2015

Sparse and low rank signal recovery with partial knowledge

Jinchun Zhan
Iowa State University

Sparse and low rank signal recovery with partial knowledge

by

Jinchun Zhan

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:

Namrata Vaswani, Major Professor

Aditya Ramamoorthy

Nicola Elia

Leslie Hogben

Kris De Brabanter

Iowa State University

Ames, Iowa

2015

DEDICATION

I would like to dedicate this dissertation to my family without whose support I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Firstly, I would like to thank Dr. Namrata Vaswani for her patience, guidance and support throughout my research and preparation for this dissertation. With her help, I can complete and enjoy my research.

And I would like to thank Dr. Aditya Ramamoorthy, Dr. Nicola Elia, Dr. Leslie Hogben, and Dr Kris De Brabanter for their support and comments to perfect my research and this dissertation.

Also I would like to thank all my friends who make my life in Iowa State University more enjoyable.

Finally I would like to thank my family, especially Yu Jie, without whose support, I cannot complete my research.

ABSTRACT

In the first part of this work, we study sparse recovery problem in the presence of bounded noise. We obtain performance guarantees for modified-CS and for its improved version, modified-CS-Add-LS-Del, for recursive reconstruction of a time sequence of sparse signals from a reduced set of noisy measurements available at each time. Under mild assumptions, we show that the support recovery error and reconstruction error of both algorithms are bounded by a time-invariant and small value at all times.

In the second part of this work, we study batch sparse recovery problem in the presence of large and low rank noise, which is also known as the problem of Robust Principal Components Analysis (RPCA). In recent work, RPCA has been posed as a problem of recovering a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ from their sum, $\mathbf{M} := \mathbf{L} + \mathbf{S}$ and a provably exact convex optimization solution called PCP has been proposed. We study the following problem. Assume that we have a partial estimate of the column space of the low rank matrix $\mathbf{L}$, we propose here a simple but useful modification of the PCP idea, called modified-PCP, that allows us to use this knowledge. We derive its correctness result which shows that modified-PCP indeed requires significantly weaker incoherence assumptions than PCP, when the available subspace knowledge is accurate.

In the third part of this work, we study the "online" sparse recovery problem in the presence of low rank noise and bounded noise, which is also known as the "online" RPCA problem. Here we study a more general version of this problem, where the goal is to recover low rank matrix $\mathbf{L}$ and sparse matrix $\mathbf{S}$ from $\mathbf{M} := \mathbf{L} + \mathbf{S} + \mathbf{W}$ and $\mathbf{W}$ is the matrix of unstructured small noise. We develop and study a novel "online" RPCA algorithm based on the recently introduced Recursive Projected Compressive Sensing

(ReProCS) framework. The key contribution is a correctness result for this algorithm under relatively mild assumptions.

CHAPTER 1. INTRODUCTION

The static sparse reconstruction problem has been studied for a while [2, 3, 4, 5, 6]. The papers on compressive sensing (CS) from 2005 [7, 8, 9, 10, 11, 12] (and many other more recent works) provide the missing theoretical guarantees – conditions for exact recovery and error bounds when exact recovery is not possible. In more recent works, the problem of recursively recovering a time sequence of sparse signals, with slowly changing sparsity patterns has also been studied [13, 1, 14, 15, 16, 17, 18, 19]. By "recursive" reconstruction, we mean that we want to use only the current measurements' vector and the previous reconstructed signal to recover the current signal. This problem occurs in many applications such as real-time dynamic magnetic resonance imaging (MRI); single-pixel camera based real-time video imaging; recursively separating the region of the brain that is activated in response to a stimulus from brain functional MRI (fMRI) sequences [20] and recursively extracting sparse foregrounds (e.g. moving objects) from slow-changing (low-dimensional) backgrounds in video sequences [21]. For other potential applications, see [22, 23].

An important assumption introduced and empirically verified in [13, 1] is that for many natural signal/image sequences, the sparsity pattern (support set of its projection into the sparsity basis) changes slowly over time. In [14], the authors exploited this fact to reformulate the above problem as one of sparse recovery with partially known support and introduced a solution approach called modified-CS. Given the partial support knowledge $\mathcal{T}$, modified-CS tries to find a signal that is sparsest outside of $\mathcal{T}$ among all signals that satisfy the data constraint. Exact recovery conditions were obtained for modified-CS

and it was argued that these are weaker than those for simple ℓ_1 minimization (basis pursuit) under the slow support change assumption. Related ideas for support recovery with prior knowledge about the support entries, that appeared in parallel, include [24], [25]. All of [14], [24] and [25] studied the noise-free measurements' case. Later work includes [26, 27].

Error bounds for modified-CS for noisy measurements were obtained in [28], [29], [30]. When modified-CS is used for recursive reconstruction, these bounds tell us that the reconstruction error bound at the current time is proportional to the support recovery error (misses and extras in the support estimate) from the previous time. Unless we impose extra conditions, this support error can keep increasing over time, in which case the bound is not useful. Thus, for recursive reconstruction, the important question is, under what conditions can we obtain time-invariant bounds on the support error (which will, in turn, imply time-invariant bounds on the reconstruction error)? In other words, when can we ensure "stability" over time? Notice that, even if we did nothing, i.e. we set $\hat{\mathbf{x}}_t = 0$, the support error will be bounded by the support size. If the support size is bounded, then this is a naive stability result too, but is not useful. Here, we look for results in which the support error bound is small compared to the support size.

Stability over time has not been studied much for recursive recovery of sparse signal sequences. To the best of our knowledge, it has only been addressed in [1], and in very recent work [19]. The result of [19] is for exact dynamic support recovery in the noise-free case and it studies a different problem: the MMV version of the recursive recovery problem. The result from [1] for Least Squares CS-residual (LS-CS) stability) holds under mostly mild assumptions; its one limitation is that it assumes that support changes occur every p frames. But from testing the slow support change assumption for real data (medical image sequences), it has been observed that support changes usually occur at *every* time, e.g. Fig. 1.1. *This important case is the focus of current work.* We explain the differences of our results w.r.t. the LS-CS result in detail later in Sec 2.5.5.

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. Accurately computing the PCs in the presence of outliers is called robust PCA. 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 [31], an outlier can be nicely modeled as a sparse vector. The robust PCA problem occurs in various applications ranging from video analysis to recommender system design in the presence of outliers, e.g. for Netflix movies, to anomaly detection in dynamic networks [32]. In video analysis, background image sequences are well modeled as forming a low-rank but dense matrix because they change slowly over time and the changes are typically global. Foreground is a sparse image consisting of one or more moving objects. In recent work, Candes et al and Chandrasekharan et al [32, 33] posed the robust PCA problem as one of separating a low-rank matrix $\mathbf{L}$ (true data matrix) and a sparse matrix $\mathbf{S}$ (outliers' matrix) from their sum, $\mathbf{M} := \mathbf{L} + \mathbf{S}$. They showed that by solving the following convex optimization called principal components' pursuit (PCP)

$$\begin{aligned} &\text{minimize}_{\tilde{\mathbf{L}},\tilde{\mathbf{S}}} \quad \|\tilde{\mathbf{L}}\|_* + \lambda\|\tilde{\mathbf{S}}\|_1 \\ &\text{subject to} \quad \tilde{\mathbf{L}} + \tilde{\mathbf{S}} = \mathbf{M} \end{aligned} \tag{1.1}$$

it is possible to recover $\mathbf{L}$ and $\mathbf{S}$ exactly with high probability under mild assumptions. This was among the first recovery guarantees for a practical (polynomial complexity) robust PCA algorithm. Since then, the batch robust PCA problem, or what is now also often called the sparse+low-rank recovery problem, has been studied extensively, e.g. see [34, 35, 36, 37, 38, 39, 40, 41, 42, 43].

In this work, we introduce modified-CS-add-LS-del which is a modified-CS based algorithm for recursive recovery with an improved support estimation step and we explain how to set its parameters in practice. The main contribution of this work is to obtain conditions for stability of modified-CS and modified-CS-add-LS-del for recursive recovery

of a time sequence of sparse signals. Under mild assumptions, we show that the support recovery error and the reconstruction error of both algorithms is bounded by a time-invariant value at all times. The support error bound is proportional to the maximum allowed support change size. Under slow support change, this bound is small compared to the support size, making our result meaningful. Similar arguments can be made for the reconstruction error also. The assumptions we need are: weaker restricted isometry property (RIP) conditions [10] on the measurement matrix than what ℓ_1 minimization for noisy data (henceforth referred to as *noisy* ℓ_1) needs; bounded cardinality of the support and support change; all but a small number of existing nonzero entries are above a threshold in magnitude; appropriately set support estimation thresholds; and a special start condition. Here and elsewhere in the paper noisy ℓ_1 (or simple CS) refers to the solution of (2.1).

A second main contribution of this work is to show two examples of signal change assumptions under which the required conditions hold and prove stability results for these. The first case is a simple signal change model that helps to illustrate the key ideas and allows for easy comparison of the results. The second set of assumptions is realistic, but more complicated to state. We use MRI image sequences to demonstrate that these assumptions are indeed valid for real data. The essential requirement in both cases is that, for any new element that is added to the support, either its initial magnitude is large enough, or for the first few time instants, its magnitude increases at a large enough rate; and a similar assumption for magnitude decrease and removal from the support.

Let S be the bound on the maximum support size and S_a the bound on the maximum number of support additions or removals. All our results require s-RIP to hold with $s = S + kS_a$ where k is a constant. On the other hand, noisy ℓ_1 needs s-RIP for $s = 2S$ [12] which is a stronger requirement when $S_a \ll S$ (slow support change).

In the second part we study the following problem. Suppose that we have a partial estimate of the column space of the low rank matrix L. How can we use this information

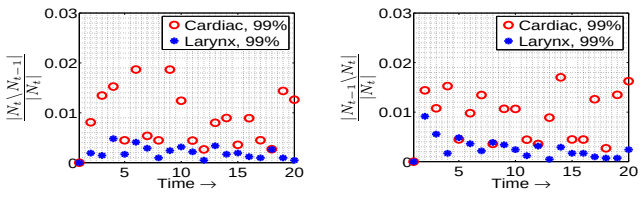


Figure 1.1: Slow support change in medical image sequences. We used the two-level Daubechies-4 2D discrete wavelet transform (DWT) as the sparsity basis. Since real image sequences are only approximately sparse, we use N_t to denote the 99%-energy support of the DWT of these sequences. The support size, $|N_t|$, was 6-7% of the image size for both sequences. We plot the number of additions (left) and the number of removals (right) as a fraction of $|N_t|$. *Notice that all changes are less than 2% of the support size.*

to improve the PCP solution, i.e. allow recovery under weaker assumptions? We propose here a simple but useful modification of the PCP idea, called *modified-PCP*, that allows us to use this knowledge. We derive its correctness result (Theorem 3.1.1) that provides explicit bounds on the various constants and on the matrix size that are needed to ensure exact recovery with high probability. Our result is used to argue that modified-PCP indeed requires significantly weaker incoherence assumptions than PCP, as long as the available subspace knowledge is accurate. To prove the result, we use the overall proof approach of [32] with some changes (see Sec 3.3).

An important problem where partial subspace knowledge is available is in online or recursive robust PCA for sequentially arriving time series data, e.g. for video based foreground and background separation. In this case, as explained in [44], the subspace spanned by a set of consecutive columns of L does not remain fixed, but instead changes over time and the changes are gradual. Also, often an initial short sequence of low-rank only data (without outliers) is available, e.g. in video analysis, it is easy to get an initial background-only sequence. For this application, modified-PCP can be used to design a piecewise batch solution that will be faster and will require weaker assumptions for exact recovery than PCP. This is made precise in Corollary 3.2.1 and the discussion below it.

We show extensive simulation comparisons and some real data comparisons of modified-PCP with PCP and with other existing robust PCA solutions from literature. The im-

plementation requires a fast algorithm for solving the modified-PCP program. This is developed by modifying the Inexact Augmented Lagrange Multiplier Method [45] and using the idea of [46, 47] for the sparse recovery step. The real data comparisons are for a face reconstruction / recognition application in the presence of outliers, e.g. eye-glasses or occlusions, that is also discussed in [32].

When RPCA needs to be solved in a recursive fashion for sequentially arriving data vectors it is referred to as online RPCA. Our "online" RPCA formulation assumes that (i) a short sequence of outlier-free (sparse component free) data vectors is available or that there is another way to get an estimate of the initial subspace of the true data (without outliers); and that (ii) the subspace from which $\boldsymbol{\ell}_t$ is generated is fixed or changes slowly over time. We put "online" in quotes here to stress that our problem formulation uses extra assumptions beyond what are used by RPCA (or batch RPCA). A key application of RPCA is the problem of separating a video sequence into foreground and background layers [32]. Video layering is a key first step to simplifying many video analytics and computer vision tasks, e.g., video surveillance (to track moving foreground objects), background video recovery and subspace tracking in the presence of frequent foreground occlusions or low-bandwidth mobile video chats or video conferencing (can transmit only the foreground layer). In videos, the foreground typically consists of one or more moving persons or objects and hence is a sparse image. The background images (in a static camera video) usually change only gradually over time, e.g., moving lake waters or moving trees in a forest, and the changes are global [32]. Hence they are well modeled as being dense and lying in a low-dimensional subspace that is fixed or slowly changing. Other applications where RPCA occurs include recommendation system design, survey data analysis, anomaly detection in dynamic social (or computer) networks [32] and dynamic magnetic resonance imaging (MRI) based region-of-interest tracking [48].

1.1 Notation and Problem Definition

1.1.1 Notation

We use bold lowercase for vectors, bold uppercase for matrices, calligraphic uppercase for sets or corresponding linear space.

We let $[1, m] := [1, 2, \dots m]$. We let $\emptyset$ denote an empty set. We use $\mathcal{T}^c$ to denote the complement of a set $\mathcal{T}$ w.r.t. $[1, m]$, i.e. $\mathcal{T}^c := \{i \in [1, m] : i \notin \mathcal{T}\}$. We use $|\mathcal{T}|$ to denote the cardinality of $\mathcal{T}$. The set operations $\cup$, $\cap$, $\setminus$ have their usual meanings (recall that $\mathcal{A} \setminus \mathcal{B} := \mathcal{A} \cap \mathcal{B}^c$). If two sets $\mathcal{B}$, $\mathcal{C}$ are disjoint, we just write $\mathcal{D} \cup \mathcal{B} \setminus \mathcal{C}$ instead of writing $(\mathcal{D} \cup \mathcal{B}) \setminus \mathcal{C}$.

For a vector, $\mathbf{x}$, and a set, $\mathcal{T}$, $\mathbf{x}_\mathcal{T}$ denotes the $|\mathcal{T}|$ length sub-vector containing the elements of $\mathbf{x}$ corresponding to the indices in the set $\mathcal{T}$. $\|\mathbf{x}\|_k$ denotes the ℓ_k norm of a vector $\mathbf{x}$. *If just* $\|\mathbf{x}\|$ *is used, it refers to* $\|\mathbf{x}\|_2$. Similarly, for a matrix $\mathbf{M}$, $\|\mathbf{M}\|_k$ denotes its induced k-norm, while just $\|\mathbf{M}\|$ refers to $\|\mathbf{M}\|_2$. $\mathbf{M}'$ denotes the transpose of $\mathbf{M}$ and $\mathbf{M}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{M}$ (when $\mathbf{M}$ is full column rank, $\mathbf{M}^\dagger := (\mathbf{M}'\mathbf{M})^{-1}\mathbf{M}'$). Also, $\mathbf{M}_\mathcal{T}$ denotes the sub-matrix obtained by extracting the columns of $\mathbf{M}$ corresponding to indices in $\mathcal{T}$.

We refer to the left (right) hand side of an equation or inequality as LHS (RHS).

For a matrix $\mathbf{X}$, we denote by $\mathbf{X}^*$ the transpose of $\mathbf{X}$; denote by $\|\mathbf{X}\|_\infty$ the ℓ_∞ norm of $\mathbf{X}$ reshaped as a long vector, i.e., $\max_{i,j} |\mathbf{X}_{ij}|$; denote by $\|\mathbf{X}\|$ the operator norm or 2-norm; denote by $\|\mathbf{X}\|_F$ the Frobenius norm; denote by $\|\mathbf{X}\|_*$ the nuclear norm; denote by $\|\mathbf{X}\|_1$ the ℓ_1 norm of $\mathbf{X}$ reshaped as a long vector.

Let $\mathbb{P}_\mathbf{I}$ denote the identity operator, i.e., $\mathbb{P}_\mathbf{I}(\mathbf{Y}) = \mathbf{Y}$ for any matrix $\mathbf{Y}$. Let $\|\mathbb{P}_\mathcal{A}\|$ denote the operator norm of operator $\mathbb{P}_\mathcal{A}$, i.e., $\|\mathbb{P}_\mathcal{A}\| = \sup_{\{\|\mathbf{X}\|_F = 1\}} \|\mathbb{P}_\mathcal{A}\mathbf{X}\|_F$; let $\langle \mathbf{X}, \mathbf{Y} \rangle$ denote the Euclidean inner product between two matrices, i.e., $\text{trace}(\mathbf{X}^*\mathbf{Y})$; let $\text{sgn}(\mathbf{X})$ denote the entrywise sign of $\mathbf{X}$. We let $\mathcal{P}_\Theta$ denote the orthogonal projection onto linear subspace Θ. We use Ω to denote the support set of matrix $\mathbf{S}$, i.e., $\Omega = \{(i, j) : \mathbf{S}(i, j) \neq$

0}. We also use Ω to denote the subspace spanned by all matrices supported on Ω. By $\Omega \sim \text{Ber}(\rho)$ we mean that any matrix index (i, j) has probability ρ of being in the support independent of all others.

Given two matrices $\mathbf{B}$ and $\mathbf{B}_2$, $[\mathbf{B}\ \mathbf{B}_2]$ constructs a new matrix by concatenating matrices $\mathbf{B}$ and $\mathbf{B}_2$ in the horizontal direction. Let $\mathbf{B}_{\text{rem}}$ be a matrix containing some columns of $\mathbf{B}$. Then $\mathbf{B} \setminus \mathbf{B}_{\text{rem}}$ is the matrix $\mathbf{B}$ with columns in $\mathbf{B}_{\text{rem}}$ removed.

We say that $\mathbf{U}$ is a *basis matrix* if $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. We use $\mathbf{e}_i$ to refer to the i^{th} column $\mathbf{I}$.

We use the interval notation $[a, b]$ to mean all of the integers between a and b, inclusive, and similarly for $[a, b)$ etc. For a set $\mathcal{T}$, $|\mathcal{T}|$ denotes its cardinality and $\bar{\mathcal{T}}$ denotes its complement set. We use $\emptyset$ to denote the empty set. For a vector $\mathbf{x}$, $\mathbf{x}_{\mathcal{T}}$ is a smaller vector containing the entries of $\mathbf{x}$ indexed by $\mathcal{T}$. Define $\mathbf{I}_{\mathcal{T}}$ to be an $n \times |\mathcal{T}|$ matrix of those columns of the identity matrix indexed by $\mathcal{T}$. For a matrix $\mathbf{A}$, define $\mathbf{A}_{\mathcal{T}} := \mathbf{A}\mathbf{I}_{\mathcal{T}}$. We use $'$ to denote transpose. The l_p-norm of a vector and the induced l_p-norm of a matrix are denoted by $\|\cdot\|_p$. We refer to a matrix with orthonormal columns as a *basis matrix*. Thus, for a basis matrix $\mathbf{P}$, $\mathbf{P}'\mathbf{P} = \mathbf{I}$. For matrices $\mathbf{P}$, $\mathbf{Q}$ where the columns of $\mathbf{Q}$ are a subset of the columns of $\mathbf{P}$, $\mathbf{P} \setminus \mathbf{Q}$ refers to the matrix of columns in $\mathbf{P}$ and not in $\mathbf{Q}$. For a matrix $\mathbf{H}$, $\mathbf{H} \overset{\text{EVD}}{=} \mathbf{U}\mathbf{\Lambda}\mathbf{U}'$ denotes its reduced eigenvalue decomposition. For a matrix $\mathbf{A}$, the restricted isometry constant (RIC) $\delta_s(\mathbf{A})$ is the smallest real number δ_s such that

$$(1 - \delta_s)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_s)\|\mathbf{x}\|_2^2$$

for all s-sparse vectors $\mathbf{x}$ [12]. A vector $\mathbf{x}$ is s-sparse if it has s or fewer non-zero entries. For Hermitian matrices $\mathbf{A}$ and $\mathbf{B}$, the notation $\mathbf{A} \preceq \mathbf{B}$ means that $\mathbf{B} - \mathbf{A}$ is positive semi-definite. For basis matrices $\hat{\mathbf{P}}$ and $\mathbf{P}$, $\text{dif}(\hat{\mathbf{P}}, \mathbf{P}) := \|(\mathbf{I} - \hat{\mathbf{P}}\hat{\mathbf{P}}')\mathbf{P}\|_2$ quantifies error between their range spaces.

1.1.2 Problem definition

The first type of problems that we study here are as following. We assume the following observation model:

$$\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t + \mathbf{w}_t, \ \|\mathbf{w}_t\| \leq \epsilon \tag{1.2}$$

where $\mathbf{x}_t$ is an m length sparse vector with support set $\mathcal{N}_t$, i.e. $\mathcal{N}_t := \{i : (\mathbf{x}_t)_i \neq 0\}$; $\mathbf{A}_t$ is a $n_t \times m$ measurement matrix; $\mathbf{y}_t$ is the n_t length observation vector at time t (with $n_t < m$); and $\mathbf{w}_t$ is the observation noise. For $t > 0$, we fix $n_t = n$.

Our goal is to recursively estimate $\mathbf{x}_t$ using $\mathbf{y}_1, \ldots \mathbf{y}_t$. By *recursively*, we mean, use only $\mathbf{y}_t$ and the estimate from $t-1$, $\hat{\mathbf{x}}_{t-1}$, to compute the estimate at t.

Remark 1.1.1 (Why bounded noise)**.** *All results for bounding ℓ_1 minimization error in noise, and hence all results for bounding modified-CS error in noise, either assume a deterministic noise bound and then bound $\|\hat{\mathbf{x}} - \mathbf{x}\|$, e.g., [12], [49], [28, 50], [51]; or assume unbounded, e.g. Gaussian, noise and then bound $\|\hat{\mathbf{x}} - \mathbf{x}\|$ with "large" probability, e.g. [52], [53, Sec IV], [1, Section III-A], [51]. The latter approach is not useful for recovering a time sequence of sparse signals because the error bound will hold for all times $0 \leq t < \infty$ with probability zero.*

One way to get a meaningful error stability result with unbounded, e.g. Gaussian noise, is to compute or bound the expected value of the error at each time, i.e. compute $\mathbb{E}[(\hat{\mathbf{x}}_t - \mathbf{x}_t)(\hat{\mathbf{x}}_t - \mathbf{x}_t)']$ or bound some norm of it. This is possible to do, for example, for a Kalman filter applied to a linear system model with additive Gaussian noise; and hence in that case, one can assume Gaussian noise and still get a time-invariant bound on the expected value of the error under mild assumptions. However, for ℓ_1 minimization based methods, such as modified-CS, there is no easy way to compute or bound the expected value of the error. Moreover, even if one could do this for a given time, it would not tell us anything about the support recovery error (for the given noise sequence realization) and hence would not be useful for analyzing modified-CS.

As a sidenote, we should point out that, in most applications, the noise is typically bounded (because of finite sensing power available). One often chooses to model the noise as Gaussian because it simplifies performance analysis.

The second type of problems that we study here are as following. We are given a data matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ that satisfies

$$\mathbf{M} = \mathbf{L} + \mathbf{S} \tag{1.3}$$

where $\mathbf{S}$ is a sparse matrix with support set Ω and $\mathbf{L}$ is a low rank matrix with rank r and with reduced singular value decomposition (SVD)

$$\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \tag{1.4}$$

We assume that we are given an $n_1 \times r_{\mathbf{G}}$ basis matrix $\mathbf{G}$ so that $(I - \mathbf{G}\mathbf{G}^*)\mathbf{L}$ has rank smaller than r. The goal is to recover $\mathbf{L}$ and $\mathbf{S}$ from $\mathbf{M}$ using $\mathbf{G}$.

We explain the above a little more. With $\mathbf{G}$ as above, $\mathbf{U}$ can be rewritten as

$$\mathbf{U} = [\underbrace{(\mathbf{G}\mathbf{R} \setminus \mathbf{U}_{\text{extra}})}_{\mathbf{U}_0} \ \mathbf{U}_{\text{new}}], \tag{1.5}$$

where $\mathbf{U}_{\text{new}} \in \mathbb{R}^{n_1 \times r_{\text{new}}}$ and $\mathbf{U}_{\text{new}}^* \mathbf{G} = 0$; $\mathbf{R}$ is a rotation matrix and $\mathbf{U}_{\text{extra}}$ contains r_{extra} columns of $\mathbf{G}\mathbf{R}$. Let r_0 be the number of columns in $\mathbf{U}_0$. Then, clearly, $r_0 = r_{\mathbf{G}} - r_{\text{extra}}$ and $r = r_0 + r_{\text{new}}$.

We use $\mathbf{V}_{\text{new}}$ to denote the right singular vectors of the reduced SVD of $\mathbf{L}_{\text{new}} := (\mathbf{I} - \mathbf{G}\mathbf{G}^*)\mathbf{L} = \mathbf{U}_{\text{new}}\mathbf{U}_{\text{new}}^*\mathbf{L}$. In other words,

$$\mathbf{L}_{\text{new}} := (\mathbf{I} - \mathbf{G}\mathbf{G}^*)\mathbf{L} \overset{\text{SVD}}{=} \mathbf{U}_{\text{new}}\mathbf{\Sigma}_{\text{new}}\mathbf{V}_{\text{new}}^* \tag{1.6}$$

From the above model, it is clear that

$$\mathbf{L}_{\text{new}} + \mathbf{G}\mathbf{X}^* + \mathbf{S} = \mathbf{M} \tag{1.7}$$

for $\mathbf{X} = \mathbf{L}^*\mathbf{G}$. We propose to recover $\mathbf{L}$ and $\mathbf{S}$ using $\mathbf{G}$ by solving the following **Modified PCP** (mod-PCP) program

$$\begin{aligned} &\text{minimize}_{\tilde{\mathbf{L}}_{\text{new}},\tilde{\mathbf{S}},\tilde{\mathbf{X}}} && \|\tilde{\mathbf{L}}_{\text{new}}\|_* + \lambda\|\tilde{\mathbf{S}}\|_1 \\ &\text{subject to} && \tilde{\mathbf{L}}_{\text{new}} + \mathbf{G}\tilde{\mathbf{X}}^* + \tilde{\mathbf{S}} = \mathbf{M} \end{aligned} \tag{1.8}$$

Denote a solution to the above by $\hat{\mathbf{L}}_{\text{new}}, \hat{\mathbf{S}}, \hat{\mathbf{X}}$. Then, $\mathbf{L}$ is recovered as $\hat{\mathbf{L}} = \hat{\mathbf{L}}_{\text{new}} + \mathbf{G}\hat{\mathbf{X}}^*$. Modified-PCP is inspired by an approach for sparse recovery using partial support knowledge called modified-CS [54].

The third type of problems that we study here are as following. At time t we observe a data vector $\mathbf{m}_t \in \mathcal{R}^n$ that satisfies

$$\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{x}_t + \mathbf{w}_t, \text{ for } t = t_{\text{train}} + 1, t_{\text{train}} + 2, \ldots, t_{\max}. \tag{1.9}$$

For $t = 1, 2, \ldots, t_{\text{train}}$, $\mathbf{x}_t = 0$, i.e., $\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{w}_t$. Here $\boldsymbol{\ell}_t$ is a vector that lies in a low-dimensional subspace that is fixed or slowly changing in such a way that the matrix $\mathbf{L}_t := [\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots, \boldsymbol{\ell}_t]$ is a low-rank matrix for all but very small values of t; $\mathbf{x}_t$ is a sparse (outlier) vector; and $\mathbf{w}_t$ is small modeling error or noise. We use $\mathcal{T}_t$ to denote the support set of $\mathbf{x}_t$ and we use $\mathbf{P}_t$ to denote a basis matrix for the subspace from which $\boldsymbol{\ell}_t$ is generated. For $t > t_{\text{train}}$, the goal of online RPCA is to recursively estimate $\boldsymbol{\ell}_t$ and its subspace range($\mathbf{P}_t$), and $\mathbf{x}_t$ and its support, $\mathcal{T}_t$, as soon as a new data vector $\mathbf{m}_t$ arrives or within a short delay [1]. Sometimes, e.g., in video analytics, it is often also desirable to get an improved offline estimate of $\mathbf{x}_t$ and $\boldsymbol{\ell}_t$ when possible. We show that this is an easy by-product of our solution approach.

The initial t_{train} outlier-free measurements are used to get an accurate estimate of the initial subspace via PCA. For video surveillance, this assumption corresponds to having a short initial sequence of background only images, which can often be obtained.

In many applications, it is actually the sparse outlier $\mathbf{x}_t$ that is the quantity of interest. The above problem can thus also be interpreted as one of *online sparse matrix recovery*

[1] By definition, a subspace of dimension $r > 1$ cannot be estimated immediately since it needs at least r data points to estimate

in large but structured noise $\boldsymbol{\ell}_t$ *and unstructured small noise* $\mathbf{w}_t$. The unstructured noise, $\mathbf{w}_t$, often models the modeling error. For example, when some of the corruptions/outliers are small enough to not significantly increase the subspace recovery error, these can be included into $\mathbf{w}_t$ rather than $\mathbf{x}_t$. Another example is when the $\boldsymbol{\ell}_t$'s form an approximately low-rank matrix.

1.2 Dissertation Organization

The dissertation is organized as follows. Recursive sparse recovery in bounded noise and corresponding results are introduced in Chapter 2. Batch sparse recovery in large and structured noise and corresponding results are discussed in Chapter 3. Recursive (online) sparse recovery in large and structured noise and bounded noise and correspond results are demonstrated in Chapter 4. Finally, conclusions are summarized in Chapter 5.

CHAPTER 2. RECURSIVE SPARSE RECOVERY IN BOUNDED NOISE

2.1 Related Work And Organization

"Recursive sparse reconstruction" also sometimes refers to homotopy methods, e.g. [55], whose goal is to use the past reconstructions and homotopy to speed up the current optimization, but not to achieve accurate recovery from fewer measurements than what noisy ℓ_1 needs. The goals in the above works are quite different from ours.

Iterative support estimation approaches (using the recovered support from the first iteration for a second weighted ℓ_1 step and doing this iteratively) have been studied in recent work [56, 57, 58, 59]. This is done for iteratively improving the recovery of a *single* signal.

This chapter is organized as follows. The algorithms – modified-CS and modified-CS-add-LS-del – are introduced in Sec 2.2. This section also includes definitions for certain quantities and sets used later in the paper. In Sec 2.3, we provide stability results for modified-CS and modified-CS-add-LS-del that do not assume anything about signal change over time except a bound on the number of small magnitude nonzero coefficients and a bound on maximum number of support additions or removals per unit time. In Sec 2.4, we give a simple set of signal change assumptions and give stability results for both algorithms under these and other simple assumptions. In Sec 2.5, we do the same for a realistic signal change model. The results are discussed in Sec 2.4.4 and 2.5.4 respectively. In Sec 2.6, we demonstrate that the signal model assumptions of

Sec 2.5 are indeed valid for medical imaging data. In Sec 4.4, we explain how to set the algorithm parameters automatically for both modified-CS and modified-CS-add-LS-del. In this section, we also give simulation experiments that back up some of our discussions from earlier sections.

2.2 Modified-CS And Modified-CS-add-LS-del For Recursive Reconstruction

2.2.1 Modified-CS

Modified-CS was first proposed in [14] as a solution to the problem of sparse reconstruction with partial, and possibly erroneous, knowledge of the support. Denote this "known" support by $\mathcal{T}$. Modified-CS tries to find a signal that is sparsest outside of the set $\mathcal{T}$ among all signals satisfying the data constraint. In the noisy case, it solves $\min_\beta \|(\beta)_{\mathcal{T}^c}\|_1$ s.t. $\|\mathbf{y}_t - \mathbf{A}\beta\| \leq \epsilon$. For recursively reconstructing a time sequence of sparse signals, we use the support estimate from the previous time, $\hat{\mathcal{N}}_{t-1}$, as the set $\mathcal{T}$. The simplest way to estimate the support is by thresholding the output of modified-CS. We summarize the complete algorithm in Algorithm 1.

At the initial time, $t = 0$, we let $\mathcal{T}$ be the empty set, $\emptyset$, i.e. we solve noisy ℓ_1. Alternatively, as explained in [14], we can use prior knowledge of the initial signal's support as the set $\mathcal{T}$ at $t = 0$, e.g. for wavelet sparse images with no (or a small) black background, the set of indices of the approximation coefficients can form the set $\mathcal{T}$. This prior knowledge is usually not as accurate.

We explain how the parameter α can be set in practice in Sec 2.7.1.

2.2.2 Limitation: biased solution

Modified-CS uses single step thresholding for estimating the support $\hat{\mathcal{N}}_t$. The threshold, α, needs to be large enough to ensure that all (or most) removed elements are

Algorithm 1 Modified-CS

For $t \geq 0$, do

1. *Noisy ℓ_1.* If $t = 0$, set $\mathcal{T}_t = \emptyset$ and compute $\hat{\mathbf{x}}_{t,modcs}$ as the solution of

$$\min_{\beta} \|(\beta)\|_1 \text{ s.t. } \|\mathbf{y}_0 - \mathbf{A}_0\beta\| \leq \epsilon \tag{2.1}$$

2. *Modified-CS.* If $t > 0$, set $\mathcal{T}_t = \hat{\mathcal{N}}_{t-1}$ and compute $\hat{\mathbf{x}}_{t,modcs}$ as the solution of

$$\min_{\beta} \|(\beta)_{\mathcal{T}_t^c}\|_1 \text{ s.t. } \|\mathbf{y}_t - \mathbf{A}_t\beta\| \leq \epsilon \tag{2.2}$$

3. *Estimate the Support.* Compute $\tilde{\mathcal{T}}_t$ as

$$\tilde{\mathcal{T}}_t = \{i \in [1, m] : |(\hat{\mathbf{x}}_{t,modcs})_i| > \alpha\} \tag{2.3}$$

4. Set $\hat{\mathcal{N}}_t = \tilde{\mathcal{T}}_t$. Output $\hat{\mathbf{x}}_{t,modcs}$. Feedback $\hat{\mathcal{N}}_t$.

correctly deleted and there are no (or very few) false detections. But this means that the new additions to the support set will either have to be added at a large value, or their magnitude will need to increase to a large value quickly enough to ensure correct detection within a small delay. This issue is further exaggerated by the fact that $\hat{\mathbf{x}}_{t,modcs}$ is a biased estimate of $\mathbf{x}_t$. Along $\mathcal{T}_t^c$, the values of $\hat{\mathbf{x}}_{t,modcs}$ will be biased toward zero (because we minimize $\|(\beta)_{\mathcal{T}_t^c}\|_1$), while, along $\mathcal{T}_t$, they may be biased away from zero. This will create the following problem. The set $\mathcal{T}_t$ contains the set $\Delta_{e,t}$ which needs to be deleted. Since the estimates along $\Delta_{e,t}$ may be biased away from zero, one will need a higher threshold to delete them. But that would make detection more difficult, especially since the estimates along $\Delta_t \subseteq \mathcal{T}_t^c$ will be biased towards zero. A similar issue for noisy CS, and a possible solution (Gauss-Dantzig selector), was first discussed in [52].

2.2.3 Modified-CS with Add-LS-Del

The bias issue can be partly addressed by replacing the support estimation step of Modified-CS by a three step Add-LS-Del procedure summarized in Algorithm 2. It

involves a support addition step (that uses a smaller threshold - α_{add}), as in (2.4), followed by LS estimation on the new support estimate, $\mathcal{T}_{\text{add},t}$, as in (2.5), and then a deletion step that thresholds the LS estimate, as in (2.6). This can be followed by a second LS estimation using the final support estimate, as in (2.7), although this last step is not critical. The addition step threshold, α_{add}, needs to be just large enough to ensure that the matrix used for LS estimation, $A_{\mathcal{T}_{\text{add},t}}$ is well-conditioned. If α_{add} is chosen properly and if n is large enough, the LS estimate on $\mathcal{T}_{\text{add},t}$ will have smaller error and will be less biased than the modified-CS output. As a result, deletion will be more accurate when done using this estimate. This also means that one can use a larger deletion threshold, α_{del}, which will ensure quicker deletion of extras.

Related ideas were introduced in our older work [1, 13] for KF-CS and LS-CS, and in [60, 49] for a greedy algorithm for static sparse reconstruction.

We explain how to automatically set the parameters for both modified-CS-add-LS-del and modified-CS in Sec 2.7.1.

2.2.4 Some definitions

Definition 2.2.1. *For any matrix, A, the left S-restricted isometry constant (left-RIC) $\delta_{S,left}(A)$ and right S-restricted isometry constant (right-RIC) $\delta_{S,right}(A)$ are the smallest real numbers satisfying*

$$(1 - \delta_{S,left}(A))\|c\|^2 \leq \|A_{\mathcal{T}}c\|^2 \leq (1 + \delta_{S,right}(A))\|c\|^2 \tag{2.8}$$

for all sets $\mathcal{T} \subset [1, m]$ of cardinality $|\mathcal{T}| \leq S$ and all real vectors c of length $|\mathcal{T}|$. The restricted isometry constant (RIC)[10] is the larger of the two, i.e.,

$$\delta_S = \max\{\delta_{S,left}(A), \delta_{S,right}(A)\}.$$

Definition 2.2.2. *The restricted orthogonality constant (ROC) [10], $\theta_{S_1,S_2}(A)$, is the smallest real number satisfying*

$$|{c_1}' {A_{\mathcal{T}_1}}' A_{\mathcal{T}_2} c_2| \leq \theta_{S_1,S_2}\|c_1\| \ \|c_2\| \tag{2.9}$$

Algorithm 2 Modified-CS-Add-LS-Del

For $t \geq 0$, do

1. *Noisy ℓ_1.* If $t = 0$, set $\mathcal{T}_t = \emptyset$ and compute $\hat{\mathbf{x}}_{t,modcs}$ as the solution of (2.1).
2. *Modified-CS.* If $t > 0$, set $\mathcal{T}_t = \hat{\mathcal{N}}_{t-1}$ and compute $\hat{\mathbf{x}}_{t,modcs}$ as the solution of (2.2).
3. *Additions / LS.* Compute $\mathcal{T}_{\text{add},t}$ and the LS estimate using it:

$$\hat{\mathcal{A}}_t = \{i : |(\hat{\mathbf{x}}_{t,modcs})_i| > \alpha_{\text{add}}\}$$
$$\mathcal{T}_{\text{add},t} = \mathcal{T}_t \cup \hat{\mathcal{A}}_t \tag{2.4}$$
$$(\hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}} = A_{\mathcal{T}_{\text{add},t}}{}^{\dagger} y_t, \quad (\hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}^c_{\text{add},t}} = 0 \tag{2.5}$$

4. *Deletions / LS.* Compute $\tilde{\mathcal{T}}_t$ and LS estimate using it:

$$\hat{\mathcal{R}}_t = \{i \in \mathcal{T}_{\text{add},t} : |(\hat{\mathbf{x}}_{t,\text{add}})_i| \leq \alpha_{\text{del}}\}$$
$$\tilde{\mathcal{T}}_t = \mathcal{T}_{\text{add},t} \setminus \hat{\mathcal{R}}_t \tag{2.6}$$
$$(\hat{\mathbf{x}}_t)_{\tilde{\mathcal{T}}_t} = A_{\tilde{\mathcal{T}}_t}{}^{\dagger} y_t, \quad (\hat{\mathbf{x}}_t)_{\tilde{\mathcal{T}}_t^c} = 0 \tag{2.7}$$

5. Set $\hat{\mathcal{N}}_t = \tilde{\mathcal{T}}_t$. Feedback $\hat{\mathcal{N}}_t$. Output $\hat{\mathbf{x}}_t$.

for all disjoint sets $\mathcal{T}_1, \mathcal{T}_2 \subset [1, m]$ with $|\mathcal{T}_1| \leq S_1$, $|\mathcal{T}_2| \leq S_2$ and $S_1 + S_2 \leq m$, and for all vectors c_1, c_2 of length $|\mathcal{T}_1|$, $|\mathcal{T}_2|$ respectively.

In this work, we need the same condition on the RIC and ROC of all measurement matrices A_t for $t > 0$. Thus, in the rest of this paper, we let

$$\delta_S := \max_{t>0} \delta_S(A_t), \text{ and } \theta_{S_1,S_2} := \max_{t>0} \theta_{S_1,S_2}(A_t).$$

If we need the RIC of ROC of any other matrix, then we specify it explicitly.

As seen above, we use α to denote the support estimation threshold used by modified-CS and we use $\alpha_{\text{add}}, \alpha_{\text{del}}$ to denote the support addition and deletion thresholds used by modified-CS-add-LS-del. We use $\hat{\mathcal{N}}_t$ to denote the support estimate at time t.

Definition 2.2.3 ($\mathcal{T}_t$, Δ_t, $\Delta_{e,t}$)**.** *We use $\mathcal{T}_t := \hat{\mathcal{N}}_{t-1}$ to denote the support estimate from the previous time. This serves as the predicted support at time t. We use $\Delta_t := \mathcal{N}_t \setminus \mathcal{T}_t$*

to denote the unknown part of support $\mathcal{N}_t$ *and* $\Delta_{e,t} := \mathcal{T}_t \setminus \mathcal{N}_t$ *to denote the "erroneous" part of support* $\mathcal{N}_t$.

With the above definition, clearly, $\mathcal{N}_t = \mathcal{T}_t \cup \Delta_t \setminus \Delta_{e,t}$.

Definition 2.2.4 ($\tilde{\mathcal{T}}_t$, $\tilde{\Delta}_t$, $\tilde{\Delta}_{e,t}$)**.** *We use* $\tilde{\mathcal{T}}_t := \hat{\mathcal{N}}_t$ *to denote the final estimate of the current support;* $\tilde{\Delta}_t := \mathcal{N}_t \setminus \tilde{\mathcal{T}}_t$ *to denote the "misses" in* $\hat{\mathcal{N}}_t$ *and* $\tilde{\Delta}_{e,t} := \tilde{\mathcal{T}}_t \setminus \mathcal{N}_t$ *to denote the "extras".*

Definition 2.2.5 (Define $\mathcal{T}_{\text{add},t}, \Delta_{\text{add},t}, \Delta_{e,\text{add},t}$)**.** *The set* $\mathcal{T}_{add,t}$ *is the support estimate obtained after the support addition step in Algorithm 2 (modified-CS-add-LS-del). It is defined in (2.4). The set* $\Delta_{add,t} := \mathcal{N}_t \setminus \mathcal{T}_{add,t}$ *denotes the set of missing elements from* $\mathcal{N}_t$ *and the set* $\Delta_{e,add,t} := \mathcal{T}_{add,t} \setminus \mathcal{N}_t$ *denotes the set of extras in it.*

Remark 2.2.6. *At certain places in the paper, we remove the subscript* t *for ease of notation.*

2.2.5 Modified-CS error bound at time t

By adapting the approach of [12], the error of modified-CS can be bounded as a function of $|\mathcal{T}_t| = |\mathcal{N}_t| + |\Delta_{e,t}| - |\Delta_t|$ and $|\Delta_t|$. This was done in [61]. We state a modified version here.

For completeness, we provide a proof for following lemma in Appendix A.0.1.

Lemma 2.2.7 (modified-CS error bound)**.** *Assume that* y_t *satisfies (1.2) and the support of* $\mathbf{x}_t$ *is* $\mathcal{N}_t$*. Consider step 2 of Algorithm 1 or 2. If* $\delta_{|\mathcal{T}_t|+3|\Delta_t|} = \delta_{|\mathcal{N}_t|+|\Delta_{e,t}|+2|\Delta_t|} < (\sqrt{2}-1)/2$*, then*

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \le C_1(|\mathcal{T}_t| + 3|\Delta_t|)\epsilon \le 7.50\epsilon, \; C_1(S) \triangleq \frac{4\sqrt{1+\delta_S}}{1-2\delta_S}.$$

Notice that the bound by $C_1(|\mathcal{T}_t| + 3|\Delta_t|)\epsilon$ will hold as long as $\delta_{|\mathcal{T}_t|+3|\Delta_t|} < 1/2$. By enforcing that $\delta_{|\mathcal{T}_t|+3|\Delta_t|} \leq 1/2c$ for a $c < 1$, we ensure that $C_1(.)$ is bounded by a fixed constant. To state the above lemma we pick $c = \sqrt{2} - 1$ and this gives $C_1(.) = 7.50$. We can state a similar result for CS [12].

Lemma 2.2.8 (CS error bound [12]). *Assume that y_t satisfies (1.2) and the support of $\mathbf{x}_t$ is $\mathcal{N}_t$. Let $\hat{\mathbf{x}}_{t,cs}$ denote the solution of (2.2) with $\mathcal{T}_t = \emptyset$. If $\delta_{2|\mathcal{N}_t|} < (\sqrt{2} - 1)/2$, then*

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,cs}\| \leq C_1(2|\mathcal{N}_t|)\epsilon \leq 7.50\epsilon$$

2.2.6 LS step error bound at time t

We can claim the following about the LS step error in step 3 of Algorithm 2.

Lemma 2.2.9. *Assume that y_t satisfies (1.2) and the support of $\mathbf{x}_t$ is $\mathcal{N}_t$. Consider step 3 of Algorithm 2.*

1. $(\mathbf{x}_t - \hat{\mathbf{x}}_{t,add})_{\mathcal{T}_{add,t}} = (A_{\mathcal{T}_{add,t}}{}' A_{\mathcal{T}_{add,t}})^{-1}[A_{\mathcal{T}_{add,t}}{}' \mathbf{w}_t + A_{\mathcal{T}_{add,t}}{}' A_{\Delta_{add,t}} (\mathbf{x}_t)_{\Delta_{add,t}}]$, $(\mathbf{x}_t - \hat{\mathbf{x}}_{t,add})_{\Delta_{add,t}} = (\mathbf{x}_t)_{\Delta_{add,t}}$, *and* $(\mathbf{x}_t - \hat{\mathbf{x}}_{t,add})_i = 0$, *if* $i \notin \mathcal{T}_{add,t} \cup \Delta_{add,t}$.

2. (a) $\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,add})_{\mathcal{T}_{add,t}}\| \leq \frac{1}{\sqrt{1-\delta_{|\mathcal{T}|}}}\epsilon + \frac{\theta_{|\mathcal{T}_{add,t}|,|\Delta_{add,t}|}}{1-\delta_{|\mathcal{T}|}}\|(\mathbf{x}_t)_{\Delta_{add,t}}\|$.

   (b) $\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,add})\| \leq \frac{1}{\sqrt{1-\delta_{|\mathcal{T}|}}}\epsilon + (1 + \frac{\theta_{|\mathcal{T}_{add,t}|,|\Delta_{add,t}|}}{1-\delta_{|\mathcal{T}|}})\|(\mathbf{x}_t)_{\Delta_{add,t}}\|$.

Proof: The first claim follows directly from the expression for $\hat{\mathbf{x}}_{t,\text{add}}$. The second claim uses the first claim and the facts that $||A_{\mathcal{T}}{}^{\dagger}||_2 \leq 1/\sqrt{1-\delta_{|\mathcal{T}|}}$, $||(A_{\mathcal{T}}{}' A_{\mathcal{T}})^{-1}|| \leq 1/(1-\delta_{|\mathcal{T}|})$ and $||A_{\mathcal{T}\cup\Delta}||_2 \leq \theta_{|\mathcal{T}|,|\Delta|}$ [1].

2.3 Stability Over Time Results Without Signal Value Change Assumptions

As suggested by an anonymous reviewer, we begin by first stating a stability over time result for modified-CS and modified-CS-add-LS-del without assuming any model on how the signal changes. This result is quite general and is applicable to various types of signal change models. In Sections 2.4 and 2.5, we specialize the proof technique to get stability results for two sets of signal change assumptions.

2.3.1 Stability over time result for Modified-CS

The following facts are immediate from Algorithm 1.

