,\text{new}})$) and it is valid to assume that g^+ is small enough so that $\check{C}\kappa_s^+ g^+ < 1$. However, f is the maximum condition number of $\text{Cov}(a_t)$ and this can be large. When it is, the second term may not be less than one. On the other hand, for projection PCA, we have $\text{SE}_k \leq \zeta_k + \zeta_* \leq \zeta_k^+ + \zeta_*^+$ with $\zeta_*^+ = r\zeta$, and $\zeta_k^+ \approx \check{C}\kappa_s^+ g^+ \zeta_{k-1}^+ + \check{C}' f (\zeta_*^+)^2$ and $\zeta_0^+ = 1$. Thus $\text{SE}_1 \lesssim \check{C}\kappa_s^+ g^+ + \check{C}' f(\zeta_*^+)^2 + \zeta_*^+$. The first term in this bound is similar to that of PCA, but the second term is much smaller. The third term is negligibly small (under the slow subspace change assumption). Thus, in this case, it is easier to ensure that the bound is less than one.

Moreover, our goal is to show that within a finite delay after a subspace change time, the subspace error decays down from one to a value proportional to ζ. For projection PCA, this can be done because we can separately bound the subspace error of the existing subspace, ζ_*, and of the newly added one, ζ_k, and then bound the total subspace error, $\text{SE}_{(t)}$, by $\zeta_* + \zeta_k$ for $t \in \mathcal{I}_{j,k}$. Assuming that, by $t = t_j$, ζ_* is small enough, i.e. $\zeta_* \leq r_* \zeta$ with $\zeta < 0.00015/r^2 f$, we can show that within K iterations, ζ_k also becomes small enough so that $\text{SE}_{(t)} \leq (r_* + c)\zeta$. However, for PCA, it is not possible to separate the subspace error in this fashion. For $k > 1$, all we can claim is that $\text{SE}_k \lesssim \check{C}\kappa_s^+ f \ \text{SE}_{k-1}$. Since f can be large (larger than $1/\kappa_s^+$), this cannot be used to show that SE_k decreases with k.

2) Why multiple iterations to get a final estimate of $P_{j,new}$: The reason is again because e_t and L_t are correlated and so the dominant perturbation terms are $(1/\alpha)\sum_t \Phi_0 L_t e_t' \Phi_0$ and its transpose. The magnitude of e_t is smaller than a constant times that of L_t. The constant is less than one, but, at the first projection PCA time, it is not of the order of ζ. With the first projection PCA, we get the first estimate of P_{new}, $\hat{P}_{\text{new},1}$. Using this in the projected CS steps ensures a smaller e_t for the second interval and thus a smaller perturbation seen by the second projection PCA. This results in a reduced perturbation seen by the second projection PCA step and thus an improved estimate $\hat{P}_{j,\text{new},2}$. This makes e_t even smaller for the third interval and so on. Within K steps, we can show that it is small enough to ensure that $\text{SE}_{(t)}$ is proportional to ζ.

3) Why not use all $k\alpha$ frames at $t = t_j + k\alpha - 1$: Another possible way to implement projection PCA is to use the past $k\alpha$ estimates $\hat{L}_t$ at the k^{th} projection PCA time, $t = t_j + k\alpha - 1$. This may actually result in an improved algorithm. We believe that it can also be analyzed using the approaches developed in this paper. However, the analysis will be more complicated. We briefly try to explain why. The perturbation seen at $t = t_j + k\alpha - 1$, $\mathcal{H}_k$, will now satisfy $\mathcal{H}_k \approx (1/(k\alpha)) \sum_{k'=1}^{k} \sum_{t\in\mathcal{I}_{j,k'}} \Phi_0(-L_t e_t' - e_t L_t' + e_t e_t')\Phi_0$ instead of just being approximately equal to the last ($k' = k$) term. Bounds on each of these terms will hold with a different probability. Thus, proving a lemma similar to Lemma 8.7 will be more complicated.

References

[1] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Allerton Conference on Communication, Control, and Computing*, 2010.

[2] ——, "Recursive sparse recovery in large but correlated noise," in *48th Allerton Conference on Communication Control and Computing*, 2011.

[3] S. Roweis, "Em algorithms for pca and spca," *Advances in Neural Information Processing Systems*, pp. 626–632, 1998.

[4] F. D. L. Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.

[5] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of ACM*, vol. 58, no. 3, 2011.

[6] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, 2011.