Proposition 2.3.1 (simple facts)**.** *Consider Algorithm 1.*

1. *An $i \in \mathcal{N}_t$ will definitely get detected in step 3 if $|(\mathbf{x}_t)_i| > \alpha + \|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\|_\infty$.*
2. *Similarly, all $i \in \Delta_{e,t}$ (the zero elements of $\mathcal{T}_t$) will definitely get deleted in step 3 if $\alpha \geq \|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\|_\infty$.*

Using the above facts and Lemma 2.2.7 and an induction argument, we get the following result.

Theorem 2.3.2. *Consider Algorithm 1. Assume that the support size of $\mathbf{x}_t$ is bounded by S and there are at most S_a additions and removals at all times. Assume that y_t satisfies (1.2). If the following hold*

1. (support estimation threshold) *set $\alpha = 7.50\epsilon$,*
2. (number of measurements) $\delta_{S+6S_a} \leq 0.207$,
3. (number of small magnitude entries) $|\mathcal{B}_t| \leq S_a$, *where* $\mathcal{B}_t = \{i \in \mathcal{N}_t : |(\mathbf{x}_t)_i| \leq \alpha + 7.50\epsilon\}$,

4. (initial time) *at* $t = 0$, n_0 *is large enough to ensure that* $|\tilde{\Delta}_t| = 0$, $\tilde{\Delta}_{e,t} = 0$.

then for all t,

1. $|\tilde{\Delta}_t| \le S_a$, $|\tilde{\Delta}_{e,t}| = 0$, $|\tilde{\mathcal{T}}_t| \le S$,
2. $|\Delta_t| \le 2S_a$, $|T_t| \le S$, $|\Delta_{e,t}| \le S_a$,
3. *and* $\|\mathbf{x}_t - \hat{x}_t\| \le 7.50\epsilon$.

The proof is provided in Appendix A.0.2.

2.3.2 Stability over time result for Modified-CS-add-LS-del

A result similar to the one above can also be proved for modified-CS-add-LS-del.

Theorem 2.3.3. *Consider Algorithm 2. Assume that the support size of* $\mathbf{x}_t$ *is bounded by* S *and there are at most* S_a *additions and removals at all times. Assume that* y_t *satisfies (1.2). If the following hold*

1. (addition and deletion thresholds)
   (a) α_{add} *is large enough so that at most* f *false additions per unit time,*
   (b) $\alpha_{del} = 1.12\epsilon + 0.261\sqrt{S_a}(\alpha_{add} + 7.50\epsilon)$,
2. (number of measurements) $\delta_{S+6S_a} \le 0.207$, $\delta_{S+2S_a+f} \le 0.207$,
3. (number of small magnitude entries) $|\mathcal{B}_t| \le S_a$, *where* $\mathcal{B}_t = \{i \in \mathcal{N}_t : |(\mathbf{x}_t)_i| \le \max\{\alpha_{add} + 7.50\epsilon, 2\alpha_{del}\}\}$,
4. (initial time) *at* $t = 0$, n_0 *is large enough to ensure that* $|\tilde{\Delta}_t| = 0$, $\tilde{\Delta}_{e,t} = 0$.

then for all t,

1. $|\tilde{\Delta}_t| \le S_a$, $\tilde{\Delta}_{e,t} = 0$, $|\tilde{\mathcal{T}}_t| \le S$,

2. $|\Delta_t| \le 2S_a$, $|\Delta_{e,t}| \le S_a$, $|T_t| \le S$,

3. $|\Delta_{add,t}| \le S_a$, $|\Delta_{e,add,t}| \le S_a + f$, $|\mathcal{T}_{add,t}| \le S + S_a + f$,

4. $\|\mathbf{x}_t - \hat{x}_{t,modcs}\| \le 7.50\epsilon$

5. *and* $\|\mathbf{x}_t - \hat{x}_t\| \le 1.12\epsilon + 1.261\sqrt{2\alpha_{del}S_a}$.

Proof is provided in Appendix A.0.3.

2.3.3 Discussion

Notice that the support error bound in both results above is $2S_a$. Under slow support change, $S_a \ll S$, this bound is small compared to the support size S, making the result a meaningful one. Also, the reconstruction error is upper bounded by a constant times ϵ. Under a high enough signal-to-noise ratio (SNR), this bound is also small compared to the signal power.

If $f = S_a$ in Theorem 2.3.3, both Modified-CS and Modified-CS-add-LS-del need $\delta_{S+6S_a} \le 0.207$. Consider noisy ℓ_1, i.e. (2.1). Since it is not a recursive approach (each time instant is handled separately), Lemma 2.2.8 is also a stability result for it. From Lemma 2.2.8, it needs $\delta_{2S} \le 0.207$ to get the same error bound. When $S_a \ll S$, clearly it requires a stronger condition than either of the modified-CS algorithms.

Remark 2.3.4. *Consider the noise-free case, i.e. the case when $\epsilon = 0$, $y_t = A_t\mathbf{x}_t$, with the number of support additions and removals per unit time at most S_a. In this case, our results say the following: as long as the signal change assumptions hold, $\delta_{S+kS_a} < 0.207$ is sufficient for both algorithms. It is easy to show that $\delta_{S+S_a,left} < 1$ is also necessary for both algorithms. We give a proof for this in Appendix A.0.8. Thus the sufficient condition that our results need are of the same order in both S and S_a as the necessary condition and hence these results cannot be improved much. Thus, for example, RIP of order $S + k\sqrt{S_a}$ or $\sqrt{S + kS_a}$ will not work. This remark is inspired by a concern of an anonymous reviewer.*

2.4 Stability Results: Simple But Restrictive Signal Change Assumptions

In this section, we assume a very simple but restrictive signal change model that allows for slow nonzero coefficient magnitude increase after a new coefficient is added and slow decrease in magnitude before a coefficient is removed.

2.4.1 Simple but restrictive signal change assumptions

We use a single parameter, r, for the newly added elements' magnitude and for the magnitude increase and decrease rate of all elements at all times. We also fixes the number of support additions and removals to be S_a.

Model 1. *Assume the following.*

1. (addition and increase) *At each $t > 0$, S_a new coefficients get added to the support at magnitude r. Denote this set by $\mathcal{A}_t$. At each $t > 0$, the magnitude of S_a coefficients out of all those which had magnitude $(j-1)r$ at $t-1$ increases to jr. This occurs for all $2 \leq j \leq d$. Thus the maximum magnitude reached by any coefficient is $M := dr$.*

2. (decrease and removal) *At each $t > 0$, the magnitude of S_a coefficients out of all those which had magnitude $(j+1)r$ at $t-1$ decreases to jr. This occurs for all $1 \leq j \leq (d-2)$. At each $t > 0$, S_a coefficients out of all those which had magnitude r at $t-1$ get removed from the support (magnitude becomes zero). Denote this set by $\mathcal{R}_t$.*

3. (initial time) *At $t = 0$, the support size is S and it contains $2S_a$ elements each with magnitude $r, 2r, \ldots (d-1)r$, and $(S - (2d-2)S_a)$ elements with magnitude M.*

Fig. 2.1 illustrates the above signal change assumptions. To understand its implications, define the following sets. For $0 \leq j \leq d-1$, let

$$\mathcal{D}_t(j) := \{i : |x_{t,i}| = jr,\ |x_{t-1,i}| = (j+1)r\}$$

denote the set of elements that *decrease* from $(j+1)r$ to jr at time, t. For $1 \leq j \leq d$, let

$$\mathcal{I}_t(j) := \{i : |x_{t,i}| = jr,\ |x_{t-1,i}| = (j-1)r\}$$

denote the set of elements that *increase* from $(j-1)r$ to jr at time, t. For $1 \leq j \leq d-1$, let

$$\mathcal{S}_t(j) := \{i : 0 < |x_{t,i}| < jr\}$$

denote the set of *small but nonzero* elements, with smallness threshold jr. Clearly, the newly added set, $\mathcal{A}_t = \mathcal{I}_t(1)$ and the newly removed set, $\mathcal{R}_t = \mathcal{D}_t(0)$. Also, $|\mathcal{I}_t(j)| = S_a$, $|\mathcal{D}_t(j)| = S_a$, $|\mathcal{S}_t(j)| = 2(j-1)S_a$.

Consider a $1 < j \leq d$. From Signal Change Assumptions 1, it is clear that at any time, t, S_a elements enter the small elements' set, $\mathcal{S}_t(j)$, from the bottom (set $\mathcal{A}_t$) and S_a enter from the top (set $\mathcal{D}_t(j-1)$). Similarly S_a elements leave $\mathcal{S}_t(j)$ from the bottom (set $\mathcal{R}_t$) and S_a from the top (set $\mathcal{I}_t(j)$). Thus,

$$\mathcal{S}_t(j) = \mathcal{S}_{t-1}(j) \cup (\mathcal{A}_t \cup \mathcal{D}_t(j-1)) \setminus (\mathcal{R}_t \cup \mathcal{I}_t(j)) \tag{2.10}$$

Since $\mathcal{A}_t, \mathcal{R}_t, \mathcal{D}_t(j-1), \mathcal{I}_t(j)$ are mutually disjoint, $\mathcal{R}_t \subseteq \mathcal{S}_{t-1}(j)$ and $\mathcal{I}_t(j) \subseteq \mathcal{S}_{t-1}(j)$, thus, (2.10) implies that

$$\mathcal{S}_{t-1}(j) \cup \mathcal{A}_t \setminus \mathcal{R}_t = \mathcal{S}_t(j) \cup \mathcal{I}_t(j) \setminus \mathcal{D}_t(j-1) \tag{2.11}$$

Also, clearly,

$$\mathcal{N}_t = \mathcal{N}_{t-1} \cup \mathcal{A}_t \setminus \mathcal{R}_t \tag{2.12}$$

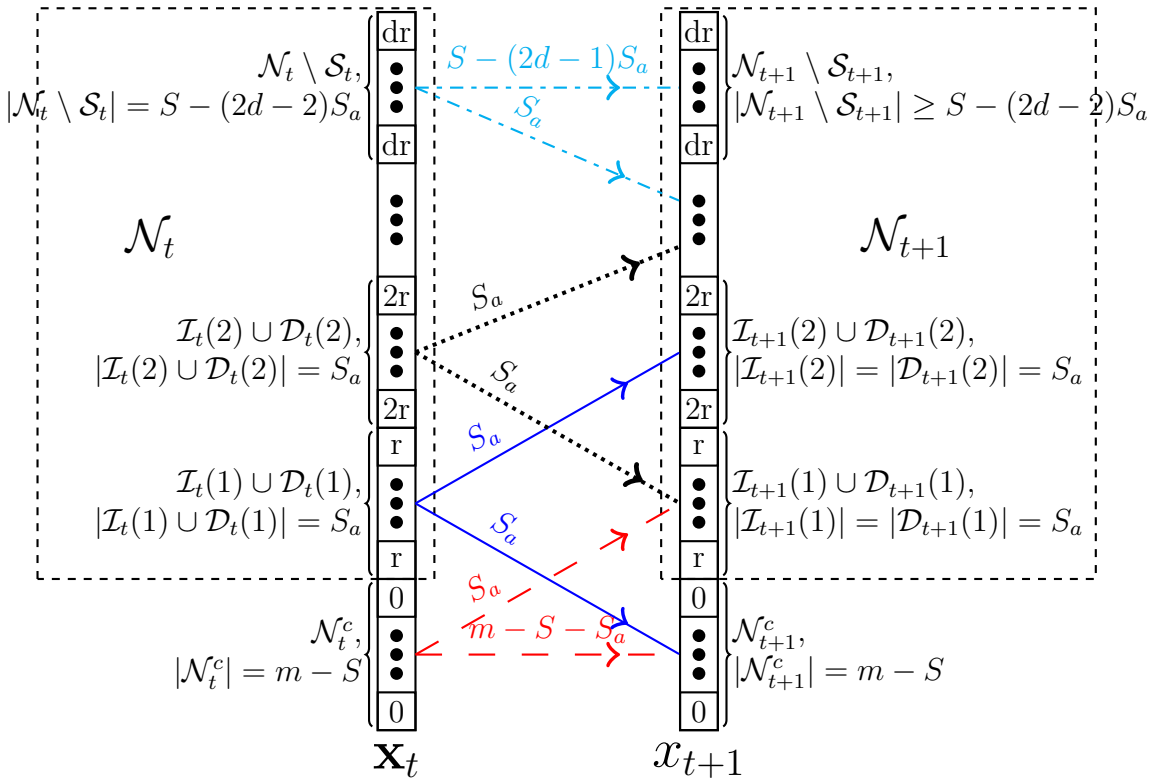


Figure 2.1: Signal Change Assumptions 1 (Values inside rectangular denote magnitudes.)

2.4.2 Stability result for modified-CS

The first step is to find sufficient conditions for a certain set of large coefficients to definitely get detected, and for the elements of Δ_e to definitely get deleted. These are obtained in Lemma 2.4.2 by using Lemma 2.2.7 and the following simple facts. Next, we use Lemma 2.4.2 to ensure that all new additions to the support get detected within a finite delay, and all removals from the support get deleted immediately.

In general, for any vector z, $\|z\|_\infty \le \|z\|$ with equality holding only if z is one-sparse (exactly one element of z is nonzero). If the energy of z is more spread out, $\|z\|_\infty$ will be smaller than $\|z\|$. Typically the error $\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}$ will not be one-sparse, but will be more spread out. The assumption below states this.

Assumption 2.4.1. *Consider Algorithm 1. Assume that the Modified-CS reconstruction error is spread out enough so that*

$$\|\mathbf{x}_t - \hat{x}_{t,modcs}\|_\infty \le \frac{\zeta_M}{\sqrt{S_a}} \|\mathbf{x}_t - \hat{x}_{t,modcs}\|$$

for some $\zeta_M \le \sqrt{S_a}$.

Combining Proposition 2.3.1 and the above assumption with Lemma 2.2.7, we get the following lemma.

Lemma 2.4.2. *Consider Algorithm 1. Assume Assumption 2.4.1. Assume that* $|\mathcal{N}_t| = S_{\mathcal{N}_t}$, $|\Delta_{e,t}| \le S_{\Delta_{e,t}}$ *and* $|\Delta_t| \le S_{\Delta_t}$.

1. *All elements of the set* $\{i \in \mathcal{N}_t : |(\mathbf{x}_t)_i| \ge b_1\}$ *will get detected in step 3 if*
   - $\delta_{S_{\mathcal{N}_t}+S_{\Delta_{e,t}}+2S_{\Delta_t}} \le 0.207$, *and* $b_1 > \alpha + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$.
2. *In step 3, there will be no false additions, and all the true removals from the support (the set* $\Delta_{e,t}$*) will get deleted at the current time, if*
   - $\delta_{S_{\mathcal{N}_t}+S_{\Delta_{e,t}}+2S_{\Delta_t}} \le 0.207$, *and* $\alpha \ge \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$.

We use the above lemma to obtain sufficient conditions to ensure the following: for some $d_0 \leq d$, at all times, t, (i) only coefficients with magnitude less than $d_0 r$ are part of the final set of misses, $\tilde{\Delta}_t$ and (ii) the final set of extras, $\tilde{\Delta}_{e,t}$, is an empty set. In other words, we find conditions to ensure that $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ and $|\tilde{\Delta}_{e,t}| = 0$. Using Signal Change Assumptions 1, $|\mathcal{S}_t(d_0)| = 2(d_0 - 1)S_a$ and thus $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ will imply that $|\tilde{\Delta}_t| \leq 2(d_0 - 1)S_a$.

Theorem 2.4.3 (Stability of modified-CS)**.** *Consider Algorithm 1. Assume Signal Change Assumptions 1 on $\mathbf{x}_t$. Also assume that y_t satisfies (1.2). Assume that Assumption 2.4.1 holds. If, for some $d_0 \leq d$, the following hold*

1. (support estimation threshold) *set* $\alpha = \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$
2. (number of measurements) $\delta_{S+(2k_1+1)S_a} \leq 0.207$,
3. (new element increase rate) $r \geq G$, *where*

$$G \triangleq \frac{\alpha + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon}{d_0} \epsilon \tag{2.13}$$

4. (initial time) *at $t = 0$, n_0 is large enough to ensure that $\tilde{\Delta}_0 \subseteq \mathcal{S}_0(d_0)$, $|\tilde{\Delta}_0| \leq 2(d_0 - 1)S_a$, $|\tilde{\Delta}_{e,0}| = 0$ and $|\tilde{\mathcal{T}}_0| \leq S$*

where

$$k_1 \triangleq \max(1, 2d_0 - 2) \tag{2.14}$$

then,

1. *at all $t \geq 0$, $|\tilde{\mathcal{T}}_t| \leq S$, $|\tilde{\Delta}_{e,t}| = 0$, $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ and so $|\tilde{\Delta}_t| \leq 2(d_0 - 1)S_a$,*
2. *at all $t > 0$, $|\mathcal{T}_t| \leq S$, $|\Delta_{e,t}| \leq S_a$, and $|\Delta_t| \leq k_1 S_a$,*
3. *at all $t > 0$, $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon$*

Proof: The proof is given in Appendix A.0.4. It follows using induction.

Remark 2.4.4. *The condition 4 is not restrictive. It is easy to see that this will hold if n_0 is large enough to ensure that $\delta_{2S}(A_0) \leq 0.207$.*

2.4.3 Stability result for Modified-CS with Add-LS-Del

The first step to show stability is to find sufficient conditions for (a) a certain set of large coefficients to definitely get detected, and (b) to definitely not get falsely deleted, and (c) for the zero coefficients in $\mathcal{T}_{\text{add}}$ to definitely get deleted. These can be obtained using Lemma 2.2.7 and simple facts similar to Proposition 2.3.1.

As explained before, we can assume that the modified-CS reconstruction error is not one-sparse but is more spread out. The same assumption should also be valid for the LS step error. We state these next.

Assumption 2.4.5. *Consider Algorithm 2. Assume that the Modified-CS reconstruction error is spread out enough so that Assumption 2.4.1 holds and assume that the LS step error along $\mathcal{T}_{add,t}$ is spread out enough so that*

$$\|(\mathbf{x}_t - \hat{\mathbf{x}}_{add,t})_{\mathcal{T}_{add,t}}\|_\infty \leq \frac{\zeta_L}{\sqrt{S_a}} \|(\mathbf{x}_t - \hat{\mathbf{x}}_{add,t})_{\mathcal{T}_{add,t}}\|$$

at all times, t, for some $\zeta_L \leq \sqrt{S_a}$.

Combining the above assumption with Lemmas 2.2.7 and 2.2.9, we get the following lemmas.

Lemma 2.4.6 (Detection condition)**.** *Consider Algorithm 2. Assume Assumption 2.4.5. Assume that $|\mathcal{N}_t| = S_{\mathcal{N}_t}$, $|\Delta_{e,t}| \leq S_{\Delta_{e,t}}$, $|\Delta_t| \leq S_{\Delta_t}$. Pick a $b_1 > 0$. All elements of the set $\{i \in \Delta : |(\mathbf{x}_t)_i| \geq b_1\}$ will get detected in step 3 if*

- $\delta_{S_{\mathcal{N}_t}+S_{\Delta_{e,t}}+2S_{\Delta_t}} \leq 0.207$, *and* $b_1 > \alpha_{add} + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$.

Lemma 2.4.7 (Deletion and No false-deletion condition)**.** *Consider Algorithm 2. Assume Assumption 2.4.5. Assume that* $|\mathcal{T}_{add,t}| \le S_{\mathcal{T}_{add,t}}$ *and* $|\Delta_{add,t}| \le S_{\Delta_{add,t}}$.

1. *Pick a* $b_1 > 0$. *No element of the set* $\{i \in \mathcal{T}_{add,t} : |(\mathbf{x}_t)_i| \ge b_1\}$ *will get (falsely) deleted in step 4 if*
   - $\delta_{S_{\mathcal{T}_{add,t}}} < 1/2$ *and* $b_1 > \alpha_{del} + \frac{\zeta_L}{\sqrt{S_a}}(\sqrt{2}\epsilon + 2\theta_{S_{\mathcal{T}_{add,t}},S_{\Delta_{add,t}}}\|(\mathbf{x}_t)_{\Delta_{add,t}}\|)$.
2. *All elements of* $\Delta_{e,add}$ *will get deleted in step 4 if*
   - $\delta_{S_{\mathcal{T}_{add,t}}} < 1/2$ *and* $\alpha_{del} \ge \frac{\zeta_L}{\sqrt{S_a}}(\sqrt{2}\epsilon + 2\theta_{S_{\mathcal{T}_{add,t}},S_{\Delta_{add,t}}}\|(\mathbf{x}_t)_{\Delta_{add,t}}\|)$.

Using the above lemmas, we can obtain sufficient conditions to ensure that, for some $d_0 \le d$, at each time t, $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ (so that $|\tilde{\Delta}_t| \le (2d_0 - 2)S_a$) and $|\tilde{\Delta}_{e,t}| = 0$.

Theorem 2.4.8 (Stability of modified-CS with add-LS-del)**.** *Consider Algorithm 2. Assume Signal Change Assumptions 1 on* $\mathbf{x}_t$. *Also assume that* y_t *satisfies (1.2). Assume that Assumption 2.4.5 holds. If, for some* $1 \le d_0 \le d$, *the following hold*

1. (addition and deletion thresholds)
   - (a) α_{add} *is large enough so that there are at most* f *false additions per unit time,*
   - (b) $\alpha_{del} = \sqrt{\frac{2}{S_a}}\zeta_L\epsilon + 2k_3\theta_{S+S_a+f,k_2S_a}\zeta_L r$,
2. (number of measurements)
   - (a) $\delta_{S+S_a(1+2k_1)} \le 0.207$,
   - (b) $\delta_{S+S_a+f} < 1/2$,
   - (c) $\theta_{S+S_a+f,k_2S_a} < \frac{1}{2}\frac{d_0}{4k_3\zeta_L}$,
3. (new element increase rate) $r \ge \max(G_1, G_2)$, *where*

$$G_1 \triangleq \frac{\alpha_{add} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon}{d_0}$$
$$G_2 \triangleq \frac{2\sqrt{2}\zeta_L\epsilon}{\sqrt{S_a}(d_0 - 4k_3\theta_{S+S_a+f,k_2S_a}\zeta_L)} \tag{2.15}$$

4. (initial time) n_0 *is large enough to ensure that* $\tilde{\Delta}_0 \subseteq \mathcal{S}_0(d_0)$, $|\tilde{\Delta}_0| \leq (2d_0 - 2)S_a$, $|\tilde{\Delta}_{e,0}| = 0$, $|\tilde{\mathcal{T}}_0| \leq S$,

where

$$\begin{aligned} k_1 &\triangleq \max(1, 2d_0 - 2) \\ k_2 &\triangleq \max(0, 2d_0 - 3) \\ k_3 &\triangleq \sqrt{\sum_{j=1}^{d_0-1} j^2 + \sum_{j=1}^{d_0-2} j^2} \end{aligned} \tag{2.16}$$

then, at all $t \geq 0$,

1. $|\tilde{\mathcal{T}}_t| \leq S$, $|\tilde{\Delta}_{e,t}| = 0$, *and* $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ *and so* $|\tilde{\Delta}_t| \leq (2d_0 - 2)S_a$,

2. $|\mathcal{T}_t| \leq S$, $|\Delta_{e,t}| \leq S_a$, *and* $|\Delta_t| \leq k_1 S_a$,

3. $|\mathcal{T}_{add,t}| \leq S + S_a + f$, $|\Delta_{e,add,t}| \leq S_a + f$, *and* $|\Delta_{add,t}| \leq k_2 S_a$,

4. $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq C_1(S + S_a + 2k_1 S_a)\epsilon \leq 7.50\epsilon$,

5. $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\| \leq 1.261 k_3 \sqrt{S_a} r + 1.12\epsilon$.

Proof: The proof is given in Appendix A.0.5.

2.4.4 Discussion

Notice that, with Signal Change Assumptions 1, at all times, t, the signals have the same support set size, $|\mathcal{N}_t| = S$ and the same signal power, $\|\mathbf{x}_t\|^2 = (S-(2d-2)S_a)M^2 + 2S_a \sum_{j=1}^{d-1} j^2 r^2$. As in the previous section, here again the support error bound in both results above is proportional to S_a. Under slow support change, this means that the support error is small compared to the support size. To make the comparison of the above two results simpler, let us fix $d_0 = 2$ and let $f = S_a$ in Theorem 2.4.8. Consider the conditions on the number of measurements. Modified-CS needs $\delta_{S+5S_a} \leq 0.207$.

Modified-CS-add-LS-del needs $\delta_{S+5S_a} \leq 0.207$; $\delta_{S+2S_a} < 0.5$ (this is implied by the first condition) and $\theta_{S+2S_a,S_a} \leq \frac{1}{4\zeta_L}$. Since $\theta_{S+2S_a,S_a} \leq \delta_{S+3S_a}$, the third condition is also implied by the first as long as $\zeta_L \leq 1.2$. In simulation tests (described in Sec 2.5.4) we observed that this was usually true. Then, both modified-CS and modified-CS-add-LS-del need the same condition on the number of measurements: $\delta_{S+5S_a} \leq 0.207$. Consider noisy ℓ_1 i.e. (2.1). As explained earlier, Lemma 2.2.8 serves as a stability result for it. From Lemma 2.2.8, iy needs $\delta_{2S} \leq 0.207$ to get the same error bound which is significantly stronger when $S_a \ll S$.

Let us compare the requirement on r. In Theorem 2.4.8 for modified-cs-add-ls-del, since $\theta_{S+S_a+f,k_2S_a} \leq \frac{1}{2}\frac{d_0}{4k_3\zeta_L}$, so $G_2 \leq \frac{4\sqrt{2}\zeta_L}{\sqrt{S_a}d_0}\epsilon < \frac{5.7\epsilon}{d_0} < \frac{7.50\epsilon}{d_0} < G_1$ and thus G_1 is what decides the minimum allowed value of r. Thus, it needs $r \geq G_1 = \frac{1}{d_0}[\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon]$. On the other hand, modified-CS needs $r \geq G = \frac{1}{d_0}[2\frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon]$. If α_{add} is close to zero, this means that the minimum magnitude increase rate, r, required by Theorem 2.4.8 is almost half of that required by Theorem 2.4.3. In our simulation experiments, α_{add} was typically quite small: it was usually close to a small constant times $\epsilon/\sqrt{n}$ (see Sec 4.4).

Remark 2.4.9. *From the above results, observe that, if the rate of magnitude change, r, is smaller, $r \geq G_1$ or $r \geq G$ will hold for a larger value of d_0. This means that the support error bound, $(2d_0 - 2)S_a$, will be larger. This, in turn, decides what conditions on the RIC and ROC are needed (in other words, how many measurements, n_t, needed). Smaller r means a larger d_0 is needed which, in turn, means that stronger conditions on the RIC and ROC (larger n_t) are needed. Thus, for a given $n_t = n$, as r is reduced, the algorithm will stabilize to larger and larger support error levels (larger d_0) and finally become unstable (because the given n does not satisfy the conditions on δ, θ for the larger d_0).*

2.5 Stability Results: Realistic Signal Change Assumptions

We introduce the signal change assumptions in the next subsection and then give the results in the following two subsections. The discussion of the results and a comparison with the results of LS-CS [1] is provided in the two subsequent subsections.

2.5.1 Realistic signal change assumptions

Briefly, we assume the following. At any time the signal vector $\mathbf{x}_t$ is a sparse vector with support set $\mathcal{N}_t$ of size S or less. At most S_a elements get added to the support at each time t and at most S_a elements get removed from it. At time $t = t_j$, a new element j gets added at an initial magnitude a_j, and its magnitude increases for the next $d_j \geq d_{\min}$ time units. Its magnitude increase at time τ (for any $t_j < \tau \leq t_j + d_j$ is $r_{j,\tau}$. Also, at each time t, at most S_a elements out of the "large elements" set (defined in the signal model) leave the set and begin to decrease. These elements keep decreasing and get removed from the support in at most b time units. In the model as stated above, we are implicitly allowing an element j to get added to the support at most once. In general, j can get added, then removed and then added again. To allow for this, we let $\mathbf{t_j}$ be the *set* of time instants at which j gets added; we replace a_j by $a_{j,t}$ and we replace d_j by $d_{j,t}$ (both of which are nonzero only for $t \in \mathbf{t_j}$).

As demonstrated in Section 2.6, the above assumptions are practically valid for MRI sequences.

Model 2. *Assume the following.*

1. *At the initial time, $t = 0$, the support set, $\mathcal{N}_0$, contains S_0 nonzero elements, i.e. $|\mathcal{N}_0| = S_0$.*

2. *At time t, $S_{a,t}$ elements are added to the support set. Denote this set by $\mathcal{A}_t$. At time t, a new element j gets added to the support at an initial magnitude $a_{j,t}$ and*

its magnitude increases for at least the next $d_{\min} > 0$ *time instants. At time* τ *(for* $t < \tau \leq t + d_{\min}$*), the magnitude of element* j *increases by* $r_{j,\tau} \geq 0$.

- $a_{j,t}$ *is nonzero only if element* j *got added at time* t*, for all other times, we set it to zero.*

3. *We define the "large set" as*

$$\mathcal{L}_t := \{j \notin \cup_{\tau=t-d_{\min}+1}^{t} \mathcal{A}_\tau : |(\mathbf{x}_t)_j| \geq \ell\},$$

for a given constant ℓ*. Elements in* $\mathcal{L}_{t-1}$ *either remain in* $\mathcal{L}_t$ *(while increasing or decreasing or remaining constant) or decrease enough to leave* $\mathcal{L}_t$.

4. *At time* t*,* $S_{d,t}$ *elements out of* $\mathcal{L}_{t-1}$ *decrease enough to leave* $\mathcal{L}_{t-1}$*. Denote this set* $\mathcal{B}_t$*. All these elements continue to keep decreasing and become zero (removed from support) within at most* b *time units. Also, at time* t*,* $S_{r,t}$ *elements out of these decreasing elements are removed from the support. Denote this set by* $\mathcal{R}_t$.

5. *At all times* t*,* $0 \leq S_{a,t} \leq S_a$*,* $0 \leq S_{d,t} \leq \min\{S_a, |\mathcal{L}_{t-1}|\}$*,* $0 \leq S_{r,t} \leq S_a$ *and the support size,* $S_t := |\mathcal{N}_t| \leq S$ *for constants* S *and* S_a *such that* $S + S_a \leq m$.

Fig.2.2 illustrates the above assumptions. We should reiterate that the above is not a generative model. It is only a set of assumptions on signal change. One possible generative model that satisfies these assumptions is given in Appendix A.0.9.

Remark 2.5.1. *It is easy to see that Signal Change Assumptions 1 are a special case of Signal Change Assumptions 2 with* $a_{j,t} = r_{j,t} = r$*,* $d_{\min} = d$*,* $b = d$*,* $S_0 = S$*,* $S_{a,t} = S_{d,t} = S_{r,t} = S_a$*,* $\ell = dr$.

From the above assumptions, the newly added elements' set $\mathcal{A}_t := \mathcal{N}_t \setminus \mathcal{N}_{t-1}$; the newly removed elements' set $\mathcal{R}_t := \mathcal{N}_{t-1} \setminus \mathcal{N}_t$; the set of elements that begin to start

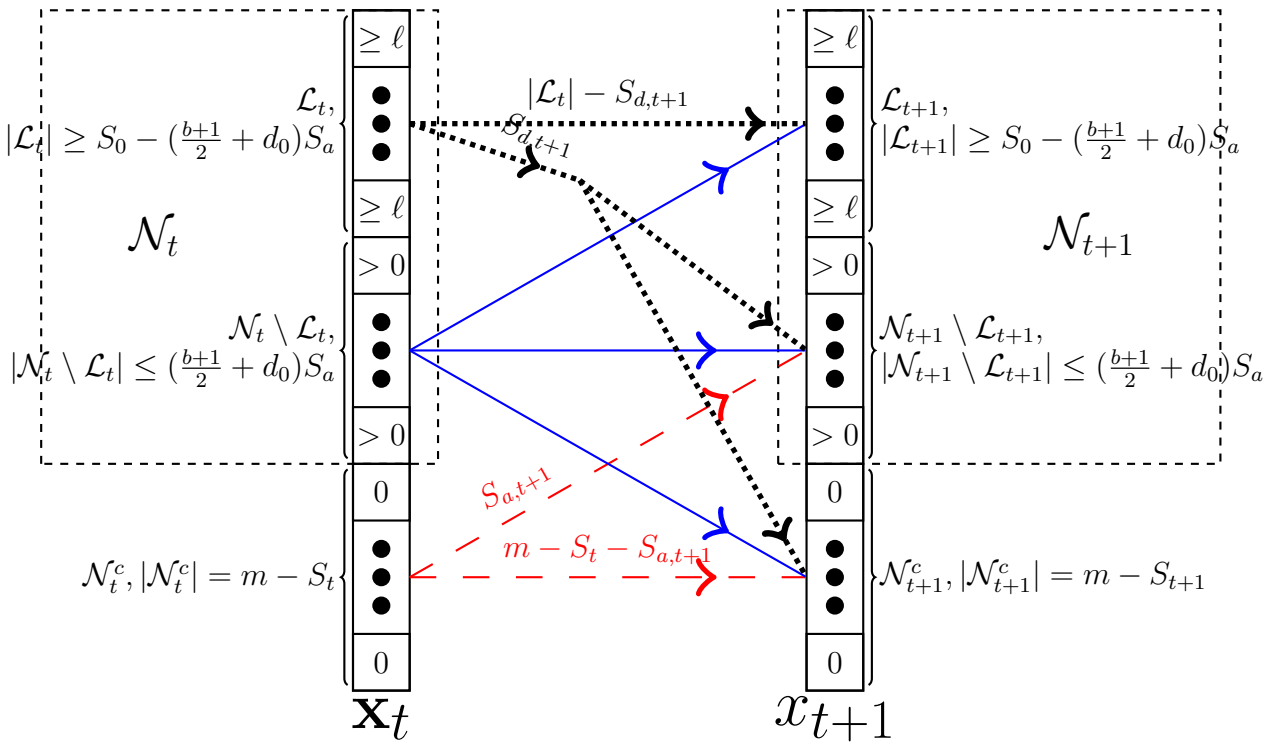


Figure 2.2: Signal Change Assumptions 2 (Values inside rectangular denote magnitudes.)

decreasing at t, $\mathcal{B}_t := \mathcal{L}_{t-1} \setminus \mathcal{L}_t$. Define the following sets: the set of increasing (actually non-decreasing) elements at t,

$$\mathcal{I}_t := \{j \in \mathcal{N}_t : |(\mathbf{x}_t)_j| \geq |(x_{t-1})_j|\};$$

and the set of small and decreasing elements,

$$\mathcal{SD}_t := \mathcal{L}_t^c \cap |\{i \in \mathcal{N}_t : 0 < |(x_t)_i| < |(x_{t-1})_i|\}|.$$

Notice that $\mathcal{I}_t$ also includes j if its magnitude does not change from $t-1$ to t.

Condition 2 of the above model implies that (i) $|\mathcal{A}_t| = S_{a,t}$; (ii) if $j \in \mathcal{A}_{t-t_0}$ (i.e. if j is added at $t - t_0$) for a $t_0 \leq d_{\min}$, then $|(\mathbf{x}_t)_j| = a_{j,t-t_0} + \sum_{\tau=t-t_0+1}^{t} r_{j,\tau}$; and (iii) $\mathcal{A}_t \subseteq \mathcal{I}_t \cap \mathcal{I}_{t+1} \cdots \cap \mathcal{I}_{t+d_{\min}}$ (all newly added elements increase for at least $d_{\min}$ time instants).

Condition 3 implies that $\mathcal{L}_{t-1} \subseteq \mathcal{L}_t \cup \mathcal{SD}_t$. It also implies that $(\cup_{\tau=t-d_{\min}+1}^{t} \mathcal{A}_\tau) \cap \mathcal{L}_t = \emptyset$. This, along with condition 2 means that $\cup_{\tau=t-d_{\min}}^{t} \mathcal{A}_\tau \subseteq \mathcal{I}_t$.

Condition 4 implies that $|\mathcal{B}_t| = S_{d,t}$; $\mathcal{L}_{t-1} \setminus \mathcal{B}_t \subseteq \mathcal{L}_t$; $\mathcal{SD}_t = \mathcal{SD}_{t-1} \cup \mathcal{B}_t \setminus \mathcal{R}_t$; $\sum_{\tau=1}^{t} S_{r,\tau} \geq \sum_{\tau=1}^{t-b} S_{d,\tau}$; $|\mathcal{SD}_t| \leq \sum_{\tau=t-b+1}^{t} S_{d,\tau}$; and $|\mathcal{R}_t| = S_{r,t}$.

Condition 5, along with the above, implies that $|\mathcal{SD}_t| \leq bS_a$.

Finally, it is easy to see that $\mathcal{N}_t = \mathcal{I}_t \cup \mathcal{L}_t \cup \mathcal{SD}_t$. The sets $\mathcal{I}_t$, $\mathcal{L}_t$ are not disjoint, but both are disjoint with $\mathcal{SD}_t$.

The above model tells us the following. Consider an element j that got added at time t, i.e. $j \in \mathcal{A}_t$. At $\tau = t, t+1, ...t+d_{\min}-1$, $j \in \mathcal{I}_\tau$ and $j \notin \mathcal{L}_\tau$. At $\tau = t + d_{\min}$, $j \in \mathcal{I}_\tau$; if $|(x_\tau)_j| \geq \ell$ then $j \in \mathcal{L}_\tau$ as well. For $\tau > t + d_{\min}$, what happens depends on $\tau - 1$. If $j \in \mathcal{L}_{\tau-1}$, then either $j \in \mathcal{L}_\tau$ or it decreases enough to enter the small and decreasing set, i.e. $j \in \mathcal{B}_\tau \subseteq \mathcal{SD}_\tau$. If $j \in \mathcal{SD}_{\tau-1}$, then either it keeps decreasing or gets removed, i.e. either $j \in \mathcal{SD}_\tau$ or $j \in \mathcal{R}_\tau \subseteq \mathcal{N}_\tau^c$. If $j \in \mathcal{L}_{\tau-1}^c \cap \mathcal{I}_{\tau-1}$, then, if $|(x_\tau)_j| \geq \ell$ then $j \in \mathcal{L}_\tau \cap \mathcal{I}_\tau$, else $j \in \mathcal{L}_\tau^c \cap \mathcal{I}_\tau$.

We now discuss sufficient conditions for condition 5 of the signal model to hold.

Remark 2.5.2. *Since* $S_t = S_{t-1} + S_{a,t} - S_{r,t} = S_0 + \sum_{\tau=1}^{t} S_{a,\tau} - \sum_{\tau=1}^{t} S_{r,\tau}$, *thus,* $S_t \leq S$ *holds if* $S_0 \leq S$ *and* $\sum_{\tau=1}^{t} S_{a,\tau} \leq \sum_{\tau=1}^{t-b} S_{d,\tau}$.

Notice that an element j could get added, then removed and added again later. Let

$$\mathbf{t_j} := \{t : a_{j,t} \neq 0\}$$

denote the set of time instants at which j gets added. Clearly, $\mathbf{t_j} = \emptyset$ if j never got added. Let

$$a_{\min} := \min_{j:\mathbf{t_j}\neq\emptyset} \min_{t\in\mathbf{t_j}, t\neq 0} a_{j,t}$$

denote the minimum of $a_{j,t}$ over all elements j that got added at $t > 0$. We are excluding coefficients that never got added and those that got added at $t = 0$. Let

$$r_{\min}(d) := \min_{j:\mathbf{t_j}\neq\emptyset} \min_{t\in\mathbf{t_j}, t\neq 0} \min_{\tau\in[t+1,t+d]} r_{j,\tau}$$

denote the minimum, over all elements j that got added at $t > 0$, of the minimum of $r_{j,\tau}$ over the first d time instants after j got added.

Define

$$\ell := a_{\min} + d_{\min} r_{\min}(d_{\min}). \tag{2.17}$$

With ℓ defined this way, clearly, $\mathcal{N}_t = (\cup_{\tau=t-d_{\min}+1}^{t} \mathcal{A}_\tau) \cup \mathcal{L}_t \cup \mathcal{SD}_t$ where the three sets are mutually disjoint.

Also, with ℓ as above, it is clear that for $t > d_{\min}$, $\mathcal{L}_t = \mathcal{L}_{t-1} \cup \mathcal{A}_{t-d_{\min}-1} \setminus \mathcal{B}_t$, and for $t \leq d_{\min}$, $\mathcal{L}_t = \mathcal{L}_{t-1} \setminus \mathcal{B}_t$. Here, by definition, $\mathcal{L}_{t-1}$ and $\mathcal{A}_{t-d_{\min}-1}$ are disjoint and $\mathcal{B}_t \subseteq \mathcal{L}_{t-1}$. Thus,

$$|\mathcal{L}_t| = |\mathcal{L}_0| + \sum_{\tau=1}^{t-d_{\min}} S_{a,\tau} - \sum_{\tau=1}^{t} S_{d,\tau}$$

Also notice that $|\mathcal{L}_0| \leq S_0$. Using these facts and Remark 2.5.2, we can conclude the following.

Remark 2.5.3. *Let* $\ell := a_{\min} + d_{\min} r_{\min}(d_{\min})$. *Then, condition 5 of Signal Change Assumptions 2 holds if*

1. $0 \leq S_{a,t} \leq S_a$ *and* $0 \leq S_{d,t} \leq S_a$,

2. $(d_{\min} + b + 1)S_a \leq |\mathcal{L}_0| \leq S_0 \leq S$, *and*

3. $\sum_{\tau=1}^{t} S_{a,\tau} \leq \sum_{\tau=1}^{t-b} S_{d,\tau} \leq |\mathcal{L}_0| + \sum_{\tau=1}^{t-b-d_{\min}-1} S_{a,\tau}$.

The leftmost lower bound of the second condition ensures that the upper bound of the third condition is not smaller than the lower bound. The upper bound of the third condition ensures that $S_{d,t} \leq |\mathcal{L}_{t-1}|$ *always (it is actually written to ensure* $S_{d,t-b} \leq |\mathcal{L}_{t-b-1}|$*).* $S_0 \leq S$ *and the lower bound of the third condition ensures that* $S_t \leq S$ *(as explained in Remark 2.5.2).*

A simpler sufficient condition is as follows.

Remark 2.5.4. *Let* $\ell := a_{\min} + d_{\min} r_{\min}(d_{\min})$. *Then, condition 5 of Signal Change Assumptions 2 holds if* $(d_{\min} + b + 1)S_a \leq |\mathcal{L}_0| \leq S_0 \leq S$; $S_{d,t} = S_a$ *for all* t; *and for* $1 \leq t \leq b$, $S_{a,t} = 0$, *and for* $t > b$, $S_{a,t} = S_a$.

In the above model, we only assume that all coefficients will get removed in at most b time units. However, it can happen that some coefficients get removed earlier than that and hence it is fair to include this in the signal model. We do this below.

Model 3. *Assume Signal Change Assumptions 2 with the following extra assumption.*

- *Out of the* $S_{d,t}$ *elements that started decreasing at time* t, *at least* $\frac{\tau}{b} S_{d,t}$ *of them get removed by* $t + \tau$ *for* $\tau < b$.

All implications of the above model are the same as those of Signal Change Assumptions 2, except that now, $|\mathcal{SD}_t| \leq S_{d,t} + \frac{b-1}{b} S_{d,t-1} + \ldots \frac{1}{b} S_{d,t-b+1} \leq \frac{b+1}{2} S_a$; while for Signal Change Assumptions 2, $|\mathcal{SD}_t| \leq bS_a$.

2.5.2 Modified-CS stability result

For the above signal model, we can claim the following.

Theorem 2.5.5. *Consider Algorithm 1. Assume Signal Change Assumptions 3 on* $\mathbf{x}_t$. *Also assume that* y_t *satisfies (1.2). Assume that Assumption 2.4.1 holds. If there exists a* $d_0 \leq d_{\min}$ *such that the following hold:*

1. *algorithm parameters*

   (a) $\alpha = \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$,

2. *number of measurements*

   (a) $\delta_{S+3(\frac{(b+1)}{2}+d_0+1)S_a} \leq 0.207$,

3. *initial magnitude and magnitude increase rate:*

$$\min\{\ell, \min_{j:\mathbf{t_j}\neq\emptyset} \min_{t\in\mathbf{t_j}} (a_{j,t} + \sum_{\tau=t+1}^{t+d_0} r_{j,\tau})\}$$
$$> \alpha + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon,$$

4. *at* $t = 0, n_0$ *is large enough to ensure that* $|\tilde{\Delta}_t| \leq \frac{b+1}{2} S_a + d_0 S_a, |\tilde{\Delta}_{e,t}| = 0$,

then, for all t,

1. $|\tilde{\Delta}_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a$, $|\tilde{\Delta}_{e,t}| = 0$, $|\tilde{\mathcal{T}}_t| \leq S$,

2. $|\Delta_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a + S_a$, $|\mathcal{T}_t| \leq S$, $|\Delta_{e,t}| \leq S_a$,

3. *and* $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\| \leq 7.50\epsilon$

Proof: See Appendix A.0.6.

Corollary 2.5.6. *Under Signal Change Assumptions 2, the result of Theorem 2.5.5 changes in the following way: replace* $\frac{(b+1)}{2} S_a$ *by* bS_a *everywhere in the result.*

Remark 2.5.7. *Condition 4 of the above result is not restrictive. It is easy to see that it will hold if* $\delta_{2S}(A_0) \leq 0.207$ *and if* $|\mathcal{L}_0| \geq [S_0 - (\frac{(b+1)}{2}S_a + d_0 S_a)]$.

Remark 2.5.8. *A simpler sufficient condition for condition 3 is:* $\min(\ell, a_{\min} + d_0 r_{\min}(d_0)) > \alpha + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$.

2.5.3 Modified-CS-Add-LS-Del stability result

Finally we study Modified-CS-Add-LS-Del.

Theorem 2.5.9. *Consider Algorithm 2. Assume Signal Change Assumptions 3 on* $\mathbf{x}_t$. *Also assume that* y_t *satisfies (1.2). Assume that Assumption 2.4.5 holds. If there exists a* $d_0 \leq d_{\min}$ *such that the following hold:*

1. *algorithm parameters*
   (a) α_{add} *is large enough so that there are at most* f *false adds at time* t, *i.e.* $|\hat{\mathcal{A}}_t \setminus \mathcal{N}_t| \leq f$
   (b) $\alpha_{del} = 1.12\frac{\zeta_L}{\sqrt{S_a}}\epsilon + 0.261\zeta_L h$, *where* $h^2 = (\frac{(b+1)}{2} + d_0)(\alpha_{add} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon)^2$
2. *number of measurements*
   (a) $\delta_{S+3(\frac{(b+1)}{2}S_a + d_0 S_a + S_a)} \leq 0.207$
   (b) $\delta_{S+S_a+f} \leq 0.207$
   (c) $\theta_{S+S_a+f, \frac{(b+1)}{2}S_a + d_0 S_a} \leq 0.207$
3. *initial magnitude and magnitude increase rate:*

$$\min\{\ell, \min_{j:\mathbf{t_j}\neq\emptyset} \min_{t\in\mathbf{t_j}} (a_{j,t} + \sum_{\tau=t+1}^{t+d_0} r_{j,\tau})\}$$
$$> \max\{\alpha_{add} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon, 2\alpha_{del}\} \tag{2.18}$$

4. *at* $t = 0, n_0$ *is large enough to ensure that* $|\tilde{\Delta}_t| \leq \frac{b+1}{2}S_a + d_0 S_a, |\tilde{\Delta}_{e,t}| = 0$,

then

1. $\tilde{\Delta}_t \subseteq \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \ldots \mathcal{A}_{t-d_0+1}$
2. $|\tilde{\Delta}_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a$, $|\tilde{\Delta}_{e,t}| = 0$, $|\tilde{\mathcal{T}}_t| \leq S$
3. $|\Delta_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a + S_a$, $|\mathcal{T}_t| \leq S$
4. $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon$,
5. $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\| \leq 1.12\epsilon + 1.261\sqrt{(\frac{(b+1)}{2} + d_0)(\alpha_{del} + 7.50\epsilon)S_a}$.

Proof: See Appendix A.0.7.

Remark 2.5.10. *Claims similar to Corollary 2.5.6 and Remarks 2.5.7 and 2.5.8 hold for the above result also.*

2.5.4 Discussion

Remark 2.5.11. *Notice that Signal Change Assumptions 2 or 3 allow for both slow and fast signal magnitude increase or decrease. Slow magnitude increase/decrease would happen, for example, in an imaging problem when one object slowly morphs into another with gradual intensity changes. Or, in case of brain regions becoming "active" in response to stimuli, the activity level gradually increases from zero to a certain maximum value within a few milliseconds (10-12 frames of fMRI data), and similarly the "activity" level decays to zero within a few milliseconds. In both of the above examples, a new coefficient will get added to the support at time t at a small magnitude $a_{j,t}$ and increase by $r_{j,\tau}$ per unit time for sometime after that. Similarly for the decay to zero of the brain's activity level. On the other hand, the signal model also allows support changes resulting from motion of objects, e.g. translation. In this case, the signal magnitude changes will typically not be slow. As the object moves, a set of new pixels enter the support and another set leave. The entering pixels may have large enough pixel intensity and their*

intensity may never change. For our model this means that the pixel enters the support at a large enough initial magnitude $a_{j,t}$ but its magnitude never changes i.e. $r_{j,\tau} = 0$ for all τ. If all pixels exit the support without their magnitude first decreasing, then $b = 1$.

The only thing that the above results (Theorem 2.5.5 and 2.5.9) require is that (i) for any element j that is added, either $a_{j,t}$ is large enough or $r_{j,\tau}$ is large enough for the initial few (d_0) time instants so that condition 3 holds; and (ii) a decaying coefficient decays to zero within a short delay, b. (i) ensures that every newly added support element gets detected either immediately or within a finite delay; while (ii) ensures removal within finite delay of a decreasing element. For the moving object case, this translates to requiring that $a_{j,t}$ be large enough. For the first two examples above, this translates to requiring that $r_{j,\tau}$ be large enough for the first few time instants after j gets added and that b be small enough.

Recall that $\delta_S := \max_{t>0} \delta_S(A_t)$. Other than the above assumption, the results also need that the support estimation thresholds are set appropriately; enough number of measurements, n_t, are available at all times $t > 0$ so that condition 2 holds (this number depends on the support size, S, the support change size, S_a and on b); and condition 4 holds.

For the above results, the support errors are bounded by a constant times S_a. Thus, under slow support change, the bound is small compared to the support size, S_t, making the above a meaningful result. The reconstruction error is bounded by a constant times ϵ. Under high enough SNR, this bound is small compared to the signal power. In fact, for Signal Change Assumptions 2 or 3, the signal power is not bounded. To compare the results, let us fix some of the parameters. Suppose that $b = 3$, $f = S_a$, $S_0 = S$, $S_{a,t} = S_{r,t} = S_{d,t} = S_a$. Let $d_0 = 2$. The modified-CS result says the following. If

1. $\delta_{S+15S_a} \leq 0.207$, and
2. LHS of condition 3 $> \frac{\zeta_M}{\sqrt{S_a}} 15\epsilon$,

then $|\tilde{\Delta}_t| \leq 4S_a$ and $|\tilde{\Delta}_{e,t}| = 0$ and $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon$. The Modified-CS-add-LS-del result says the following. If

1. $\delta_{S+15S_a} \leq 0.207$(the other two conditions are implied by this), and
2. LHS of condition 3 $>$ $\max(\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon, 2.24\frac{\zeta_L}{\sqrt{S_a}}\epsilon + 0.522\zeta_L h)$, where $h^2 = 4(\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon)^2$.

then $|\tilde{\Delta}_t| \leq 4S_a$ and $|\tilde{\Delta}_{e,t}| = 0$ and $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon$.

The CS result from Lemma 2.2.8 says the following. If

1. $\delta_{2S} \leq 0.207$

then $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,cs}\| \leq 7.50\epsilon$.

Thus, both modified-CS and modified-CS-add-LS-del need the same restricted isometry condition (condition on the number of measurements). Under the slow support change assumption, $S_a \ll S_t \leq S$. In this case, both the modified-CS algorithms hold under a weaker restricted isometry condition (potentially fewer number of measurements required) than what noisy ℓ_1 needs for the same error bound. Next we compare the lower bounds on the LHS of condition 3 needed by modified-CS and by modified-CS-add-LS-del. This requires knowing ζ_M and ζ_L. To get an idea of the values of ζ_M and ζ_L, we did simulations based on Signal Change Assumptions 2 with $S = 0.1m, S_{a,t} = S_{d,t} = S_{r,t} = S_a = 0.01m, b = d_{\min} = 3, r_{j,t} = 1, a_{j,t} = 1$ (we generated it using the generative model given in Appendix A of [30]). The measurement matrices A_t were zero mean random Gaussian $n_t \times m$ matrices with columns normalized to unit norm. For $t = 0$, $n_0 = 160$; for $t > 0$, $n_t = n = 57$. The measurement noise, $(\mathbf{w}_t)_j \sim^{i.i.d.} uniform(-c_t, c_t)$ for $1 \leq j \leq m$. For $t = 0$, $c_t = 0.01266$; for $t > 0$, $c_t = 0.1266$. We used the same measurement Gaussian matrix A for $t > 0$. We generated 500 realizations respectively with different choices of m, and used both algorithms for reconstruction. When $m = 200$, we got, $\zeta_M = 0.9328\sqrt{S_a}, \zeta_L = 0.8734\sqrt{S_a}$; when $m = 1000$, $\zeta_M = 0.8295\sqrt{S_a}, \zeta_L = 0.8628\sqrt{S_a}$; when $m = 2000$, $\zeta_M = 0.8497\sqrt{S_a}, \zeta_L = 0.8628\sqrt{S_a}$.

For our comparison, we pick the largest values we got from the above experiment: let $\zeta_M = 0.9328\sqrt{S_a}$ and $\zeta_L = 0.8734\sqrt{S_a}$. With these values, modified-CS needs LHS of condition $3 > 13.99\epsilon$ and modified-CS-Add-LS-Del needs LHS of condition $3 > \max\{\alpha_{\text{add}} + 7.00\epsilon, 10.978\epsilon + 3.246\alpha_{\text{add}}\} = 10.978\epsilon + 3.246\alpha_{\text{add}}$. With α_{add} small enough, clearly modified-CS-add-LS-del requires a weaker assumption. As explained earlier and also in [30], α_{add} is a small threshold that is typically proportional to the noise bound $c, i.e., \epsilon/\sqrt{n}$. Thus the mod-CS-Add-LS-Del condition is weaker.

The comparison between modified-CS and modified-CS-add-LS-del above is not as clear-cut as that in the simple model case (Signal Change Assumptions 1). The reason is that the simple model tells us exactly how many support additions and removals occur at each time; and it also tells us the exact number of elements with a certain magnitude. As a result, it is possible to get a better bound on $\|x_{\Delta_{t,\text{add}}}\|_2$: this is needed to bound the LS step error. The LS error decides the value of α_{del} and α_{del}, in turn, decides the lower bound on the LHS of condition 3. The current Signal Change Assumptions 2 or 3 are much more flexible, but this also means that they not give us exact magnitude information. As a result, the bounds are looser and so the advantage of modified-CS-add-ls-del is not demonstrated as clearly.

Remark 2.5.12. *Finally, we explain why condition 1a of Theorem 2.5.9 is stated the way it is. Because of how the modified-CS error is bounded, we cannot get a bound on the reconstruction error for the j^{th} coefficient, $|(\hat{\mathbf{x}}_t)_j - (x_t)_j|$. We can only bound this error by its infinity norm. Thus, the only way to get an explicit value for α_{add} is to let it equal the upper bound on $\|\hat{\mathbf{x}}_t - x_t\|_\infty$ and this will ensure $f = 0$ false adds. However, the key point of the add-LS-del procedure is that one can pick an addition threshold that is smaller than this but results in some false adds, f. As long as f is small enough so that $A_{T_{add}}$ is well conditioned (condition 2b holds), the LS step error will be much smaller. With α_{del} chosen appropriately, one can still delete all of these false adds (as well as all elements of the removed set) in the deletion step.*

2.5.5 Comparison with the LS-CS result of [1]

In [1], we obtained a stability result for LS-CS which was a worse algorithm than modified-CS: it required stronger conditions for exact recovery, and was worse is simulation experiments as shown in [14, 30]. The same signal model and the same strategy as that of [1] can be used for modified-CS as well and we will, in fact, get a stronger stability result for it: the modified-CS result will not need condition 3b of the LS-CS stability result (Theorem 2 of [1]).

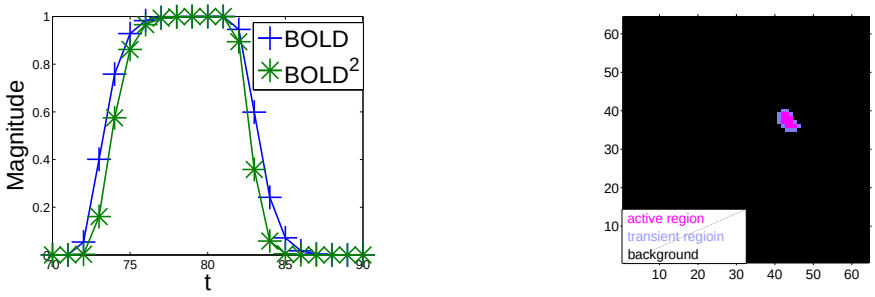
The most important difference between the LS-CS result from [1] and our results is that [1] assumed S_a support changes every p frames and the result required a lower bound on p. With this, one could ensure that all newly added support elements got detected before the next support change time. This meant that one could delete the false adds and removals after all new adds got detected, but before the next change time. At this time, the signal recovery is very accurate (because of zero misses) and hence, for the result of [1], a very small deletion threshold could suffice. However, as explained earlier (see Fig 1.1), support change every so often is not a practically valid assumption in most applications. In this work, we allow the support to change at every time which is more realistic, but is also more difficult to analyze. With this, one always has some misses at each time instant (except in the simplest case where all new elements are added at very large magnitudes). Thus, one cannot wait for all the missed elements to get detected before deleting the false adds and removals and hence one requires a larger deletion threshold.

A third difference is that the signal change model of [1] fixed the number of support additions and removals at each time to be just S_a; it fixed the initial magnitude and the rate of magnitude increase for a new support element j to both be a_j at all times; and, for decreasing coefficients, it assumed a very specific and fixed rate of magnitude decrease. None of these is a very practical assumption. Our realistic signal change models (Signal Change Assumptions 2 or 3) allow all these things to vary with time.

2.6 Model Verification

We verified that two different types of MRI image sequences – a larynx (vocal tract) MRI sequence and a brain functional MRI sequence – do indeed satisfy Signal Change Assumptions 2. First we describe model verification for the larynx sequence. We used a 10 frame sequence and extracted out a 36x36 region of this sequence selected as the region that includes the part where most of the changes were visible. As shown in earlier work [14], this sequence is approximately sparse in the 2D discrete wavelet transform (DWT) domain. A two level db4 wavelet was used there. We computed this 2D DWT, re-arranged it as a vector and computed its 99.9% energy support set. All elements not in this set were set to zero. This gave us an exactly sparse sequence $\mathbf{x}_t$. Its dimension $m = 36^2 = 1296$. For this sequence, we observed the following. The support size $\mathcal{N}_t$ satisfied $|\mathcal{N}_t| \leq S = 113$ for all t. The number of additions from $t-1$ to t satisfied $|\mathcal{N}_t \setminus \mathcal{N}_{t-1}| \leq 21$ and the number of removals, $|\mathcal{N}_{t-1} \setminus \mathcal{N}_t| \leq 26$. Thus, $S_a = 26$. Also, the initial nonzero value, $a_{j,t}$, ranged from 13 to 37, the rate of magnitude increase, $r_{j,t}$, ranged from 1 to 37, and the duration for which the increase occurred, $d_{j,t}$, ranged from 0 to 4. Also, the maximum delay between the time that a coefficient began to decrease and when it was removed was $b = 7$.

Next we consider a 64x64 functional MRI sequence. fMRI is a technique that is used to investigate brain function. The sequence we study here is for the brain responding to a certain type of stimulus (light being turned on and off). This sequence consisted of a rest state brain sequence to which activation was added based on the models suggested in [62]. The goal is to be able to accurately extract out the activation region from this sequence. As is done in [20], one can use the undersampled ReProCS algorithm to extract out the sparse activation regions from the low rank background brain image sequence, as long as an initial background brain training sequence is available. In our example, the activation started at frame 71. For the purpose of ReProCS, the active region "image"



(a) (b)

Figure 2.3: (a): plot of the BOLD signal and of its square. (b): active, transient and inactive brain regions.

(the image that is zero everywhere except in the active region), is the sparse signal of interest. For a 23 pixel region that is known to correspond to the part of the brain that is affected by the above stimulus, the activation was added follows [62]. The 23 pixel region was split into 2 sub-regions so that the activation intensity was smallest at the boundary of the region and slowly increased as one moved inwards. We show the 2 regions in Fig 2.3b. $\mathcal{R}_1$ is the innermost region, $\mathcal{R}_2$ is the outermost. The activation in these regions satisfied the following model. For $j \in \mathcal{R}_1$, $(x_t)_j = b(t)M_a$. For $j \in \mathcal{R}_2$, $(x_t)_j = 0.2b(t)^2 M_a$. Here $M_a = 1783$ is the maximum magnitude in the active region and $b(t)$ is the blood oxygenation level dependent (BOLD) signal taken from [62]. It is plotted in Fig 2.3a. This image sequence was of size 64x64, i.e. its dimension $m = 64^2 = 4096$. We computed its 99.9% energy support and set all elements not in this set to zero. This gave us our sparse sequence $\mathbf{x}_t$. The support size of $\mathbf{x}_t$, $\mathcal{N}_t$, satisfied $|\mathcal{N}_t| \leq S = 23$ for all t. The number of additions from $t-1$ to t satisfied $|\mathcal{N}_t \setminus \mathcal{N}_{t-1}| \leq S_a = 13$ and the number of removals, $|\mathcal{N}_{t-1} \setminus \mathcal{N}_t| \leq S_a = 13$. Also, the initial nonzero value, $a_{j,t}$, ranged from 57 to 97, the rate of magnitude increase, $r_{j,t}$, ranged from 1 to 637, and the duration for which the increase occurred, $d_{j,t}$, ranged from 6 to 7. Also, the maximum delay between the time that a coefficient began to decrease and when it was removed was $b = 7$.

2.7 Setting Algorithm Parameters And Simulation Results

2.7.1 Setting algorithm parameters automatically

Algorithm 1 has one parameter α. Algorithm 2 has two parameters α_{add}, α_{del}. We explain here how to set these thresholds automatically. It is often fair to assume that the noise bound on ϵ is known, e.g. it can be estimated using a short initial noise-only training sequence. We assume this here. In cases where it is not known or can change with time, one can approximate it by $\|y_{t-1} - A_{t-1}\hat{\mathbf{x}}_{t-1}\|_2$ (assuming accurate recovery at $t-1$).

Define the minimum nonzero value at time t, $x_{\min,t} = \min_{j\in\mathcal{N}_t} |(\mathbf{x}_t)_j|$. This can be estimated as $\hat{\mathbf{x}}_{\min,t} = \min_{j\in\tilde{\mathcal{T}}_{t-1}} |(\hat{\mathbf{x}}_{t-1})_j|$.

When setting the thresholds automatically, they will change with time. We set $\alpha_{\text{add},t}$ using the following heuristic. By Lemma 2.2.9, we have $(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}} = (A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}[A_{\mathcal{T}_{\text{add},t}}{}'\mathbf{w}_t + A_{\mathcal{T}_{\text{add},t}}{}' A_{\Delta_{\text{add},t}}(\mathbf{x}_t)_{\Delta_{\text{add},t}}]$. To ensure that this is bounded, we need $\|A_{\mathcal{T}_{\text{add},t}}{}^{\dagger}\|$ and $\|(A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}\|$ to be bounded. Since $\|A_{\mathcal{T}_{\text{add},t}}{}^{\dagger}\| = \frac{1}{\sigma_{\min}(A_{\mathcal{T}_{\text{add},t}})}$ and $\|(A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}\| = \frac{1}{\sigma^2_{\min}(A_{\mathcal{T}_{\text{add},t}})}$, we pick $\alpha_{\text{add},t}$ as smallest number such that $\sigma_{\min}(A_{\mathcal{T}_{\text{add},t}}) \geq 0.4$.

If one could set α_{del} equal to the lower bound on $x_{\min,t} - \|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\|_\infty$, there will be zero misses. Using this idea, we let $\alpha_{\text{del},t}$ be an estimate of the lower bound of this quantity. Notice that

$$
\begin{aligned}
&\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\|_\infty \leq \|(A^{\dagger}_{\mathcal{T}_{yy,t}} A_{\Delta_{\text{add}}} x_{t,\Delta_{\text{add}}} + A^{\dagger}_{\mathcal{T}_{\text{add},t}}\mathbf{w}_t\|_\infty \\
&\leq \|(A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}\|_\infty \|A_{\mathcal{T}_{\text{add},t}} A_{\Delta_{\text{add}}} x_{t,\Delta_{\text{add}}}\|_\infty + \|A^{\dagger}_{\mathcal{T}_{\text{add},t}}\mathbf{w}_t\|_\infty \\
&\approx \|(A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}\|_\infty C_1 \theta_{|\mathcal{T}_{\text{add},t}|,|\Delta_{\text{add}}|} C_2 \hat{x}_{\min} + \|A^{\dagger}_{\mathcal{T}_{\text{add},t}}\hat{w}_t\|_\infty,
\end{aligned}
$$

where C_1, C_2 are some constant larger than 1. Here we use the fact that for any matrix B, $\|B\|_\infty \leq C_1\|B\|$ for some constant C_1 and that only small elements are missed and hence we can approximate $\|x_{t,\Delta_{\text{add}}}\|_\infty$ by C_2 times $\hat{\mathbf{x}}_{\min,t}$ where C_2 is a small constant

larger than 1. We cannot compute $\theta_{|\mathcal{T}_{\text{add},t}|,\Delta_{\text{add}}}$, but it is fair to assume that it is small (significantly smaller than one). If we assume that

$$C_1 C_2 \|(A_{\mathcal{T}_{\text{add},t}}{}' A_{\mathcal{T}_{\text{add},t}})^{-1}\|_\infty \theta_{|\mathcal{T}_{\text{add},t}|,|\Delta_{\text{add}}|} \le 0.3,$$

then the above bound simplifies to $0.3\hat{\mathbf{x}}_{\text{min},t} + \|A^\dagger_{\mathcal{T}_{\text{add},t}} \hat{w}_t\|_\infty$. We can approximate $\hat{w}_t$ by $y_t - A\hat{\mathbf{x}}_{t,\text{modcs}}$. Thus, we set $\alpha_{\text{del},t} = 0.7\hat{x}_{\text{min},t} - \|A^\dagger_{\mathcal{T}_{\text{add},t}}(y_t - A\hat{\mathbf{x}}_{t,\text{modcs}})\|_\infty$.

For Algorithm 1, we set α_t as follows. If $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{modcs}}\|_\infty \le C x_{\text{min},t}$ for some $C < 1$, then setting $\alpha_t = (1-C)x_{\text{min},t}$ will ensure that there are no misses. If this bound holds for most entries i, then most entries will be correctly recovered, i.e., there will be few misses. If we ensure $\sigma_{\min}(A_{\tilde{\mathcal{T}}_t}) \ge 0.4$ then the number of extras will be bounded. To try to ensure that both the above hold, we let α_t to be the smallest value such that $\min_{j \in \tilde{\mathcal{T}}_t} |(\hat{\mathbf{x}}_{t,\text{modcs}})_j|_j \ge (1-C)\hat{\mathbf{x}}_{\text{min},t} = 0.5\hat{\mathbf{x}}_{\text{min},t}$ (we pick $C = 0.5$), and $\sigma_{\min}(A_{\tilde{\mathcal{T}}_t}) \ge 0.4$.

To get a more robust estimate of the minimum nonzero value of $\mathbf{x}_t$, we use a short-time average of $\{\hat{\mathbf{x}}_{\text{min},\tau}, t - t_0 \le \tau \le t\}$ as the estimate of $x_{\text{min},t}$. In our experiments, $t_0 = 10$.

2.7.2 Simulation results

In the discussion so far, we only compared sufficient conditions required by different algorithms. The general conclusion obtained by comparing the sufficient conditions was that modified-CS-add-LS-del is the best algorithm followed by modified-CS and then noisy ℓ_1. In this section, we use simulations to demonstrate the same thing. We compared noisy ℓ_1 (simple CS), i.e. solution of (2.1) at each time instant, modified-CS(mod-CS) as given in Algorithm 1, and modified-CS-add-LS-del (mod-CS-Add-LS-Del) as given in Algorithm 2. The parameters for the algorithms were set as explained in Sec 2.7.1 above.

The data was generated as follows. We used Signal Model 2 generated as explained in Appendix A.0.9 with $m = 200$, $S = 20$, $d_{\min} = 3$, $a_{\min} = r_{\min}(d_{\min}) = r$, $S_a = 2$, $b = 3$, $\ell = a_{\min} + d_{\min} r_{\min}(d_{\min}) = 4r$ and r was varied. The measurement matrices A_t

were zero mean random Gaussian $n_t \times m$ matrices with columns normalized to unit norm. We used $n_0 = 160$ and $n_t = n = 57$ for $t > 0$. The measurement noise, $(\mathbf{w}_t)_j \sim^{i.i.d.} uniform(-c_t, c_t)$ for $1 \leq j \leq m$. For $t = 0$, $c_t = 0.01266$; for $t \geq 1$, $c_t = c = 0.1266$. Here $\sim^{i.i.d.}$ means that $(\mathbf{w}_t)_j$ are independent and identically distributed (i.i.d.) both for different j's and for different t's.

In the first set of experiments shown in Fig. 2.4, we used the same measurement matrix $A_t = A$ for all $t \geq 1$. In the second experiment shown in Fig. 2.5, A_t was time varying.

The normalized mean squared error (NMSE), $\frac{\mathbb{E}[\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2]}{\mathbb{E}[\|\mathbf{x}_t\|^2]}$, the normalized mean extras, $\frac{\mathbb{E}[|\tilde{\mathcal{N}}_t \backslash \mathcal{N}_t|]}{\mathbb{E}[|\mathcal{N}_t|]}$, and the normalized mean misses, $\frac{\mathbb{E}[|\mathcal{N}_t \backslash \tilde{\mathcal{N}}_t|]}{\mathbb{E}[|\mathcal{N}_t|]}$ are used to compare the reconstruction performance. Here $\mathbb{E}[.]$ denotes the empirical mean over the 500 realizations. Consider the results of Fig 2.4. Clearly, both mod-CS and mod-CS-Add-LS-Del significantly outperform noisy ℓ_1 (simple CS). This is because for $t > 0$, the number of measurements, $n_t = 57$ is too small for a 200 length 20 sparse signal. When $a_{\min} = r_{\min}(d_{\min}) = r$ is large enough, both mod-CS and mod-CS-Add-LS-Del are stable at 5% error or less. When r is reduced, mod-CS becomes unstable. Of course when r is reduced even further to $r = 0.2$, both become unstable (not shown). If Fig 2.5, we show results for the case when A_t changes with time and all other parameters are the same as Fig 2.4 (a). Clearly in this case, the performance of both mod-CS and mod-CS-add-LS-del has improved significantly.

In Fig. 2.6, we plot the average value of $\alpha_{\text{add},t}$ for the simulations corresponding to Fig 2.5. As can be seen, this threshold is close to $4c = 4\epsilon/\sqrt{n}$ at all times.

For solving the minimization problems given in (2.1) and (2.2), we used the YALL1 software, which is provided in http://yall1.blogs.rice.edu/. Both the modified-CS algorithms and noisy ℓ_1 took roughly the same amount of time. For the results of Fig. 2.5, when running the code in MATLAB on the same server, noisy ℓ_1 needed 0.0466 seconds per frame; mod-CS needed 0.0432 seconds per frame and mod-CS-Add-LS-Del needed

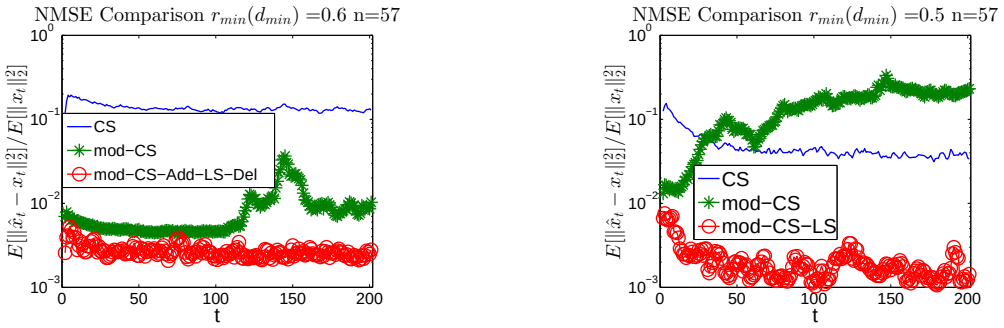


Figure 2.4: Error Comparison with Fixed Measurement Matrix. "CS" in the figures refers to noisy ℓ_1, i.e. the solution of (2.1) at each time.

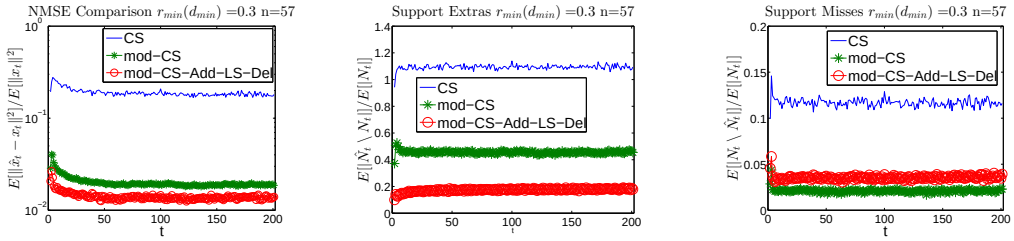


Figure 2.5: Error Comparison with Time Varying Measurement Matrices. "CS" in the figures refers to noisy ℓ_1, i.e. the solution of (2.1) at each time.

0.0517 seconds. These numbers are computed by averaging over all 500 realizations and over the 200 time instants per realization.

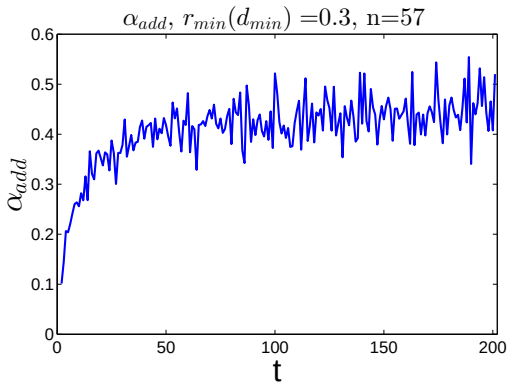


Figure 2.6: Mean of α_{add} over time.

CHAPTER 3. BATCH SPARSE RECOVERY IN LARGE AND STRUCTURED NOISE - MODIFIED PCP

3.1 Correctness Result

We first state the assumptions required for the result and then give the main result and discuss it.

3.1.1 Assumptions

As explained in [32], we need that $\mathbf{S}$ is not low rank in order to separate it from $\mathbf{L}_{\text{new}}$. One way to ensure that $\mathbf{S}$ is full rank w.h.p. is by selecting the support of $\mathbf{S}$ uniformly at random [32]. We assume this here too. In addition, we need a denseness assumption on $\mathbf{G}$ and on the left and right singular vectors of $\mathbf{L}_{\text{new}}$.

Let $n_{(1)} = \max(n_1, n_2)$ and $n_{(2)} = \min(n_1, n_2)$. Assume that following hold with a constant ρ_r that is small enough (we set its values later in Assumption 3.1.2).

$$\max_i \|[\mathbf{G}\ \mathbf{U}_{\text{new}}]^* \mathbf{e}_i\|^2 \leq \frac{\rho_r n_{(2)}}{n_1 \log^2 n_{(1)}}, \tag{3.1}$$

$$\max_i \|\mathbf{V}_{\text{new}}^* \mathbf{e}_i\|^2 \leq \frac{\rho_r n_{(2)}}{n_2 \log^2 n_{(1)}}, \tag{3.2}$$

and

$$\|\mathbf{U}_{\text{new}} \mathbf{V}_{\text{new}}^*\|_\infty \leq \sqrt{\frac{\rho_r}{n_{(1)} \log^2 n_{(1)}}}. \tag{3.3}$$

3.1.2 Main result

We state the main result in a form that is slightly different from that of [32]. It eliminates the parameter μ and combines the bound on μr directly with the incoherence assumptions (μ is a parameter defined in [32] to quantify the denseness of $\mathbf{U}$ and $\mathbf{V}$ and the incoherence between their rows) . We state it this way because it is easier to interpret and compare with the result of PCP. In particular, the dependence of the result on $n_{(2)}$ is clearer this way. The corresponding result for PCP in the same form is an immediate corollary.

Theorem 3.1.1. *Consider the problem of recovering* $\mathbf{L}$ *and* $\mathbf{S}$ *from* $\mathbf{M}$ *using partial subspace knowledge* $\mathbf{G}$ *by solving modified-PCP (1.8). Assume that* Ω*, the support set of* $\mathbf{S}$*, is uniformly distributed with size* m *satisfying*

$$m \leq 0.4\rho_s n_1 n_2 \tag{3.4}$$

Assume that $\mathbf{L}$ *satisfies (3.1), (3.2) and (3.3) and* ρ_s, ρ_r*, are small enough and* n_1, n_2 *are large enough to satisfy Assumption 3.1.2 given below. Then, Modified-PCP (1.8) with* $\lambda = 1/\sqrt{n_{(1)}}$ *recovers* $\mathbf{S}$ *and* $\mathbf{L}$ *exactly with probability at least* $1 - 23n_{(1)}^{-10}$.

Assumption 3.1.2. *Assume that* ρ_s, ρ_r *and* n_1, n_2 *satisfy:*

(a) $\rho_r \leq \min\{10^{-4}, 7.2483 \times 10^{-5} C_{03}^{-4}\}$

(b) $\rho_s = \min\{1 - 1.5b_1(\rho_r), 0.0156\}$ *where* $b_1(\rho_r) := \max\left\{60\rho_r^{1/2}, 11C_{01}\rho_r^{1/2}, 0.11\right\}$

(c) $n_{(1)} \geq \max\left\{\exp(0.5019\rho_r), \exp(253.9618C_{01}\rho_r), 1024\right\}$

(d) $n_{(2)} \geq 100\log^2 n_{(1)}$,

(e) $\frac{(n_1+n_2)^{1/6}}{\log(n_1+n_2)} > \frac{10.5}{(\rho_s)^{1/6}(1-5.6561\sqrt{\rho_s})}$,

(f) $\frac{n_{(1)}n_{(2)}}{500\log n_{(1)}} > 1/\rho_s^2$

where C_{01}, C_{03} are numerical constants from Lemma B.0.6 ([63, Theorem 4.1]) and Lemma B.0.8 ([63, Theorem 6.3]) respectively. Their expressions were not specified in the original paper.

Proof: We prove this result in Sec 3.3.

3.1.3 Discussion w.r.t. PCP

The PCP program of [32] is (1.8) with no subspace knowledge available, i.e. $\mathbf{G}_{PCP} = [\,]$ (empty matrix). With this, Theorem 3.1.1 simplifies to the corresponding result for PCP. Thus, $\mathbf{U}_{\text{new},PCP} = \mathbf{U}$ and $\mathbf{V}_{\text{new},PCP} = \mathbf{V}$ and so PCP needs

$$\max_i \|\mathbf{U}^*\mathbf{e}_i\|^2 \leq \frac{\rho_r n_{(2)}}{n_1 \log^2 n_{(1)}}, \tag{3.5}$$

$$\max_i \|\mathbf{V}^*\mathbf{e}_i\|^2 \leq \frac{\rho_r n_{(2)}}{n_2 \log^2 n_{(1)}}, \tag{3.6}$$

and

$$\|\mathbf{U}\mathbf{V}^*\|_\infty \leq \sqrt{\frac{\rho_r}{n_{(1)} \log^2 n_{(1)}}}. \tag{3.7}$$

Notice that the second and third conditions needed by modified-PCP, i.e. (3.2) and (3.3), are always weaker than (3.6) and (3.7) respectively. They are much weaker when r_{new} is small compared to r. When $r_{\text{extra}} = 0$, range($\mathbf{G}$) = range($\mathbf{U}_0$) and so the first condition is the same for both modified-PCP and PCP. When $r_{\text{extra}} > 0$ but is small, the first condition for modified-PCP is slightly stronger. However, as we argue below the third condition is the hardest to satisfy and hence in all cases except when r_{extra} is very large, the modified-PCP requirements are weaker. We demonstrate this via simulations and for some real data in Sec 3.4.2 (see Fig 3.1b and Fig 3.3b) and 3.4.5.

The third condition constrains the inner product between the rows of two basis matrices $\mathbf{U}$ and $\mathbf{V}$ while the first and second conditions only constrain the norm of the rows of a basis matrix. On first glance it may seem that the third condition is

implied by the first two using the Cauchy-Schwartz inequality. However that is not the case. Using Cauchy-Schwartz inequality, the first two conditions only imply that $\|\mathbf{UV}^*\|_\infty \leq \sqrt{\frac{\rho_r}{n_{(1)}\log^2 n_{(1)}}}\frac{\sqrt{\rho_r n_{(2)}}}{\log n_{(1)}}$ which is looser than what the third condition requires.

3.2 Online Robust PCA

Consider the online / recursive robust PCA problem where data vectors $\mathbf{y}_t := \mathbf{s}_t + \boldsymbol{\ell}_t$ come in sequentially and their subspace can change over time. Starting with an initial knowledge of the subspace, the goal is to estimate the subspace spanned by $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots \boldsymbol{\ell}_t$ and to recover the $\mathbf{s}_t$'s. Assume the following subspace change model introduced in [44]: $\boldsymbol{\ell}_t = \mathbf{P}_{(t)} 1.5_t$ where $\mathbf{P}_{(t)} = \mathbf{P}_j$ for all $t_j \leq t < t_{j+1}$, $j = 0, 1, \ldots J$. At the change times, $\mathbf{P}_j$ changes as $\mathbf{P}_j = [(\mathbf{P}_{j-1}\mathbf{R}_j \setminus \mathbf{P}_{j,\text{old}})\ \mathbf{P}_{j,\text{new}}]$ where $\mathbf{P}_{j,\text{new}}$ is a $n \times c_{j,\text{new}}$ basis matrix that satisfies $\mathbf{P}^*_{j,\text{new}}\mathbf{P}_{j-1} = 0$; $\mathbf{R}_j$ is a rotation matrix; and $\mathbf{P}_{j,\text{old}}$ is a $n \times c_{j,\text{old}}$ matrix that contains a subset of columns of $\mathbf{P}_{j-1}\mathbf{R}_j$. Also assume that $c_{j,\text{new}} \leq c$ and $\sum_j (c_{j,\text{new}} - c_{j,\text{old}}) \leq c_{dif}$. Let $r_j := \text{rank}(\mathbf{P}_j)$. Clearly, $r_j = r_{j-1} + c_{j,\text{new}} - c_{j,\text{old}}$ and so $r_j \leq r_{\max} = r_0 + c_{dif}$.

For the above model, the following is an easy corollary.

Corollary 3.2.1 (modified-PCP for online robust PCA)**.** *Let* $\mathbf{M}_j := [\mathbf{y}_{t_j}, \mathbf{y}_{t_j+1}, \ldots \mathbf{y}_{t_{j+1}-1}]$, $\mathbf{L}_j := [\boldsymbol{\ell}_{t_j}, \boldsymbol{\ell}_{t_j+1}, \ldots \boldsymbol{\ell}_{t_{j+1}-1}]$, $\mathbf{S}_j := [\mathbf{s}_{t_j}, \mathbf{s}_{t_j+1}, \ldots \mathbf{s}_{t_{j+1}-1}]$ *and let* $\mathbf{L}_{full} := [\mathbf{L}_1, \mathbf{L}_2, \ldots \mathbf{L}_J]$ *and* $\mathbf{S}_{full} := [\mathbf{S}_1, \mathbf{S}_2, \ldots \mathbf{S}_J]$. *Suppose that the following hold.*

1. $\mathbf{S}_{full}$ *satisfies the assumptions of Theorem 3.1.1.*
2. *The initial subspace range*$(\mathbf{P}_0)$ *is exactly known, i.e. we are given* $\hat{\mathbf{P}}_0$ *with range*$(\hat{\mathbf{P}}_0) =$ *range*$(\mathbf{P}_0)$.
3. *For all* $j = 1, 2, \ldots J$, *(3.1), (3.2), and (3.3) hold with* $n_1 = n$, $n_2 = t_{j+1} - t_j$, $\mathbf{G} = \mathbf{P}_{j-1}$, $\mathbf{U}_{new} = \mathbf{P}_{j,new}$ *and* $\mathbf{V}_{new}$ *being the matrix of right singular vectors of* $\mathbf{L}_{new} = (\mathbf{I} - \mathbf{P}_{j-1}\mathbf{P}^*_{j-1})\mathbf{L}_j$.

4. *We solve modified-PCP at every* $t = t_{j+1}$, *using* $\mathbf{M} = \mathbf{M}_j$ *and with* $\mathbf{G} = \mathbf{G}_j = \hat{\mathbf{P}}_{j-1}$ *where* $\hat{\mathbf{P}}_{j-1}$ *is the matrix of left singular vectors of the reduced SVD of* $\hat{\mathbf{L}}_{j-1}$ *(the low-rank matrix obtained from modified-PCP on* $\mathbf{M}_{j-1}$*). At* $t = t_1$ *we use* $\mathbf{G} = \hat{\mathbf{P}}_0$.

Then, modified-PCP recovers $\mathbf{S}_{full}, \mathbf{L}_{full}$ *exactly and in a piecewise batch fashion with probability at least* $(1 - 23n^{-10})^J$.

Proof. Denote by Θ_0 the event that $\text{range}(\hat{\mathbf{P}}_0) = \text{range}(\mathbf{P}_0)$. For $j = 1, 2, \dots J$, denote by Θ_j the event that the program (1.8) succeeds for the matrix $\mathbf{M} = \mathbf{M}_j$, i.e. $\mathbf{S}_j$ and $\mathbf{L}_j$ are exactly recovered. Clearly, Θ_j also implies that $\text{range}(\hat{\mathbf{P}}_j) = \text{range}(\mathbf{P}_j)$. Using Theorem 3.1.1 and the model, we then get that probability $\mathbb{P}(\Theta_j | \Theta_0, \Theta_1, \dots \Theta_{j-1}) \geq 1 - 23n^{-10}$. Also, by assumption, $\mathbb{P}(\Theta_0) = 1$. Thus by chain rule, $\mathbb{P}(\Theta_0, \Theta_1, \Theta_2, \cdots, \Theta_J) \geq (1 - 23n^{-10})^J$. □

Discussion w.r.t. PCP. For the data model above, two possible corollaries for PCP can be stated.

Corollary 3.2.2 (PCP for online robust PCA)**.** *If* $\mathbf{S}_{full}$ *satisfies the assumptions of Theorem 3.1.1 and if (3.1), (3.2), and (3.3) hold with* $n_1 = n$, $n_2 = t_{J+1} - t_1$, $\mathbf{G}_{PCP} = [\,]$, $\mathbf{U}_{new,PCP} = \mathbf{U} = [\mathbf{P}_0, \mathbf{P}_{1,new}, \dots \mathbf{P}_{J,new}]$ *and* $\mathbf{V}_{new,PCP} = \mathbf{V}$ *being the right singular vectors of* $\mathbf{L}_{full} := [\mathbf{L}_1, \mathbf{L}_2, \dots \mathbf{L}_J]$, *then, we can recover* $\mathbf{L}_{full}$ *and* $\mathbf{S}_{full}$ *exactly with probability at least* $(1 - 23n^{-10})$ *by solving PCP (1.1) with input* $\mathbf{M}_{full}$. *Here* $\mathbf{M}_{full} := \mathbf{L}_{full} + \mathbf{S}_{full}$.

When we compare this with the result for modified-PCP, the second and third condition are even more significantly weaker than those for PCP. The reason is that $\mathbf{V}_{\text{new}}$ contains at most c columns while $\mathbf{V}$ contains at most $r_0 + Jc$ columns. The first conditions cannot be easily compared. The LHS contains at most $r_{\max} + c = r_0 + c_{dif} + c$ columns for modified-PCP, while it contains $r_0 + Jc$ columns for PCP. However, the RHS for PCP is also larger. If $t_{j+1} - t_j = d$, then the RHS is also J times larger for PCP than for modified-PCP. The above advantage for mod-PCP comes with two caveats. First,

modified-PCP assumes knowledge of the subspace change times while PCP does not need this. Secondly, modified-PCP succeeds w.p. $(1-23n^{-10})^J \geq 1-23Jn^{-10}$ while PCP succeeds w.p. $1-23n^{-10}$. Alternatively if PCP is solved at every $t=t_{j+1}$ using $\mathbf{M}_j$, we get the following corollary

Corollary 3.2.3 (PCP for $\mathbf{M}_j$)**.** *Solve PCP, i.e. (1.1), at $t=t_{j+1}$ using M_j. If $\mathbf{S}_{full}$ satisfies the assumptions of Theorem 3.1.1 and if (3.1), (3.2), and (3.3) hold with $n_1=n$, $n_2=t_{j+1}-t_j$, $\mathbf{G}_{PCP}=[\,]$, $\mathbf{U}_{new,PCP}=\mathbf{P}_j$ and $\mathbf{V}_{new,PCP}=\mathbf{V}_j$ being the right singular vectors of $\mathbf{L}_j$ for all $j=1,2,\ldots,J$, then, we can recover $\mathbf{L}_{full}$ and $\mathbf{S}_{full}$ exactly with probability at least $(1-23n^{-10})^J$.*

When we compare this with modified-PCP, the second and third condition are significantly weaker than those for PCP when $c_{j,\text{new}} \ll r_j$. The first condition is exactly the same when $c_{j,\text{old}}=0$ and is only slightly stronger as long as $c_{j,\text{old}} \ll r_j$.

Discussion w.r.t. ReProCS. In [21, 64, 44], Qiu et al studied the online / recursive robust PCA problem and proposed a novel recursive algorithm called ReProCS. With the subspace change model described above, they also needed the following "slow subspace change" assumption: $\|P^*_{j,\text{new}}\boldsymbol{\ell}_t\|$ is small for sometime after t_j and increases gradually. Modified-PCP does not need this. Moreover, even with perfect initial subspace knowledge, ReProCS cannot achieve exact recovery of $\mathbf{s}_t$ or $\boldsymbol{\ell}_t$ while, as shown above, modified-PCP can. On the other hand, ReProCS is a recursive algorithm while modified-PCP is not; and for highly correlated support changes of the $\mathbf{s}_t$'s, ReProCS outperforms modified-PCP (see Sec 3.4). The reason is that correlated support change results in $\mathbf{S}$ also being rank deficient, thus making it difficult to separate it from $\mathbf{L}_{\text{new}}$ by modified-PCP.

Discussion w.r.t. the work of Feng et al. Recent work of Feng et. al. [65, 66] provides two asymptotic results for online robust PCA. The first work [65] does not model the outlier as a sparse vector but just as a vector that is "far" from the low-dimensional data

subspace. In [66], the authors reformulate the PCP program and use this to develop a recursive algorithm that comes "close" to the PCP solution asymptotically.

3.3 Proof of Theorem 3.1.1: Main Lemmas

Our proof adapts the proof approach of [32] to our new problem and the modified-PCP solution. The main new lemma is Lemma 3.3.7 in which we obtain different and weaker conditions on the dual certificate to ensure exact recovery. This lemma is given and proved in Sec 3.3.5. In addition, we provide a proof for two key statements from [32] for which either a proof is not immediate (Lemma 3.3.1) or for which the cited reference does not work (Lemma 3.3.2). These lemmas are given below in Sec 3.3.1 and proved in the Appendix.

We state Lemma 3.3.1 and Lemma 3.3.2 in Sec 3.3.1. We give the overall proof architecture next in Sec 4.4. Some definitions and basic facts are given in Sec 4.5.2 and 3.3.3. In Sec 3.3.5, we obtain sufficient conditions (on the dual certificate) under which $\mathbf{S}, \mathbf{L}_{\text{new}}$ is the unique minimizer of modified-PCP. In Sec 3.3.6, we construct a dual certificate that satisfies the required conditions with high probability (w.h.p.). Here, we also give the two main lemmas to show that this indeed satisfies the required conditions. The proof of all the four lemmas from this section is given in the Appendix.

Whenever we say "with high probability" or w.h.p., we mean with probability at least $1 - O(1)n_{(1)}^{-10}$.

3.3.1 Two lemmas

Lemma 3.3.1. *Denote by $\mathbb{P}_{Unif}$ and $\mathbb{P}_{Ber}$ the probabilities calculated under the uniform and Bernoulli models and let "Success" be the event that $(\mathbf{L}_{new}, \mathbf{S}, \mathbf{L}^*\mathbf{G})$ is the unique solution of modified-PCP (1.8). Then*

$$\mathbb{P}_{Unif(m_0)}(Success) \geq \mathbb{P}_{Ber(\rho_0)}(Success) - e^{-2n_1n_2\epsilon_0^2}, \text{ where } \rho_0 = \frac{m_0}{n_1n_2} + \epsilon_0.$$

The proof is given in Appendix B.0.11. A similar statement is given in Appendix A.1 of [32] but without a proof. The expression for the second term on the right hand side given there is $e^{-\frac{2n_1n_2\epsilon_0^2}{\rho_0}}$ which is different from the one we derive.

Lemma 3.3.2. *Let* $\mathbf{E}$ *be a* $n_1 \times n_2$ *random matrix with entries i.i.d. (independently identically distributed) as*

$$\mathbf{E}_{ij} = \begin{cases} 1, & w.\ p.\ \rho_s/2, \\ 0, & w.\ p.\ 1-\rho_s, \\ -1, & w.\ p.\ \rho_s/2. \end{cases} \tag{3.8}$$

If $\rho_s < 0.03$ *and* $\frac{(n_1+n_2)^{1/6}}{\log(n_1+n_2)} > \frac{10.5}{(\rho_s)^{1/6}(1-5.6561\sqrt{\rho_s})}$, *then*

$$\mathbb{P}(\|\mathbf{E}\| \geq 0.5\sqrt{n_{(1)}}) \leq n_{(1)}^{-10}.$$

The proof is provided in Appendix B.0.12 and uses the result of [67]. In [32], the authors claim that using [68], $\|\mathbf{E}\| > 0.25\sqrt{n_{(1)}}$ w.p. less than $n_{(1)}^{-10}$. While the claim is correct, it is not possible to prove it using any of the results from [68]. Using ideas from [68], one can only show that the above holds when $n_{(2)}$ is upper bounded by a constant times $\log n_{(1)}$ (see the Appendix of [69]) which is a strong extra assumption.

3.3.2 Proof architecture

The proof of the theorem involves 4 main steps.

(a) The first step is to show that when the locations of the support of $\mathbf{S}$ are Bernoulli distributed with parameter ρ_s and the signs of $\mathbf{S}$ are i.i.d ± 1 with probability 1/2 (and independent from the locations), and all the other assumptions

on $\mathbf{L}, n_1, n_2, \rho_s, \rho_r$ in Theorem 3.1.1 are satisfied, then Modified-PCP (1.8) with $\lambda = 1/\sqrt{n_{(1)}}$ recovers $\mathbf{S}$ exactly (and hence also $\mathbf{L} = \mathbf{M} - \mathbf{S}$) with probability at least $1 - 22n_{(1)}^{-10}$.

(b) By [32, Theorem 2.3], the previous claim also holds for the model in which the signs of $\mathbf{S}$ are fixed and the locations of its nonzero entries are sampled from the Bernoulli model with parameter $\rho_s/2$, and all the other assumptions on $\mathbf{L}, n_1, n_2, \rho_s, \rho_r$ from Theorem 3.1.1 are satisfied.

(c) By Lemma 3.3.1 with $\epsilon_0 = 0.1\rho_s$, $m_0 = \lfloor 0.4\rho_s n_1 n_2 \rfloor$, since $n_1 n_2 > 500 \log n_1/\rho_s^2$ (Assumption 3.1.2(f)), the previous claim holds with probability at least $1 - 23n_{(1)}^{-10}$ for the model in which the signs of $\mathbf{S}$ are fixed and the locations of its nonzero entries are sampled from the Uniform model with parameter m_0, and all the other assumptions on $\mathbf{L}, n_1, n_2, \rho_s, \rho_r$ from Theorem 3.1.1 are satisfied.