[5] If e_t and L_t were uncorrelated, as is assumed in most work analyzing finite sample PCA, e.g. see [29], and since L_t is zero mean, these terms would be close to zero w.h.p. (due to law of large numbers) and the dominant perturbation term in either case would depend only on $\sum e_t e_t'$.

[7] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *European Conference on Computer Vision*, 2002, pp. 707–720.

[8] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, vol. 2, Oct 2003, pp. 1494 –1501.

[9] Y. Li, L. Xu, J. Morphett, and R. Jacobs, "An integrated algorithm of incremental and robust pca," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2003, pp. 245–248.

[10] J. Wright and Y. Ma, "Dense error correction via l1-minimization," *IEEE Trans. on Info. Th.*, vol. 56, no. 7, pp. 3540–3560, 2010.

[11] B. Lois, N. Vaswani, and C. Qui, "Recursive projected modified compressed sensing for undersampled measurements," *http://www.public.iastate.edu/%7Eblois/ReProModCSLong.pdf*.

[12] T. Zhang and G. Lerman, "A novel m-estimator for robust pca," *arXiv:1112.4863v1*, 2011.

[13] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," *IEEE Tran. on Information Theorey*, vol. 58, no. 5, 2012.

[14] M. McCoy and J. Tropp, "Two proposals for robust pca using semidefinite programming," *arXiv:1012.1086v3*, 2010.

[15] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex deconvolution, with applications," *arXiv:1205.1580*.

[16] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, no. 6, 2012.

[17] Y. Hu, S. Goud, and M. Jacob, "A fast majorize-minimize algorithm for the recovery of sparse and low-rank matrices," *IEEE Transactions on Image Processing*, vol. 21, no. 2, p. 742=753, Feb 2012.

[18] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. of Neural Information Processing Systems(NIPS)*, 2011.

[19] E. Richard, P.-A. Savalle, and N. Vayatis, "Estimation of simultaneously sparse and low rank matrices," *arXiv:1206.6474, appears in Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*.

[20] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with outliers," *arXiv:1011.1518*.

[21] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *arXiv:1204.6537*.

[22] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *arXiv:1202.4596*.

[23] A. Ganesh, K. Min, J. Wright, and Y. Ma, "Principal component pursuit with reduced linear measurements," *arXiv:1202.6445*.

[24] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[25] E. Candes, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, pp. 589–592, 2008.

[26] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. iii," *SIAM Journal on Numerical Analysis*, Mar. 1970.

[27] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[28] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of Computational Mathematics*, vol. 12, no. 4, 2012.

[29] B. Nadler, "Finite sample approximation results for principal component analysis: A matrix perturbation approach," *The Annals of Statistics*, vol. 36, no. 6, 2008.

[30] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Th.*, vol. 51(12), pp. 4203 – 4215, Dec. 2005.

[31] Y. Jin and B. Rao, "Algorithms for robust linear regression by exploiting the connection to sparse signal recovery," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2010.

[32] K. Mitra, A. Veeraraghavan, and R. Chellappa, "A robust regression using sparse learing for high dimensional parameter estimation problems," in *IEEE Intl. Conf. Acous. Speech. Sig.Proc.(ICASSP)*, 2010.

[33] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[34] C. Qiu and N. Vaswani, "Support-predicted modified-cs for principal components' pursuit," in *IEEE Intl. Symp. on Information Theory (ISIT)*, 2011.

[35] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford University Press, 2001.

[36] J. Laska, M. Davenport, and R. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice," in *Asilomar Conf. on Sig. Sys. Comp.*, Nov 2009, pp. 1556 –1560.

[37] N. H. Nguyen and T. D. Tran, "Robust lasso with missing and grossly corrupted observations," *To appear in IEEE Transaction on Information Theory*, 2012.

[38] G. Li and Z. Chen., "Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and monte carlo," *Journal of the American Statistical Association*, vol. 80, no. 391, pp. 759–766, 1985.

[39] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Th.*, vol. 52(2), pp. 489–509, February 2006.

[40] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52(4), pp. 1289–1306, April 2006.

[41] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *Annals of Statistics*, 2006.

[42] C. Qiu and N. Vaswani, "Recursive sparse recovery in large but structured noise – part 2," *arXiv: 1303.1144 [cs.IT]*, 2013.

[43] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Commun. ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[44] E. Candes and Y. Plan, "Matrix completion with noise," *arXiv:0903.3131*, 2009.

[45] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Processing*, September 2010.