(d) By [32, Theorem 2.2], the previous claim also holds for the model in which the signs of $\mathbf{S}$ are fixed and the locations of its nonzero entries are sampled from the Uniform model with parameter $m \leq m_0 = 0.4\rho_s n_1 n_2$, and all the other assumptions on $\mathbf{L}, n_1, n_2, \rho_s, \rho_r$ from Theorem 3.1.1 are satisfied.

Thus, all we need to do is to prove step (a). To do this we start with the KKT conditions and strengthen them to get a set of easy to satisfy sufficient conditions on the dual certificate under which $\mathbf{L}_{\text{new}}, \mathbf{S}$ is the unique minimizer of (1.8). This is done in Sec 3.3.5. Next, we use the golfing scheme [70, 32] to construct a dual certificate that satisfies the required conditions (Sec. 3.3.6).

3.3.3 Basic facts

We state some basic facts which will be used in the following proof.

Definition 3.3.3 (Sub-gradient [71]). *Consider a convex function $f : \mathbb{O} \to \mathbb{R}$ on a convex set of matrices $\mathbb{O}$. A matrix $\mathbf{Y}$ is called its sub-gradient at a point $\mathbf{X}_0 \in \mathbb{O}$ if*

$$f(\mathbf{X}) - f(\mathbf{X}_0) \geq \langle \mathbf{Y}, (\mathbf{X} - \mathbf{X}_0) \rangle.$$

for all $\mathbf{X} \in \mathbb{O}$. The set of all sub-gradients of f at $\mathbf{X}_0$ is denoted by $\partial f(\mathbf{X}_0)$.

It is known [72, 73] that

$$\partial \|\mathbf{L}_{\text{new}}\|_* = \{\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W} : \mathbb{P}_{T_{\text{new}}}\mathbf{W} = 0, \|\mathbf{W}\| \leq 1\}.$$

and

$$\partial \|\mathbf{S}\|_1 = \{\mathbf{F} : \mathbb{P}_{\Omega}\mathbf{F} = \text{sgn}(\mathbf{S}), \|\mathbf{F}\|_\infty \leq 1\}.$$

Definition 3.3.4 (Dual norm [39]). *The matrix norm $\|\cdot\|_\heartsuit$ is said to be dual to matrix norm $\|\cdot\|_\spadesuit$ if, for all $\mathbf{Y}_1 \in \mathbb{R}^{n_1 \times n_2}$, $\|\mathbf{Y}_1\|_\heartsuit = \sup_{\|\mathbf{Y}_2\|_\spadesuit \leq 1} \langle \mathbf{Y}_1, \mathbf{Y}_2 \rangle$.*

Proposition 3.3.5 (Proposition 2.1 of [74]). *The following pairs of matrix norms are dual to each other:*

- *$\|\cdot\|_1$ and $\|\cdot\|_\infty$;*
- *$\|\cdot\|_*$ and $\|\cdot\|$;*
- *$\|\cdot\|_F$ and $\|\cdot\|_F$.*

For all these pairs, the following hold.

1. *$|\langle \mathbf{Y}, \mathcal{Z} \rangle| \leq \|\mathbf{Y}\|_\spadesuit \|\mathcal{Z}\|_\heartsuit$.*
2. *Fixing any $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$, there exists $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2}$ (that depends on $\mathbf{Y}$) such that*

$$\langle \mathbf{Y}, \mathcal{Z} \rangle = \|\mathbf{Y}\|_\spadesuit \|\mathcal{Z}\|_\heartsuit.$$

3. *In particular, we can get $\langle \mathbf{Y}, \mathcal{Z} \rangle = \|\mathbf{Y}\|_1 \|\mathcal{Z}\|_\infty$ by setting $\mathcal{Z} = sgn(\mathbf{Y})$, we can get $\langle \mathbf{Y}, \mathcal{Z} \rangle = \|\mathbf{Y}\|_* \|\mathcal{Z}\|$ by setting $\mathcal{Z} = \mathbf{U}_Y\mathbf{V}_Y^*$ where $\mathbf{U}_Y\mathbf{\Sigma}_Y\mathbf{V}_Y^*$ is the SVD of $\mathbf{Y}$, and we can get $\langle \mathbf{Y}, \mathcal{Z} \rangle = \|\mathbf{Y}\|_F \|\mathcal{Z}\|_F$ by letting $\mathcal{Z} = \mathbf{Y}$.*

For any matrix $\mathbf{Y}$, we have

$$\|\mathbf{Y}\|_F^2 = \mathrm{tr}(\mathbf{Y}^*\mathbf{Y}) = \sum_{i,j} |\mathbf{Y}_{ij}|^2 \leq (\sum_{i,j} |\mathbf{Y}_{ij}|)^2 = \|\mathbf{Y}\|_1^2$$

and

$$\|\mathbf{Y}\|_F^2 = \mathrm{tr}(\mathbf{Y}^*\mathbf{Y}) = \sum_{i} \sigma_i^2(\mathbf{Y}) \leq (\sum_{i} \sigma_i(\mathbf{Y}))^2 = \|\mathbf{Y}\|_*^2$$

Let Υ be the linear space of matrices with column span equal to that of the columns of $\mathbf{P}_1$ and row span equal to that of the columns of $\mathbf{P}_2$ where $\mathbf{P}_1$ and $\mathbf{P}_2$ are basis matrices. Then, for a matrix $\mathbf{M}$,

$$\mathcal{P}_{\Upsilon^\perp}\mathbf{M} = (\mathbf{I} - \mathbf{P}_1\mathbf{P}_1^*)\mathbf{M}(\mathbf{I} - \mathbf{P}_2\mathbf{P}_2^*) \text{ and } \mathcal{P}_{\Upsilon}\mathbf{M} = \mathbf{M} - \mathcal{P}_{\Upsilon^\perp}\mathbf{M}.$$

Let Υ be the linear space of matrices with column span equal to that of the columns of $\mathbf{P}_1$. Then,

$$\mathcal{P}_{\Upsilon^\perp}\mathbf{M} = (\mathbf{I} - \mathbf{P}_1\mathbf{P}_1^*)\mathbf{M} \text{ and } \mathcal{P}_{\Upsilon}\mathbf{M} = \mathbf{P}_1\mathbf{P}_1^*\mathbf{M}$$

For a matrix $\mathbf{x}\mathbf{y}^*$ where $\mathbf{x}$ and $\mathbf{y}$ are vectors,

$$\|\mathbf{x}\mathbf{y}^*\|_F^2 = \|\mathbf{x}\|^2\|\mathbf{y}\|^2.$$

If an operator $\mathcal{A}$ is linear and bounded, then [75]

$$\|\mathcal{A}^*\mathcal{A}\| = \|\mathcal{A}\|^2.$$

3.3.4 Definitions

Here we define the following linear spaces of matrices.

Denote by Γ the linear space of matrices with column span equal to that of the columns of $\mathbf{G}$, i.e.

$$\Gamma := \{\mathbf{G}\mathbf{Y}^*, \mathbf{Y} \in \mathbb{R}^{n_2 \times r_G}\}, \tag{3.9}$$

and by $\Gamma^\perp$ its orthogonal complement.

Define also the following linear spaces of matrices

$$T_{\text{new}} := \{\mathbf{U}_{\text{new}}\mathbf{Y}_1^* + \mathbf{Y}_2\mathbf{V}_{\text{new}}^*,\ \mathbf{Y}_1 \in \mathbb{R}^{n_2\times r_{\text{new}}}, \mathbf{Y}_2 \in \mathbb{R}^{n_1\times r_{\text{new}}}\},$$

$$\Pi := \{[\mathbf{G}\ \mathbf{U}_{\text{new}}]\mathbf{Y}_1^* + \mathbf{Y}_2\mathbf{V}_{\text{new}}^*,\ \mathbf{Y}_1 \in \mathbb{R}^{n_2\times (r_G+r_{\text{new}})}, \mathbf{Y}_2 \in \mathbb{R}^{n_1\times r_{\text{new}}}\},$$

Notice that $T_{\text{new}} \cup \Gamma = \Pi$.

Remark 3.3.6. *For the matrix $\mathbf{e}_i\mathbf{e}_j^*$, together with (3.1) and (3.2), we have*

$$\begin{aligned} &\|\mathbb{P}_{\Pi^\perp}\mathbf{e}_i\mathbf{e}_j^*\|_F^2 \\ =&\|(\mathbf{I} - [\mathbf{G}\ \mathbf{U}_{new}][\mathbf{G}\ \mathbf{U}_{new}]^*)\mathbf{e}_i\|^2\|(\mathbf{I} - \mathbf{V}_{new}\mathbf{V}_{new}^*)\mathbf{e}_j\|^2 \\ \geq&(1 - \rho_r/\log^2 n_{(1)})^2, \end{aligned} \tag{3.10}$$

where $\rho_r/\log^2 n_{(1)} \leq 1$ as assumed. Using $\|\mathbb{P}_\Pi\mathbf{e}_i\mathbf{e}_j^*\|_F^2 + \|\mathbb{P}_{\Pi^\perp}\mathbf{e}_i\mathbf{e}_j^*\|_F^2 = 1$, we have

$$\|\mathbb{P}_\Pi\mathbf{e}_i\mathbf{e}_j^*\|_F \leq \sqrt{\frac{2\rho_r}{\log^2 n_{(1)}}}. \tag{3.11}$$

3.3.5 Dual certificates

We modify Lemma 2.5 of [32] to get the following lemma which gives us sufficient conditions on the dual certificate needed to ensure that modified-PCP succeeds.

Lemma 3.3.7. *If $\|\mathbb{P}_\Omega\mathbb{P}_\Pi\| \leq 1/4$, $\lambda < 3/10$, and there is a pair $(\mathbf{W}, \mathbf{F})$ obeying*

$$\mathbf{U}_{new}\mathbf{V}_{new}^* + \mathbf{W} = \lambda(sgn(\mathbf{S}) + \mathbf{F} + \mathbb{P}_\Omega\mathbf{D})$$

with $\mathbb{P}_\Pi\mathbf{W} = \mathbf{0}$, $\|\mathbf{W}\| \leq \frac{9}{10}$, $\mathbb{P}_\Omega\mathbf{F} = \mathbf{0}$, $\|\mathbf{F}\|_\infty \leq \frac{9}{10}$, and $\|\mathbb{P}_\Omega\mathbf{D}\|_F \leq \frac{1}{4}$, then $(\mathbf{L}_{new}, \mathbf{S}, \mathbf{L}^*\mathbf{G})$ is the unique solution to Modified-PCP (1.8).

Proof. Any feasible perturbation of $(\mathbf{L}_{\text{new}}, \mathbf{S}, \mathbf{L}^*\mathbf{G})$ will be of the form

$$(\mathbf{L}_{\text{new}} + \mathbf{H}_1, \mathbf{S} - \mathbf{H}, \mathbf{L}^*\mathbf{G} + \mathbf{H}_2),\ \text{with}\ \mathbf{H}_1 + \mathbf{G}\mathbf{H}_2^* = \mathbf{H}.$$

Let $\mathbf{G}_\perp$ be a basis matrix that is such that $[\mathbf{G}\ \mathbf{G}_\perp]$ is a unitary matrix. Then, $\mathbf{H}_1 = \mathbf{H} - \mathbf{G}\mathbf{H}_2^* = \mathbf{G}_\perp\mathbf{G}_\perp^*\mathbf{H} + \mathbf{G}\mathbf{G}^*\mathbf{H} - \mathbf{G}\mathbf{H}_2^*$. Notice that

- $\mathbf{L}_{\text{new}} = \mathbf{G}_\perp \mathbf{G}_\perp^* \mathbf{L}_{\text{new}}$ and $\mathbf{G}_\perp \mathbf{G}_\perp^* \mathbf{H} = \mathbb{P}_{\Gamma^\perp} \mathbf{H}$.
- For any two matrices $\mathbf{Y}_1$ and $\mathbf{Y}_2$,

$$\|\mathbf{G}_\perp \mathbf{Y}_1 + \mathbf{G}\mathbf{Y}_2\|_* \geq \|\mathbf{G}_\perp \mathbf{Y}_1\|_*$$

  where equality holds if and only if $\mathbf{Y}_2 = \mathbf{0}$. To see why this holds, let the full SVD of $\mathbf{Y}_1, \mathbf{Y}_2$ be $\mathbf{Y}_1 \overset{\text{SVD}}{=} \mathcal{Q}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^*$ and $\mathbf{Y}_2 \overset{\text{SVD}}{=} \mathcal{Q}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^*$. Since $[\mathbf{G}\ \mathbf{G}_\perp]$ is a unitary matrix, $\mathbf{G}_\perp \mathbf{Y}_1 + \mathbf{G}\mathbf{Y}_2 \overset{\text{SVD}}{=} [\mathbf{G}_\perp \mathcal{Q}_1\ \mathbf{G}\mathcal{Q}_2] \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} [\mathbf{V}_1\ \mathbf{V}_2]^*$. Thus, $\|\mathbf{G}_\perp \mathbf{Y}_1 + \mathbf{G}\mathbf{Y}_2\|_* = \text{trace}(\mathbf{\Sigma}_1) + \text{trace}(\mathbf{\Sigma}_2) \geq \text{trace}(\mathbf{\Sigma}_1) = \|\mathbf{G}_\perp \mathbf{Y}_1\|_*$ where equality holds if and only if $\mathbf{\Sigma}_2 = \mathbf{0}$, or equivalently, $\mathbf{Y}_2 = \mathbf{0}$.

Thus,

$$\begin{aligned} &\|\mathbf{L}_{\text{new}} + \mathbf{H}_1\|_* \\ &= \|\mathbf{G}_\perp(\mathbf{G}_\perp^* \mathbf{L}_{\text{new}} + \mathbf{G}_\perp^* \mathbf{H}) + \mathbf{G}(\mathbf{G}^* \mathbf{H} - \mathbf{H}_2^*)\|_* \\ &\geq \|\mathbf{G}_\perp(\mathbf{G}_\perp^* \mathbf{L}_{\text{new}} + \mathbf{G}_\perp^* \mathbf{H})\|_* = \|\mathbf{L}_{\text{new}} + \mathbb{P}_{\Gamma^\perp} \mathbf{H}\|_* \end{aligned} \tag{3.12}$$

where equality holds if and only if $\mathbf{H}_2 = \mathbf{G}^* \mathbf{H}$.

Recall that $T_{\text{new}} \cup \Gamma = \Pi$. Choose a $\mathbf{W}_a$ so that $\langle \mathbf{W}_a, \mathbb{P}_{\Pi^\perp} \mathbf{H} \rangle = \|\mathbb{P}_{\Pi^\perp} \mathbf{H}\|_* \|\mathbf{W}_a\|$. This is possible using Proposition 3.3.5. Let

$$\mathbf{W}_0 = \mathbb{P}_{\Pi^\perp} \mathbf{W}_a / \|\mathbf{W}_a\|.$$

Thus, $\mathbf{W}_0$ satisfies $\mathbb{P}_{T_{\text{new}}} \mathbf{W}_0 = \mathbf{0}$ and $\|\mathbf{W}_0\| \leq 1$ and so it belongs to the sub-gradient set of the nuclear norm at $\mathbf{L}_{\text{new}}$. Also,

$$\begin{aligned} \langle \mathbf{W}_0, \mathbb{P}_{\Gamma^\perp} \mathbf{H} \rangle &= \frac{1}{\|\mathbf{W}_a\|} \langle \mathbb{P}_{\Pi^\perp} \mathbf{W}_a, \mathbb{P}_{\Gamma^\perp} \mathbf{H} \rangle \\ &= \frac{1}{\|\mathbf{W}_a\|} \langle \mathbf{W}_a, \mathbb{P}_{\Pi^\perp} \mathbb{P}_{\Gamma^\perp} \mathbf{H} \rangle \\ &= \frac{1}{\|\mathbf{W}_a\|} \langle \mathbf{W}_a, \mathbb{P}_{\Pi^\perp} \mathbf{H} \rangle = \|\mathbb{P}_{\Pi^\perp} \mathbf{H}\|_*. \end{aligned}$$

Let $\mathbf{F}_0 = -\text{sgn}(\mathbb{P}_{\Omega^\perp}\mathbf{H})$. Thus, $\mathbb{P}_\Omega \mathbf{F}_0 = \mathbf{0}$, $\|\mathbf{F}_0\|_\infty = 1$ and so it belongs to the sub-gradient set of the 1-norm at $\mathbf{S}$. Also,

$$\langle \mathbf{F}_0, \mathbf{H}\rangle = \langle \mathbf{F}_0, \mathbb{P}_{\Omega^\perp}\mathbf{H}\rangle = -\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1.$$

Thus,

$$\begin{aligned}
&\|\mathbf{L}_{\text{new}} + \mathbf{H}_1\|_* + \lambda\|\mathbf{S} - \mathbf{H}\|_1 \\
\geq &\|\mathbf{L}_{\text{new}} + \mathbb{P}_{\Gamma^\perp}\mathbf{H}\|_* + \lambda\|\mathbf{S} - \mathbf{H}\|_1 \\
&\text{(using (3.12))} \\
\geq &\|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W}_0, \mathbb{P}_{\Gamma^\perp}\mathbf{H}\rangle \\
&- \lambda\langle \text{sgn}(\mathbf{S}) + \mathbf{F}_0, \mathbf{H}\rangle \\
&\text{(by definition of sub-gradient)} \\
= &\|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1 + \|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_* + \lambda\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1 \\
&+ \langle \mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* - \lambda\text{sgn}(\mathbf{S}), \mathbf{H}\rangle \\
&\text{(using } \mathbf{W}_0 \text{ and } \mathbf{F}_0 \text{ as defined above)} \\
\geq &\|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1 + \|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_* + \lambda\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1 \\
&- \max(\|\mathbf{W}\|, \|\mathbf{F}\|_\infty)(\|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_* + \lambda\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1) + \lambda\langle \mathbb{P}_\Omega\mathbf{D}, \mathbf{H}\rangle \\
&\text{(by the lemma's assumption and Proposition 3.3.5)} \\
\geq &\|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1 + \frac{1}{10}\Big(\|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_* + \lambda\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1\Big) \\
&- \frac{\lambda}{4}\|\mathbb{P}_\Omega\mathbf{H}\|_F \\
&\text{(by Proposition 3.3.5 and assumption } \|\mathbb{P}_\Omega\mathbf{D}\|_F \leq \frac{1}{4}\text{)}
\end{aligned}$$

Observe now that

$$\begin{aligned}
\|\mathbb{P}_\Omega\mathbf{H}\|_F &\leq \|\mathbb{P}_\Omega\mathbb{P}_\Pi\mathbf{H}\|_F + \|\mathbb{P}_\Omega\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_F \\
&\leq \frac{1}{4}\|\mathbf{H}\|_F + \|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_F \\
&\leq \frac{1}{4}\|\mathbb{P}_\Omega\mathbf{H}\|_F + \frac{1}{4}\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_F + \|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_F
\end{aligned}$$

and, therefore,

$$\begin{aligned}\|\mathbb{P}_{\Omega}\mathbf{H}\|_F &\leq \frac{1}{3}\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_F + \frac{4}{3}\|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_F \\ &\leq \frac{1}{3}\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1 + \frac{4}{3}\|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_*\end{aligned}$$

In conclusion,

$$\begin{aligned}&\|\mathbf{L}_{\text{new}} + \mathbb{P}_{T^\perp}\mathbf{H}\|_* + \lambda\|\mathbf{S} - \mathbf{H}\|_1 \\ &\geq \|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1 + \left((\frac{1}{10} - \frac{\lambda}{3})\|\mathbb{P}_{\Pi^\perp}\mathbf{H}\|_* + \frac{\lambda}{60}\|\mathbb{P}_{\Omega^\perp}\mathbf{H}\|_1\right) \\ &> \|\mathbf{L}_{\text{new}}\|_* + \lambda\|\mathbf{S}\|_1\end{aligned}$$

The last inequality holds because $\|\mathbb{P}_{\Omega}\mathbb{P}_{\Pi}\| < 1$ and this implies that $\Pi \cap \Omega = \{0\}$ and so at least one of $\mathbb{P}_{\Pi^\perp}\mathbf{H}$ or $\mathbb{P}_{\Omega^\perp}\mathbf{H}$ is strictly positive for $\mathbf{H} \neq \mathbf{0}$. Thus, the cost function is strictly increased by any feasible perturbation. Since the cost is convex, this proves the lemma. □

Lemma 3.3.7 is equivalently saying that $(\mathbf{L}_{\text{new}}, \mathbf{S}, \mathbf{L}^*\mathbf{G})$ is the unique solution to Modified-PCP (1.8) if there is a $\mathbf{W}$ satisfying:

$$\begin{cases}\mathbf{W} \in \Pi^\perp, \\ \|\mathbf{W}\| \leq 9/10, \\ \|\mathbb{P}_{\Omega}(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* - \lambda\text{sgn}(\mathbf{S}) + \mathbf{W})\|_F \leq \lambda/4, \\ \|\mathbb{P}_{\Omega^\perp}(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W})\|_\infty < 9\lambda/10.\end{cases} \tag{3.13}$$

3.3.6 Construction of the required dual certificate

The golfing scheme is introduced by [76, 70]; here we use it with some modifications similar to those in [32] to construct dual certificate. Assume that $\Omega \backsim \text{Ber}(\rho_s)$ or equivalently, $\Omega^c \backsim \text{Ber}(1 - \rho_s)$.

Notice that Ω^c can be generated as a union of j_0 i.i.d. sets $\{\bar{\Omega}_j\}_{j=1}^{j_0}$, where $\bar{\Omega}_j \overset{i.i.d}{\sim}$ Ber$(q), 1 \leq j \leq j_0$ with q, j_0 satisfying $\rho_s = (1-q)^{j_0}$. This is true because

$$\mathbb{P}((i,j) \in \Omega) = \mathbb{P}((i,j) \notin \bar{\Omega}_1 \cup \bar{\Omega}_2 \cup \cdots \bar{\Omega}_{j_0}) = (1-q)^{j_0}.$$

As there is overlap between $\bar{\Omega}_j's$, we have $q \geq (1-\rho_s)/j_0$.

Let $\mathbf{W} = \mathbf{W}^L + \mathbf{W}^S$, where $\mathbf{W}^L, \mathbf{W}^S$ are constructed similar to [32] as:

- *Construction of* $\mathbf{W}^L$ *via the golfing scheme.* Let $\mathbf{Y}_0 = 0$,

  $$\mathbf{Y}_j = \mathbf{Y}_{j-1} + q^{-1}\mathbb{P}_{\bar{\Omega}_j}\mathbb{P}_{\Pi}(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* - \mathbf{Y}_{j-1}),$$

  and $\mathbf{W}^L = \mathbb{P}_{\Pi^\perp}\mathbf{Y}_{j_0}$. Notice that $\mathbf{Y}_j \in \Omega^\perp$.

- *Construction of* $\mathbf{W}^S$ *via the method of least squares.* Assume that $\|\mathbb{P}_\Omega\mathbb{P}_\Pi\| \leq 1/4$. We prove that this holds in Lemma 3.3.9 below. With this, $\|\mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega\| = \|\mathbb{P}_\Omega\mathbb{P}_\Pi\|^2 \leq 1/16$ and so $\|\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega\| \geq 1 - 1/16 > 0$. Thus this operator, which maps the subspace Ω onto itself, is invertible. Let $(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}$ denote its inverse and let

  $$\mathbf{W}^S = \lambda\mathbb{P}_{\Pi^\perp}(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S}).$$

  Using the Neumann series, notice that [32]

  $$(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S}) = \sum_{k\geq 0}(\mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^k\text{sgn}(\mathbf{S}).$$

Thus [32],

$$\mathbb{P}_\Omega\mathbf{W}^S = \lambda\text{sgn}(\mathbf{S}).$$

This follows because $(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)$ is an operator mapping Ω onto itself, and so $(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S}) = \mathbb{P}_\Omega(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S})$ [1]. With this, $\mathbb{P}_\Omega\mathbf{W}^S = \lambda\mathbb{P}_\Omega(\mathbf{I} - \mathbb{P}_\Pi)\mathbb{P}_\Omega(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S}) = \lambda(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\text{sgn}(\mathbf{S}) = \lambda\text{sgn}(\mathbf{S})$.

[1] This is also clear from the Neumann series

Clearly, $\mathbf{W} = \mathbf{W}^L + \mathbf{W}^S$ is a dual certificate if

$$\begin{cases} \|\mathbf{W}^L + \mathbf{W}^S\| < 9/10, \\ \|\mathbb{P}_\Omega(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W}^L)\|_F \leq \lambda/4, \\ \|\mathbb{P}_{\Omega^\perp}(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W}^L + \mathbf{W}^S)\|_\infty < 9\lambda/10. \end{cases} \tag{3.14}$$

Next, we present the two lemmas that together prove that (3.14) holds w.h.p..

Lemma 3.3.8. *Assume $\Omega \sim Ber(\rho_s)$. Let $j_0 = 1.3\lceil \log n_{(1)} \rceil$. Under the other assumptions of Theorem 3.1.1, the matrix $\mathbf{W}^L$ obeys, with probability at least $1 - 11n_{(1)}^{-10}$,*

(a) $\|\mathbf{W}^L\| < 1/16$,

(b) $\|\mathbb{P}_\Omega(\mathbf{U}_{new}\mathbf{V}_{new}^* + \mathbf{W}^L)\|_F < \lambda/4$,

(c) $\|\mathbb{P}_{\Omega^\perp}(\mathbf{U}_{new}\mathbf{V}_{new}^* + \mathbf{W}^L)\|_\infty < 2\lambda/5$.

This is similar to [32, Lemma 2.8]. The proof is in the Appendix.

Lemma 3.3.9. *Assume $\Omega \sim Ber(\rho_s)$, and the signs of $\mathbf{S}$ are independent of Ω and i.i.d. symmetric. Under the other assumptions of Theorem 3.1.1, with probability at least $1 - 11n_{(1)}^{-10}$, the following is true*

(a) $\|\mathbb{P}_\Omega \mathbb{P}_\Pi\| \leq 1/4$ *and so $\mathbf{W}_S$ constructed earlier is well defined.*

(b) $\|\mathbf{W}^S\| < 67/80$,

(c) $\|\mathbb{P}_{\Omega^\perp}\mathbf{W}^S\|_\infty < \lambda/2$.

This is similar to [32, Lemma 2.9]. The proof is in the Appendix.

Table 3.1: Speed comparison of different algorithms. Sequence length refers to the length of sequence for training plus the length of test sequence.

DataSet	Image Size	Sequence Length	mod-PCP	PCP	ReProCS	GRASTA	RSL	DEC	GOSUS [77]
Yale Face	122×160	$48 + 24$	2.7 sec	9.8 sec	0.5 sec	50.2 sec	141.7 sec	21.3 sec	
Lake	72×90	$1420 + 80$	2.2 sec	1.7 sec	9.3 sec	338.7 sec	26.7 sec		
Fig. 3.6a	256×1	200+2400	2.7 sec	6.2 sec	12.0 sec	5.7 sec	25.4 sec		576.9 sec
Fig. 3.6b	256×1	200+8000	9.7 sec	18.9 sec	24.8 sec	12.6 sec	67.7 sec		1735.6 sec
Fig. 3.6c	256×1	200+8000	13.1 sec	18.7 sec	26.1 sec	12.7 sec	74.8 sec		1972.5 sec

3.4 Solving The Modified-PCP Program And Experiments With It

We first give below the algorithm used to solve modified-PCP. Next, we give recovery error comparisons for static simulated and real data. Finally we show some online robust PCA experiments, both on simulated and real data.

3.4.1 Algorithm for solving Modified-PCP

We give below an algorithm based on the Inexact Augmented Lagrange Multiplier (ALM) method [45] to solve the modified-PCP program, i.e. solve (1.8). This algorithm is a direct modification of the algorithm designed to solve PCP in [45] and uses the idea of [46, 47] for the sparse recovery step.

For the modified-PCP program (1.8), the Augmented Lagrangian function is:

$$\begin{aligned}\mathbb{L}(\tilde{\mathbf{L}}_{\text{new}}, \tilde{\mathbf{S}}, \mathbf{Y}, \tau) = &\|\tilde{\mathbf{L}}_{\text{new}}\|_* + \lambda\|\tilde{\mathbf{S}}\|_1 + \langle \mathbf{Y}, \mathbf{M} - \tilde{\mathbf{L}}_{\text{new}} - \tilde{\mathbf{S}} \\ &- \mathbf{G}\tilde{\mathbf{X}}^* \rangle + \frac{\tau}{2}\|\mathbf{M} - \tilde{\mathbf{L}}_{\text{new}} - \tilde{\mathbf{S}} - \mathbf{G}\tilde{\mathbf{X}}^*\|_F^2,\end{aligned}$$

Thus, with similar steps in [45], we have following algorithm. In Algorithm 3, Lines 3 solves $\tilde{\mathbf{S}}_{k+1} = \arg\min_{\tilde{\mathbf{S}}} \|\tilde{\mathbf{L}}_{\text{new},k}\|_* + \lambda\|\tilde{\mathbf{S}}\|_1 + \langle \mathbf{Y}_k, \mathbf{M} - \tilde{\mathbf{L}}_{\text{new},k} - \tilde{\mathbf{S}} - \mathbf{G}\tilde{\mathbf{X}}_k^* \rangle + \frac{\tau}{2}\|\mathbf{M} - \tilde{\mathbf{L}}_{\text{new},k} - \tilde{\mathbf{S}} - \mathbf{G}\tilde{\mathbf{X}}_k^*\|_F^2$; Line 4-6 solve $[\tilde{\mathbf{L}}_{\text{new},k+1}, \tilde{\mathbf{X}}_{k+1}] = \arg\min_{\tilde{\mathbf{L}}_{\text{new}}, \tilde{\mathbf{X}}} \|\tilde{\mathbf{L}}_{\text{new}}\|_* + \lambda\|\tilde{\mathbf{S}}_{k+1}\|_1 + \langle \mathbf{Y}_k, \mathbf{M} - \tilde{\mathbf{L}}_{\text{new}} - \tilde{\mathbf{S}}_{k+1} - \mathbf{G}\tilde{\mathbf{X}}^* \rangle + \frac{\tau}{2}\|\mathbf{M} - \tilde{\mathbf{L}}_{\text{new}} - \tilde{\mathbf{S}}_{k+1} - \mathbf{G}\tilde{\mathbf{X}}_k^*\|_F^2$. The soft-thresholding operator is

Algorithm 3 Algorithm for solving Modified-PCP (1.8)

Input: Measurement matrix $\mathbf{M} \in \mathcal{R}^{n_1 \times n_2}$, $\lambda = 1/\sqrt{\max\{n_1, n_2\}}$, $\mathbf{G}$.

1: $\mathbf{Y}_0 = \mathbf{M}/\max\{\|\mathbf{M}\|, \|\mathbf{M}\|_\infty/\lambda\}$; $\tilde{\mathbf{S}}_0 = 0$; $\tau_0 > 0$; $v > 1$; $k = 0$.
2: **while** not converged **do**
3: $\quad \tilde{\mathbf{S}}_{k+1} = \mathfrak{S}_{\lambda\tau_k^{-1}}[\mathbf{M} - \mathbf{G}\tilde{\mathbf{X}}_k - \tilde{\mathbf{L}}_{\text{new},k} + \tau_k^{-1}\mathbf{Y}_k]$.
4: $\quad (\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}, \tilde{\mathbf{V}}) = \text{svd}((I - \mathbf{G}\mathbf{G}^*)(\mathbf{M} - \tilde{\mathbf{S}}_{k+1} + \tau_k^{-1}\mathbf{Y}_k))$;
5: $\quad \tilde{\mathbf{L}}_{\text{new},k+1} = \tilde{\mathbf{U}}\mathfrak{S}_{\tau_k^{-1}}[\tilde{\mathbf{\Sigma}}]\tilde{\mathbf{V}}^T$.
6: $\quad \tilde{\mathbf{X}}_{k+1} = \mathbf{G}^*(\mathbf{M} - \tilde{\mathbf{S}}_{k+1} + \tau_k^{-1}\mathbf{Y}_k)$
7: $\quad \mathbf{Y}_{k+1} = \mathbf{Y}_k + \tau_k(\mathbf{M} - \tilde{\mathbf{S}}_{k+1} - \tilde{\mathbf{L}}_{\text{new},k+1} - \mathbf{G}\tilde{\mathbf{X}}_{k+1})$.
8: $\quad \tau_{k+1} = \min(v\tau_k, \bar{\tau})$.
9: $\quad k \leftarrow k + 1$.
10: **end while**

Output: $\hat{\mathbf{L}}_{\text{new}} = \tilde{\mathbf{L}}_{\text{new},k}, \hat{\mathbf{S}} = \tilde{\mathbf{S}}_k, \hat{L} = \mathbf{M} - \tilde{\mathbf{S}}_k$.

defined as

$$\mathfrak{S}_\epsilon[x] = \begin{cases} x - \epsilon, & \text{if } x > \epsilon; \\ x + \epsilon, & \text{if } x < -\epsilon; \\ 0, & \text{otherwise,} \end{cases} \tag{3.15}$$

Parameters are set as suggested in [45], i.e., $\tau_0 = 1.25/\|\mathbf{M}\|, v = 1.5, \bar{\tau} = 10^7\tau_0$ and iteration is stopped when $\|\mathbf{M} - \tilde{\mathbf{S}}_{k+1} - \tilde{\mathbf{L}}_{\text{new},k+1} - \mathbf{G}\tilde{\mathbf{X}}_{k+1}\|_F/\|\mathbf{M}\|_F < 10^{-7}$.

3.4.2 Simulated data

The data were generated as follows. For the sparse matrix $\mathbf{S}$, we generated a support set of size m uniformly at random and assigned values ± 1 with equal probability to entries in the support set. We generated the matrix $[\mathbf{G}\ \mathbf{U}_{\text{new}}]$ by orthonormalizing an $n_1 \times (r_0 + r_{\text{extra}} + r_{\text{new}})$ matrix with entries i.i.d. Gaussian $\mathcal{N}(0, 1/n_1)$; we set $\mathbf{U}_0$ as the first r_0 columns of this matrix, $\mathbf{G}_{\text{extra}}$ as the next r_{extra} columns and $\mathbf{U}_{\text{new}}$ as the last r_{new} columns. Then, we set $\mathbf{G} = [\mathbf{U}_0,\ \mathbf{G}_{\text{extra}}]$. This matrix has $r_G = r_0 + r_{\text{extra}}$ columns. We generated a matrix $\mathbf{Y}_1$ of size $r_G \times d$ and a matrix $\mathbf{Y}_2$ of size $(r_0 + r_{\text{new}}) \times n_2$ with entries i.i.d. $\mathcal{N}(0, 1/n_1)$. We set $\mathbf{M}_G = \mathbf{G}\mathbf{Y}_1$ as training data and $\mathbf{M} = [\mathbf{U}_0\ \mathbf{U}_{\text{new}}]\mathbf{Y}_2 + \mathbf{S}$. The matrix $\mathbf{M}_G$ is $n_1 \times d$ and the $\mathbf{M}$ is $n_1 \times n_2$. We computed $\mathbf{G}$ as the left singular vectors with nonzero singular values of $\mathbf{M}_G$ and this was used as the partial subspace

knowledge for modified-PCP. For modified-PCP, we solved (1.8) with $\mathbf{M}$ and $\mathbf{G}$ using Algorithm 3. For PCP, we solved (1.1) with $\mathbf{M}$ using the Inexact Augmented Lagrangian Multiplier algorithm from [45]. This section provides a simulation comparison of what we conclude from the theoretical results. In the theorems, both modified-PCP and PCP use the same matrix $\mathbf{M}$, but modified-PCP is given extra information (partial subspace knowledge). In the first set of simulations, we also compare with PCP when it is also given access to the initial data $\mathbf{M}_G$, i.e. we also solve PCP using $[\mathbf{M}_G\ \mathbf{M}]$. We refer to this as PCP($[\mathbf{M}_G\ \mathbf{M}]$).

Sparse recovery error is calculated as $\|\mathbf{S}-\hat{\mathbf{S}}\|_F^2/\|\mathbf{S}\|_F^2$ averaged over 100 Monte Carlo trials. For the simulated data, we also compute the smallest value of ρ_r required to satisfy the sufficient conditions – (3.1), (3.2), (3.3) for mod-PCP and (3.5), (3.6), (3.7) for PCP. We denote the respective values of ρ_r by $\rho_r([\mathbf{G}\ \mathbf{U}_{\text{new}}])$, $\rho_r(\mathbf{V}_{\text{new}})$, $\rho_r(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}})$, $\rho_r(\mathbf{U})$, $\rho_r(\mathbf{V})$ and $\rho_r(\mathbf{UV})$. Also,

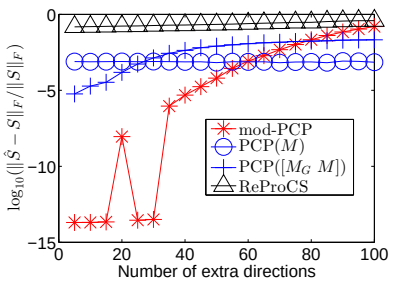
$$\rho_r(\text{mod-PCP}) = \max\{\rho_r([\mathbf{G}\ \mathbf{U}_{\text{new}}]), \rho_r(\mathbf{V}_{\text{new}}), \rho_r(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}})\}$$

and

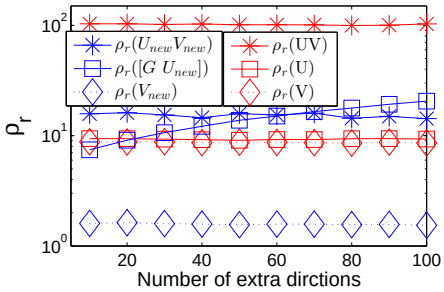
$$\rho_r(\text{PCP}) = \max\{\rho_r(\mathbf{U}), \rho_r(\mathbf{V}), \rho_r(\mathbf{UV})\}.$$

In Fig. 3.1, we show comparisons with increasing number of extra directions r_{extra}. We used $n_1 = 200$, $d = 200$, $n_2 = 120$, $m = 0.075 n_1 n_2$, $r = 20$, $r_0 = 0.9r = 18$, $r_{\text{new}} = 0.1r = 2$ and r_{extra} ranging from 0 to $n_2 - r = 100$. As we can see from Fig. 3.1a, for $r_{\text{extra}} < 60$, mod-PCP performs better than PCP with or without training data $\mathbf{M}_G$. Fig. 3.1b shows that mod-PCP allows a larger value of ρ_r (needs weaker assumptions) than PCP. Notice that the recovery error of PCP($[\mathbf{M}_G\ \mathbf{M}]$) is larger than that of PCP($\mathbf{M}$). This is because the rank of $[\mathbf{M}_G\ \mathbf{M}]$ is larger than that of $\mathbf{M}$ because of the extra directions. In the rest of the simulations, we only compare with PCP($\mathbf{M}$).

In Fig. 3.2, we show comparisons with increasing number of new directions r_{new} (or equivalently decreasing $r_0 = r - r_{\text{new}}$). We used $n_1 = 200$, $d = 200$, $n_2 = 120$,



(a) Recovery result comparison



(b) Comparing the value of ρ_r

Figure 3.1: Comparison with increasing r_{extra} ($n_1 = 200$, $d = 200$, $n_2 = 120$, $m = 0.075 n_1 n_2$, $r = 20$, $r_0 = 18$, $r_{\text{new}} = 2$). In (b), we plot the value of ρ_r needed to satisfy (3.1), (3.2), (3.3) and (3.5), (3.6), (3.7). We denote the respective values of ρ_r by $\rho_r([\mathbf{G}\ \mathbf{U}_{\text{new}}])$, $\rho_r(\mathbf{V}_{\text{new}})$, $\rho_r(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}})$, $\rho_r(\mathbf{U})$, $\rho_r(\mathbf{V})$ and $\rho_r(\mathbf{UV})$. Notice that $\rho_r(\mathbf{UV})$ is the largest, i.e. (3.7) is the hardest to satisfy. Notice also that $\rho_r(\text{mod-PCP}) = \max\{\rho_r([\mathbf{G}\ \mathbf{U}_{\text{new}}]), \rho_r(\mathbf{V}_{\text{new}}), \rho_r(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}})\}$ is significantly smaller than $\rho_r(\text{PCP}) = \max\{\rho_r(\mathbf{U}), \rho_r(\mathbf{V}), \rho_r(\mathbf{UV})\}$.

$m = 0.075 n_1 n_2$, $r = 30$, $r_{\text{extra}} = 5$ and r_{new} ranging from 1 to 20 (thus r_0 ranges from 29 to 10). As we can see, mod-PCP performs better than PCP.

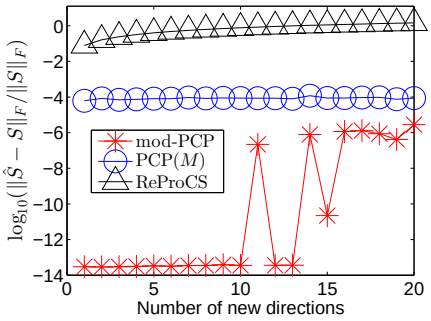
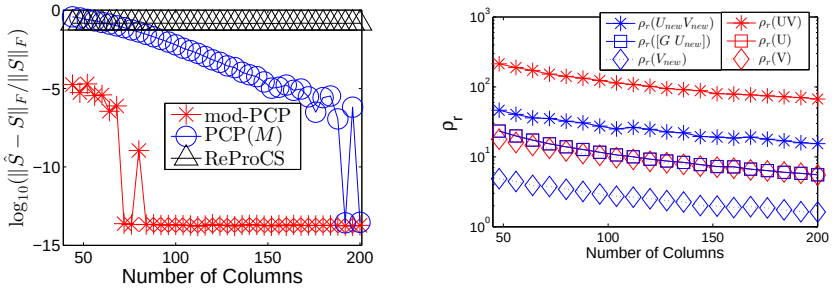


Figure 3.2: Comparison with increasing r_{new} ($n_1 = 200$, $d = 200$, $n_2 = 120$, $m = 0.075 n_1 n_2$, $r = 30$, $r_{\text{extra}} = 5$).

In Fig 3.3, we show a comparison for increasing number of columns n_2. For this figure, we used $n_1 = 200, d = 60$, $r_G = r_0 = 18$, $r_{\text{new}} = 2, m = 0.075 n_1 n_2$, and n_2 ranging from 40 to 200. Notice that this is the situation where $n_2 \leq n_1$ so that $n_{(2)} = n_2$ and $n_{(1)} = n_1$. This situation typically occurs for time series applications, where one would like to use fewer columns to still get exact/accurate recovery. We compare mod-PCP and PCP. As we can see from Fig. 3.3a, PCP needs many more columns than mod-PCP for exact recovery. Here we say exact recovery when $\|\mathbf{S} - \hat{\mathbf{S}}\|_F^2 / \|\mathbf{S}\|_F^2$ is less than 10^{-6}.

Fig. 3.3b is the corresponding comparison of ρ_r(mod-PCP) and ρ_r(PCP) for this dataset and the conclusion is similar.



(a) Recovery result comparison (b) Comparing the value of ρ_r

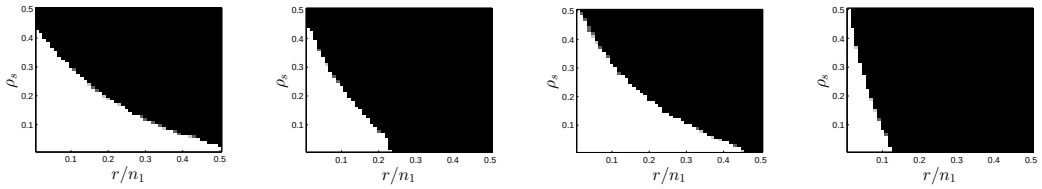
Figure 3.3: Comparison with increasing n_2 ($n_1 = 200, d = 60$, $r_G = r_0 = 18$, $r_{\text{new}} = 2, m = 0.075 n_1 n_2$).

As pointed out by an anonymous reviewer, notice that there are jumps in Fig 3.1a, 3.2, 3.3a. The reason for these is as follows. The guarantees for modified-PCP (and also for PCP) hold only with high probability. So there is always a small chance that modified-PCP fails. In these figures we plot the Monte Carlo based normalized mean squared error (NMSE). The averaging is done over 100 realizations. This number is small enough that even if modified-PCP does not give exact recovery and has larger recovery error for one out of all the realizations, it increases the NMSE by a significant amount. This is what happened in Fig 3.1a for $r_{\text{extra}} = 20$ or in Fig 3.2 for $r_{\text{new}} = 11$ or 14 and in Fig. 3.3a for n_2 just large enough for exact recovery. Notice that, in all of these figures, the "bad" case happens just before the phase transition from near-zero error to large error. These are precisely the cases for which the probability of exact recovery is smaller and hence there is a higher chance of modified-PCP failing.

Also, because of the above reason, the phase transition plots given next are more useful in evaluating algorithms such as modified-PCP or PCP that work with high probability. The phase transition plots will also have a jump for the above case, except that the jump will be from probability of failure = zero to probability of failure = 0.01 (in case of one "bad" case out of 100) and back to zero. The jump from 0 to 0.01 and back to 0 is so small that it is not even visible.

We generated phase transition plots similar to those for PCP in [32]. We used the approach outlined in [32] to generate $\mathbf{L}, \mathbf{S}$ and $\mathbf{M}$ i.e. we let $n_1 = n_2 = 400$ and $\mathbf{L} = \mathbf{X}\mathbf{Y}^*$, where $\mathbf{X}$ and $\mathbf{Y}$ are independent $n_1 \times r$ i.i.d. $\mathcal{N}(0, 1/n_1)$ matrix and independent $n_2 \times r$ i.i.d. $(0, 1/n_2)$ matrices respectively. The support Ω of $\mathbf{S}$ is of size m and uniformly distributed and for $(i, j) \in \Omega$, $\mathbb{P}(\mathbf{S}_{ij} = 1) = \mathbb{P}(\mathbf{S}_{ij} = -1) = 1/2$. For mod-PCP, we used $r_{\text{new}} = \lfloor 0.15r \rfloor$, $r_{\text{extra}} = \lfloor 0.15r \rfloor$ and we generated $\mathbf{G}$ as follows. We let $\mathbf{U}_0$ be the first $(r - r_{\text{new}})$ columns of the orthonormalized $\mathbf{X}$, and we generated $\mathbf{G}_{\text{extra}}$ as the first r_{extra} columns of the orthonomalized $(\mathbf{I} - \mathbf{U}\mathbf{U}^*)\mathbf{X}_1$. Here $\mathbf{U}$ is the matrix of left singular vectors of $\mathbf{L}$ and $\mathbf{X}_1$ is a $n_1 \times 2r_{\text{extra}}$ i.i.d. $\mathcal{N}(0, 1/n_1)$ matrix. We set $\mathbf{G} = [\mathbf{U}_0, \ \mathbf{G}_{\text{extra}}]$.

To show the advantages of mod-PCP with less columns, we also did a comparison with the same parameters above but with $n_1 = 400, n_2 = 200$. Fig. 3.4 shows the fraction of correct recoveries across 10 trials (as was also done in [32]). Recoveries are considered correct if $\|\hat{L} - L\|_F / \|L\|_F \leq 10^{-3}$. As we can see from Fig. 3.4, mod-PCP is always better than PCP since r_{new} and r_{extra} are small. But the difference is much more significant when $n_2 = n_1/2$ than when $n_2 = n_1$.



(a) mod-PCP, $n_2 = 400$ (b) PCP, $n_2 = 400$ (c) mod-PCP, $n_2 = 200$ (d) PCP, $n_2 = 200$

Figure 3.4: Phase transition plots with $r_{\text{new}} = \lfloor 0.15r \rfloor$, $r_{\text{extra}} = \lfloor 0.15r \rfloor$, $n_1 = 400$

3.4.3 Real data (face reconstruction application)

As stated in [32], robust PCA is useful in face recognition to remove sparse outliers, like cast shadows, specularities or eyeglasses, from a sequence of images of the same face. As explained there, without outliers, face images arranged as columns of a matrix are known to form an approximately low-rank matrix. Here we use the images from the

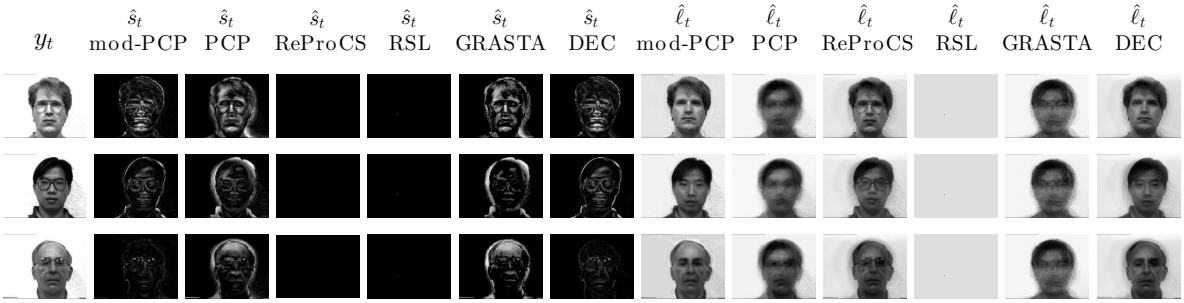
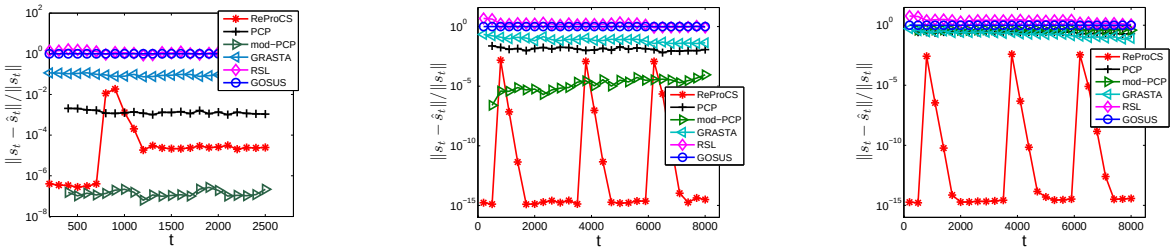


Figure 3.5: Yale Face Image result comparison



(a) Uniformly distributed $\mathbf{s}_t$ (b) Correlated $\mathbf{s}_t$ with $|\mathbf{s}_t|$ small (c) Correlated $\mathbf{s}_t$ with $|\mathbf{s}_t|$ large

Figure 3.6: NRMSE of sparse part comparison with online model ($n = 256$, $J = 3$, $r_0 = 40$, $t_0 = 200$, $c_{j,\text{new}} = 4$, $c_{j,\text{old}} = 4$, $j = 1, 2, 3$)

Yale Face Database [78] that is also used in [32]. Outlier-free training data consisting of face images taken under a few illumination conditions, but all without eyeglasses, is used to obtain a partial subspace estimate. The test data consists of face images under different lighting conditions and with eyeglasses or other outliers. For test data, the goal is to reconstruct a clear face image with the cast shadows, eyeglasses or other outliers removed. Thus, the clear face image should be a column of the estimated low-rank matrix while the cast shadows or eyeglasses should be a column of the sparse matrix.

Each image is of size 243×320, which we reduce to 122×160. All images are re-arranged as long vectors and a mean image is subtracted from each of them. The mean image is computed as the empirical mean of all images in the training data. For the training data, $\mathbf{M}_G$, we use images of subjects with no glasses, which is 12 subjects out of 15 subjects. We keep four face images per subject – taken with center-light, right-light, left-light, and normal-light – for each of these 12 subjects. Thus the training data matrix

$\mathbf{M}_G$ is 19520×48. We compute $\mathbf{G}$ by keeping its left singular vectors corresponding to 99% energy. This results in $r_G = 38$. We use another two face images per subject for each of the twelve subjects, some with glasses and some without, as the test data, i.e. the measurement matrix $\mathbf{M}$. Thus $\mathbf{M}$ is 19520×24.

In the experiments, we compare modified-PCP with PCP [32] and ReProCS [21, 64] and also with some of the other algorithms compared in [64]: robust subspace learning (RSL) [79], which is a batch robust PCA algorithm that was compared against in [32], and GRASTA [80], which is a very recent online robust PCA algorithm. We also compare against Dense Error Correction (DEC) [81, 82] since this first addressed this application using ℓ_1 minimization. To implement Dense Error Correction (DEC) [81, 82], we normalize each column of $\mathbf{M}_G$ to get the dictionary $(\mathbf{D})_{n_1 \times 48}$, and we solve

$$(\hat{\mathbf{x}}_i, \hat{\mathbf{s}}_i) = \arg\min_{\tilde{\mathbf{x}}, \tilde{\mathbf{s}}} \|\tilde{\mathbf{x}}\|_1 + \|\tilde{\mathbf{s}}\|_1 \text{ subject to } \mathbf{M}_i = \mathbf{D}\tilde{\mathbf{x}} + \tilde{\mathbf{s}}$$

using YALL-1. Here $\mathbf{M}_i$ is the ith column of $\mathbf{M}$. The solution gives us $\hat{\mathbf{s}}_i$ and $\hat{\boldsymbol{\ell}}_i = \mathbf{D}\hat{\mathbf{x}}_i$.

For PCP and RSL, we use the test dataset only, i.e., $\mathbf{M}$, which is a 19520×24 matrix, as the measurement matrix. DEC, ReProCS and GRASTA are provided the same partial knowledge that mod-PCP gets. Fig. 3.5 shows 3 cases where mod-PCP successfully removes the glasses into $(\hat{S})_i$ and gives the clearest estimate of the person's face without glasses as $(\hat{L})_i$. In the total 24 test frames, both mod-PCP and DEC remove the glasses (for those having glasses) or remove nothing (for those not having glasses) correctly in 14 of them, but the result of DEC has extra shadows in the face estimate. The other algorithms succeed for none of the 24 frames. Both ReProCS and GRASTA assume that the initial subspace estimate is accurate and "slow subspace change" holds, neither of which happen here and this is the reason that neither of them work. RSL does not converge for this data set because the available number of frames is too small. The time taken by each algorithm is shown in Table 3.1.

3.4.4 Online robust PCA: simulated data comparisons

For simulation comparisons for online robust PCA, we generated data as explained in [83]. The data was generated using the model given in Section 3.2, with $n = 256$, $J = 3$, $r_0 = 40$, $t_0 = 200$ and $c_{j,\text{new}} = 4$, $c_{j,\text{old}} = 4$, for each $j = 1, 2, 3$. The coefficients, $1.5_{t,*} = \mathbf{P}_{j-1}^{*}\boldsymbol{\ell}_t$ were i.i.d. uniformly distributed in the interval $[-\gamma, \gamma]$; the coefficients along the new directions, $1.5_{t,\text{new}} := \mathbf{P}_{j,\text{new}}^{*}\boldsymbol{\ell}_t$ generated i.i.d. uniformly distributed in the interval $[-\gamma_{\text{new}}, \gamma_{\text{new}}]$ (with a $\gamma_{\text{new}} \leq \gamma$) for the first 1700 columns after the subspace change and i.i.d. uniformly distributed in the interval $[-\gamma, \gamma]$ after that. We vary the value of γ_{new}; small values mean that "slow subspace change" required by ReProCS holds. The sparse matrix $\mathbf{S}$ was generated in two different ways to simulate uncorrelated and correlated support change. For partial knowledge, $\mathbf{G}$, we first did SVD decomposition on $[\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \cdots, \boldsymbol{\ell}_{t_0}]$ and kept the directions corresponding to singular values larger than $\mathbf{E}(z^2)/9$, where $z \sim \text{Unif}[-\gamma_{\text{new}}, \gamma_{\text{new}}]$. We solved PCP and modified-PCP every 200 frames by using the observations for the last 200 frames as the matrix $\mathbf{M}$. The ReProCS algorithm of [44, 83] was implemented with $\alpha = 100$. The averaged sparse part errors with three different sets of parameters over 20 Monte Carlo simulations are displayed in Fig. 3.6a, Fig. 3.6b, and Fig. 3.6c, and the corresponding averaged time spent for each algorithm is shown in Table 3.1. For all three figures, we used $t_1 = t_0 + 6\alpha + 1$, $t_2 = t_0 + 12\alpha + 1$ and $t_3 = t_0 + 18\alpha + 1$ and $\gamma = 5$.

In the first case, Fig. 3.6a, we used $\gamma_{\text{new}} = \gamma$ and so "slow subspace change" does not hold. For the sparse vectors $\mathbf{s}_t$, each index is chosen to be in support with probability 0.0781. The nonzero entries are uniformly distributed between $[20, 60]$. Since "slow subspace change" does not hold, ReProCS does not work well. Since the support is generated independently over time, this is a good case for both PCP and mod-PCP. Mod-PCP has the smallest sparse recovery error. In the second case, Fig. 3.6b, we used $\gamma_{\text{new}} = 1$ and thus "slow subspace change" holds. For sparse vectors, $\mathbf{s}_t$, the support is generated in a correlated fashion. We used support size $s = 5$ for each $\mathbf{s}_t$; the support

remained constant for 25 columns and then moved down by $s = 5$ indices. Once it reached n, it rolled back over to index one. Because of the correlated support change, PCP does not work. In this case, both mod-PCP and ReProCS work but PCP does not. In the third case, Fig. 3.6c, the parameters are the same as in the second case, except that the support size is $s = 10$ in each column and it moves down by $s/2 = 5$ indices every 25 columns. In this case, the sparse vectors are much more correlated over time, resulting in sparse matrix $\mathbf{S}$ that is even more low rank, thus neither mod-PCP nor PCP work for this data. In this case, only ReProCS works. Thus from simulations, modified-PCP is able to handle correlated support change better than PCP but worse than ReProCS. Modified-PCP also works when slow subspace change does not hold; this is a situation where ReProCS fails. Of course, modified-PCP, GRASTA and ReProCS are provided the same partial subspace knowledge $\mathbf{G}$ while PCP and RSL do not get this information.

In Fig. 4.4, as noted by an anonymous reviewer, one can see jumps in the ReProCS error at the time instants at which there is a subspace change. This is due to how the ReProCS algorithm works - it detects subspace change within a short delay of the change and then slowly improves its estimate of the new subspace. For a detailed explanation, please see [44].

3.4.5 Online robust PCA: comparisons for video layering

The lake sequence is similar to the one used in [64]. The background consists of a video of moving lake waters. The foreground is a simulated moving rectangular object. The sequence is of size $72 \times 90 \times 1500$, and we used the first 1420 frames as training data (after subtracting the empirical mean of the training images), i.e. $\mathbf{M}_G$. The rest 80 frames (after subtracting the same mean image) served as the background $\mathbf{L}$ for the test data. For the first frame of test data, we generated a rectangular foreground support with upper left vertex $(1, j_0)$ and lower right vertex $(i_1, 25 + j_0)$, where $j_0 \sim \text{Unif}[1, 30]$

and $i_1 \sim \text{Unif}[7, 16]$, and the foreground moves to the right 1 column each time. Then we stacked each image as a long vector $\boldsymbol{\ell}_t$ of size 6480×1. For each index i belonging to the support set of foreground $\mathbf{s}_t$, we assign $(\mathbf{s}_t)_i = 185 - (\boldsymbol{\ell}_t)_i$. We set $\mathbf{M} = \mathbf{L} + \mathbf{S}$. For mod-PCP, ReProCS and GRASTA, we used the approach used in [64] to estimate the initial background subspace (partial knowledge): do SVD on $\mathbf{M}_G$ and keep the left singular vectors corresponding to 95% energy as the matrix $\mathbf{G}$. The averaged normalized mean squared error (NMSE) of the sparse part over 50 Monte Carlo realizations is shown in Fig. 3.7a. The averaged time spent for each algorithm is shown in Table 3.1. As can be seen, in this case, both mod-PCP and ReProCS perform almost equally well, with ReProCS being slightly better.

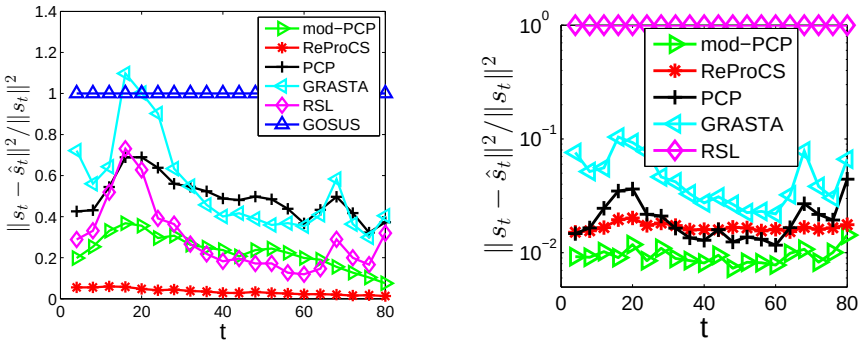
To show the advantage of mod-PCP, we did another experiment. In this case, the support of the foreground was uniformly generated with $m = \lfloor 0.2 n_1 n_2 \rfloor$ nonzero pixels. Everything else was the same as in the above experiment. Notice that in this case the support size of the foreground is 20% while in the previous correlated motion case, it was much smaller, only 3% on average. The corresponding foreground NMSE comparison is shown in Fig. 3.7b. Figs. 3.7a and 3.7b again show that when there are many small and fast-moving foreground objects, modified-PCP is the best algorithm, whereas when there is one (or a few) slow-moving foreground object(s) ReProCS is slightly better than modified-PCP.

On our webpage, we have also shown comparisons on a real video sequence consisting of multiple and small-sized moving persons. This is the airport escalator sequence that was originally downloaded from http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html, but is now unavailable at that website. We provide the video and our experimental results comparing all the methods on our webpage at http://www.ece.iastate.edu/~jzhan/data/. In this video, the background consists of a moving escalator and the foreground is moving passengers. We used the first 100 frames of this sequence as training data (after subtracting the empirical mean of the training images),

i.e., $\mathbf{M}_G$. The same training data was provided to ReProCS and GRASTA as well. This is a sequence for which modified-PCP is better than or as good as all other algorithms. It is significantly better than ReProCS.

Next we compute the value of ρ_r for the lake video sequence. We calculated prior knowledge $\mathbf{G}$ as explained above. We calculated the singular vectors $\mathbf{U}, \mathbf{V}$ by doing SVD decomposition on $\mathbf{L}$ and keeping all the directions with corresponding singular values larger than 10^{-10} (we choose 10^{-10} because it is the precision that MATLAB can achieve for SVD decomposition); calculate $\mathbf{U}_{\text{new}}, \mathbf{V}_{\text{new}}$ by doing SVD decomposition of $(I - \mathbf{G}\mathbf{G}^*)\mathbf{L}$ and keeping all the directions with singular values larger than 10^{-10}. With this, we get $\rho_r(\text{PCP}) = 1.8584 \times 10^4$ and $\rho_r(\text{mod-PCP}) = 1.7785 \times 10^4$.

We also calculate ρ_r for fountain02 sequence (available on http://changedetection.net/). The image size is 288×432, and we resize it to 96×144. For the first 600 background images we form a low rank matrix $[\mathbf{M}_G\ \mathbf{L}]$ by stacking each image as a column (the first 300 columns belong to $\mathbf{M}_G$ and the rest belong to $\mathbf{L}$). With the same steps for lake sequence, we get $\rho_r(\text{PCP})$ is 4.311×10^4 and $\rho_r(\text{mod-PCP})$ is 1.7866×10^4.



(a) Correlated $\mathbf{s}_t$ (b) Uniformly distributed $\mathbf{s}_t$

Figure 3.7: Lake sequence NMSE comparison. (a) shows comparisons for one slow-moving foreground object; (b) shows comparisons for a large number of small-sized fast-moving foreground objects (total foreground support size is much larger for (b)).

CHAPTER 4. RECURSIVE (ONLINE) SPARSE RECOVERY IN LARGE AND STRUCTURED NOISE AND BOUNDED NOISE

4.1 Introduction

4.1.1 Related work

Solutions for online RPCA have been analyzed in recent works [44], [65], [83, 84, 85]. The work of [44] introduced the Recursive Projected Compressive Sensing (ReProCS) algorithmic framework and obtained a partial result for it. Another approach for online RPCA (defined differently from above) and a partial result for it were provided in [65]. We use the term *partial result* to refer to a performance guarantee that depends on intermediate algorithm estimates satisfying certain properties. We will see examples of this in Sec. 4.2.7 when we discuss the above results. In very recent work [83, 84, 85], a *correctness result* for ReProCS was obtained. The term *correctness result* refers to a complete performance guarantee, i.e., a guarantee that only puts assumptions on the input data (here $\mathbf{m}_t$) and/or on the algorithm initialization, but not on intermediate algorithm estimates.

Other somewhat related work includes [66] (online PCA with contaminated data that is not modeled as being sparse) and [86] (modified-PCP, a piecewise batch method). All the above results are discussed Sec. 4.2.7. Some other works, such as [80](GRASTA), [87] (adaptive-iSVD), [88] (incremental Robust Subspace Learning) or [77] (GOSUS),

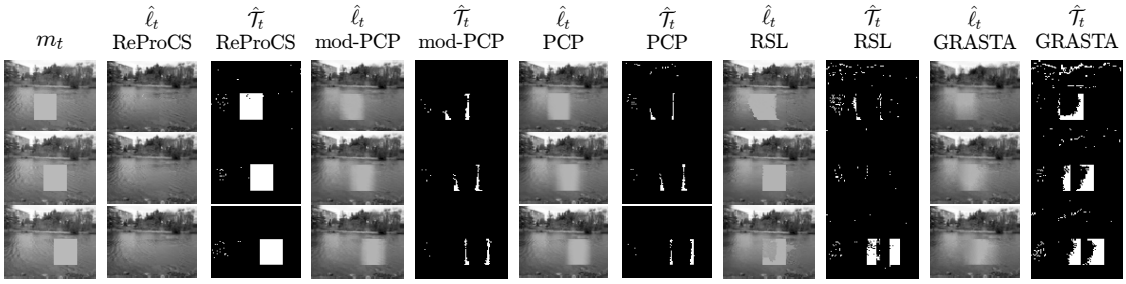


Figure 4.1: The first column shows the video of a moving rectangular object against moving lake waters' background. The object and its motion are simulated while the background is real. In the next two columns, we show the recovered background ($\hat{\boldsymbol{\ell}}_t$) and the recovered foreground support ($\hat{\mathcal{T}}_t$) using Automatic ReProCS-cPCA (labeled ReProCS in the figure). The algorithm parameters are set differently for the experiments (see Sec. 4.8) than in our theoretical result. Notice that the foreground support is recovered mostly correctly with only a few extra pixels and the background appears correct too (does not contain the moving block). The quantitative comparison is shown later in Fig. 4.4. The next few columns show background and foreground-support recovery using some of the existing methods discussed in Sec. 4.1.1.

[89, 90], [91], [92], [93] only provide an online RPCA algorithm without guarantees. We do not discuss these here. As demonstrated by the experimental comparisons shown in [64] and in [86, Fig 6], when the outlier support is large and changes in a correlated fashion over time, ReProCS-based algorithms significantly outperform most of these, besides also outperforming batch methods such as PCP and robust subspace learning (RSL) [32, 79]. This is also evident from Fig. 4.1 and Fig. 4.4.

4.1.2 Contributions

In this work we develop and study an algorithm based on the ReProCS idea introduced and studied in [44, 83, 84, 85]. We call it Automatic ReProCS with cluster PCA (ReProCS-cPCA). This is an improved ReProCS algorithm compared to the ones studied in previous work. (1) It is able to automatically detect subspace changes within a short delay; is able to correctly estimate the number of directions added or deleted; and is also able to correctly estimate the clusters of eigenvalues along the existing directions. This is important because it is impractical to assume that a subspace change time or the

exact number of added or removed directions is known. Additionally, these estimates themselves are relevant for applications such as understanding dynamic social networks' structural changes in the presence of outliers. While many heuristics exist to detect sudden subspace changes, we provide an approach for correctly detecting slow subspace changes within a short delay. (2) Moreover it is able to accurately estimate both the newly added subspace as well as the newly deleted subspace. The latter is done by re-estimating the current subspace using an approach called cluster PCA (cPCA). The basic cPCA idea was introduced in [44]. The current work uses that idea to develop an automatic algorithm. The cPCA step ensures that the estimated subspace dimension does not keep increasing with time. (3) The current algorithm also returns more accurate offline estimates. The algorithms studied in [44, 84] could not do (1) and (3). The algorithms studied in [83, 84, 85] did not do (2) and (3).

The *main* contribution of this work is a correctness result (complete performance guarantee) for the proposed algorithm under relatively mild assumptions on $\boldsymbol{\ell}_t$, $\mathbf{x}_t$, and $\mathbf{w}_t$. To our knowledge, this and [83, 84, 85] are the first correctness results for online RPCA. The result obtained here removes two key limitations of [83, 84, 85]. (1) First, we obtain a result for the case where the $\boldsymbol{\ell}_t$'s can be correlated over time (follow an autoregressive (AR) model) where as the result of [83, 84, 85] needed mutual independence of the $\boldsymbol{\ell}_t$'s. This models mostly static backgrounds in which changes are only due to independent variations at each time, e.g., light flickers. However, a large class of background image sequences change due to factors that are correlated over time, e.g., moving waters. This can be better modeled using an AR model. (2) Second, with one extra assumption – that the eigenvalues of the covariance matrix of $\boldsymbol{\ell}_t$ are clustered for a period of time after the previous subspace change has stabilized – we are able to remove another significant limitation of [83, 84, 85]. That result needed the rank of $\mathbf{L}$ to grow as $\mathcal{O}(\log n)$ while our result allows it to grow as $\mathcal{O}(n)$. Batch methods such as PCP allow the rank to grow almost linearly with n. The clustered eigenvalues assumption is valid

for data that has variability at different scales - large scale variations would result in the first (largest eigenvalues') cluster and the smaller scale variations would form the later clusters.

Because we use extra assumptions – accurate initial subspace knowledge, slow subspace change, and clustered eigenvalues – we are able to remove an important limitation of batch methods [32, 33, 94]. As we explain in Sec. 4.2.7, our result requires an order-wise looser bound on the number of time instants for which a particular index i can be outlier-corrupted compared to these results. In other words, it allows significantly more correlated changes of the outlier support over time. This is important in practice, e.g., in video, foreground objects do not randomly jump around; in social networks, once an anomalous pattern starts to occur, it remains on many of the same edges for a while. The clustered eigenvalues assumption is discussed above. Accurate initial subspace knowledge and slow subspace change were discussed earlier (just above Sec. 1.1.2).

The novelty in the proof techniques used in this work is summarized in Sec. 4.4.1. The proof relies on the $\sin\theta$ theorem [95] (that bounds the effect of a perturbation on a Hermitian matrix's top eigenvectors) and the matrix Azuma inequality [96].

4.1.3 Notation

We use the interval notation $[a, b]$ to mean all of the integers between a and b, inclusive, and similarly for $[a, b)$ etc. For a set $\mathcal{T}$, $|\mathcal{T}|$ denotes its cardinality and $\bar{\mathcal{T}}$ denotes its complement set. We use $\emptyset$ to denote the empty set.

We use $'$ to denote a vector or matrix transpose. The l_p-norm of a vector and the induced l_p-norm of a matrix are denoted by $\|\cdot\|_p$. For a vector $\boldsymbol{x}$ and set $\mathcal{T}$, $\boldsymbol{x}_{\mathcal{T}}$ is a smaller vector containing the entries of $\boldsymbol{x}$ indexed by entries in $\mathcal{T}$. We use $\mathcal{I}$ to denote the identity matrix. Define $\mathbf{I}_{\mathcal{T}}$ to be an $n \times |\mathcal{T}|$ matrix of those columns of the identity matrix indexed by entries in $\mathcal{T}$. For a matrix $\boldsymbol{A}$, define $\boldsymbol{A}_{\mathcal{T}} := \boldsymbol{A}\boldsymbol{I}_{\mathcal{T}}$. For matrices $\boldsymbol{P}$, $\boldsymbol{Q}$ where the columns of $\boldsymbol{Q}$ are a subset of the columns of $\boldsymbol{P}$, $\boldsymbol{P} \setminus \boldsymbol{Q}$ refers to the

matrix of columns in $\boldsymbol{P}$ and not in $\boldsymbol{Q}$. For a matrix $\boldsymbol{H}$, $\boldsymbol{H} \overset{\text{EVD}}{=} \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}'$ denotes its reduced eigenvalue decomposition. For Hermitian matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the notation $\boldsymbol{A} \preceq \boldsymbol{B}$ means that $\boldsymbol{B} - \boldsymbol{A}$ is positive semi-definite.

For a matrix $\boldsymbol{A}$, the restricted isometry constant (RIC) $\delta_s(\boldsymbol{A})$ is the smallest real number δ_s such that

$$(1 - \delta_s)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1 + \delta_s)\|\boldsymbol{x}\|_2^2$$

for all s-sparse vectors $\boldsymbol{x}$ [12]. A vector $\boldsymbol{x}$ is s-sparse if it has s or fewer non-zero entries.

We refer to a matrix with orthonormal columns as a *basis matrix*. Thus, for a basis matrix $\boldsymbol{P}$, $\boldsymbol{P}'\boldsymbol{P} = \boldsymbol{I}$. For basis matrices $\hat{\mathbf{P}}$ and $\mathbf{P}$, $\text{dif}(\hat{\mathbf{P}}, \mathbf{P}) := \|(\mathbf{I} - \hat{\mathbf{P}}\hat{\mathbf{P}}')\mathbf{P}\|_2$ quantifies error between their range spaces.

4.1.4 Paper organization

This paper is organized as follows. We discuss the data models and the main results for the proposed algorithm in Sec. 4.2. The Automatic ReProCS-cPCA algorithm is developed in Sec. 4.3. The stepwise algorithm is summarized in Algorithm 4. The proof outline of our main result is given in Sec. 4.4. This section also helps understand the algorithm better and explains the novelty in the proof techniques. The lemmas for proving the main result, the proof of the main result and the proofs of the main lemmas are given in Sec. 4.5. The key lemmas needed to prove the main lemmas are proved in Sec. 4.6 (lemmas for analyzing the projection-PCA based subspace addition step) and in Sec. 4.7 (lemmas for analyzing the cluster PCA based subspace deletion step). These are the long sections that contain the new proofs that rely on the matrix Azuma inequality [96]. This is needed because the $\boldsymbol{\ell}_t$'s are now correlated over time. Simulation experiments comparing the proposed algorithm to some existing batch and online RPCA algorithms are described in Sec. 4.8. Conclusions are given in Sec. 4.9.

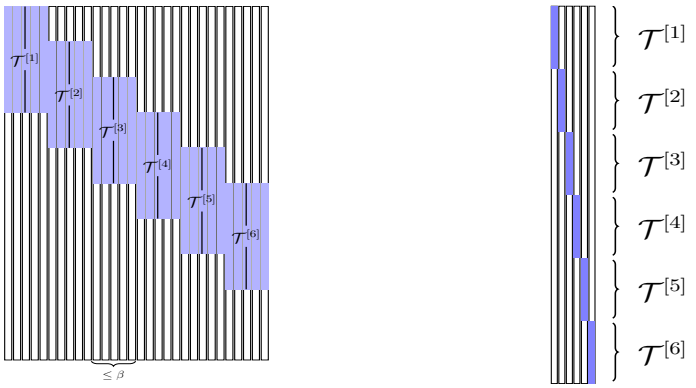
4.2 Data Models And Main Results

In this section, we give the data models and correctness results for our proposed algorithm, Automatic ReProCS-cPCA, and for its simplification, Automatic ReProCS. The algorithm itself is developed in Sec 4.3 and the complete stepwise algorithm is summarized in Algorithm 4. We give below the model on the outlier support sets $\mathcal{T}_t$, the model on $\boldsymbol{\ell}_t$, and the denseness assumption. Using these, we state the result for Automatic ReProCS in Sec. 4.2.5. In Sec. 4.2.6, we state the clustering assumption and give the correctness result for Automatic ReProCS-cPCA. The results are discussed in Sec. 4.2.7.

4.2.1 Model on the outlier support set, $\mathcal{T}_t$

We give here one simple and practically relevant special case of the most general assumptions (Model 10) on the outlier support sets $\mathcal{T}_t$. It requires that the $\mathcal{T}_t$'s have *some* changes over time and have size less than s. An example of this is a video application consisting of a foreground with a 1D object of length s or less that remains static for at most β frames at a time. When it moves, it moves *downwards (or upwards, but always in one direction)* by at least $\frac{s}{\rho}$ pixels, and at most $\frac{s}{\rho_2}$ pixels. Once it reaches the bottom of the scene, it disappears. The maximum motion is such that, if the object were to move at each frame, it still does not go from the top to the bottom of the scene in a time interval of length α. This is ensured if $\frac{s}{\rho_2}\alpha \leq n$. Anytime after it has disappeared another object could appear. A visual depiction of this model is shown in Fig. 4.2. We have used this "one object moving in one direction" example to only explain the idea in a simple fashion. Instead, one could also have multiple moving objects and arbitrary motions, as long as the union of their supports follows the assumptions of Model 4 below or those given later in Model 10. These models were introduced in [85].

Model 4 (model on $\mathcal{T}_t$)**.** *Let t^k, with $t^k < t^{k+1}$, denote the times at which $\mathcal{T}_t$ changes*



(a) $\rho = 3$ and $\beta = 5$ case (b) $\rho = 1$ and $\beta = 1$ case

Figure 4.2: Examples of Model 4. (a) shows a 1D object of length s that moves by at least $s/3$ pixels at least once every 5 frames (i.e., $\rho = 3$ and $\beta = 5$). (b) shows the object moving by s pixels at every frame (i.e., $\rho = 1$ and $\beta = 1$). (b) is an example of the best case for our result - the case with the smallest ρ, β ($\mathcal{T}_t$'s mutually disjoint)

and let $\mathcal{T}^{[k]}$ *denote the distinct sets. For an integer* α*,*

1. *assume that* $\mathcal{T}_t = \mathcal{T}^{[k]}$ *for all times* $t \in [t^k, t^{k+1})$ *with* $(t^{k+1} - t^k) < \beta$ *and* $|\mathcal{T}^{[k]}| \leq s$*;*

2. *let* ρ *be a positive integer so that for any* k*,* $\mathcal{T}^{[k]} \cap \mathcal{T}^{[k+\rho]} = \emptyset$*; assume that* $\rho^2\beta \leq 0.0001\alpha$*;*

3. *for any* k*,* $\sum_{i=k+1}^{k+\alpha} \left|\mathcal{T}^{[i]} \setminus \mathcal{T}^{[i+1]}\right| \leq n$ *and for any* $k < i \leq k + \alpha$*,* $(\mathcal{T}^{[k]} \setminus \mathcal{T}^{[k+1]}) \cap (\mathcal{T}^{[i]} \setminus \mathcal{T}^{[i+1]}) = \emptyset$ *(one way to ensure the first condition is to require that for all* i*,* $|\mathcal{T}^{[i]} \setminus \mathcal{T}^{[i+1]}| \leq \frac{s}{\rho_2}$ *with* $\frac{s}{\rho_2}\alpha \leq n$*).*

In this model, k *takes values* $1, 2, \ldots$*; the largest value it can take is* $t_{\max}$*. We set* α *in the Theorem.*

4.2.2 Model on ℓ_t

A common model for data that lies in a low-dimensional subspace is to assume that, at all times, it is independent and identically distributed (iid) Gaussian with zero mean and a fixed *low-rank* covariance matrix $\mathbf{\Sigma}$. However this can be restrictive since, in many applications, data statistics change with time, albeit slowly. To model this perfectly,

one would need to assume that $\boldsymbol{\ell}_t$ is zero mean with covariance matrix $\boldsymbol{\Sigma}_t$ at time t. If $\boldsymbol{\Sigma}_t \overset{\text{EVD}}{=} \mathbf{P}_t \boldsymbol{\Lambda}_t \mathbf{P}_t'$, this means that both $\mathbf{P}_t$ and $\boldsymbol{\Lambda}_t$ can change at each time t, though slowly. This is the most general model but it has an identifiability problem if the goal is to estimate the subspace from which $\boldsymbol{\ell}_t$ was generated, range($\mathbf{P}_t$). The subspace cannot be estimated with one data point. If it is r-dimensional, it needs at least r data points. So, if $\mathbf{P}_t$ changes at each time, it is not clear how one can estimate all the subspaces. To resolve this issue, a general enough but tractable option is to assume that $\mathbf{P}_t$ is piecewise constant with time and $\boldsymbol{\Lambda}_t$ can change at each time. To ensure that $\boldsymbol{\Sigma}_t$ changes "slowly", we assume that, when $\mathbf{P}_t$ changes, the eigenvalues along the newly added directions are small initially for the first d frames, and after that they can increase gradually or suddenly to any large value. One precise model for this is specified next.

The model below assumes boundedness of $\boldsymbol{\ell}_t$. This is more practically valid than the usual Gaussian assumption since most sensor data or noise is bounded. We also replace independence of $\boldsymbol{\ell}_t$'s by an AR model with independent perturbations $\boldsymbol{\nu}_t$ and we place the above assumptions on $\boldsymbol{\nu}_t$. As explained earlier, this is a more practical model and includes independence as a special case.

Model 5 (Model on $\boldsymbol{\ell}_t$)**.** *Assume the following.*

1. *Let* $\boldsymbol{\ell}_0 = \mathbf{0}$ *and for* $t = 1, 2, \ldots t_{\max}$*, assume that*

$$\boldsymbol{\ell}_t = b\boldsymbol{\ell}_{t-1} + \boldsymbol{\nu}_t$$

   for a $b < 1$*. Assume that the* $\boldsymbol{\nu}_t$ *are zero mean, mutually independent and bounded random vectors with covariance matrix*

$$\text{Cov}(\boldsymbol{\nu}_t) = \boldsymbol{\Sigma}_t \overset{\text{EVD}}{=} \boldsymbol{P}_t \boldsymbol{\Lambda}_t \boldsymbol{P}_t'.$$

2. *Let* $t_1, t_2, \ldots t_J$ *denote the subspace change times. The basis matrices* $\mathbf{P}_t$ *change as*

$$\boldsymbol{P}_t = \begin{cases} [(\boldsymbol{P}_{t-1}\boldsymbol{R}_t \setminus \boldsymbol{P}_{t,old})\ \boldsymbol{P}_{t,new}] & \text{if } t = t_1, t_2, \ldots\ t_J \\ \boldsymbol{P}_{t-1} & \text{otherwise.} \end{cases}$$

where $\boldsymbol{R}_t$ *is a rotation matrix,* $\mathbf{P}_{t_j,new}$ *and* $\mathbf{P}_{t_j,old}$ *are basis matrices of size* $n \times r_{j,new}$ *and* $n \times r_{j,old}$ *respectively,* $\boldsymbol{P}_{t_j,old}$ *contains a subset of columns of* $\boldsymbol{P}_{t_j-1}\boldsymbol{R}_t$*, and* $\mathbf{P}_{t_j,new}{}'\mathbf{P}_{t_j-1} = \mathbf{0}$ *(new directions are orthogonal to previous subspace).*

3. *Define*

$$\lambda^- := \lambda_{\min}\left(\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\boldsymbol{\Lambda}_t\right) \text{ and } \lambda^+ := \lambda_{\max}\left(\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\boldsymbol{\Lambda}_t\right).$$

The eigenvalues' matrices $\boldsymbol{\Lambda}_t$ *are such that (i)* $\lambda_{\max}(\boldsymbol{\Lambda}_t) \le \lambda^+$ *and (ii) for a* $d < t_{j+1} - t_j$*,*

$$\begin{aligned} &0 < \lambda^- \le \lambda^-_{new} \le \lambda^+_{new} \le 3\lambda^- \text{ where} \\ &\lambda^-_{new} := \min_j \min_{t\in[t_j,t_j+d]} \lambda_{\min}\left(\boldsymbol{P}_{t_j,new}{}'\boldsymbol{\Sigma}_t\boldsymbol{P}_{t_j,new}\right), \\ &\lambda^+_{new} := \max_j \max_{t\in[t_j,t_j+d]} \lambda_{\max}\left(\boldsymbol{P}_{t_j,new}{}'\boldsymbol{\Sigma}_t\boldsymbol{P}_{t_j,new}\right). \end{aligned} \tag{4.1}$$

4. *Assume that* $d \ge (K+2)\alpha$*. This also implies that* $t_{j+1} - t_j > d \ge (K+2)\alpha$*. We set* K *and* α *in the Theorem. This along with (4.1) quantifies "slow subspace change".*

5. *Other assumptions: (i) define* $t_0 := 1$ *and assume that* $t_{\text{train}} \in [t_0, t_1)$*; (ii) for* $j = 0, 1, 2, \ldots, J$*, define* $r_j := rank(\boldsymbol{P}_{t_j})$*,* $r_{j,new} := rank(\boldsymbol{P}_{t_j,new})$*,* $r_{j,old} := rank(\boldsymbol{P}_{t_j,old})$ *Clearly,* $r_j = r_{j-1} + r_{j,new} - r_{j,old}$*. Assume that* $r_{j,new}$ *is small enough compared to* $r_{j,old}$ *so that* $r_j \le r$ *and* $r_{j,new} \le r_{new}$ *for all* j *for constants* r *and* r_{new}*. Assume that* $r + r_{new} < \min(n, t_{j+1} - t_j)$ *and* $r_{new} \le r_0$*.*

6. *Since the* $\boldsymbol{\nu}_t$*'s are bounded random variables, there exists a* $\gamma < \infty$ *and a* $\gamma_{new} \le \gamma$ *such that*

$$\max_t \|\mathbf{P}_t{}'\boldsymbol{\nu}_t\|_2 \le \gamma, \ \max_j \max_{t\in[t_j,t_j+d]} \|\mathbf{P}_{t_j,new}{}'\boldsymbol{\nu}_t\|_\infty \le \gamma_{new}.$$

We assume an upper bound on γ_{new} *in the Thoerem.*

A visual depiction of Model 5 is shown in Figure 4.3. The above model is similar to the ones introduced in [44, 85]. Various low-rank and "slow changing" models on $\boldsymbol{\Sigma}_t$

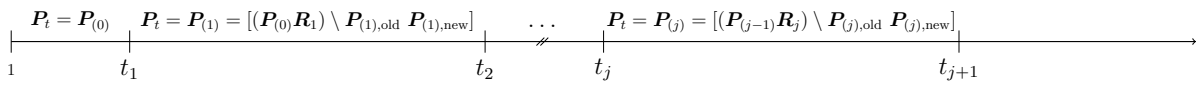


Figure 4.3: A diagram of Model 5

are special cases of the above model. One interesting special case is one that allows the variance along new directions to increase slowly as follows: for $t \in [t_j, t_j+d]$, let $\boldsymbol{\Lambda}_{t,\text{new}} := \boldsymbol{P}_{t_j,\text{new}}{}'\boldsymbol{\Sigma}_t\boldsymbol{P}_{t_j,\text{new}}$ and assume that $(\boldsymbol{\Lambda}_{t,\text{new}})_{i,i} = (v_i)^{t-t_j}q_i\lambda^-$ for $i = 1, \ldots, r_{j,\text{new}}$. Here $q_i \geq 1$ and $v_i > 1$. An upper bound on v_i of the form $q_i(v_i)^d \leq 3$ ensures that (4.1) holds.

Remark 4.2.1. *Model 5 requires the upper bound on the eigenvalues along the new directions to hold only for the first d time instants after t_j. At any time $t > t_j + d$, the eigenvalues along $\mathbf{P}_{t_j,new}$ could increase to any large value up to λ^+ either gradually or suddenly.*

The above model requires the directions to get deleted and added at the same set of times $t = t_j$. This is assumed for simplicity. In general, directions from range($\mathbf{P}_{t_j-1}$) could get deleted at any other time as well. The lower bound in (4.1) requires the energy of $\boldsymbol{\ell}_t$ along the new directions at *all* times $t \in [t_j, t_j + d]$ to be above λ^-. With very minor changes to the proof (of Lemma 4.5.34), we can relax this to the following: we can let λ^-_{new} be the minimum eigenvalue along the new directions of any α-frame *average* covariance matrix over the period $[t_j, t_j + d]$ and require this to be larger than λ^-. For video analytics, this translates to requiring that, after a subspace change, *enough (but not necessarily all)* background frames have "detectable" energy along the new directions, so that the minimum eigenvalue of the average covariance along the new directions is above a threshold. For the recommendation systems' application, this means that the initial set of users may only be influenced by a few, say five, factors, but as more users come in to the system, *some (not necessarily all)* of them may also get influenced by a sixth factor (newly added direction). There is a trade off between the upper bound on λ^+_{new} in (4.1) in Model 5 above and the bound on $\rho^2\beta$ assumed in Model 4. Allowing a larger value of

λ^{+}_{new} will require a tighter bound on $\rho^2\beta$. We chose one set of bounds, but many other pairs would also work. For video analytics, this means that if the background subspace changes are faster, then we also need the foreground objects to be moving more so we can 'see' enough of the background behind them.

4.2.3 Denseness

To separate sparse $\mathbf{x}_t$'s from the $\boldsymbol{\ell}_t$'s, the basis vectors for the subspace from which the $\boldsymbol{\ell}_t$'s are generated cannot be sparse. We quantify this using an incoherence condition similar to [32].

Model 6 (Denseness)**.** *Let μ be the smallest real number such that $\max_i \|\boldsymbol{P}_{t_j}{}'\mathbf{I}_i\|_2^2 \leq \frac{\mu r_j}{n}$ and $\max_i \|\boldsymbol{P}_{t_{j,new}}{}'\mathbf{I}_i\|_2^2 \leq \frac{\mu r_{j,new}}{n}$ for all j ($\mathbf{I}_i$ is the i^{th} column of the identity matrix; thus $\mathbf{P}'I_i$ is the i-th row of $\mathbf{P}$). Assume that*

$$2sr\mu \leq 0.09n \text{ and } 2sr_{new}\mu \leq 0.0004n.$$

Fact 4.2.2. *Model 6 is one way to ensure that $\|\boldsymbol{P}_{t_j}{}'\mathbf{I}_{\mathcal{T}}\|_2 \leq 0.3$ and $\|\boldsymbol{P}_{t_{j,new}}{}'\mathbf{I}_{\mathcal{T}}\|_2 \leq 0.02$ for all sets $\mathcal{T}$ with $|\mathcal{T}| \leq 2s$. This follows using the fact that for an $r \times s$ matrix M, $\|M\|_2 \leq \sqrt{s}\max_i \|M_i\|_2$ where M_i is the i-th column vector of M.*

4.2.4 Assumption on the unstructured noise $\mathbf{w}_t$

Model 7. *Assume that the noise $\mathbf{w}_t$ is zero mean, mutually independent over time, and bounded with $\|\mathbf{w}_t\|_2 \leq \epsilon_w$.*

4.2.5 Main result for Automatic ReProCS

In this section, we give a correctness result for Automatic ReproCS, i.e., for Algorithm 4 with the cluster PCA (cPCA) step removed. This is exactly the algorithm studied in our earlier work [85]. The result given in [85] for it required mutual independence of the $\boldsymbol{\ell}_t$'s over time. For the video application, this means that background changes at

different times are due to independent causes, e.g., independent light flickers. This is often a restrictive assumption. The current result replaces this requirement with an autoregressive model which is a much better model for background changes due to correlated factors such as moving lake or sea waters.

The main idea of Automatic ReProCS is as follows. It estimates the initial subspace as the top r_0 left singular vectors of $[\boldsymbol{m}_1, \boldsymbol{m}_2, \dots, \boldsymbol{m}_{t_{\text{train}}}]$. At time t, if the previous subspace estimate, $\hat{\mathbf{P}}_{t-1}$, is accurate enough, because of the "slow subspace change" assumption, projecting $\mathbf{m}_t = \mathbf{x}_t + \boldsymbol{\ell}_t + \mathbf{w}_t$ onto its orthogonal complement nullifies most of $\boldsymbol{\ell}_t$. Specifically, we compute $\boldsymbol{y}_t := \boldsymbol{\Phi}_t \mathbf{m}_t$ where $\boldsymbol{\Phi}_t := \mathbf{I} - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}'$. Clearly, $\boldsymbol{y}_t = \boldsymbol{\Phi}_t \mathbf{x}_t + \boldsymbol{b}_t$ with $\|\boldsymbol{b}_t\|_2$ being small. Thus recovering $\mathbf{x}_t$ from $\boldsymbol{y}_t$ is a traditional sparse recovery problem in small noise [12]. We recover $\mathbf{x}_t$ by l_1 minimization with the constraint $\|\boldsymbol{y}_t - \boldsymbol{\Phi}_t x\|_2 \le \xi$ and estimate its support by thresholding using a threshold ω. We use the estimated support, $\hat{\mathcal{T}}_t$, to get an improved debiased estimate of $\mathbf{x}_t$, denoted $\hat{\mathbf{x}}_t$, by least squares (LS) estimation on $\hat{\mathcal{T}}_t$ [52]. We then estimate $\boldsymbol{\ell}_t$ as $\hat{\boldsymbol{\ell}}_t = \mathbf{m}_t - \hat{\mathbf{x}}_t$. The estimates $\hat{\boldsymbol{\ell}}_t$ are used in the subspace estimation step which involves (i) detecting subspace change; and (ii) K steps of projection-PCA, each done with a new set of α frames of $\hat{\boldsymbol{\ell}}_t$, to get an accurate enough estimate of the new subspace. This step is explained in detail later in Sec. 4.3. Automatic ReProCS has four algorithm parameters - α, K, ξ, ω - whose values will be set in the result below.

Theorem 4.2.3. *Consider Algorithm 4 without the cluster PCA step. Assume that, for* $t > t_{\text{train}}$, $\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{w}_t + \mathbf{x}_t$ *and, for* $t \le t_{\text{train}}$, $\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{w}_t$. *Pick a* ζ *that satisfies*

$$\zeta \le \min\left\{ \frac{10^{-4}}{(r_0 + Jr_{new})^2}, \frac{0.003\lambda^-}{(r_0 + Jr_{new})^2\lambda^+}, \frac{1}{(r_0 + Jr_{new})^3\gamma^2}, \frac{0.05\lambda^-}{(r_0 + Jr_{new})^3\gamma^2} \right\}.$$

Let $b_0 = 0.1$. *Suppose that the following hold.*

1. *enough initial training data is available:* $t_{\text{train}} \ge \frac{32(2(r_0+Jr_{new})\gamma^2)^2}{(1-b_0)^2(0.001 r_{new}\zeta\lambda^-)^2}(11\log n + \log 8)$
2. *algorithm parameters are set as:*

$$\xi = \xi_{\text{cor}} := \epsilon_w + \frac{2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{new}}{1-b_0};\ \omega = 7\xi;\ K = \left\lceil \frac{\log(0.85 r_{new}\zeta)}{\log(0.2)} \right\rceil;$$

$\alpha = \alpha_{add}$ *where* $\alpha_{add} \geq 32 \frac{1.2^2(2\sqrt{\zeta}+\sqrt{r_{new}}\gamma_{new}+2\epsilon_w)^4}{(1-b_0)^6} \frac{(1-b_0^2)^2}{(0.001 r_{new}\zeta\lambda^-)^2}(11\log n + \log((52K + 44)J))$

3. *model on* $\mathcal{T}_t$*: Model 4 holds;*

4. *model on* $\boldsymbol{\ell}_t$*:*

   Model 5 holds with $\mathbf{P}_{t_j,new}'[\mathbf{P}_0, \mathbf{P}_{t_1,new}, \mathbf{P}_{t_2,new}, \dots \mathbf{P}_{t_{j-1},new}] = \mathbf{0}$*,* $b \leq b_0 = 0.1$*, and with* $\sqrt{r_{new}}\gamma_{new}$ *small enough so that* $14\xi \leq \min_t \min_{i\in\mathcal{T}_t} |(x_t)_i|$*;*

   Model 6 (denseness) holds with r *replaced by* $(r_0 + Jr_{new})$*.*

5. *model on* $\mathbf{w}_t$*: Model 7 holds with* $\epsilon_w^2 \leq 0.03\zeta\lambda^-$

6. *independence: Let* $\mathcal{T} := \{\mathcal{T}_{\tilde{t}}\}_{\tilde{t}=1,2,\dots,t_{\max}}$*. Assume that* $\mathcal{T}, \boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_{t_{\max}}, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots, \boldsymbol{\nu}_{t_{\max}}$ *are mutually independent random variables.*

Then, with probability $\geq 1 - 2n^{-10}$*, at all times* t*,*

1. $\mathcal{T}_t$ *is exactly recovered, i.e.* $\hat{\mathcal{T}}_t = \mathcal{T}_t$ *for all* t*;*

2. $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \leq 1.34\left(2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{\text{new}} + \epsilon_w\right)$ *and* $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\|_2 \leq \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \epsilon_w$*;*

3. *the subspace error* $\text{SE}_t := \|(\mathbf{I} - \hat{\boldsymbol{P}}_t\hat{\boldsymbol{P}}_t')\mathbf{P}_t\|_2 \leq 10^{-2}\sqrt{\zeta}$ *for all* $t \in [t_j + d, t_{j+1})$*.*

4. *the subspace change time estimates satisfy* $t_j \leq \hat{t}_j \leq t_j + 2\alpha$*; and its estimates of the number of new directions are correct:* $\hat{r}_{j,new,k} = r_{j,new}$ *for* $j = 1, \dots, J$*.*

Proof: The above result follows as a corollary of the more general result, Theorem 4.2.8, that is given below. For its proof, please see Appendix C.6.

Remark 4.2.4. *Consider condition 6). If it is not practical to assume that* $\mathbf{w}_t$*'s are independent of* $\mathcal{T}$ *(e.g., if* $\mathbf{w}_t$ *contains the smaller magnitude outlier entries and* $\mathbf{x}_t$ *the larger ones and so* $\mathcal{T}_t = support(\mathbf{x}_t)$ *cannot be independent of* $\mathbf{w}_t$*), the following weaker assumption can be used with small changes to the proof (see Fact 4.6.1 in Sec. 4.6.2). Let* $Q := \{\mathcal{T}, \{\mathbf{w}_t\}_{t=1,2,\dots,t_{\max}}\}$*. Assume that* $Q, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots, \boldsymbol{\nu}_{t_{\max}}$ *are mutually independent.*

Theorem 4.2.3 says the following. If an accurate estimate of the initial subspace is available (t_{train} is large enough); the algorithm parameters are set appropriately; the outlier support at time t, $\mathcal{T}_t$, has enough changes over time; $\boldsymbol{\ell}_t$ follows an AR model with parameter $b \leq b_0 = 0.1$ (i.e., the $\boldsymbol{\ell}_t$'s are not too correlated over time); the low-dimensional subspace from which $\boldsymbol{\nu}_t$ is generated (this is also approximately the subspace from which $\boldsymbol{\ell}_t$ is generated) is fixed or changes "slowly" enough, i.e. (i) the delay between change times is large enough ($t_{j+1} - t_j > d \geq (K+2)\alpha$) and (ii) the eigenvalues along the newly added directions are small enough for d frames after a subspace change; the basis vectors whose span defines the low-dimensional subspaces are dense enough; the noise $\mathbf{w}_t$ is small enough; then, with high probability (whp), the error in estimating $\boldsymbol{\ell}_t$ or $\mathbf{x}_t$ will be bounded by a small value at all times t. Also, whp, the outlier support will be exactly recovered at all times; and the error in estimating the low-dimensional subspace will decay to a small constant times $\sqrt{\zeta}$ within a finite delay of a subspace change. Moreover, subspace changes will get detected within a short delay, and the dimension of the newly added subspaces will get correctly estimated.

The condition "$14\xi \leq \min_t \min_{i \in \mathcal{T}_t} |(x_t)_i|$" in condition 4) can be interpreted either as another slow subspace change condition or as a requirement that the minimum magnitude nonzero entry of $\mathbf{x}_t$ (the smallest magnitude outlier) be large enough compared to $\epsilon_w + \sqrt{r_{\text{new}}}\gamma_{\text{new}}$. Interpreted this way, it says the following. If $\boldsymbol{\ell}_t$ is the true data, $\mathbf{m}_t - \boldsymbol{\ell}_t = \mathbf{w}_t + \mathbf{x}_t$ is the vector of corruptions with $\mathbf{w}_t$ being the small corruptions and the nonzero entries of $\mathbf{x}_t$ being the large ones (outliers). We need $\mathbf{w}_t$ to be small enough to not affect subspace recovery error too much ($\|\mathbf{w}_t\|_2 \leq \epsilon_w \leq \sqrt{0.03\zeta\lambda^-}$) and we need the nonzero entries of $\mathbf{x}_t$ to be large enough to be detectable ($\min_t \min_{i \in \mathcal{T}_t} |(x_t)_i| \geq 14\xi \approx 14(\epsilon_w + \sqrt{r_{\text{new}}}\gamma_{\text{new}})$).

4.2.6 Eigenvalues' clustering assumption and main result for Automatic ReProCS-cPCA

The ReProCS algorithm studied above (which is the same as the one introduced in [85]) does not include a step to delete old directions from the subspace estimate. As a result, its estimated subspace dimension can only increase over time. This necessitates a bound on the number of subspace changes, J. The bound is imposed by the denseness assumption - notice that Theorem 4.2.3 requires the bound in Model 6 to hold with r replaced by $r_0 + Jr_{\text{new}}$. In this section, we relax this requirement by analyzing automatic ReProCS-cPCA (Algorithm 4) which includes cluster PCA to delete the old directions from the subspace estimate. This is done by re-estimating the current subspace.

In order to be able to design an accurate algorithm to delete the old directions by re-estimating the current subspace, we need one of the following for a period of d_2 frames within the interval $[t_j, t_{j+1})$. We either need the condition number of $\mathbf{\Lambda}_t$ (or equivalently of $\mathbf{\Sigma}_t$) to be small, or we need a generalization of it: we need its eigenvalues to be "clustered" into a few (at most ϑ) clusters in such a way that the condition number within each cluster is small and the distance between consecutive clusters is large (clusters are well separated). The problem with requiring a small upper bound on the condition number of $\mathbf{\Sigma}_t$ is that it disallows situations where the $\boldsymbol{\ell}_t$'s constitute large but structured noise. This is why the "clustered" generalization is needed. This would be valid for data that has variations at different scales. For example, for data that has variations at two scales, there would be two clusters, the large scale variations would form the first cluster and the small scale ones the second cluster. These clusters would naturally be well separated.

Let ϑ denote the maximum number of clusters. As we will explain in Sec. 4.3, the subspace deletion via re-estimation step is done after the new directions are accurately estimated. As explained later, with high probability (whp), this will not happen until $t_j + K\alpha$. Thus, we assume that the clustering assumption holds for the period $[t_j + K\alpha +$

$1, t_j + K\alpha + d_2]$ with $d_2 > (\vartheta + 3)\alpha$ and $t_{j+1} - t_j > K\alpha + d_2$. In the algorithm, cluster PCA is done starting at $\hat{t}_j + K\alpha$.

Model 8. *Assume the following.*

1. *Assume that $t_{j+1} - t_j > K\alpha + d_2$ for an integer $d_2 \geq (\vartheta + 3)\alpha$ (where ϑ is defined below). Assume that for all $t \in [t_j + K\alpha + 1, t_j + K\alpha + d_2]$, $\mathbf{\Lambda}_t$ is constant; let $\mathbf{\Lambda}_{(j)}$ be this constant matrix and assume that $\lambda_{\min}(\mathbf{\Lambda}_{(j)}) \geq \lambda^-$.*

2. *Define a partition of the index set $\{1, 2, \ldots r_j\}$ into sets $\mathcal{G}_{j,1}, \mathcal{G}_{j,2}, \ldots, \mathcal{G}_{j,\vartheta_j}$ as follows. Sort the eigenvalues of $\mathbf{\Lambda}_{(j)}$ in decreasing order of magnitude. To define $\mathcal{G}_{j,1}$, start with the first (largest) eigenvalue and keep adding smaller eigenvalues to the set. Stop when the ratio of the maximum to the minimum eigenvalue first exceeds $g^+ = 3$ or when there are no more nonzero eigenvalues. Suppose this happens for the i-th eigenvalue. Then, define $\mathcal{G}_{j,1} = \{1, 2, \ldots i-1\}$. For $\mathcal{G}_{j,2}$, start with the i-th eigenvalue and repeat the same procedure. Keep doing this until there are no more nonzero eigenvalues. Let ϑ_j denote the number of clusters for the j-th subspace and let $\vartheta := \max_j \vartheta_j$. Define*

$$\lambda^+_{j,k} := \max_{i \in \mathcal{G}_{j,k}} \lambda_i \left(\mathbf{\Lambda}_{(j)}\right), \quad \lambda^-_{j,k} := \min_{i \in \mathcal{G}_{j,k}} \lambda_i \left(\mathbf{\Lambda}_{(j)}\right)$$

Assume that the clusters are well-separated, i.e.,

$$\frac{\lambda^+_{j,k+1}}{\lambda^-_{j,k}} \leq \chi^+ = 0.2 \tag{4.2}$$

Fact 4.2.5. *The above way of defining the clusters is one way to ensure that the condition number of the eigenvalues within each cluster (ratio of the maximum to minimum eigenvalue of the cluster) is below $g^+ = 3$, i.e., for all $k = 1, 2, \ldots, \vartheta_j$,*

$$\frac{\lambda^+_{j,k}}{\lambda^-_{j,k}} \leq g^+ = 3. \tag{4.3}$$

A model similar to Model 8 was first introduced in [44] where the cluster PCA idea was introduced.

Remark 4.2.6. *The case when, for the entire period* $[t_j + K\alpha + 1, t_j + K\alpha + d_2]$*, the condition number of* $\mathbf{\Sigma}_t$ *is below* g^+ *is a special case of Model 8 with* $\vartheta_j = \vartheta = 1$ *and* $\chi^+ = 0$*.*

Remark 4.2.7. *Model 5 requires the eigenvalues along* $\mathbf{P}_{t_j,new}$ *to be small for* $t \in [t_j, t_j + d]$ *with* $d \geq (K+2)\alpha$ *while Model 8 requires all eigenvalues to be constant for* $t \in [t_j + K\alpha + 1, t_j + K\alpha + d_2]$*. Taken together, this means that for all* $t \in [t_j, t_j + K\alpha + d_2]$*, we are requiring that the eigenvalues along* $\mathbf{P}_{t_j,new}$ *be small. However after* $t = t_j + K\alpha + d_2$*, there is no constraint on its eigenvalues until* $t = t_{j+1} + K\alpha$ *at which time Model 8 again requires all eigenvalues to be constant. Thus, in the interval* $[t_j + K\alpha + d_2 + 1, t_{j+1} + K\alpha]$*, or in later intervals of the form* $[t_{j+j'} + K\alpha + d_2 + 1, t_{j+j'+1} + K\alpha]$ *for any* $j' > 0$*, the eigenvalues along* $\mathbf{P}_{t_j,new}$ *could increase to any large value up to* λ^+ *either gradually or suddenly. Or they could also decrease to any small value.*

With small changes to the proof, one can relax the $\mathbf{\Lambda}_t$ constant requirement to the following. Let ClustInterval denote the interval $[t_j + K\alpha + 1, t_j + K\alpha + d_2]$ and let t_0 denote the first time instant of ClustInterval. Define a partition of the index set $\{1, 2, \ldots r_j\}$ into sets $\mathcal{G}_{j,1}, \mathcal{G}_{j,2}, \ldots, \mathcal{G}_{j,\vartheta_j}$ as in Model 8 but by using $\mathbf{\Lambda}_{t_0}$ to replace $\mathbf{\Lambda}_{(j)}$. Assume that for all $k = 1, 2, \ldots, \vartheta_j$, $\lambda_{j,k}^- \leq \min_{i \in \mathcal{G}_{j,k}} \min_{t \in \text{ClustInterval}} \lambda_i(\mathbf{\Lambda}_t) \leq \max_{i \in \mathcal{G}_{j,k}} \max_{t \in \text{ClustInterval}} \lambda_i(\mathbf{\Lambda}_t) \leq \lambda_{j,k}^+$.

At the cost of making our model more complicated, the requirement discussed in Remark 4.2.7 can also be relaxed, i.e., we can allow the eigenvalues along $\mathbf{P}_{t_j,\text{new}}$ to increase to a large value before imposing Model 8. To do this we need to assume an upper bound on d. Suppose that $(K+2)\alpha \leq d \leq (K+3)\alpha$. Suppose also that we allow a period of $\Delta = 4\alpha$ frames for the new eigenvalues to increase. We can assume Model 8 holds for the period $[t_j + K\alpha + 3\alpha + \Delta + 1, t_j + K\alpha + 3\alpha + \Delta + d_2]$ with $d_2 > (\vartheta + 3)\alpha$.

In addition, we would also need $t_{j+1} - t_j > (K+3)\alpha + \Delta + d_2$. With this, we would run the cluster PCA algorithm starting at $\hat{t}_j + K\alpha + 3\alpha + \Delta$ instead of at $\hat{t}_j + K\alpha$ as we do now.

We give below a correctness result for Automatic ReproCS-cPCA (Algorithm 4) that uses the above model. It has one extra parameter, $\hat{g}^+$, other than the four used by Automatic ReProCS. $\hat{g}^+$ is used to estimate the eigenvalue clusters automatically from an empirical covariance matrix computed using an appropriate set of $\hat{\boldsymbol{\ell}}_t$'s.

Theorem 4.2.8. *Consider Algorithm 4. Assume that, for $t > t_{\text{train}}$, $\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{w}_t + \mathbf{x}_t$ and, for $t \le t_{\text{train}}$, $\mathbf{m}_t = \boldsymbol{\ell}_t + \mathbf{w}_t$. Pick a ζ that satisfies*

$$\zeta \le \min\left\{ \frac{10^{-4}}{(r+r_{new})^2}, \frac{0.003\lambda^-}{(r+r_{new})^2\lambda^+}, \frac{1}{(r+r_{new})^3\gamma^2}, \frac{0.05\lambda^-}{(r+r_{new})^3\gamma^2} \right\}.$$

Let $b_0 = 0.1$. Suppose that the following hold.

1. *enough initial training data is available:* $t_{\text{train}} \ge \frac{32(2r\gamma^2)^2}{(1-b_0)^2(0.001 r_{new}\zeta\lambda^-)^2}(11\log n + \log 8)$

2. *algorithm parameters are set as:*
   $\xi = \xi_{\text{cor}} := \epsilon_w + \frac{2\sqrt{\zeta}+\sqrt{r_{new}}\gamma_{new}}{1-b_0}$; $\omega = 7\xi$; $K = \left\lceil \frac{\log(0.85 r_{new}\zeta)}{\log(0.2)} \right\rceil$; $\hat{g}^+ := \frac{g^+ + 0.06}{1-0.06} = 3.26$;
   $\alpha = \max\{\alpha_{add}, \alpha_{del}\}$ *where* $\alpha_{add} \ge 32\frac{1.2^4(2\sqrt{\zeta}+\sqrt{r_{new}}\gamma_{new}+2\epsilon_w)^4}{(1-b_0)^6}\frac{(1-b_0^2)^2}{(0.001 r_{new}\zeta\lambda^-)^2}(11\log n +$
   $\log((52K+44)J))$ *and* $\alpha_{del} \ge 32\frac{1.2^4 r^2\gamma^4}{(1-b_0)^6}\frac{(1-b_0^2)^2}{(0.001 r_{new}\zeta\lambda^-)^2}(11\log n + \log((52\vartheta + 36)J))$;

3. *model on $\mathcal{T}_t$: Model 4 holds;*

4. *model on $\boldsymbol{\ell}_t$:*
   Model 5 holds with $b \le b_0 = 0.1$ and with $\sqrt{r_{new}}\gamma_{new}$ small enough so that $14\xi \le \min_t \min_{i\in\mathcal{T}_t} |(x_t)_i|$;
   Model 8 holds with $|\mathcal{G}_{j,k}| \ge 0.15(r + r_{new})$;
   Model 6 (denseness) holds.

5. *model on $\mathbf{w}_t$: Model 7 holds with $\epsilon_w^2 \le 0.03\zeta\lambda^-$*

6. *independence: Let* $\mathcal{T} := \{\mathcal{T}_{\tilde{t}}\}_{\tilde{t}=1,2,\ldots,t_{\max}}$. *Assume that* $\mathcal{T}, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{t_{\max}}, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots, \boldsymbol{\nu}_{t_{\max}}$ *are mutually independent random variables.*

Then, with probability $\geq 1 - 3n^{-10}$, *at all times* t,

1. $\mathcal{T}_t$ *is exactly recovered, i.e.* $\hat{\mathcal{T}}_t = \mathcal{T}_t$ *for all* t;
2. $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \leq 1.34\left(2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{\text{new}} + \epsilon_w\right)$ *and* $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\|_2 \leq \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \epsilon_w$;
3. *the subspace error* $\text{SE}_t := \|(\mathbf{I} - \hat{\boldsymbol{P}}_t\hat{\boldsymbol{P}}_t{}')\mathbf{P}_t\|_2 \leq 10^{-2}\sqrt{\zeta}$ *for all* $t \in [t_j + d, t_{j+1})$;
4. *the subspace change time estimates given by Algorithm 4 satisfy* $t_j \leq \hat{t}_j \leq t_j + 2\alpha$;
5. *its estimates of the number of new directions are correct:* $\hat{r}_{j,new,k} = r_{j,new}$ *for* $j = 1, \ldots, J$;
6. *eigenvalue clusters are recovered exactly:* $\hat{\mathcal{G}}_{j,k} = \mathcal{G}_{j,k}$ *for all* j *and* k; *thus its estimate of the number of deleted directions is also correct.*

Proof: The proof outline is given in Section 4.4. The proof is given in Sections 4.5, 4.6, 4.7.

Remark 4.2.9. *Notice that the lower bound* $|\mathcal{G}_{j,k}| \geq 0.15(r + r_{new})$ *can hold only if the number of clusters* ϑ_j *is at most 6. This is one choice that works along with the given bounds on other quantities such as* $\rho^2\beta$. *It can be made larger if we assume a tighter bound on* $\rho^2\beta$ *for example. But what will remain true is that our result requires the number of clusters to be* $\mathcal{O}(1)$.

Remark 4.2.10. *The independence assumption can again be replaced by the weaker one of Remark 4.2.4.*

The extra assumption needed by the above result compared to Theorem 4.2.3 is the clustering one. Using this, ReProCS-cPCA is able to correctly estimate the current subspace. Thus,for $t \in [t_j, \hat{t}_j + \alpha]$, $\hat{\mathbf{P}}_{t-1}$ is an accurate estimate of range$(\mathbf{P}_{t_j-1})$ where as when

using ReProCS (and Theorem 4.2.3), it is an estimate of range($[\mathbf{P}_0, \mathbf{P}_{t_1,\text{new}}, \mathbf{P}_{t_2,\text{new}}, \dots, \mathbf{P}_{t_{j-1},\text{new}}]$). Because of this, (i) the above result needs a much weaker denseness assumption, (ii) it does not need a bound on J, and (iii) it requires the new directions to only be orthogonal to range($\mathbf{P}_{t_j-1}$).We discuss the results in detail in Sec. 4.2.7.

Corollary 4.2.11. *The following conclusions also hold under the assumptions of Theorem 4.2.8 with probability at least* $1 - 3n^{-10}$.

1. *The recovery error satisfies* $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\|_2 \le \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \epsilon_w$ *and*

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \le \begin{cases} 1.34\left(2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{\text{new}} + \epsilon_w\right) & t \in [t_j, (\hat{u}_j+1)\alpha] \\ 1.34(2.15\sqrt{\zeta} + 0.19\cdot(0.1)^{k-1} & t \in [(\hat{u}_j+k-1)\alpha+1, (\hat{u}_j+k)\alpha], \\ \quad \cdot\sqrt{r_{new}}\gamma_{\text{new}} + \epsilon_w) & \quad k = 2,3,\dots,K \\ 2.67(\sqrt{\zeta} + \epsilon_w) & t \in \left[\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta+1)\alpha\right] \\ 2.67(\frac{r}{r+r_{new}}\sqrt{\zeta} + \epsilon_w) & t \in \left[\hat{t}_j + K\alpha + (\vartheta+1)\alpha + 1, t_{j+1} - 1\right]; \end{cases}$$

2. *The subspace error satisfies,*

$$\text{SE}_t \le \begin{cases} 1 & t \in \left[t_j, \hat{t}_j + \alpha\right] \\ 10^{-2}\sqrt{\zeta} + 0.19\cdot 0.1^{k-1} & t \in \left[\hat{t}_j + (k-1)\alpha + 1, \hat{t}_j + k\alpha\right], \ k = 2,3,\dots,K \\ 10^{-2}\sqrt{\zeta} & t \in \left[\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta+1)\alpha\right] \\ 10^{-2}\frac{r}{r+r_{new}}\sqrt{\zeta} & t \in \left[\hat{t}_j + K\alpha + (\vartheta+1)\alpha + 1, t_{j+1} - 1\right]; \end{cases}$$

Online matrix completion (MC). MC can be interpreted as a special case of RPCA and hence the same is true for online MC and online RPCA [32, 85]. In [85], we explicitly stated results for both. In a similar fashion, an analog of either of the above results can also be obtained for online MC.

Offline RPCA. In certain applications such as video analytics, an improved offline estimate of both the background and the foreground is desirable. In some other applications, there is no real need for an online solution. We show here that, with a delay of at

most $(K+2)\alpha$ frames, by using essentially the same ReProCS algorithm with one extra step, it is possible to recover $\mathbf{x}_t$ and $\boldsymbol{\ell}_t$ with close to zero error.

Corollary 4.2.12 (Offline RPCA)**.** *Consider the estimates given in the last two lines of Algorithm 4. Under the assumptions of Theorem 4.2.8, with probability at least* $1-3n^{-10}$*, at all times* t*,* $\|\mathbf{x}_t - \hat{\mathbf{x}}_t^{\text{offline}}\|_2 \leq 2.67(\sqrt{\zeta}+\epsilon_w)$*,* $\|\hat{\boldsymbol{\ell}}_t^{\text{offline}} - \boldsymbol{\ell}_t\|_2 \leq 2.67(\sqrt{\zeta}+2\epsilon_w)$*, and all its other conclusions hold.*

Observe that the offline recovery error can be made smaller and smaller by reducing ζ (this, in turn, will result in an increased delay between subspace change times). As can be seen from the last two lines of Algorithm 4, the offline estimates are obtained at $t = \hat{t}_j + K\alpha$. Since $\hat{t}_j \leq t_j + 2\alpha$, this means that the offline estimates are obtained after a delay of at most $(K+2)\alpha$ frames.

4.2.7 Discussion

Online versus offline. We analyze an online algorithm that is faster and needs less storage. It needs to store only a few $n \times \alpha$ or $n \times r$ matrices, while PCP needs to store matrices of size $n \times t_{\max}$. Other results for online algorithms include correctness results from [83, 84, 85] (discussed below), and partial results of Qiu et al. [44] and Feng et al. [65]. In [65], Feng et al. proposed a method for online RPCA and proved a partial result for their algorithm. Their approach was to reformulate the PCP program and to use this reformulation to develop a recursive algorithm that converged asymptotically to the solution of PCP as long as the basis estimate $\hat{\boldsymbol{P}}_t$ was full rank at each time t. Since this result assumed something about an intermediate algorithm estimate, $\hat{\boldsymbol{P}}_t$, it was a *partial* result. In [44], Qiu et al. obtained a performance guarantee for ReProCS and ReProCS-cPCA that also needed intermediate algorithm estimates to satisfy certain properties. In particular, their result required that the basis vectors for the currently unestimated subspace, range$((I - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}')\mathbf{P}_{t_j,\text{new}})$, be dense vectors. Thus, their result

was also a *partial* result. In the current work, we remove this requirement and provide a *correctness result* for both ReProCS and ReProCS-cPCA. The assumption that helps us get this is Model 4 on $\mathcal{T}_t$ (or its generalization given in Model 10 later). Secondly, unlike [44], we provide a correctness result for an automatic algorithm that does not assume knowledge of subspace change times, number of directions added or removed, or of the eigenvalue-based subspace clusters. Thirdly, we allow the $\boldsymbol{\ell}_t$'s to follow an AR model where as [44] required independence over time.

To our knowledge, our work and [83, 84, 85] are the only correctness results for an online RPCA method. Our work significantly improves upon the results of [83, 84, 85]. We allow the $\boldsymbol{\ell}_t$'s to be correlated over time and use a first order AR model to model the correlation. As discussed earlier, this is significantly more practically valid than the independence assumption used in [83, 84, 85]. It includes independence as a special case. Moreover, with the extra clustering assumption, we are able to analyze Automatic ReProCS-cPCA in Theorem 4.2.8. It needs a much weaker rank-sparsity assumption than what is needed by the result of [85], and it does not need a bound on J. We discuss this below.

Bounds on rank and sparsity. Let $\mathbf{L} := [\ell_1, \ell_2 \dots \ell_{t_{\max}}]$, $\mathbf{S} := [x_1, x_2 \dots x_{t_{\max}}]$, $r_{\text{mat}} := \text{rank}(\mathbf{L})$ and let s_{mat} be the number of nonzero entries in $\mathbf{S}$. With our models, $s_{\text{mat}} \le s t_{\max}$ and $r_{\text{mat}} \le r_0 + J r_{\text{new}}$ with both bounds being tight. Models 4 and 6 constrain s and s, r, r_{new} respectively. Model 4 needs $s \le \rho_2 n/\alpha$ and Model 6 needs $rs \in \mathcal{O}(n)$ and $r_{\text{new}} s \in \mathcal{O}(n)$. Using the expression for α, it is easy to see that if $J \in \mathcal{O}(n)$, $r_{\text{new}} \in \mathcal{O}(1)$ and $r \in \mathcal{O}(\log n)$, then $\frac{1}{\alpha} \in \mathcal{O}(\frac{\zeta^2}{r^2 \log n}) = \mathcal{O}(\frac{1}{(\log n)^9})$ Alternatively, if $r \in \mathcal{O}(1)$, then $\frac{1}{\alpha} \in \mathcal{O}(\frac{1}{\log n})$. Thus, Theorem 4.2.8 definitely holds in two regimes of interest. The first is $J \in \mathcal{O}(n)$, $r_{\text{new}} \in \mathcal{O}(1)$, $r \in \mathcal{O}(\log n)$, $s_{\text{mat}} \in \mathcal{O}(\frac{n t_{\max}}{(\log n)^9})$ and $r_{\text{mat}} \in \mathcal{O}(n)$. The second is $J \in \mathcal{O}(n)$, $r_{\text{new}} \in \mathcal{O}(1)$, $r \in \mathcal{O}(1)$, $s_{\text{mat}} \in \mathcal{O}(\frac{n t_{\max}}{\log n})$ and $r_{\text{mat}} \in \mathcal{O}(n)$. The second regime is more favorable when comparing bounds on s_{mat} and r_{mat}, but, it also implies that the dimension of the subspace at any given time is $\mathcal{O}(1)$.

This can be restrictive. The first regime allows the subspace dimension at any time to be $\mathcal{O}(\log n)$ which is more reasonable, but, because of this, it needs a tighter bound on s and hence on s_{mat}.

In either regime, our requirements are weaker than those of the PCP results from [33, 94]: they need $r_{\text{mat}} s = \mathcal{O}(n)$ which implies $r_{\text{mat}} s_{\text{mat}} \in \mathcal{O}(nt_{\max})$; thus if $s_{\text{mat}} \in \mathcal{O}(\frac{nt_{\max}}{\log n})$, they would require r_{mat} to be $\mathcal{O}(\log n)$. In the first regime, our conditions are slightly stronger than those of the PCP result from [32] while in the second, they are comparable: [32] needs $r_{\text{mat}} \in \mathcal{O}(\frac{n}{(\log n)^2})$ and $s_{\text{mat}} \in \mathcal{O}(nt_{\max})$.

Either set of requirements for Theorem 4.2.8 is significantly weaker than what is needed by Theorem 4.2.3 or by the results of [83, 84, 85]: both need $r_{\text{mat}} \in \mathcal{O}(\log n)$. This is because both analyze ReProCS without the cluster PCA based subspace deletion step. Suppose that $r_{j,\text{new}} = r_{\text{new}}$ for each j. For ReProCS without cluster PCA, this means that the dimension of the estimated subspace grows by r_{new} with each subspace change time. Thus, the maximum dimension of the estimated subspace is $r_{\text{mat}} = r_0 + Jr_{\text{new}}$ and this is what was used in place of r in the denseness assumption as well in the bound on ζ. This is why these results need r_{mat} to be $\mathcal{O}(\log n)$. However, in Theorem 4.2.8, we analyze ReProCS with cluster PCA. Cluster PCA is used to re-estimate the current subspace and thus effectively delete the subspace corresponding to the old directions. This ensures that the rank of the estimated subspace is also bounded by the rank of the true subspace at any time, i.e. by r. Thus, Theorem 4.2.8 only needs $r \in \mathcal{O}(\log n)$ while r_{mat} can as large as $\mathcal{O}(n)$.

No bound on the number of subspace changes, J. Notice that the result for ReProCS-cPCA given in Theorem 4.2.8 does not require an upper bound on the number of subspace changes, J. On the other hand, the results for ReProCS (both Theorem 4.2.3 and the results from [83, 84, 85]) require a bound on J that is imposed by the denseness assumption: they need $(r_0 + Jr_{\text{new}})2s\mu \leq 0.09n$. All results for PCP need a bound on r_{mat}. Under our model of subspace change, r_{mat} is at most $r_0 + Jr_{\text{new}}$

with the bound being tight and hence the PCP results also need a bound on J. Of course, even for Theorem 4.2.8, J does affect bounds on other quantities: the result needs $t_{j+1} - t_j > d > K\alpha + (\vartheta + 3)\alpha$ where α is an algorithm parameter that depends linearly on $\log J$. Thus, for any given value of J, the delay between subspace change times, $t_{j+1} - t_j$, and the duration for which the eigenvalues along the new directions need to be small (quantified in (4.1)), d, need to grow as $\log J$.

Assumptions on how often the outlier support $\mathcal{T}_t$ needs to change. An important advantage of our work over PCP and other batch methods is that we allow more correlated changes of the set of outliers over time. From the assumption on $\mathcal{T}_t$, it is easy to see that we allow the number of outliers per row of $\mathbf{L}$ to be $\mathcal{O}(t_{\max})$, as long as the sets follow Model 4[1]. This is the same as what our previous results [83, 84, 85] also allowed. On the other hand, the PCP results from [33, 94] need this number to be $\mathcal{O}(\frac{t_{\max}}{r_{\max}})$ which is stronger. The PCP result from [32] needs that the set $\cup_{t=1}^{t_{\max}} \mathcal{T}_t$ should be generated uniformly at random which is even stronger.

Other assumptions. The above advantages are obtained because we use extra assumptions on $\boldsymbol{\ell}_t$. We assume (i) accurate knowledge of the initial subspace (or available outlier free data from which this can be obtained), (ii) slow subspace change as quantified by (4.1) and the lower bound on the delay between subspace change times, and (iii) for a period of time after the previous subspace change has stabilized, we assume that the eigenvalues along the various subspace directions can be clustered into a few clusters. The result of [85] required (i) and (ii) but not (iii). On the other hand, the PCP results [32, 33, 94] do not need any of the above. But they need other extra assumptions. They require denseness of the right singular vectors of $\mathbf{L}$ and a bound on the maximum absolute entry of the matrix UV' where U is the matrix of left singular vectors of $\mathbf{L}$ and V is the matrix

[1] In a period of length α, the set $\mathcal{T}_t$ can occupy index i for at most $\rho\beta$ time instants, and this pattern is allowed to repeat every α time instants. So an index can be in the support for a total of $\rho\beta\frac{t_{\max}}{\alpha}$ time instants and the model assumes $\rho\beta \leq \frac{0.0001\alpha}{\rho}$ for a constant ρ. Thus an index i can be part of the support $\mathcal{T}_t$ for at most $\frac{0.0001}{\rho} t_{\max} \in \mathcal{O}(t_{\max})$ time instants.

of its right singular vectors. In our notation range(U) = range$([P_0, P_{1,\text{new}}, \dots P_{J,\text{new}}])$. We assume denseness of U but not of the right singular vectors.

Setting algorithm parameters. Our result needs five algorithm parameters to be appropriately set. Some of these require knowing at least an upper bound on the model parameters. Our result needs to know upper bounds on $\gamma, \gamma_{\text{new}}$, r_0, r, r_{new}, b, and g^+. The PCP results need this for none [32] or at most one [33, 94] algorithm parameter. We briefly explain in Sec. 4.8.1 how to set algorithm parameters automatically for practical experiments.

Other work. A recent work that uses knowledge of the initial subspace estimate but performs recovery in a piecewise batch fashion is modified-PCP [97]. Like PCP, the result for modified PCP also needs uniformly randomly generated support sets which is stronger than what we need. But, like PCP, it does not need the other extra assumptions that ReProCS needs. Another somewhat related work is the algorithm and correctness result of Feng et al. [66] on online PCA with contaminated data. This does not model the outlier as a sparse vector but defines anything that is far from the data subspace as an outlier.

4.3 Automatic ReProCS-cPCA

The automatic ReProCS-cPCA algorithm is summarized in Algorithm 4. It proceeds as follows. It begins by estimating the initial subspace as the top r_0 left singular vectors of $[\boldsymbol{m}_1, \boldsymbol{m}_2, \dots, \boldsymbol{m}_{t_{\text{train}}}]$. Let $\hat{\mathbf{P}}_t$ denote the basis matrix for the subspace estimate at time t. At time t, if the previous subspace estimate, $\hat{\mathbf{P}}_{t-1}$, is accurate enough, because of the "slow subspace change" assumption, projecting $\mathbf{m}_t = \mathbf{x}_t + \boldsymbol{\ell}_t + \mathbf{w}_t$ onto its orthogonal complement nullifies most of $\boldsymbol{\ell}_t$. Specifically, we compute $\boldsymbol{y}_t := \boldsymbol{\Phi}_t \mathbf{m}_t$ where $\boldsymbol{\Phi}_t := \mathbf{I} - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}'$. Clearly, $\boldsymbol{y}_t = \boldsymbol{\Phi}_t \mathbf{x}_t + \boldsymbol{b}_t$ where $\boldsymbol{b}_t := \boldsymbol{\Phi}_t \boldsymbol{\ell}_t + \boldsymbol{\Phi}_t \mathbf{w}_t$ and it can be argued that $\|\boldsymbol{b}_t\|_2$ is small: $\|\boldsymbol{\Phi}_t \boldsymbol{\ell}_t\|_2$ is small due to the slow subspace change assumption and

$\|\mathbf{w}_t\|_2 \leq \epsilon_w$. Thus recovering $\mathbf{x}_t$ from $\boldsymbol{y}_t$ becomes a traditional sparse recovery problem in small noise [12]. We recover $\mathbf{x}_t$ by l_1 minimization with the constraint $\|\boldsymbol{y}_t - \boldsymbol{\Phi}_t x\|_2 \leq \xi$ and estimate its support by thresholding using a threshold ω. We use the estimated support, $\hat{\mathcal{T}}_t$, to get an improved debiased estimate of $\mathbf{x}_t$, denoted $\hat{\mathbf{x}}_t$, by least squares (LS) estimation on $\hat{\mathcal{T}}_t$. We then estimate $\boldsymbol{\ell}_t$ as $\hat{\boldsymbol{\ell}}_t = \mathbf{m}_t - \hat{\mathbf{x}}_t$. By the denseness assumption given in Model 6, it can be argued that the restricted isometry constant (RIC) of $\boldsymbol{\Phi}_t$ will be small. Under the theorem's assumptions, we can bound it by 0.14. This ensures that a sparse $\mathbf{x}_t$ is indeed accurately recoverable from $\boldsymbol{y}_t$. With the support estimation threshold ω set as in Theorem 4.2.8, it can be argued that the support will be exactly recovered, i.e., $\hat{\mathcal{T}}_t = \mathcal{T}_t$. Let $\mathbf{e}_t := \boldsymbol{\ell}_t - \hat{\boldsymbol{\ell}}_t$. With this, it is clear that $\mathbf{e}_t = (\hat{\mathbf{x}}_t - \mathbf{x}_t) - \mathbf{w}_t$ satisfies

$$\mathbf{e}_t = \mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\boldsymbol{b}_t - \mathbf{w}_t = \mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\boldsymbol{\Phi}_t(\boldsymbol{\ell}_t + \mathbf{w}_t) - \mathbf{w}_t. \quad (4.4)$$

Using the bound on the RIC of $\boldsymbol{\Phi}_t$, clearly $\|(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}^{-1}\|_2 \leq (1-0.14)^{-1} < 1.2$. Thus, $\|\mathbf{e}_t\|_2 \leq 1.2\|\boldsymbol{b}_t\|_2 + \epsilon_w$, i.e., it is small too. In other words, $\boldsymbol{\ell}_t$ is accurately recovered.

The estimates $\hat{\boldsymbol{\ell}}_t$ are used in the subspace estimation step which involves (i) detecting subspace change; (ii) K steps of projection-PCA, each done with a new set of α frames of $\hat{\boldsymbol{\ell}}_t$, to get an accurate enough estimate of the newly added subspace; and (iii) cluster PCA to delete the old subspace by re-estimating the current subspace. At the end of the projection PCA step, the estimated subspace dimension is at most $r + r_{\text{new}}$, and after cluster PCA, it comes down to at most r.

Subspace update. In the subspace update step, the algorithm switches between the "detect" phase, the "pPCA" phase and the "cPCA" phase. It starts in the "detect" phase. When a subspace change is detected, i.e. at $t = \hat{t}_j$, it enters the "pPCA" phase. After K iterations of projection-PCA, i.e. at $t = \hat{t}_j + K\alpha$, the new subspace has been accurately estimated. At this time, it enters the "cPCA" phase. At $t = \hat{t}_j + K\alpha + (\vartheta+1)\alpha$, cluster PCA is done. At this time, it enters the "detect" phase again and remains in it until the next subspace change is detected. We detect the j-th subspace change as

follows. Let $\hat{\mathbf{P}}_* := \hat{\mathbf{P}}_{\hat{t}_{j-1}+K\alpha+(\vartheta+1)\alpha}$. We detect change by comparing the eigenvalues of $\frac{1}{\alpha}\sum_t(\mathcal{I} - \hat{\boldsymbol{P}}_*\hat{\boldsymbol{P}}_*{}')\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t'(\mathcal{I} - \hat{\boldsymbol{P}}_*\hat{\boldsymbol{P}}_*{}')$ to a chosen threshold at every $t = u\alpha$ when the algorithm is in the "detect" phase.

Projection-PCA (p-PCA). We use projection-PCA to estimate the newly added subspace. The reason this cannot be done using standard PCA is as follows [44]. Let $\sum_t$ denote a sum over an α length time interval. Because of how $\boldsymbol{\ell}_t$ is recovered, the error, $\mathbf{e}_t$, in the estimate of $\boldsymbol{\ell}_t$, $\hat{\boldsymbol{\ell}}_t$, is correlated with $\boldsymbol{\ell}_t$. This is evident from (4.4). Due to this, the dominant terms in the perturbation seen by standard PCA, $\frac{1}{\alpha}\sum_t \hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t{}' - \frac{1}{\alpha}\sum_t \boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'$, are $\frac{1}{\alpha}\sum_t \boldsymbol{\ell}_t\mathbf{e}_t{}'$ and its transpose[2]. Thus, when the condition number of $\mathrm{Cov}(\boldsymbol{\ell}_t)$ is large, it is not possible to argue that the perturbation will be small compared to the smallest eigenvalue of $\mathrm{Cov}(\boldsymbol{\ell}_t)$. With a large perturbation, either the $\sin\theta$ theorem [95] (that bounds the subspace error between the eigenvectors of the true and estimated sample covariance matrices) cannot be applied or it gives a very large and useless bound.

Projection-PCA addresses the above issue as follows. Consider the j-th subspace change. Let $\boldsymbol{P}_* := \mathbf{P}_{t_{j-1}}$, $\mathbf{P}_{\text{new}} := \mathbf{P}_{t_j,\text{new}}$, and $\hat{\mathbf{P}}_* := \hat{\mathbf{P}}_{\hat{t}_{j-1}+K\alpha+(\vartheta+1)\alpha}$. Denote the time at which this change is detected by $\hat{t}_j$. As explained in [85], it is easy to show that, whp, $t_j \le \hat{t}_j \le t_j+2\alpha$. After $\hat{t}_j$ we use SVD on K different sets of α frames of the $\hat{\boldsymbol{\ell}}_t$'s projected orthogonal to $\hat{\mathbf{P}}_*$ to get K estimates of the new subspace range($\boldsymbol{P}_{\text{new}}$). We get the k-th estimate, $\hat{\mathbf{P}}_{\text{new},k}$, as the left singular vectors of $(\mathbf{I} - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')[\hat{\boldsymbol{\ell}}_{\hat{t}_j+(k-1)\alpha+1}, \ldots, \hat{\boldsymbol{\ell}}_{\hat{t}_j+k\alpha}]$ with singular values above a threshold. After each projection-PCA step, we update $\hat{\mathbf{P}}_t$ as $\hat{\mathbf{P}}_t = [\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},k}]$. This ensures that the error $\mathbf{e}_t$ is smaller for the next projection-PCA interval compared to the previous one and hence the subspace estimates also improve with each iteration. The above is done K times with K chosen so that, by $t = \hat{t}_j + K\alpha$, the error in estimating the new subspace is below $r_{\text{new}}\zeta$, which ensures $\mathrm{SE}_t \le r\zeta + r_{\text{new}}\zeta$.

Cluster PCA for deleting directions by re-estimating the subspace. The next step is to delete the subspace range($\mathbf{P}_{j,\text{old}}$) from $\hat{\mathbf{P}}_t$. The goal of doing this is to reduce the

[2]When $\boldsymbol{\ell}_t$ and $\mathbf{e}_t$ are uncorrelated and one of them is zero mean, it can be argued by law of large numbers that, whp, these two terms will be close to zero and $\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t{}'$ will be the dominant term.

subspace error from $(r+r_{\text{new}})\zeta$ to $r\zeta$. The simplest way to do this would be to re-estimate range(P_t) by standard PCA, i.e. compute the eigenvectors of $\frac{1}{\alpha}\sum_{t=\hat{t}_j+K\alpha+1}^{t=\hat{t}_j+K\alpha+\alpha}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t'$ with eigenvalues above a threshold. However, since $\boldsymbol{\ell}_t$ and $\mathbf{e}_t$ are correlated, this will cause a problem similar to the one described above. It will work only if the condition number of Cov($\boldsymbol{\ell}_t$) is small. This is impractical though since we assume that $\boldsymbol{\ell}_t$ can be large but structured noise. Hence we re-estimate the subspace by developing a generalization of the projection-PCA idea that we call *cluster PCA (cPCA)*. This relies on the clustering assumption given in Model 8.

cPCA proceeds as follows. We first estimate the clusters as follows. We compute the empirical covariance matrix of $\hat{\boldsymbol{\ell}}_t$'s after the new subspace is accurately estimated: $\hat{\boldsymbol{\Sigma}}_{\text{sample}} = \frac{1}{\alpha}\sum_{t=\hat{t}_j+K\alpha+1}^{t=\hat{t}_j+K\alpha+\alpha}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t'$ and obtain its EVD. Let $\hat{\lambda}_i$ denote its i-th largest eigenvalue. To get the first cluster $\hat{\mathcal{G}}_{j,1}$, we start with the index of the first (largest) eigenvalue and keep adding indices of the smaller eigenvalues to it until $\frac{\hat{\lambda}_1}{\hat{\lambda}_{i+1}} > \hat{g}^+$ but $\frac{\hat{\lambda}_1}{\hat{\lambda}_i} \leq \hat{g}^+$ or until the next eigenvalue $\hat{\lambda}_{i+1} < 0.25\hat{\lambda}^-_{\text{train}}$. We set $\hat{\mathcal{G}}_{j,1} = \{1, 2, \ldots i\}$. To get the second cluster we repeat the same procedure but starting with the $(i+1)$-th eigenvalue. We repeat this until there is no eigenvalue larger than $0.25\hat{\lambda}^-_{\text{train}}$. Observe that $\hat{g}^+$ is set to a value that is a little larger than g^+ (see Theorem 4.2.8). This is needed to allow for the fact that $\hat{\lambda}_i$ is not equal to the i-th eigenvalue of $\boldsymbol{\Lambda}_{(j)}$ but is within a small margin of it. For the same reason, we need to also use a "zeroing" threshold of $0.25\hat{\lambda}^-_{\text{train}}$ (notice that $\hat{\boldsymbol{\Sigma}}_{\text{sample}}$ is not exactly low rank). This, along with appropriately setting $\hat{g}^+$, and with using the separation condition from Model 8 ensures that, whp, all the clusters are correctly recovered.

Let $\boldsymbol{G}_{j,k} := (\mathbf{P}_j)_{\hat{\mathcal{G}}_{j,k}}$. Next, we estimate the subspace corresponding to the first cluster, range($\boldsymbol{G}_{j,1}$) by standard PCA on $[\hat{\boldsymbol{\ell}}_{\hat{t}_j+(K+1)\alpha+1}, \ldots, \hat{\boldsymbol{\ell}}_{\hat{t}_j+(K+1)\alpha+\alpha}]$, i.e., by computing its top $|\hat{\mathcal{G}}_{j,1}|$ left singular vectors. Since the cluster's condition number is small (bounded by g^+), this works. Denote the basis for the estimated subspace by $\hat{\boldsymbol{G}}_{j,1}$. To estimate the subspace corresponding to the second cluster, we project the next set of α $\hat{\boldsymbol{\ell}}_t$'s orthogonal

to $\hat{\boldsymbol{G}}_{j,1}$, followed by standard PCA to compute the top $|\hat{\mathcal{G}}_{j,2}|$ left singular vectors [44]. To estimate the k-th cluster's subspace, we do a similar thing but with projecting orthogonal to the estimated subspace corresponding to the previous $k-1$ clusters [44].

4.4 Proof Outline for Theorem 4.2.8 and Corollary 4.2.11

The proof proceeds by induction. Consider the j-the subspace change interval. Let $\boldsymbol{P}_* := \mathbf{P}_{t_{j-1}} = \mathbf{P}_{t_j-1}$, $\mathbf{P}_{\text{new}} := \mathbf{P}_{t_j,\text{new}}$, and $\hat{\mathbf{P}}_* := \hat{\mathbf{P}}_{\hat{t}_{j-1}+K\alpha+(\vartheta+1)\alpha}$. Assume that there have been no (false) change detects in the interval $[\hat{t}_{j-1}+K\alpha+(\vartheta+1)\alpha+1, t_j-1]$. Thus, $\hat{\mathbf{P}}_{t_j-1} = \hat{\mathbf{P}}_*$. Assume also that the subspace, $\text{range}(\mathbf{P}_{t_j-1}) = \text{range}(\mathbf{P}_*)$, has been accurately recovered, i.e., $\text{SE}_{t_j-1} = \text{dif}(\hat{\mathbf{P}}_*, \mathbf{P}_*) \leq r\zeta$. Conditioned on this, we use the following steps to show that, whp, the same conclusions hold at $t = t_{j+1}-1$ as well.

1. First, we show that the subspace change is detected within a short delay of t_j. We show that $t_j \leq \hat{t}_j \leq t_j + 2\alpha$ whp. This is done in Lemma 4.5.28.

2. At $t = \hat{t}_j + \alpha$, the first projection-PCA step is done to get the first estimate, $\hat{\mathbf{P}}_{\text{new},1}$, of $\text{range}(\mathbf{P}_{\text{new}})$. This computes the top singular vectors of $[\hat{\boldsymbol{\ell}}_{\hat{t}_j+1}, \hat{\boldsymbol{\ell}}_{\hat{t}_j+2}, \ldots, \hat{\boldsymbol{\ell}}_{\hat{t}_j+\alpha}]$ projected orthogonal to $\text{range}(\hat{\mathbf{P}}_*)$. In the interval $[t_j, \hat{t}_j+\alpha-1]$, the new subspace is not estimated at all, i.e., $\hat{\mathbf{P}}_t = \hat{\mathbf{P}}_*$ while $\mathbf{P}_t = [\mathbf{P}_* \ \mathbf{P}_{\text{new}}]$ and so $\text{SE}_t \leq 1$. Thus, the noise seen by the projected sparse recovery step, $\boldsymbol{b}_t$, is the largest in this interval. Hence the error $\mathbf{e}_t$ is also the largest for the $\hat{\boldsymbol{\ell}}_t$'s used in the first projection-PCA step. However, due to slow subspace change, even this error is not too large. Because of this, and because $\mathbf{P}_{\text{new}}$ is dense, we can argue that $\hat{\mathbf{P}}_{\text{new},1}$ is a good estimate. We show that $\text{dif}([\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},1}], \mathbf{P}_{\text{new}}) \leq 0.19 < 1$. Thus, at this time, $\text{SE}_t = \text{dif}([\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},1}], [\mathbf{P}_* \ \mathbf{P}_{\text{new}}]) \leq r\zeta + 0.19$. This is shown in Lemmas 4.5.29 and 4.5.21.

3. At $t = \hat{t}_j + k\alpha$, for $k = 1, 2, \ldots, K$, the k-th projection-PCA step is done to get the k-th estimate, $\hat{\mathbf{P}}_{\text{new},k}$. This computes the top singular vectors of $[\hat{\boldsymbol{\ell}}_{\hat{t}_j+(k-1)\alpha+1},$

$\hat{\boldsymbol{\ell}}_{\hat{t}_j+(k-1)\alpha+2}, \ldots, \hat{\boldsymbol{\ell}}_{\hat{t}_j+k\alpha}]$ projected orthogonal to range($\hat{\mathbf{P}}_*$). After the first projection-PCA step, $\hat{\mathbf{P}}_t = [\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},1}]$ and this reduces $\boldsymbol{b}_t$ and hence $\mathbf{e}_t$ for the $\hat{\boldsymbol{\ell}}_t$'s in the next α frames. This fact, along with the fact that $\mathbf{e}_t$ is approximately sparse with support $\mathcal{T}_t$ and $\mathcal{T}_t$ follows Model 4, in turn, imply that the perturbation seen by the second projection-PCA step is even smaller. So $\hat{\mathbf{P}}_{\text{new},2}$ is a more accurate estimate of range($\boldsymbol{P}_{\text{new}}$) than $\hat{\mathbf{P}}_{\text{new},1}$. Repeating the same argument, the third estimate is even better and so on. Under the theorem's assumptions, we can show that $\text{dif}([\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},k}], \mathbf{P}_{\text{new}}) \leq 0.19 \cdot 0.1^{k-1} + 0.15 r_{\text{new}} \zeta$ and so, at $t = \hat{t}_j + k\alpha$, $\text{SE}_t \leq r\zeta + 0.19 \cdot 0.1^{k-1} + 0.15 r_{\text{new}} \zeta$. This is shown in Lemmas 4.5.29 and 4.5.21. The most important idea here is to use the fact that $\mathbf{e}_t$ is approximately supported on $\mathcal{T}_t$ (shown in Lemma 4.5.25) and the support change model on $\mathcal{T}_t$ (this is used in Lemma 4.5.22).

4. The above is repeated K times with K set to ensure that, by $t = \hat{t}_j + K\alpha$, $\text{dif}([\hat{\mathbf{P}}_* \ \hat{\mathbf{P}}_{\text{new},K}], \mathbf{P}_{\text{new}}) \leq r_{\text{new}} \zeta$ and so, at this time, $\text{SE}_t \leq (r + r_{\text{new}})\zeta$.

5. In the interval $[\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta + 1)\alpha]$, cluster PCA is done to delete range($P_{t_j,\text{old}}$). At the end of this step, we can show that the bound on SE_t has reduces from $(r + r_{\text{new}})\zeta$ to $r\zeta$. This is proved in Lemmas 4.5.30, 4.5.31 and 4.5.21.

6. Finally, we also argue that there are no (false) subspace change detects for any $t \in [\hat{t}_j + K\alpha + (\vartheta + 1)\alpha + 1, t_{j+1} - 1]$. This ensures that $\hat{t}_{j+1} \geq t_{j+1}$. This is done in Lemma 4.5.27.

To prove the theorem, we first show that the initial subspace is recovered accurately enough, i.e., $\text{SE}_t \leq r\zeta$ at $t = t_{\text{train}} + 1$, whp. This is done in Lemma 4.5.20. Then, repeating the above argument for each subspace change period, we can obtain the subspace error bounds of the theorem. We set t_{train} and α to ensure that the probability of the good events is at least $1 - 3n^{-10}$. The sparse recovery error bounds can be obtained

by using these bounds and quantifying the discussion of Sec. 4.3. This is done in Lemma 4.5.25.

The main part of the proof is the analysis of the projection-PCA steps (for subspace addition) and the cluster PCA steps (for subspace deletion). We explain its key ideas next. Assume for this approximate analysis that $\mathbf{w}_t = 0$ and that $\mathrm{dif}(\hat{\mathbf{P}}_*, \mathbf{P}_*) = 0$ (previous subspace is perfectly estimated). In the k-th projection-PCA step the goal is to bound $\zeta_{\mathrm{new},k} := \mathrm{dif}([\hat{\mathbf{P}}_*, \hat{\mathbf{P}}_{\mathrm{new},k}], \mathbf{P}_{\mathrm{new}})$ conditioned on "accurate recovery so far". Here "accurate recovery so far" means $\mathrm{dif}(\hat{\mathbf{P}}_*, \mathbf{P}_*) \approx 0$ and $\zeta_{\mathrm{new},k-1} \leq \zeta^+_{\mathrm{new},k-1}$. Before $k = 1$, there is no estimate of $\mathbf{P}_{\mathrm{new}}$ and thus we have $\zeta_{\mathrm{new},0} \leq \zeta^+_{\mathrm{new},0} = 1$.

We first use the $\sin\theta$ theorem [95] (Theorem C.1.3) to get a bound on $\zeta_{\mathrm{new},k}$. This is done in Lemma 4.5.33. We then bound the terms in this bound using the matrix Azuma inequality from [96] (Corollaries C.1.13 and C.1.14). This is done in Lemmas 4.5.34, 4.5.35 and 4.5.36. Using the $\sin\theta$ theorem followed by using matrix Azuma for lower bounding $\lambda_{\min}(\frac{1}{\alpha}\sum_t (I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\boldsymbol{\ell}_t\boldsymbol{\ell}_t'(I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}'))$, we can conclude that

$$\begin{aligned}\zeta_{\mathrm{new},k} &\lesssim \frac{\|\mathrm{perturbation}\|_2}{\frac{1}{1-b^2}\lambda^-_{\mathrm{new}} - \epsilon - \|\mathrm{perturbation}\|_2} \\ &\lesssim \frac{2\big\|\frac{1}{\alpha}\sum_t (I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\boldsymbol{\ell}_t\mathbf{e}_t'\big\|_2 + \big\|\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t'\big\|_2}{\frac{1}{1-b^2}\lambda^-_{\mathrm{new}} - \epsilon - (2\big\|\frac{1}{\alpha}\sum_t (I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\boldsymbol{\ell}_t\mathbf{e}_t'\big\|_2 + 2\big\|\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t'\big\|_2)} \end{aligned} \tag{4.5}$$

Here $\mathrm{perturbation} = \frac{1}{\alpha}\sum_t (I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t'(I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}') - \frac{1}{\alpha}\sum_t (I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\boldsymbol{\ell}_t\boldsymbol{\ell}_t'(I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')$. Since $\sum_t (\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t' - \boldsymbol{\ell}_t\boldsymbol{\ell}_t') = \sum_t (\boldsymbol{\ell}_t\mathbf{e}_t' + \mathbf{e}_t\boldsymbol{\ell}_t' + \mathbf{e}_t\mathbf{e}_t')$, the bound used in the second inequality above follows. The next task is to bound the two perturbation terms using the matrix Azuma inequality. This is done in Lemma 4.5.36. As explained in Sec 4.3, under "accurate recovery so far", it can be shown that $\mathbf{e}_t$ satisfies (4.4) and that $\big\|[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\big\|_2 \leq 1.2$. This is proved in Lemma 4.5.25. Notice that, when $\mathbf{w}_t = 0$, $\mathbf{e}_t$ is exactly supported on $\mathcal{T}_t$. Using the expression for $\mathbf{e}_t$, expanding $\boldsymbol{\ell}_t$ in terms of $\boldsymbol{\nu}_\tau$'s, processing as explained in Sec 4.6.1, and applying the matrix Azuma inequality, one can show that, whp,

$$\Big\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} (\mathbf{I} - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*{}')\boldsymbol{\ell}_t\mathbf{e}_t'\Big\|_2 \leq 4\epsilon + \frac{1}{\alpha}\frac{b^2}{1-b^2}(r\gamma^2) + \mathrm{LargeTerm}_k$$

where $t_0 = \hat{t}_j + (k-1)\alpha + 1$ is the first time instant of the k-th projection-PCA interval and

$$\text{LargeTerm}_k := \left\| \frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} \sum_{\tau=t_0}^{t} b^{2t-2\tau} \mathbf{P}_{\text{new}} \mathbf{\Lambda}_{t,\text{new}} \mathbf{P}'_{\text{new}} (\mathbf{I} - \hat{\mathbf{P}}_* \hat{\mathbf{P}}_*{}' - \hat{\mathbf{P}}_{\text{new},k-1} \hat{\mathbf{P}}_{\text{new},k-1}{}') \mathbf{I}_{\mathcal{T}_t} [(\mathbf{\Phi}_t)_{\mathcal{T}_t}{}' (\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}' \right\|_2.$$

In the above, ϵ is very small (comes from applying Azuma for zero-mean terms). The second term is also very small since $1/\alpha \leq (r_{\text{new}}\zeta)^2$. Thus, LargeTerm_k is the only significant term. To bound it, for $k = 1$, we use the fact that $\hat{\mathbf{P}}_{\text{new},k-1} = \hat{\mathbf{P}}_{\text{new},0} = [.]$ and hence $(I - \hat{\mathbf{P}}_* \hat{\mathbf{P}}_*{}' - \hat{\mathbf{P}}_{\text{new},k-1} \hat{\mathbf{P}}_{\text{new},k-1}{}') \mathbf{P}_{\text{new}} \approx \mathbf{P}_{\text{new}}$ and $\mathbf{P}_{\text{new}}$ is dense. From Model 6, $\left\| \mathbf{P}_{\text{new}}{}' \mathbf{I}_{\mathcal{T}_t} \right\|_2 \leq 0.02$. Thus, using $\left\| [(\mathbf{\Phi}_t)_{\mathcal{T}_t}{}' (\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1} \right\|_2 \leq 1.2$ and slow subspace change, (4.1), we get that, for $k = 1$,

$$\begin{aligned} \left\| \frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} (I - \hat{\mathbf{P}}_* \hat{\mathbf{P}}_*{}') \boldsymbol{\ell}_t \mathbf{e}'_t \right\|_2 \lesssim \left\| \text{LargeTerm}_1 \right\|_2 &\leq \frac{1}{1-b^2} 1.2 \cdot 0.02 \cdot \lambda^+_{\text{new}} \\ &\leq \frac{1}{1-b^2} 1.2 \cdot 0.02 \cdot 3\lambda^- = 0.072 \frac{1}{1-b^2} \lambda^-. \end{aligned}$$

For $k > 1$, we cannot show that $(\mathbf{I} - \hat{\mathbf{P}}_* \hat{\mathbf{P}}_*{}' - \hat{\mathbf{P}}_{\text{new},k-1} \hat{\mathbf{P}}_{\text{new},k-1}{}') \mathbf{P}_{\text{new}}$ is dense[3]. Thus we use a different approach. We apply the Cauchy-Schwartz inequality (Lemma C.1.6) with $\boldsymbol{X}_t := \sum_{\tau=t_0}^{t} b^{2t-2\tau} \mathbf{P}_{\text{new}} \mathbf{\Lambda}_{t,\text{new}} \mathbf{P}'_{\text{new}} (I - \hat{\mathbf{P}}_* \hat{\mathbf{P}}_*{}' - \hat{\mathbf{P}}_{\text{new},k-1} \hat{\mathbf{P}}_{\text{new},k-1}{}')$ and $\boldsymbol{Y}_t := \mathbf{I}_{\mathcal{T}_t} [(\mathbf{\Phi}_t)_{\mathcal{T}_t}{}' (\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}'$, followed by using Model 4 on $\mathcal{T}_t$ to bound $\lambda_{\max}(\frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{Y}_t \boldsymbol{Y}'_t)$.

It is easy to see that $\lambda_{\max}(\frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{X}_t \boldsymbol{X}'_t) \leq \max_t \|\boldsymbol{X}_t\|_2^2$ and $\|\boldsymbol{X}_t\|_2 \leq \frac{1}{1-b^2} \lambda^+_{\text{new}} \zeta^+_{\text{new},k-1}$ $\leq 3\zeta^+_{\text{new},k-1} \frac{1}{1-b^2} \lambda^-$.

We bound $\lambda_{\max}(\frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{Y}_t \boldsymbol{Y}'_t)$ by using Model 4 on support change. This is done in Lemma 4.5.22. This lemma exploits the fact that $\frac{1}{\alpha} \sum_t \boldsymbol{Y}_t \boldsymbol{Y}'_t = \frac{1}{\alpha} \sum_t \mathbf{I}_{\mathcal{T}_t} ([(\mathbf{\Phi}_t)_{\mathcal{T}_t}{}' (\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1})^2$ $\mathbf{I}_{\mathcal{T}_t}{}'$ is a block-banded matrix and, for each block, the summation is not over α frames but only over β frames with β being much smaller. For example, if Model 4 holds with $\rho = 1$,

[3] The partial result of [44] assumed that this holds and then used the above approach to get a performance guarantee.

this matrix is block diagonal; if it holds with $\rho = 2$, then it is block-tridiagonal and so on. Thus, using $\|[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \leq 1.2$, we can show that $\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{Y}_t \boldsymbol{Y}_t') \leq \frac{1}{\alpha}\rho^2\beta(1.2)^2 \leq 0.0001 \cdot (1.2)^2$.

By Cauchy-Schwartz and the above bounds, we can conclude that, for $k > 1$,

$$\big\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}(I - \hat{\mathbf{P}}_*\hat{\mathbf{P}}_*')\boldsymbol{\ell}_t\mathbf{e}_t'\big\|_2 \lesssim \big\|\text{LargeTerm}_k\big\|_2 \leq \sqrt{0.0001 \cdot (1.2)^2} \cdot 3 \cdot \zeta^+_{\text{new},k-1}\frac{1}{1-b^2}\lambda^-$$
$$= 0.036 \cdot \zeta^+_{\text{new},k-1}\frac{1}{1-b^2}\lambda^-$$

Using an approach similar to the one outlined above one can also bound the $\mathbf{e}_t\mathbf{e}_t'$ term. This is actually easier to bound because one does not need Cauchy-Schwartz. For $k = 1$, we get

$$\big\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\mathbf{e}_t\mathbf{e}_t'\big\|_2 \lesssim \rho^2\beta(1.2)^2 \cdot 0.02^2\frac{1}{1-b^2}3\lambda^- \leq 0.0001 \cdot 1.44 \cdot 0.02^2 \cdot 3\frac{1}{1-b^2}\lambda^-$$
$$< 0.00002\frac{1}{1-b^2}\lambda^-.$$

and for $k > 1$,

$$\big\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\mathbf{e}_t\mathbf{e}_t'\big\|_2 \lesssim \rho^2\beta(1.2)^2 \cdot (\zeta^+_{\text{new},k-1})^2\frac{1}{1-b^2}3\lambda^- \leq 0.0001 \cdot 1.44 \cdot 3 \cdot (\zeta^+_{\text{new},k-1})^2\frac{1}{1-b^2}\lambda^-$$
$$< 0.075(\zeta^+_{\text{new},k-1})^2\frac{1}{1-b^2}\lambda^-$$

Using the above bounds in (4.5) and using $\lambda^-_{\text{new}} \geq \lambda^-$, we can conclude that,

$$\zeta^+_{\text{new},1} \lesssim 0.19, \quad \zeta^+_{\text{new},k} \lesssim \frac{2 \cdot 0.036 \cdot \zeta^+_{\text{new},k-1} + 0.075(\zeta^+_{\text{new},k-1})^2}{1 - \text{NumeratorTerm}}$$

Here NumeratorTerm refers to the expression from the numerator. From the above, it is easy to see that $\zeta^+_{\text{new},2} \lesssim 0.19$ and, proceeding similarly, $\zeta^+_{\text{new},k} \lesssim 0.19$. Using this to get a loose bound on NumeratorTerm, we can conclude that $\zeta^+_{\text{new},k} \lesssim 0.1\zeta^+_{\text{new},k-1} \leq 0.19 \cdot 0.1^{k-1}$.

The above approximate analysis ignores the fact that range$(\hat{\mathbf{P}}_*) \neq$ range$(\mathbf{P}_*)$. It also ignores the unstructured noise term $\mathbf{w}_t$ and the other small terms that come with each application of matrix Azuma. With incorporating all this, and with using dif$(\hat{\mathbf{P}}_*, \mathbf{P}_*) \leq$

$r\zeta$ (instead of zero), we can conclude that $\zeta_{\text{new},k} \leq \zeta^{+}_{\text{new},k} \leq 0.19 \cdot 0.1^{k-1} + 0.15 r_{\text{new}}\zeta$. By picking K carefully, we get that $\zeta_{\text{new},K} \leq r_{\text{new}}\zeta$ and thus $\text{SE}_t \leq (r + r_{\text{new}})\zeta$ after the K-the projection PCA step.

The analysis of cluster PCA is a significant generalization of the above ideas. The slow subspace change assumption is replaced by the clustering assumption at various places in its proof.

4.4.1 Novelty in proof techniques

This work has two key contributions - it analyzes ReProCS with the deletion step (done via cluster PCA), and it obtains a complete result for ReProCS and ReProCS-cPCA for the case when the $\boldsymbol{\ell}_t$'s are correlated over time.

While the overall proof structure described above is similar to that used in [85], the proof approach for proving the "main lemmas" is quite different for the correlated $\boldsymbol{\ell}_t$'s case. The first such difference is seen in Fact 4.5.26 which shows how to bound $\|(I - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}{}')\boldsymbol{\ell}_t\|_2$ for when $\boldsymbol{\ell}_t$ is correlated over time. This is used to prove Lemma 4.5.25. The second and most significant difference is in proving the matrix-Azuma-based lemmas for projection-PCA and for cluster PCA. These are proved in Sec 4.6 and 4.7. The matrix Azuma inequality [96, Theorem 7.1] is significantly harder to apply than the matrix Hoeffding [96]. There are two reasons for this. First we need to get the sums of conditional expectations of quantities needed to apply this result in a form that can be bounded easily. The simplest way of doing this can lead to loose bounds. To get the desired bounds, we need to rewrite $\boldsymbol{\ell}_t$ in terms of past $\boldsymbol{\nu}_t$'s and use the fact that $b^{\alpha} < (r_{\text{new}}\zeta)$ (is very small) and that $\sum_{\tau=t-\alpha+1}^{t} b^t \leq 1/(1-b) \leq 1/(1-b_0) < 1.12$. In words, the contribution of very old $\boldsymbol{\nu}_t$'s is negligible and the contribution due to the last α $\boldsymbol{\nu}_t$'s is only slightly larger than that of one $\boldsymbol{\nu}_t$.

The third main difference is the analysis of the automatic cluster estimation step and of the cluster PCA algorithm for deleting the subspace. The fact that the former

is correct whp is shown in Lemma 4.5.30. This uses Lemma 4.5.38 and the separation condition from Model 8 to show that, whp, the clusters obtained by using a threshold of $\hat{g}^+$ on the condition numbers of the eigenvalues of the empirical covariance matrix computed with the $\hat{\boldsymbol{\ell}}_t$'s are exactly the same as the true clusters defined in Model 8. The analysis of cluster PCA (Lemma 4.5.31) relies on matrix-Azuma-based Lemmas 4.5.39, 4.5.40, and 4.5.41. These are new too and are proved using a significant generalization of the approach used for analyzing the projection-PCA step.

4.5 Proof of Theorem 4.2.8 And Corollary 4.2.11

We first give the most general denseness assumption and the most general model on $\mathcal{T}_t$ in Sec. 4.5.1 below. Next, we define quantities that will be used in the proofs in Sec. 4.5.2. The basic lemmas that are used several times in the proof are stated next in Sec. 4.5.3. The five main lemmas leading to the proof and the proof itself are given in Sec. 4.5.4. We then give the seven key lemmas that are used to prove the main lemmas in Sec. 4.5.5, followed by the proofs of the main lemmas in Sec. 4.5.6. The proofs of the key lemmas are the long ones and these are given in Sec. 4.6 and 4.7.

4.5.1 Generalizations

Consider the denseness assumption in Model 6. This can be generalized as follows.

Model 9. *For a basis matrix $\boldsymbol{P}$, define the (un)denseness coefficient*

$$\kappa_s(\boldsymbol{P}) := \max_{|\mathcal{T}|\le s} \|\mathbf{I}_{\mathcal{T}}{}'\boldsymbol{P}\|_2$$

Assume that

$$\kappa_{2s,*} := \max_j \kappa_{2s}(\boldsymbol{P}_{t_j}) \le 0.3 \quad \textit{and} \quad \kappa_{2s,\text{new}} := \max_j \kappa_{2s}(\boldsymbol{P}_{t_j,new}) \le 0.02. \tag{4.6}$$

Lemma 4.5.1. *Model 6 is a special case of Model 9.*

Proof. Recall Model 6. For any basis matrix $\mathbf{P}$, $[\kappa_1(\mathbf{P})]^2 = \max_i \|\mathbf{P}'\mathbf{I}_i\|_2^2$. Using the triangle inequality, it is easy to show that $\kappa_s(\boldsymbol{P}) \leq \sqrt{s}\kappa_1(\boldsymbol{P})$ [44]. Using this, the claim follows. □

The proof of Theorem 4.2.8 only uses (4.6) for the denseness assumption.

The reason for defining the (un)denseness coefficient $\kappa_s(\mathbf{P})$ as above is the following lemma from [44].

Lemma 4.5.2 ([44]). *For a basis matrix $\boldsymbol{P}$, $\delta_s(\mathbf{I} - \boldsymbol{P}\boldsymbol{P}') = (\kappa_s(\boldsymbol{P}))^2$.*

Next consider the support change model given in Model 4. This is one special case of the most general model that works for our result. This model was introduced in [85]. We explain it here. What we need to prevent is $\mathcal{T}_t$ occupying the same indices for too many time instants in a given interval. If $\mathcal{T}_t$ does not change 'enough' in a time interval of length α, we will be unable to see enough entries of a given index of $\boldsymbol{\ell}_t$ in order to be able to accurately fill in the missing ones. The following model quantifies 'enough' for our purposes. The number of time instants for which an index is part of $\mathcal{T}_t$ is determined both by how often this set changes, and by how much it moves when it changes. The latter is parameterized by ρ which controls how much the set moves when it changes. For example $\rho = 1$ would require that distinct sets be disjoint, and $\rho = 2$ would mean that at least half of the set is displaced whenever it changes. The parameter $h^+ \in (0, 1)$ represents the maximum fraction of time for which the set remains in a given area in a time interval of length α. The smaller h^+, the more frequently the set will need to change in order to satisfy the model. Our result requires a bound on the product $\rho^2 h^+$.

Model 10. *Let ρ be a positive integer. Split $[1, t_{\max}]$ into intervals of length α. Use $\mathcal{J}_u := [(u-1)\alpha + 1, u\alpha]$ to denote the u-th interval. For a given interval, $\mathcal{J}_u$, let $\mathcal{T}_{(i),u}$ for $i = 1, \ldots, l_u$ be mutually disjoint subsets of $\{1, \ldots, n\}$ and let $\mathcal{J}_{(i),u}, i = 1, 2, \ldots, l_u$ be*

a *partition*[4] *of the interval* $\mathcal{J}_u$ *so that*

$$\text{for all } t \in \mathcal{J}_{(i),u},\ \mathcal{T}_t \subseteq \mathcal{T}_{(i),u} \cup \mathcal{T}_{(i+1),u} \cup \cdots \cup \mathcal{T}_{(i+\rho-1),u} \tag{4.7}$$

Define

$$h_u\left(\alpha; \{\mathcal{T}_{(i),u}\}_{i=1,\ldots,l_u}, \{\mathcal{J}_{(i),u}\}_{i=1,\ldots,l_u}\right) := \max_{i=1,2,\ldots l_u} \left|\mathcal{J}_{(i),u}\right| \tag{4.8}$$

and define $h_u^*(\alpha)$ *as the minimum over all choices of* $\mathcal{T}_{(i),u}$ *and over all choices of the partition* $\mathcal{J}_{(i),u}$.

$$h_u^*(\alpha) := \min_{\substack{\text{all choices of mutually disjoint } \mathcal{T}_{(i),u}, i=1,2,\ldots l_u \\ \text{and all choices of mutually disjoint } \mathcal{J}_{(i),u}, i=1,2,\ldots l_u \\ \text{so that } \cup_{i=1}^{l_u} \mathcal{J}_{(i),u} = \mathcal{J}_u \text{ and (4.7) holds}}} h_u\left(\alpha; \{\mathcal{T}_{(i),u}\}_{i=1,\ldots,l_u}, \{\mathcal{J}_{(i),u}\}_{i=1,\ldots,l_u}\right) \tag{4.9}$$

Assume that $|\mathcal{T}_t| \leq s$ *and that for all* $u = 1, \ldots, \lceil \frac{t_{\max}}{\alpha} \rceil$,

$$h_u^*(\alpha) \leq h^+\alpha \quad \text{with } h^+ \leq \frac{0.0001}{\rho^2}.$$

In the above model, $h_u^*(\alpha)$ provides a bound on how long $\mathcal{T}_t$ remains in a given "area", $\mathcal{T}_{(i),u} \cup \mathcal{T}_{(i+1),u} \cup \cdots \cup \mathcal{T}_{(i+\rho-1),u}$ during the interval $\mathcal{J}_u$, for the best allocation of $\mathcal{T}_t$'s to a given "area" and the best choice of the "areas."

Notice that (4.7) can always be trivially satisfied by choosing $l_u = 1$, $\mathcal{T}_{(1),u} = \{1, \ldots, n\}$ and $\mathcal{J}_{(1),u} = \mathcal{J}_u$, but this will give $h_u(\alpha; .) = \alpha$ and hence is not a good choice. This is why we take a minimum over all choices.

Lemma 4.5.3. *[[85]] Model 4 is a special case of Model 10 above with* $h^+ = \frac{\beta}{\alpha}$.

4.5.2 Definitions

Remark 4.5.4. *Recall that* ϑ *is the maximum number of clusters from Model 8. For ease of notation, henceforth, we will assume that there are* ϑ *clusters for all* j. *If* $\vartheta_j < \vartheta$, *it will just mean that the last* $(\vartheta - \vartheta_j + 1)$ *clusters are empty.*

[4] *i.e. the* $\mathcal{J}_{(i),u}$*'s are mutually disjoint intervals and their union equals* $\mathcal{J}_u$

Definition 4.5.5. *Define* $\mathbf{b}_t := \mathbf{\Phi}_t\mathbf{m}_t - \mathbf{\Phi}_t\mathbf{x}_t = \mathbf{\Phi}_t(\boldsymbol{\ell}_t + \mathbf{w}_t)$. *This is the "noise" seen by the projected sparse recovery step of the algorithm.*

Define $\boldsymbol{e}_t$ *to be the error made in estimating* $\boldsymbol{\ell}_t$. *That is* $\mathbf{e}_t := \boldsymbol{\ell}_t - \hat{\boldsymbol{\ell}}_t$. *Thus, from the algorithm,* $\mathbf{e}_t = (\hat{\mathbf{x}}_t - \mathbf{x}_t) - \mathbf{w}_t$

Definition 4.5.6. *Define the intervals*

$$\mathcal{J}_u := [(u-1)\alpha + 1, u\alpha].$$

Define u_j *to be the* u *such that* $t_j \in \mathcal{J}_u$. *That is* $u_j := \left\lceil \frac{t_j}{\alpha} \right\rceil$. *For the purposes of describing events before the first subspace change, let* $u_0 := 0$.

Define $\hat{u}_j := \frac{\hat{t}_j}{\alpha}$. *Notice from the algorithm that this will be an integer, because detection only occurs when* $t \mod \alpha = 0$. *We will show that, under appropriate conditioning, whp,* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$.

For the cluster-PCA step, define the following intervals for $k = 0, 1, 2, \ldots \vartheta$.

$$\tilde{\mathcal{I}}_{j,k} := [\hat{t}_j + (K+1)\alpha + (k-1)\alpha + 1, \hat{t}_j + (K+1)\alpha + k\alpha]$$

Notice that $\tilde{\mathcal{I}}_{j,0}$ *is where the clusters are determined, and* $\tilde{\mathcal{I}}_{j,k}$ *is where cluster* k *is recovered.*

Definition 4.5.7. *Define* $\boldsymbol{P}_{(j)} := \boldsymbol{P}_{t_j}$,

$$\boldsymbol{P}_{(j),*} := \boldsymbol{P}_{(j-1)} = \boldsymbol{P}_{t_j - 1} \text{ and } \boldsymbol{P}_{(j),new} := \boldsymbol{P}_{t_j,new} \text{ for } j = 1, \ldots, J$$

$$\boldsymbol{a}_{t,*} := \mathbf{P}_{(j),*}{}'\boldsymbol{\nu}_t \text{ and } \boldsymbol{a}_{t,\text{new}} := \mathbf{P}_{(j),\text{new}}{}'\boldsymbol{\nu}_t \text{ for } t \in [t_j, t_{j+1}).$$

Notice that $\boldsymbol{a}_{t,*}$ *is a vector of length* r_{j-1}, *whose last* $(r_{j-1} - r_{j,old})$ *entries are zeroes. Also define*

$$\boldsymbol{P}_{(j),add} := [\boldsymbol{P}_{(j),*}\ \boldsymbol{P}_{(j),new}]$$

Thus, for $t \in [t_j, t_j + d]$, $\boldsymbol{\nu}_t$ can be written as

$$\boldsymbol{\nu}_t = \mathbf{P}_{(j)}\mathbf{a}_t = [\mathbf{P}_{(j),*}\ \mathbf{P}_{(j),\text{new}}] \begin{bmatrix} \boldsymbol{a}_{t,*} \\ \mathbf{a}_{t,\text{new}} \end{bmatrix}$$

and $\text{Cov}(\boldsymbol{\nu}_t) = \boldsymbol{\Sigma}_t$ can be rewritten as

$$\boldsymbol{\Sigma}_t = \mathbf{P}_{(j)}\boldsymbol{\Lambda}_t{\mathbf{P}_{(j)}}' = \begin{bmatrix}\boldsymbol{P}_{(j),*} & \boldsymbol{P}_{(j),\text{new}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_{t,*} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{t,\text{new}}\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_{(j),*}{}' \\ \boldsymbol{P}_{(j),\text{new}}{}'\end{bmatrix}$$

Notice that the last $(r_{j-1} - r_{j,\text{old}})$ diagonal entries of $\boldsymbol{\Lambda}_{t,*}$ are zeroes.

Remark 4.5.8. *From Model 5, $\boldsymbol{P}_{(j),*}$ is orthogonal to $\boldsymbol{P}_{(j),new}$.*

Definition 4.5.9. *For $j = 1, 2, \ldots, J$ and $k = 1, 2, \ldots, K$ define*

1. $\hat{\mathbf{P}}_{(j),*} := \hat{\mathbf{P}}_{\hat{t}_{j-1}+K\alpha+(\vartheta+1)\alpha}$. *If all subspace changes are correctly detected, this is the final estimate of $\boldsymbol{P}_{(j),*} = \boldsymbol{P}_{(j-1)}$ and $\hat{\mathbf{P}}_{(j),*} = \hat{\mathbf{P}}_{t_j-1}$. Let $\hat{\mathbf{P}}_{(1),*} := \hat{\mathbf{P}}_{t_{\text{train}}}$ (the initial estimate).*

2. $\hat{\mathbf{P}}_{(j),new,0} := [.]$ *and* $\hat{\mathbf{P}}_{(j),new,k} := \hat{\mathbf{P}}_{\hat{t}_j+k\alpha,new}$. *This is the k^{th} estimate of $\boldsymbol{P}_{(j),new}$ (again, conditioned on correct change time detection).*

3. $\hat{\mathbf{P}}_{(j),add} := [\hat{\mathbf{P}}_{(j),*}\ \hat{\mathbf{P}}_{(j),new,K}]$ *is the final estimate of $\boldsymbol{P}_{(j),add}$.*

Notice from the algorithm that,

1. $\hat{\mathbf{P}}_{t,*} = \hat{\mathbf{P}}_{(j),*}$ for all $t \in [\hat{t}_{j-1} + K\alpha + (\vartheta+1)\alpha, \hat{t}_j + K\alpha + (\vartheta+1)\alpha - 1]$

2. $\hat{\mathbf{P}}_{t,\text{new}} = \hat{\mathbf{P}}_{(j),\text{new},k-1}$ for all $t \in \mathcal{J}_{\hat{u}_j+k}$ for $k = 1, 2, \ldots K$, $\hat{\mathbf{P}}_{t,\text{new}} = \hat{\mathbf{P}}_{(j),\text{new},K}$ for $t \in [\hat{t}_j + K\alpha, \hat{t}_j + K\alpha + (\vartheta+1)\alpha - 1]$, and $\hat{\mathbf{P}}_{t,\text{new}} = [.]$ at all other times.

3. At all times, $\hat{\mathbf{P}}_t = [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$.

4. $\hat{\mathbf{P}}_{t-1,*} \perp \hat{\mathbf{P}}_{t,\text{new}}$ at $t = \hat{t}_j + k\alpha$ and so $\hat{\mathbf{P}}_{(j),*} \perp \hat{\mathbf{P}}_{(j),\text{new},k}$

Definition 4.5.10. *Define $\boldsymbol{G}_{j,k} := (\mathbf{P}_{t_j})_{\mathcal{G}_{j,k}}$ for $k = 1, 2, \ldots, \vartheta$. The clusters $\mathcal{G}_{j,k}$ were defined in Model 8. Thus $\mathbf{P}_{(j+1),*} = \mathbf{P}_{(j)} = \mathbf{P}_{t_j} = [G_{j,1}, G_{j,2}, \ldots G_{j,\vartheta}]$.*

Recall that $\hat{\mathbf{G}}_{j,k}$ is obtained in the cluster-PCA routine of Algorithm 4. From the definition of $\hat{\mathbf{P}}_{(j),*}$, $\hat{\mathbf{P}}_{(j+1),*} = [\hat{G}_{j,1}, \hat{G}_{j,2}, \ldots \hat{G}_{j,\vartheta}]$.

Definition 4.5.11. *Define*

1. $\zeta_{j,*} := \mathrm{dif}(\hat{\boldsymbol{P}}_{(j),*}, \boldsymbol{P}_{(j),*})$
2. $\zeta_{j,new,k} := \mathrm{dif}([\hat{\mathbf{P}}_{(j),*}\ \hat{\mathbf{P}}_{(j),new,k}], \boldsymbol{P}_{(j),new})$
3. $\zeta_{j,add} := \mathrm{dif}(\hat{\mathbf{P}}_{(j),add}, \boldsymbol{P}_{(j),add})$
4. $\tilde{\zeta}_{j,k} := \mathrm{dif}([\hat{\mathbf{G}}_{j,1} \dots \hat{\mathbf{G}}_{j,k}], \mathbf{G}_{j,k})$.

Using the previous definition, clearly $\zeta_{j+1,*} \leq \sum_{k=1}^{\vartheta} \tilde{\zeta}_{j,k}$.

Definition 4.5.12. *Define*

1. $\zeta^{+}_{j,*} := r\zeta$
2. $\zeta^{+}_{j,new,0} := 1$, $\zeta^{+}_{j,new,k} := \dfrac{b_{\boldsymbol{\mathcal{H}},k}}{b_{\boldsymbol{A}} - b_{\boldsymbol{A},\perp} - b_{\boldsymbol{\mathcal{H}},k}}$ *for* $k = 1, 2, \dots, K$ *where* $b_{\boldsymbol{A}}$, $b_{\boldsymbol{A},\perp}$, *and* $b_{\boldsymbol{\mathcal{H}},k}$ *are defined in Lemmas 4.5.34, 4.5.35, and 4.5.36 respectively. Their expressions use* ϵ *given by (4.14).*
3. $\zeta^{+}_{j,add} := (r + r_{new})\zeta$.
4. $\tilde{\zeta}^{+}_{k} := \dfrac{b_{\tilde{\boldsymbol{\mathcal{H}}},k}}{b_{\tilde{\boldsymbol{A}},k} - b_{\tilde{\boldsymbol{A}},k,\perp} - b_{\tilde{\boldsymbol{\mathcal{H}}},k}}$ *where* $b_{\tilde{\boldsymbol{\mathcal{H}}},k}$, $b_{\tilde{\boldsymbol{A}},k}$, *and* $b_{\tilde{\boldsymbol{A}},k,\perp}$ *are defined in Lemmas 4.5.39, 4.5.40, and 4.5.41 respectively.*

We will show that these are high probability upper bounds on $\zeta_{j,*}$, $\zeta_{j,new,k}$, $\zeta_{j,add}$, *and* $\tilde{\zeta}_{j,k}$ *under appropriate conditioning. We should point out that* $\zeta^{+}_{j,*}$, $\zeta^{+}_{j,add}$, *and* $\zeta^{+}_{j,new,k}$ *do not actually depend on* j. *However, when analyzing Algorithm 4 without the c-PCA step, they do depend on* j.

Definition 4.5.13. *Define the random variable*

$$X_u := \{\{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots \boldsymbol{\nu}_{u\alpha}\}, \{\mathcal{T}_t\}_{t=1,2,\dots t_{\max}}\}.$$

This is the random variable that we condition on (with appropriate choice of u*) when analyzing the subspace update steps - detection or projection-PCA or cluster-PCA.*

Definition 4.5.14. *Recall from Algorithm 4 that*

$$\text{thresh} = \frac{\hat{\lambda}^{-}_{\text{train}}}{2}.$$

Also, recall the definition of $\boldsymbol{\mathcal{D}}_u$ from Algorithm 4. For $j = 1, \ldots, J$, and for $a = u_j$ or $a = u_j + 1$, define the following events

- $\text{DET}^a_j := \{\hat{u}_j = a\}$
- $\text{PPCA}^a_{j,k} := \left\{\hat{u}_j = a \text{ and } rank(\hat{\mathbf{P}}_{(j),new,k}) = r_{j,new} \text{ and } \zeta_{j,new,k} \leq \zeta^+_{j,new,k}\right\}$ *for* $k = 1, \ldots, K$,
- $\text{CLUSTER}^a_j := \left\{\hat{u}_j = a \text{ and } \hat{\mathcal{G}}_{j,k} = \mathcal{G}_{j,k} \text{ for } k = 1, \ldots, \vartheta\right\}$
- $\text{CPCA}^a_{j,k} := \left\{\hat{u}_j = a \text{ and } \tilde{\zeta}_{j,k} \leq \tilde{\zeta}^+_k\right\}$ *for* $k = 1, \ldots, \vartheta$,
- $\text{NODETS}^a_j := \left\{\hat{u}_j = a \text{ and } \lambda_{\max}\left(\frac{1}{\alpha}\boldsymbol{\mathcal{D}}_u \boldsymbol{\mathcal{D}}_u{}'\right) < \text{thresh for all } u \in [\hat{u}_j + K + (\vartheta + 1) + 1, u_{j+1} - 1]\right\}$
- $\Gamma_{0,\text{end}} := \{\zeta_{1,*} \leq r_0\zeta\} \cap \left\{\lambda_{\max}\left(\frac{1}{\alpha}\boldsymbol{\mathcal{D}}_u \boldsymbol{\mathcal{D}}_u{}'\right) < \text{thresh for all } u \in [1, u_1 - 1]\right\}$
- $\Gamma^a_{j,0} := \Gamma_{j-1,\text{end}} \cap \text{DET}^a_j$
- $\Gamma^a_{j,k} := \Gamma^a_{j,k-1} \cap \text{PPCA}^a_{j,k}$ *for* $k = 1, 2, \ldots K$
- $\tilde{\Gamma}^a_{j,0} := \Gamma^a_{j,K} \cap \text{CLUSTER}^a_j$
- $\tilde{\Gamma}^a_{j,k} := \tilde{\Gamma}^a_{j,k-1} \cap \text{CPCA}^a_{j,k}$ *for* $k = 1, 2 \ldots \vartheta$
- $\Gamma_{j,\text{end}} := \left(\tilde{\Gamma}^{u_j}_{j,\vartheta} \cap \text{NODETS}^{u_j}_j\right) \cup \left(\tilde{\Gamma}^{u_j+1}_{j,\vartheta} \cap \text{NODETS}^{u_j+1}_j\right)$

We misuse notation as follows. Suppose that a set Γ is a subset of all possible values that a r.v. X can take. For two r.v.s' $\{X, Y\}$, when we need to say "$X \in \Gamma$ and Y can be anything" we will sometimes misuse notation and just say "$\{X, Y\} \in \Gamma$." For example, we sometimes say $X_{u_j} \in \Gamma_{j,\text{end}}$. This means $X_{u_j - 1} \in \Gamma_{j,\text{end}}$ and $\boldsymbol{a}_t$ for $t \in \mathcal{J}_{u_j}$ are unconstrained.

Definition 4.5.15. *Define*

1. *Let* $\boldsymbol{D}_{j,new} := (\boldsymbol{I} - \hat{\mathbf{P}}_{(j),*}\hat{\mathbf{P}}_{(j),*}{}')\boldsymbol{P}_{(j),new} \overset{QR}{=} \boldsymbol{E}_{j,\text{new}}\boldsymbol{R}_{j,\text{new}}$ *denote its reduced QR decomposition, i.e. let* $\boldsymbol{E}_{j,\text{new}}$ *be a basis matrix for range* $(\boldsymbol{D}_{j,new})$ *and let* $\boldsymbol{R}_{j,\text{new}} = \boldsymbol{E}_{j,\text{new}}{}'\boldsymbol{D}_{j,new}$.

2. *Let* $\boldsymbol{E}_{j,\text{new},\perp}$ *be a basis matrix for the orthogonal complement of range*$(\boldsymbol{E}_{j,\text{new}})$. *To be precise,* $\boldsymbol{E}_{j,\text{new},\perp}$ *is an* $n \times (n - r_j)$ *basis matrix so that* $[\boldsymbol{E}_{j,\text{new}}\ \boldsymbol{E}_{j,\text{new},\perp}]$ *is unitary.*

3. *For* $u = \hat{u}_j + k$ *for* $k = 1, \ldots, K$, *define* $\boldsymbol{A}_u$, $\boldsymbol{A}_{u,\perp}$, $\boldsymbol{\mathcal{A}}_u$ *as*

$$\boldsymbol{A}_u := \frac{1}{\alpha}\sum_{t\in\mathcal{J}_u} \boldsymbol{E}_{j,\text{new}}{}'(\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'(\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')\boldsymbol{E}_{j,\text{new}}$$

$$\boldsymbol{A}_{u,\perp} := \frac{1}{\alpha}\sum_{t\in\mathcal{J}_u} \boldsymbol{E}_{j,\text{new},\perp}{}'(\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'(\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')\boldsymbol{E}_{j,\text{new},\perp}$$

and let

$$\boldsymbol{\mathcal{A}}_u := \begin{bmatrix}\boldsymbol{E}_{j,\text{new}}\ \boldsymbol{E}_{j,\text{new},\perp}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_u & \mathbf{0}\\ \mathbf{0} & \boldsymbol{A}_{u,\perp}\end{bmatrix}\begin{bmatrix}\boldsymbol{E}_{j,\text{new}}{}'\\ \boldsymbol{E}_{j,\text{new},\perp}{}'\end{bmatrix}$$

4. *For* $u = \hat{u}_j + k$ *for* $k = 1, \ldots, K$, *define* $\boldsymbol{\mathcal{M}}_u$ *and* $\boldsymbol{\mathcal{H}}_u$ *as*

$$\mathbf{M}_u = (\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')\left(\frac{1}{\alpha}\sum_{t\in\mathcal{J}_u}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t{}'\right)(\mathbf{I} - \hat{\boldsymbol{P}}_{(j),*}\hat{\boldsymbol{P}}_{(j),*}{}')$$

and

$$\boldsymbol{\mathcal{H}}_u := \boldsymbol{\mathcal{M}}_u - \boldsymbol{\mathcal{A}}_u$$

Remark 4.5.16. *Recall the definition of* $\boldsymbol{\mathcal{D}}_u$ *from Algorithm 4. Conditioned on* $\Gamma_{j,0}^{\hat{u}_j}$, *for* $u = \hat{u}_j + k$, $k = 1, 2, \ldots, K$, $\hat{\mathbf{P}}_{u\alpha-1,*} = \hat{\mathbf{P}}_{(j),*}$ *and thus, for these values of* u

$$\frac{1}{\alpha}\boldsymbol{\mathcal{D}}_u\boldsymbol{\mathcal{D}}_u{}' = \mathbf{M}_u.$$

For these u*'s* $\mathbf{M}_u$ *is the matrix whose eigenvectors with eigenvalue above* thresh *form* $\hat{\mathbf{P}}_{(j),new,k}$ *(see step 3b of Algorithm 4). In other words,* $\mathbf{M}_u$ *has eigendecomposition*

$$\mathbf{M}_u \overset{\text{EVD}}{=} \begin{bmatrix}\hat{\mathbf{P}}_{(j),new,k}\,\hat{\mathbf{P}}_{(j),new,k,\perp}\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{\Lambda}}_u & \mathbf{0}\\ \mathbf{0} & \hat{\boldsymbol{\Lambda}}_{u,\perp}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{P}}_{(j),new,k}{}'\\ \hat{\mathbf{P}}_{(j),new,k,\perp}{}'\end{bmatrix}.$$

Definition 4.5.17. *Define*

1. $\kappa_{s,*} := \max_j \kappa_s(\boldsymbol{P}_{(j),*})$ *and* $\kappa_{s,\text{new}} := \max_j \kappa_s(\boldsymbol{P}_{(j),\text{new}})$.

2. $\kappa^+_{s,*} := 0.3$ *and* $\kappa^+_{s,new} := 0.0215$. *As we will show later in Lemma 4.5.23,* $\kappa^+_{s,new}$ *upper bounds* $\|\mathbf{I}_{\mathcal{T}_t}{}'\boldsymbol{D}_{j,new}\|_2$ *under appropriate conditioning.*

3. $\phi^+ := 1.2$. *As we will show later in Lemma 4.5.25, this upper bounds* $\phi_t := \|[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2$ *under appropriate conditioning.*

Definition 4.5.18. *Define* $\boldsymbol{\Phi}_{(j),0} := (I - \hat{\mathbf{P}}_{(j),*}\hat{\mathbf{P}}_{(j),*}{}')$ *and* $\boldsymbol{\Phi}_{(j),k} := (I - \hat{\mathbf{P}}_{(j),*}\hat{\mathbf{P}}'_{(j),*} - \hat{\mathbf{P}}_{(j),new,k}\hat{\mathbf{P}}_{(j),new,k}{}')$ *for* $k = 1, 2, \ldots K$.

Thus for $t \in [t_j, \hat{t}_j + \alpha]$ *(before the first proj-PCA step),* $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{(j),0}$, *for* $t \in \mathcal{J}_{\hat{u}_j+k}$ *(during interval used for k-th proj-PCA step),* $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{(j),k-1}$, *for* $t \in [\hat{t}_j + K\alpha, \hat{t}_j + K\alpha + (\vartheta+1)\alpha]$ *(after K-th proj-PCA step),* $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{(j),K}$ *and for* $t \in [\hat{t}_j + K\alpha + (\vartheta+1)\alpha, t_{j+1} - 1]$ *(after cluster-PCA step),* $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{(j+1),0}$.

Remark 4.5.19. *The proof uses Model 10 on* $\mathcal{T}_t$. *By Lemma 4.5.3, Model 4 is a special case of it. In particular, this means that (a) Model 4 also implies* $\rho^2 h^+ \leq 0.01$ *and (b) Model 4 also allows us to use the support change lemma, Lemma 4.5.22. This lemma and the sparse recovery lemma, Lemma 4.5.25, are used to get bounds on quantities containing* $\mathbf{e}_t$ *in the proof of Lemma 4.5.36.*

4.5.3 Basic lemmas

Lemma 4.5.20. *Consider Algorithm 4. Under Theorem 4.2.8 assumptions,*

$$\text{dif}(\hat{\mathbf{P}}_{t_{\text{train}}}, \mathbf{P}_{t_{\text{train}}}) \leq r_0\zeta \quad \textit{and}$$
$$0.8\lambda^- \leq \hat{\lambda}^-_{\text{train}} \leq 1.2\lambda^-$$

with probability at least $1 - n^{-10}$.

This lemma follows in a fashion analogous to the proof of the p-PCA lemma, Lemma 4.5.29 (or actually just the proof of Lemma 4.5.34 which is one of the lemmas used to prove Lemma 4.5.29). Its proof is in Appendix C.2.

Lemma 4.5.21. *[Bounds on $b_{\boldsymbol{A}}, b_{\boldsymbol{A},\perp}, b_{\boldsymbol{\mathcal{H}},k}$, $\zeta_{j,new,k}$ and $\tilde{\zeta}_k^+$] Consider the quantities defined in Defnition 4.5.11. Under the conditions of Theorem 4.2.8,*

1. $b_{\boldsymbol{A}} - b_{\boldsymbol{\mathcal{H}},1} \geq 0.8\lambda^- > 0.5\hat{\lambda}^-_{\text{train}} = \text{thresh}$ *and* $b_{\boldsymbol{A},\perp} + b_{\boldsymbol{\mathcal{H}},1} \leq 0.2\lambda^- < 0.35\hat{\lambda}^-_{\text{train}} <$ thresh.

2. $\zeta^+_{new,0} = 1$, $\zeta^+_{new,1} \leq 0.19$, $\zeta^+_{new,k} \leq 0.19 \cdot 0.1^{k-1} + 0.15 r_{new}\zeta$ *for all* $k \geq 1$.

3. $\tilde{\zeta}_k^+ \leq r_{j,k}\zeta$ *where* $r_{j,k} = |\mathcal{G}_{j,k}|$.

This lemma essentially follows using simple algebra. We provide the proof in Appendix C.3. The proof of the second part is similar to that of Lemma 6.14 of [85].

Lemma 4.5.22. *[Support change lemma [85, Lemma 5.3]] Let $s_t = |\mathcal{T}_t|$. Consider a sequence of $s_t \times s_t$ symmetric positive-semidefinite matrices $\boldsymbol{A}_t$ such that $\|\boldsymbol{A}_t\|_2 \leq \sigma^+$ for all t. Assume that the $\mathcal{T}_t$ obey Model 10. Let $\boldsymbol{M} = \sum_{t \in \mathcal{J}_u} \boldsymbol{I}_{\mathcal{T}_t} \boldsymbol{A}_t \boldsymbol{I}_{\mathcal{T}_t}{}'$ be an $n \times n$ matrix ($\mathcal{I}$ is an $n \times n$ identity matrix). Then*

$$\|\boldsymbol{M}\|_2 \leq \rho^2 h^+ \alpha \sigma^+ \leq 0.0001 \sigma^+ \alpha$$

Lemma 4.5.23. *[[85]] Assume that the assumptions of Theorem 4.2.8 hold. Conditioned on $X_{\hat{u}_j+k-1}$, for $X_{\hat{u}_j+k-1} \in \Gamma^{\hat{u}_j}_{j,k-1}$, for $\hat{u}_j = u_j$ or $\hat{u}_j = u_j + 1$,*

$$\|\mathbf{I}_{\mathcal{T}}{}' \boldsymbol{D}_{j,\text{new}}\|_2 \leq \kappa^+_{s,new} := .0215 \tag{4.10}$$

for all $\mathcal{T}$ such that $|\mathcal{T}| \leq s$.

The following summarizes many simple facts.

Fact 4.5.24.

1. *Observe that* $\Gamma^a_{j,0}$ *both for* $a = u_j$ *and* $a = u_j + 1$ *implies that* $u_j \le \hat{u}_j \le u_j + 1$. *Thus, since* $\hat{t}_j = \hat{u}_j\alpha$, *in both cases,* $t_j \le \hat{t}_j \le t_j + 2\alpha$. *So with the model assumption that* $d \ge (K+2)\alpha$, *we have that* $\mathcal{J}_{\hat{u}_j+k} \subseteq [t_j, t_j + d]$ *for* $k = 1, 2, \ldots, K$, *i.e., for all the projection-PCA intervals, (4.1) holds and we can bound* $\|\mathbf{a}_{t,\text{new}}\|_\infty$ *by* γ_{new}.

2. *Since,* $\Gamma^a_{j,K} \subseteq \Gamma^a_{j,0}$, $\Gamma^a_{j,K}$ *also implies that* $t_j \le \hat{t}_j \le t_j + 2\alpha$. *This along with* $d_2 > (\vartheta + 3)\alpha$ *implies that all the intervals used for the cluster-estimation or the cluster-PCA steps are subsets of the interval in which the clustering assumption holds, i.e.,* $[\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta + 1)\alpha] \subseteq [t_j + K\alpha + 1, t_j + K\alpha + d_2]$.

3. *Lemma 4.5.21, item 3, implies that, if* $\tilde{\zeta}_{j,k} \le \tilde{\zeta}^+_k$ *for* $k = 1, \ldots, \vartheta$, *then* $\zeta_{j+1,*} := \text{dif}(\hat{\mathbf{P}}_{(j+1),*}, \boldsymbol{P}_{(j+1),*}) \le \sum_{k=1}^{\vartheta} \tilde{\zeta}_{j,k} \le \sum_{k=1}^{\vartheta} r_{j,k}\zeta = r_j\zeta \le \zeta^+_{j+1,*}$. *This follows by triangle inequality and the fact that* $\hat{\mathbf{P}}_{(j+1),*} = [\hat{G}_{j,1}, \hat{G}_{j,2}, \ldots \hat{G}_{j,\vartheta}]$ *and* $\mathbf{P}_{(j+1),*} = \mathbf{P}_{(j)} = [G_{j,1}, G_{j,2}, \ldots G_{j,\vartheta}]$.

4. *Thus the event* $\Gamma_{j,\text{end}}$ *implies* $\zeta_{j+1,*} \le \zeta^+_{j+1,*}$. *Equivalently,* $\Gamma_{j-1,\text{end}}$ *implies* $\zeta_{j,*} \le \zeta^+_{j,*}$

5. *Thus, the event* $\Gamma^a_{j,0}$ *implies* $\zeta_{j,*} \le \zeta^+_{j,*} = r\zeta$ *for* $a = u_j$ *or* $a = u_{j+1}$.

6. *Thus the event* $\Gamma^a_{j,k-1}$ *also implies this.*

7. *Lemma 4.5.21, item 2, and the choice of* K *in the theorem imply that* $\zeta^+_{j,new,K} \le r_{new}\zeta$.

8. *Using the previous two items, the event* $\Gamma^{\hat{u}_j}_{j,K}$, *both for* $\hat{u}_j = u_j$ *and* $\hat{u}_j = u_j + 1$, *implies that* $\text{dif}(\hat{\mathbf{P}}_{(j),add}, \boldsymbol{P}_{(j),add}) \le \zeta^+_{j,*} + r_{new}\zeta = \zeta^+_{j,add}$.

9. $\frac{1}{\alpha} \le (r_{new}\zeta)^2$. *To see this, observe that the lower bound for* α *has* $(r_{new}\zeta)^2$ *in the denominator, and everything else in the expression is greater than or equal to 1. (Notice that* $\frac{\gamma_{new}{}^2}{\lambda^-} \ge 1$*)*

10. $b^{\alpha} \leq (r_{new}\zeta)$. *This follows because* $b \leq b_0 = 0.1$ *and so* $\frac{-\log(r_{new}\zeta)}{-\log b} \leq \frac{-\log(r_{new}\zeta)}{-\log b_0} = \frac{\log \frac{1}{r_{new}\zeta}}{2.3} \leq \frac{1}{2.3}\frac{1}{r_{new}\zeta} \leq \frac{1}{(r_{new}\zeta)^2} \leq \alpha$.

Lemma 4.5.25 (Sparse Recovery Lemma (similar to [44, Lemma 6.4] and [85])). *Assume that all of the conditions of Theorem 4.2.8 hold. Recall that* $\mathrm{SE}_t = \mathrm{dif}(\hat{\mathbf{P}}_t, \boldsymbol{P}_t)$.

1. *Conditioned on* $\Gamma_{j-1,\mathrm{end}}$, *for* $t \in [t_j, (\hat{u}_j + 1)\alpha]$

   (a) $\phi_t := \|[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \leq \phi^+ := 1.2$.

   (b) *the support of* $\mathbf{x}_t$ *is recovered exactly i.e.* $\hat{\mathcal{T}}_t = \mathcal{T}_t$ *and* $\mathbf{e}_t$ *satisfies:*

$$\mathbf{e}_t := \boldsymbol{\ell}_t - \hat{\boldsymbol{\ell}}_t = (\hat{\boldsymbol{x}}_t - \mathbf{x}_t) - \mathbf{w}_t = \boldsymbol{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\boldsymbol{I}_{\mathcal{T}_t}'\boldsymbol{\Phi}_t(\boldsymbol{\ell}_t + \mathbf{w}_t) - \mathbf{w}_t \tag{4.11}$$

   (c) *Furthermore,*

$$\mathrm{SE}_t \leq 1 \text{ , and}$$
$$\|\boldsymbol{e}_t\|_2 \leq \frac{\phi^+}{1-b}(2\zeta_{,*}^+\sqrt{r}\gamma + \sqrt{r_{new}}\gamma_{\mathrm{new}} + 2\epsilon_w) \leq 1.34\left(2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{\mathrm{new}} + 2\epsilon_w\right)$$

2. *For* $k = 2, 3, \ldots, K$ *and* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$, *conditioned on* $\Gamma_{j,k-1}^{\hat{u}_j}$, *for* $t \in \mathcal{J}_{\hat{u}_j+k} = [(\hat{u}_j + k - 1)\alpha + 1, (\hat{u}_j + k)\alpha]$, *the first two conclusions above hold. That is,* $\phi_t \leq \phi^+$ *and* $\mathbf{e}_t$ *satisfies (4.11). Furthermore,*

$$\mathrm{SE}_t \leq \zeta_{j,*}^+ + \zeta_{j,new,k-1}^+ \text{ , and}$$
$$\|\boldsymbol{e}_t\|_2 \leq \frac{\phi^+}{1-b}(2\zeta_{j,*}^+\sqrt{r}\gamma + \zeta_{j,new,k-1}^+\sqrt{r_{new}}\gamma_{\mathrm{new}} + 2\epsilon_w)$$
$$\leq 1.34\left(2.15\sqrt{\zeta} + 0.19\cdot(0.1)^{k-1}\sqrt{r_{new}}\gamma_{\mathrm{new}} + 2\epsilon_w\right)$$

3. *For* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$, *conditioned on* $\Gamma_{j,K}^{\hat{u}_j}$, *for* $t \in \left[\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta + 1)\alpha\right]$, *the first two conclusions above hold (*$\phi_t \leq \phi^+$ *and* $\mathbf{e}_t$ *satisfies (4.11)). Furthermore,*

$$\mathrm{SE}_t \leq \zeta_{j,add}^+ \text{ , and}$$
$$\|\boldsymbol{e}_t\|_2 \leq \frac{\phi^+}{1-b}(2\zeta_{j,add}^+\sqrt{r}\gamma + 2\epsilon_w) \leq 2.67(\sqrt{\zeta} + \epsilon_w)$$

4. *For $\hat{u}_j = u_j$ or $\hat{u}_j = u_j+1$, conditioned on $\tilde{\Gamma}^{\hat{u}_j}_{j,\vartheta}$, for $t \in \left[\hat{t}_j + K\alpha + (\vartheta+1)\alpha + 1, t_{j+1} - 1\right]$, the first two conclusions above hold ($\phi_t \leq \phi^+$ and $\mathbf{e}_t$ satisfies (4.11)). Furthermore,*

$$\mathrm{SE}_t \leq \zeta^+_{j+1,*} \text{ , and}$$
$$\|\boldsymbol{e}_t\|_2 \leq \frac{\phi^+}{1-b}(2\zeta^+_{j+1,*}\sqrt{r}\gamma + 2\epsilon_w) \leq 2.67(\sqrt{\zeta} + \epsilon_w)$$

Notice that cases 1) and 4) of the above lemma occur when the algorithm is in the detection phase; during the intervals for case 2) the algorithm is performing projection-PCA; during the interval for case 3), the algorithm is performing cluster-PCA. In case 1) new directions have been added but not estimated, so the error, $\mathbf{e}_t$, is the largest. In case 2), the error is decaying exponentially with each estimation step. Case 3) occurs after the new directions have been successfully estimated but the old directions are not deleted yet. Case 4) occurs after the latter has been done too (after cluster-PCA is done). Case 4) contains the smallest error bound, with case 3) bounds being only slightly larger. The proof of this lemma is similar to the proof of Lemma 6.15 of [85]. It is given in Appendix C.4. The main extra fact that we need to use now because the $\boldsymbol{\ell}_t$'s follow an AR model is the following.

Fact 4.5.26. *From Model 5, clearly $\|\boldsymbol{\ell}_t\|_2 \leq \frac{\sqrt{r}\gamma}{1-b}$. Moreover, $\boldsymbol{\ell}_t$ can be expanded as follows.*

$$\boldsymbol{\ell}_t = \boldsymbol{\ell}_{t,small} + \sum_{\tau=t-\alpha+1}^{t} b^{t-\tau}\mathbf{P}_\tau \boldsymbol{a}_\tau \text{ where } \boldsymbol{\ell}_{t,small} := \sum_{\tau=0}^{t-\alpha} b^{t-\tau}\boldsymbol{\nu}_\tau$$

Using the geometric series sum formula, $b^\alpha \leq r_{new}\zeta$, and the bound on ζ from the theorem,

$$\|\boldsymbol{\ell}_{t,small}\|_2 \leq \frac{b^\alpha\sqrt{r}\gamma}{1-b} \leq \frac{r_{new}\zeta\sqrt{r}\gamma}{1-b} \leq \frac{\sqrt{\zeta}}{1-b}$$

For $t \in [t_j, (\hat{u}_j+1)\alpha)$, conditioned on $\Gamma_{j-1,\mathrm{end}}$,

$$\|\boldsymbol{\Phi}_t\boldsymbol{\ell}_t\|_2 = \|\boldsymbol{\Phi}_{(j),0}\boldsymbol{\ell}_t\|_2 \leq \frac{r_{new}\zeta\sqrt{r}\gamma}{1-b} + \frac{1}{1-b}\max_{\tau\in[t-\alpha+1,t]}\|\boldsymbol{\Phi}_0\mathbf{P}_\tau\boldsymbol{a}_\tau\|_2 \leq \frac{2r\zeta\sqrt{r}\gamma + \sqrt{r_{new}}\gamma_{new}}{1-b}$$
$$\leq \frac{2\sqrt{\zeta} + \sqrt{r_{new}}\gamma_{new}}{1-b}$$

For a $t \in \mathcal{J}_{\hat{u}_j+k}$ *for* $k = 2, 3, \ldots K$, *conditioned on* $\Gamma_{j,k-1}^{\hat{u}_j}$, *for* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$,

$$\begin{aligned}\|\mathbf{\Phi}_t \boldsymbol{\ell}_t\|_2 = \|\mathbf{\Phi}_{(j),k-1} \boldsymbol{\ell}_t\|_2 &\leq \frac{r_{new}\zeta\sqrt{r}\gamma}{1-b} + \frac{1}{1-b} \max_{\tau \in [t-\alpha+1,t]} \|\mathbf{\Phi}_{k-1}\mathbf{P}_\tau \boldsymbol{a}_\tau\|_2 \\ &\leq \frac{2r\zeta\sqrt{r}\gamma + \zeta^+_{new,k-1}\sqrt{r_{new}}\gamma_{new}}{1-b}\end{aligned}$$

and the above can further be bounded by $\frac{2\sqrt{\zeta}+\zeta^+_{new,k-1}\sqrt{r_{new}}\gamma_{new}}{1-b}$.

Using $\zeta^+_{new,K} \leq r_{new}\zeta$ *(follows using Lemma 4.5.21 and expression for* K*) and the bound on* ζ, *for* $t \in [\hat{t}_j + K\alpha + 1, \hat{t}_j + K\alpha + (\vartheta+1)\alpha]$, *conditioned on* $\Gamma_{j,K}^{\hat{u}_j}$,

$$\|\mathbf{\Phi}_t \boldsymbol{\ell}_t\|_2 = \|\mathbf{\Phi}_{(j),K} \boldsymbol{\ell}_t\|_2 \leq \frac{(2r\zeta + r_{new}\zeta)\sqrt{r}\gamma}{1-b} \leq \frac{2\sqrt{\zeta}}{1-b}$$

Using Fact 4.5.24, item 3, for $t \in [\hat{t}_j + K\alpha + (\vartheta+1)\alpha + 1, t_{j+1} - 1]$, *conditioned on* $\tilde{\Gamma}_{j,\vartheta}^{\hat{u}_j}$, $\zeta_{j+1,*} \leq \zeta^+_{j+1,*} = r\zeta$ *and so*

$$\|\mathbf{\Phi}_t \boldsymbol{\ell}_t\|_2 = \|\mathbf{\Phi}_{(j+1),0} \boldsymbol{\ell}_t\|_2 \leq \frac{2r\zeta\sqrt{r}\gamma}{1-b} \leq \frac{2\sqrt{\zeta}}{1-b}$$

Recall that $\mathbf{b}_t := \mathbf{\Phi}_t(\boldsymbol{\ell}_t + \mathbf{w}_t)$. *Thus, using the above, we get that* $\|\mathbf{b}_t\|_2 \leq \|\mathbf{\Phi}_t \boldsymbol{\ell}_t\|_2 + \|\mathbf{w}_t\|_2 \leq \xi$ *(*ξ *is set in Theorem 4.2.8).*

4.5.4 Main lemmas for proving Theorem 4.2.8 and proof of Theorem 4.2.8

The first three lemmas below deal with analyzing the addition step. They have statements which are exactly the same as the corresponding lemmas in [85]. But the proofs of the key lemmas needed for proving them are very different since the $\boldsymbol{\ell}_t$'s are now correlated over time. We thus relegate the proofs of these lemmas to the appendix. The proofs of the key lemmas needed for these are given in the main text though. The fourth and the fifth lemma below deal with the deletion step (cluster-PCA) and these are new. These are proved in this section itself.

Lemma 4.5.27 (No false detection of subspace changes)**.** $\Pr\left(\text{NODETS}_j^a \mid \tilde{\Gamma}_{j,\vartheta}^a\right) = 1$ *for* $a = u_j$ *or* $a = u_j + 1$.

Lemma 4.5.28 (Subspace change detected within 2α frames)**.** *For $j = 1, \ldots, J$,*

$$\Pr\left(\mathrm{DET}_j^{u_j+1} \mid \Gamma_{j-1,\mathrm{end}}, \overline{\mathrm{DET}^{u_j}}\right) \geq p_{\mathrm{det},1} := 1 - p_{\boldsymbol{A}} - p_{\boldsymbol{\mathcal{H}}}.$$

The definitions of $p_{\boldsymbol{A}}$ and $p_{\boldsymbol{\mathcal{H}}}$ can be found in the proofs of Lemmas 4.5.34 and 4.5.36 respectively.

Lemma 4.5.29 (k-th iteration of pPCA works well)**.**

$$\Pr\left(\Gamma_{j,k}^a \mid \Gamma_{j,k-1}^a\right) = \Pr\left(\mathrm{PPCA}_{j,k}^a \mid \Gamma_{j,k-1}^a\right) \geq p_{\mathrm{ppca}} := 1 - p_{\boldsymbol{A}} - p_{\boldsymbol{A},\perp} - p_{\boldsymbol{\mathcal{H}}}$$

for $a = u_j$ or $a = u_j + 1$. The definitions of $p_{\boldsymbol{A}}$, $p_{\boldsymbol{A},\perp}$, and $p_{\boldsymbol{\mathcal{H}}}$ can be found in the proofs of Lemmas 4.5.34, 4.5.35, and 4.5.36 respectively.

Lemma 4.5.30 (Clusters are correctly estimated)**.**

$$\Pr\left(\mathrm{CLUSTER}_j^a \mid \Gamma_{j,K}^a\right) \geq p_{\mathrm{cluster}} = 1 - p_{\mathrm{cl}} - p_{\tilde{\boldsymbol{l}}\tilde{\boldsymbol{e}}} - p_{\tilde{\boldsymbol{e}}\tilde{\boldsymbol{e}}}$$

for $a = u_j$ or $a = u_j + 1$. The definition of p_{cl} can be found in the proof of Lemma 4.5.38 and definition of $p_{\tilde{\boldsymbol{l}}\tilde{\boldsymbol{e}}}$, $p_{\tilde{\boldsymbol{e}}\tilde{\boldsymbol{e}}}$ can be found in the proof of Lemma 4.5.41.

Lemma 4.5.31 (Subspaces corresponding to each cluster are correctly estimated)**.**

$$\Pr\left(\mathrm{CPCA}_{j,k}^a \mid \tilde{\Gamma}_{j,k-1}^a\right) \geq p_{\mathrm{cpca}} := 1 - p_{\tilde{\boldsymbol{A}}} - p_{\tilde{\boldsymbol{A}},\perp} - p_{\tilde{\boldsymbol{\mathcal{H}}}}$$

for $a = u_{j+1}$ or $a = u_{j+1} + 1$. The probabilities $p_{\tilde{\boldsymbol{A}}}$, $p_{\tilde{\boldsymbol{A}},\perp}$, $p_{\tilde{\boldsymbol{\mathcal{H}}}}$ are defined in the proofs of Lemmas 4.5.39, 4.5.40, and 4.5.41 respectively.

Using Fact 4.5.24, $\bigcap_{k=1}^{\vartheta} \mathrm{CPCA}_{j,k}^a$ implies that $\zeta_{(j+1),*} \leq \zeta_{(j+1),*}^+ = r\zeta$. Thus, $\Gamma_{j,\mathrm{end}}^a$ also implies this.

Corollary 4.5.32. *Let* $p_{\text{det},0} := \Pr\left(\text{DET}_j^{u_j} \mid \Gamma_{j-1,\text{end}}\right)$. *Combining Lemmas 4.5.27, 4.5.28, 4.5.29, 4.5.30, and 4.5.31 gives*

$$
\begin{aligned}
&\Pr\left(\Gamma_{j,\text{end}} \mid \Gamma_{j-1,\text{end}}\right) \\
&= \Pr\Bigg(\left(\text{DET}_j^{u_j} \bigcap_{k=1}^{K} \text{PPCA}_{j,k}^{u_j} \bigcap \text{CLUSTER}_j^{u_j} \bigcap_{k=1}^{\vartheta} \text{CPCA}_{j,k}^{u_j}\right) \bigcup \\
&\qquad \left(\overline{\text{DET}_j^{u_j}} \cap \text{DET}_j^{u_j+1} \bigcap_{k=1}^{K} \text{PPCA}_{j,k}^{u_j+1} \bigcap \text{CLUSTER}_j^{u_j+1} \bigcap_{k=1}^{\vartheta} \text{CPCA}_{j,k}^{u_j+1}\right) \mid \Gamma_{j-1,\text{end}}\Bigg) \\
&\geq p_{\text{det},0} \cdot (p_{\text{ppca}})^K \cdot (p_{\text{cluster}}) \cdot (p_{\text{cpca}})^\vartheta + (1 - p_{\text{det},0}) \cdot p_{\text{det},1} \cdot (p_{\text{ppca}})^K \cdot (p_{\text{cluster}}) \cdot (p_{\text{cpca}})^\vartheta \\
&\geq p_{\text{det},1} (p_{\text{ppca}})^K \cdot p_{\text{cluster}} (p_{\text{cpca}})^\vartheta
\end{aligned}
$$

Proof of Theorem 4.2.8 and Corollary 4.2.11. Using the fact that $\Gamma_{j-1,\text{end}} \subseteq \Gamma_{j-2,\text{end}} \subseteq \cdots \subseteq \Gamma_{1,\text{end}} \subseteq \Gamma_{0,\text{end}}$, $\Pr(\Gamma_{J,\text{end}}) = \Pr(\Gamma_{0,\text{end}}) \prod_{j=1}^{J} \Pr(\Gamma_{j,\text{end}} \mid \Gamma_{j-1,\text{end}})$.

By Lemma 4.5.20 and the argument used to prove Lemmas 4.5.25 and 4.5.27, we get that $\Pr(\Gamma_{0,\text{end}}) \geq 1 - n^{-10}$. Thus, using Corollary 4.5.32, and the lower bound on α,

$$
\begin{aligned}
\Pr(\Gamma_{J,\text{end}}) &\geq (1 - n^{-10}) \left(p_{\text{det},1} (p_{\text{ppca}})^K \cdot p_{\text{cluster}} (p_{\text{cpca}})^\vartheta\right)^J \\
&\geq (1 - n^{-10})(p_{\text{ppca}})^{(K+1)J} (p_{\text{cluster}} (p_{\text{cpca}})^\vartheta)^J \geq (1 - n^{-10})^3 \geq 1 - 3n^{-10}.
\end{aligned}
$$

By Fact 4.5.24, Lemma 4.5.25, and Lemma 4.5.21, $\Gamma_{J,\text{end}}$ implies that $\hat{\mathcal{T}}_t = \mathcal{T}_t$ for all times t; and that all the bounds on the subspace error SE_t and on $\mathbf{e}_t$ hold. □

4.5.5 Key lemmas needed for proving the main lemmas

The following lemma follows from the $\sin\theta$ theorem [95] (Theorem C.1.3 in Appendix C.1) and Weyl's inequality. It is taken from [44].

Lemma 4.5.33 ([44], Lemma 6.9)**.** *At* $u = \hat{u}_j + k$, *if* $\text{rank}(\hat{\mathbf{P}}_{(j),new,k}) = r_{j,new}$, *and if* $\lambda_{\min}(\boldsymbol{A}_u) - \|\boldsymbol{A}_{u,\perp}\|_2 - \|\boldsymbol{\mathcal{H}}_u\|_2 > 0$, *then*

$$
\zeta_{j,new,k} \leq \frac{\|\boldsymbol{\mathcal{H}}_u\|_2}{\lambda_{\min}(\boldsymbol{A}_u) - \|\boldsymbol{A}_{u,\perp}\|_2 - \|\boldsymbol{\mathcal{H}}_u\|_2}. \tag{4.12}
$$

Similarly, if $\hat{\mathcal{G}}_{j,k} = \mathcal{G}_{j,k}$ *and* $\lambda_{\min}(\tilde{\boldsymbol{A}}_{j,k}) - \|\tilde{\boldsymbol{A}}_{j,k,\perp}\|_2 - \|\tilde{\boldsymbol{\mathcal{H}}}_{j,k}\|_2 > 0$, *then*

$$\tilde{\zeta}_{j,k} \leq \frac{\|\tilde{\boldsymbol{\mathcal{H}}}_{j,k}\|_2}{\lambda_{\min}(\tilde{\boldsymbol{A}}_{j,k}) - \|\tilde{\boldsymbol{A}}_{j,k,\perp}\|_2 - \|\tilde{\boldsymbol{\mathcal{H}}}_{j,k}\|_2} \tag{4.13}$$

The next three lemmas (4.5.34, 4.5.35, and 4.5.36) each assert a high probability bound for one of the terms in (4.12). These, along with Lemma 4.5.33, are used to prove Lemmas 4.5.28 and 4.5.29. The proofs of these lemmas use the matrix Azuma inequalities (Lemmas C.1.12, C.1.13 or C.1.14 in the Appendix) and hence we refer to them as the "addition Azuma" lemmas. Let

$$\epsilon = \frac{1}{1-b^2} 0.001 r_{\text{new}} \zeta \lambda^- \tag{4.14}$$

Lemma 4.5.34. *Define*

$$b_{\boldsymbol{A}} := \frac{1}{1-b^2}\left((1-(\zeta_*^+)^2)\lambda^-_{new} - (r_{new}\zeta)^2 \frac{b^2}{1-b^2}(1-\zeta_*^+)^2\lambda^-_{new}\right) - 4\epsilon$$

For $k = 1, \ldots, K$, *for all* $X_{\hat{u}_j+k-1} \in \Gamma^{\hat{u}_j}_{j,k-1}$ *with* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$,

$$\Pr\left(\lambda_{\min}\left(\boldsymbol{A}_{\hat{u}_j+k}\right) \geq b_{\boldsymbol{A}} \;\middle|\; X_{\hat{u}_j+k-1}\right) \geq 1 - p_{\boldsymbol{A}}$$

where $p_{\boldsymbol{A}}$ *is defined in the proof.*

Lemma 4.5.35. *Define*

$$b_{\boldsymbol{A},\perp} := \frac{1}{1-b^2}(\zeta_*^+)^2\lambda^+ + \frac{0.05(r_{new}\zeta)^2 b^2 \lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

For $k = 1, \ldots, K$, *for all* $X_{\hat{u}_j+k-1} \in \Gamma^{\hat{u}_j}_{j,k-1}$ *with* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$,

$$\Pr\left(\lambda_{\max}\left(\boldsymbol{A}_{\hat{u}_j+k,\perp}\right) \leq b_{\boldsymbol{A},\perp} \;\middle|\; X_{\hat{u}_j+k-1}\right) \geq 1 - p_{\boldsymbol{A},\perp}$$

where $p_{\boldsymbol{A},\perp}$ *is defined in the proof.*

Lemma 4.5.36. *Define*

$$b_{\boldsymbol{\mathcal{H}},k} := 2b_{\boldsymbol{\ell}\mathbf{e},k} + b_{\mathbf{ee},k} + 2b_{\boldsymbol{F_k}}$$

where for $k \geq 2$,

$$b_{\boldsymbol{\ell e},k} := \frac{1}{1-b^2}(\sqrt{\rho^2 h^+}\phi^+(\zeta_*^+)^2\lambda^+ + \sqrt{\rho^2 h^+}\phi^+\zeta_{new,k-1}^+\lambda_{new}^+) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$$

$$b_{\mathbf{ee},k} := \frac{1}{1-b^2}(\rho^2 h^+ \ (\phi^+)^2(\zeta_*^+)^2\lambda^+ + \rho^2 h^+ \ (\phi^+)^2(\zeta_{new,k-1}^+)^2\lambda_{new}^+) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$
$$+ (\phi^+)^2(0.06 r_{new}\zeta\lambda^-) + 8\epsilon$$

$$b_{\boldsymbol{F},k} := \frac{1}{1-b^2}(\zeta_*^+)^2\lambda^+ + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

and for $k = 1$,

$$b_{\boldsymbol{\ell e},1} := \frac{1}{1-b^2}(\sqrt{\rho^2 h^+}\phi^+(\zeta_*^+)^2\lambda^+ + \phi^+\kappa_{s,new}^+\lambda_{new}^+) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$$

$$b_{\mathbf{ee},1} := \frac{1}{1-b^2}(\rho^2 h^+ \ (\phi^+)^2(\zeta_*^+)^2\lambda^+ + \rho^2 h^+ \ (\phi^+)^2(\kappa_{s,new}^+)^2\lambda_{new}^+) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}\rho^2 h^+$$
$$+ (\phi^+)^2(0.06 r_{new}\zeta\lambda^-) + 8\epsilon$$

$$b_{\boldsymbol{F},1} := \frac{1}{1-b^2}(\zeta_*^+)^2\lambda^+ + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

For $k = 1, \ldots, K$, *for all* $X_{\hat{u}_j+k-1} \in \Gamma_{j,k-1}^{\hat{u}_j}$ *with* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$,

$$\Pr\left(\|\boldsymbol{\mathcal{H}}_{\hat{u}_j+k}\|_2 \leq b_{\boldsymbol{\mathcal{H}},k} \;\middle|\; X_{\hat{u}_j+k-1}\right) \geq 1 - p_{\boldsymbol{\mathcal{H}}} \tag{4.15}$$

where $p_{\boldsymbol{\mathcal{H}}} := p_{\boldsymbol{\ell e}} + p_{\mathbf{ee}} + p_{\boldsymbol{F}}$ *and* $p_{\boldsymbol{\ell e}}$, $p_{\mathbf{ee}}$ *and* $p_{\boldsymbol{F}}$ *are defined in the proof.*

Fact 4.5.37. *Using* $\rho^2 h^+ \leq 10^{-4}$, $\frac{\lambda_{new}^+}{\lambda^-} \leq 3$, $\lambda_{new}^- \geq \lambda^-$, $\phi^+ = 1.2$, $\kappa_{s,new}^+ = 0.0215$, $b \leq 0.1$, $\zeta \leq \min\{\frac{10^{-4}}{(r+r_{new})^2}, \frac{0.003\lambda^-}{(r+r_{new})^2\lambda^+}\}$, $\zeta_*^+ = r\zeta$, $\epsilon = \frac{1}{1-b^2}0.001 r_{new}\zeta$,

$$b_{\boldsymbol{A}} \geq \frac{\lambda^-}{1-b^2}(0.9999 - 0.005 r_{new}\zeta)$$
$$b_{\boldsymbol{A},\perp} \leq \frac{0.008 r_{new}\zeta\lambda^-}{1-b^2}$$
$$b_{\boldsymbol{\mathcal{H}},1} \leq \frac{\lambda^-}{1-b^2}(0.156 + 0.1 r_{new}\zeta)$$
$$b_{\boldsymbol{\mathcal{H}},k} \leq \frac{\lambda^-}{1-b^2}(0.073\zeta_{new,k-1}^+ + 0.1 r_{new}\zeta)$$

The following lemma is needed for the proof of Lemma 4.5.30.

Lemma 4.5.38. *Let* $\hat{t}_{\rm cl} := \hat{t}_j + K\alpha + 1$. *Let* $q_2 := 0.05\lambda^-$.

$$\Pr\left(\|\frac{1}{\alpha}\sum_{t=\hat{t}_{cl}}^{\hat{t}_{cl}+\alpha-1} \boldsymbol{\ell}_t\boldsymbol{\ell}_t' - \frac{1}{1-b^2}\Sigma_{(j)}\| \le q_2 \mid X_{\hat{u}_j+K} \right) \ge 1 - p_{\rm cl}.$$

for all $X_{\hat{u}_j+K} \in \Gamma_{j,K}^{\hat{u}_j}$ *for* $\hat{u}_j = u_j$ *or* $u_j + 1$ *In the above,* $p_{\rm cl}$ *is defined in the proof.*

The next three lemmas are needed for the proof of Lemma 4.5.31. The third one below is also used in the proof of Lemma 4.5.30.

Lemma 4.5.39. *Define*

$$b_{\tilde{\boldsymbol{A}},k} := (1-r^2\zeta^2)(1 - \frac{(r_{new}\zeta)^2b^2}{1-b^2})\frac{1}{1-b^2}\lambda_{j,k}^- - 4\epsilon$$

For $j = 1,\ldots,J$ *and* $k = 1,\ldots,\vartheta$, *for* $a = u_j$ *or* $a = u_j + 1$, *for all* $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}_{j,k-1}^a$,

$$\mathbb{P}\left(\lambda_{\min}(\tilde{\boldsymbol{A}}_{j,k}) \ge b_{\tilde{\boldsymbol{A}},k} \mid X_{(\hat{u}_j+K+1)+k-1}\right) > 1 - p_{\tilde{\boldsymbol{A}}}$$

where $p_{\tilde{\boldsymbol{A}}}$ *is defined in the proof.*

Lemma 4.5.40. *Define*

$$b_{\tilde{\boldsymbol{A}},\perp,k} := \frac{1}{1-b^2}(2(r\zeta)^2\lambda^+ + \lambda_{k+1}^+) + \frac{0.05(r_{new}\zeta)b^2}{(1-b^2)(1-b)^2} + 4\epsilon$$

For $j = 1,\ldots,J$ *and* $k = 1,\ldots,\vartheta$, *for* $a = u_j$ *or* $a = u_j + 1$, *for all* $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}_{j,k-1}^a$,

$$\Pr\left(\lambda_{\max}(\tilde{\boldsymbol{A}}_{j,k,\perp}) \le b_{\tilde{\boldsymbol{A}},\perp,k} \mid X_{(\hat{u}_j+K+1)+k-1}\right) > 1 - p_{\tilde{\boldsymbol{A}},\perp,k}$$

where $p_{\tilde{\boldsymbol{A}},\perp,k}$ *is defined in the proof.*

Lemma 4.5.41. *Define*

$$b_{\tilde{\boldsymbol{\mathcal{H}}},k} := 2b_{\tilde{\boldsymbol{\ell}}\boldsymbol{e},k} + b_{\tilde{\boldsymbol{e}}\boldsymbol{e},k} + 2b_{\tilde{\boldsymbol{F}},k}$$

where $b_{\tilde{\boldsymbol{\ell}}\boldsymbol{e},k} := \sqrt{\rho^2h^+(\phi^+)^2}(\frac{1}{1-b^2}(r+r_{new})\zeta((r\zeta)\lambda^+ + \lambda_k^+)) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$

$b_{\tilde{\boldsymbol{e}}\boldsymbol{e},k} := \rho^2h^+\ (\phi^+)^2\frac{1}{1-b^2}(r\zeta)(r+r_{new})\zeta\lambda^+ + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + (\phi^+)^2 2(0.03\zeta\lambda^-) + 8\epsilon$

$b_{\tilde{\boldsymbol{F}},k} := \frac{1}{1-b^2}\left((r\zeta)^2\lambda^+ + \frac{(r\zeta)^2}{\sqrt{1-(r\zeta)^2}}\lambda_{k+1}^+\right) + \frac{0.05(r_{new}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$

For $k = 1, \ldots, k$, for $a = u_j$ or $a = u_j + 1$, for all $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}^a_{j,k-1}$,

$$\Pr\left(\|\tilde{\boldsymbol{\mathcal{H}}}_k\|_2 \le b_{\tilde{\boldsymbol{\mathcal{H}}},k} \;\middle|\; X_{(\hat{u}_j+K+1)+k-1}\right) \ge 1 - p_{\tilde{\boldsymbol{\mathcal{H}}}}$$

where $p_{\tilde{\boldsymbol{\mathcal{H}}}} := p_{\tilde{\boldsymbol{\ell}}\tilde{\boldsymbol{e}}} + p_{\tilde{\boldsymbol{e}}\tilde{\boldsymbol{e}}} + p_{\tilde{\boldsymbol{F}}}$ and $p_{\tilde{\boldsymbol{\ell}}\tilde{\boldsymbol{e}}}$, $p_{\tilde{\boldsymbol{e}}\tilde{\boldsymbol{e}}}$, $p_{\tilde{\boldsymbol{F}}}$ are defined in the proof.

Also, for $a = u_j$ or $a = u_j + 1$, for all $X_{\hat{u}_j+K} \in \Gamma^a_{j,K}$,

$$\Pr\left(2\|\frac{1}{\alpha}\sum_{t=\hat{t}_j+K\alpha+1}^{\hat{t}_j+(K+1)\alpha+1} \boldsymbol{\ell}_t\mathbf{e}_t'\|_2 + \|\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t'\|_2 \le b_{\tilde{\boldsymbol{\mathcal{H}}},1}\middle|\; X_{\hat{u}_j+K}\right) \ge 1 - p_{\tilde{\boldsymbol{\ell}}\tilde{\boldsymbol{e}}} - p_{\tilde{\boldsymbol{e}}\tilde{\boldsymbol{e}}}$$

(This is used in the proof of Lemma 4.5.38. It follows using the exact same approach as that used to bound $\|\tilde{\boldsymbol{\mathcal{H}}}_1\|_2$.)

Fact 4.5.42. *Using $\rho^2 h^+ \le 10^{-4}$, $\frac{\lambda_k^+}{\lambda_k^-} \le g^+ = 3$, $\phi^+ = 1.2$, $\kappa^+_{s,new} = 0.0215$, $b \le 0.1$, $\frac{\lambda^+_{k+1}}{\lambda_k^-} \le \chi^+ = 0.2$, $\zeta \le \min\{\frac{10^{-4}}{(r+r_{new})^2}, \frac{0.003\lambda^-}{(r+r_{new})^2\lambda^+}\}$, $\zeta_*^+ = r\zeta$, $\epsilon = \frac{1}{1-b^2}0.001 r_{new}\zeta$, we have*

$$b_{\tilde{\boldsymbol{A}},k} \ge \frac{\lambda_k^-}{1-b^2}(0.9999 - 0.005 r_{new}\zeta) \tag{4.16}$$

$$b_{\tilde{\boldsymbol{A}},\perp,k} \le \frac{\lambda_k^-}{1-b^2}(0.2 + 0.07 r_{new}\zeta) \tag{4.17}$$

$$b_{\tilde{\boldsymbol{\mathcal{H}}},k} \le \frac{\lambda_k^-}{1-b^2}(0.072(r + r_{new})\zeta + 0.095 r_{new}\zeta) \tag{4.18}$$

4.5.6 Proofs of the main lemmas

Lemmas 4.5.27, 4.5.28, and 4.5.29 are proved in Appendix C.5. These use the first three lemmas from the above subsection.

Proof of Lemma 4.5.30. In this proof, all of the probabilistic statements are conditioned on $X_{\hat{u}_j+K} \in \Gamma^{\hat{u}_j}_{j,K}$ for $\hat{u}_j = u_j$ or $u_j + 1$.

Let $\hat{t}_{\text{cl}} := \hat{t}_j + K\alpha + 1$. Recall from Algorithm 4 that $\hat{\boldsymbol{\Sigma}}_{\text{sample}} := \frac{1}{\alpha}\sum_{t=\hat{t}_{cl}}^{\hat{t}_{cl}+\alpha-1} \hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t'$. Define $\boldsymbol{\Sigma}_{\text{sample}} := \frac{1}{\alpha}\sum_{t=\hat{t}_{cl}}^{\hat{t}_{cl}+\alpha-1} \boldsymbol{\ell}_t\boldsymbol{\ell}_t'$.

By Lemma 4.5.38 and Lemma C.1.11, under the given conditioning, with probability (w.p.) at least $1 - p_{\text{cl}}$,

$$\lambda_{\max}(\mathbf{\Sigma}_{\text{sample}} - \frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) \leq q_2 := 0.05\lambda^- \tag{4.19}$$

Let $k_0 = 0$. Let k_i denote the last index of cluster i. Thus true cluster 1, $\mathcal{G}_{j,1} = \{1, 2, \ldots k_1\}$, true cluster 2, $\mathcal{G}_{j,2} = \{k_1 + 1, k_1 + 2, \ldots k_2\}$ and so on for all $i = 1, 2, \ldots \vartheta_j$. Recall that $\mathbf{\Sigma}_{(j)}$ has rank r_j and so $k_{\vartheta_j} = r_j$.

Consider "true cluster" 1. We need to show that "estimated cluster" 1, $\hat{\mathcal{G}}_{j,1} = \{1, 2, \ldots k_1\}$. Let $\hat{\lambda}_i := \lambda_i\left(\hat{\mathbf{\Sigma}}_{\text{sample}}\right)$. We will be done if we can show that

1. $\frac{\hat{\lambda}_1}{\hat{\lambda}_{k_1}} \leq \hat{g}^+$ and
2. $\frac{\hat{\lambda}_1}{\hat{\lambda}_{k_1+1}} > \hat{g}^+$

Define

$$q := \left\|\mathbf{\Sigma}_{\text{sample}} - \hat{\mathbf{\Sigma}}_{\text{sample}}\right\|_2$$

Using the fact that $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t - \mathbf{e}_t$ we get that

$$q \leq 2\left\|\frac{1}{\alpha}\sum_t \boldsymbol{\ell}_t \mathbf{e}_t{}'\right\| + \left\|\frac{1}{\alpha}\sum_t \mathbf{e}_t \mathbf{e}_t{}'\right\|$$

Using Lemma 4.5.41, Fact 4.5.42 and $(r + r_{\text{new}})\zeta\lambda_k^- \leq (r + r_{\text{new}})\zeta\lambda^+ \leq 0.0003\lambda^-$ (from the bound on ζ), under the given conditioning,

$$\begin{aligned} q &\leq \tfrac{\lambda_k^-}{1-b^2}(0.072(r + r_{\text{new}})\zeta + 0.095 r_{\text{new}}\zeta) \\ &< 0.01\lambda^- \end{aligned}$$

with probability at least $1 - p_{\tilde{\boldsymbol{\ell}}\mathbf{e}} - p_{\tilde{\mathbf{e}}\mathbf{e}}$ where $p_{\tilde{\boldsymbol{\ell}}\mathbf{e}}$, $p_{\tilde{\mathbf{e}}\mathbf{e}}$ are defined in Lemma 4.5.41.

Using Weyl's inequality and (4.19), for $i = 1, \ldots, n$

$$\begin{aligned} \hat{\lambda}_i := \lambda_i(\hat{\mathbf{\Sigma}}_{\text{sample}}) &\leq \lambda_i(\mathbf{\Sigma}_{\text{sample}}) + \lambda_{\max}(\hat{\mathbf{\Sigma}}_{\text{sample}} - \mathbf{\Sigma}_{\text{sample}}) \\ &\leq \lambda_i(\frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) + \lambda_{\max}(\mathbf{\Sigma}_{\text{sample}} - \frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) + q \\ &\leq \lambda_i(\frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) + q_2 + q \end{aligned}$$

and

$$\begin{aligned}\hat{\lambda}_i := \lambda_i(\hat{\mathbf{\Sigma}}_{\text{sample}}) &\geq \lambda_i(\mathbf{\Sigma}_{\text{sample}}) - \lambda_{\max}(\hat{\mathbf{\Sigma}}_{\text{sample}} - \mathbf{\Sigma}_{\text{sample}}) \\ &\geq \lambda_i(\frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) - \lambda_{\max}(\mathbf{\Sigma}_{\text{sample}} - \frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) - q \\ &\geq \lambda_i(\frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}) - q_2 - q\end{aligned}$$

The above strategy to bound $\lambda_i(\hat{\mathbf{\Sigma}}_{\text{sample}})$ was suggested in [98].

Thus, using the the fact that $\frac{\lambda_1(\mathbf{\Lambda}_{(j)})}{\lambda_{k_1}(\mathbf{\Lambda}_{(j)})} \leq g^+ = 3$ and $\lambda_{k_1}(\mathbf{\Lambda}_{(j)}) \geq \lambda^-$, we have that

$$\frac{\hat{\lambda}_1}{\hat{\lambda}_{k_1}} \leq \frac{\frac{1}{1-b^2}\lambda_1(\mathbf{\Lambda}_{(j)}) + q_2 + q}{\frac{1}{1-b^2}\lambda_{k_1}(\mathbf{\Lambda}_{(j)}) - q_2 - q} \leq \frac{g^+ + \frac{(q_2+q)(1-b^2)}{\lambda_{k_1}(\mathbf{\Lambda}_{(j)})}}{1 - \frac{(q_2+q)(1-b^2)}{\lambda_{k_1}(\mathbf{\Lambda}_{(j)})}} \leq \frac{g^+ + \frac{(q_2+q)}{\lambda^-}}{1 - \frac{(q_2+q)}{\lambda^-}} \leq \frac{3 + 0.06}{1 - 0.06} = \hat{g}^+.$$

Similarly, using the lower bound $\frac{\lambda_1(\Lambda_{(j)})}{\lambda_{k_1+1}(\Lambda_{(j)})} \geq \frac{1}{\chi^+} = 5$ from Model 8,

$$\begin{aligned}\frac{\hat{\lambda}_1}{\hat{\lambda}_{k_1+1}} &\geq \frac{\frac{1}{1-b^2}\lambda_1(\mathbf{\Lambda}_{(j)}) - q_2 - q}{\frac{1}{1-b^2}\lambda_{k_1+1}(\mathbf{\Lambda}_{(j)}) + q_2 + q} \geq \frac{\frac{\lambda_1(\mathbf{\Lambda}_{(j)})}{\lambda_{k_1+1}(\mathbf{\Lambda}_{(j)})} - \frac{(q_2+q)(1-b^2)}{\lambda_{k_1+1}(\mathbf{\Lambda}_{(j)})}}{1 + \frac{(q_2+q)(1-b^2)}{\lambda_{k_1+1}(\mathbf{\Lambda}_{(j)})}} \geq \frac{\frac{\lambda_{k_1}(\mathbf{\Lambda}_{(j)})}{\lambda_{k_1+1}(\mathbf{\Lambda}_{(j)})} - \frac{(q_2+q)}{\lambda^-}}{1 + \frac{(q_2+q)}{\lambda^-}} \\ &\geq \frac{\frac{1}{\chi^+} - \frac{(q_2+q)}{\lambda^-}}{1 + \frac{(q_2+q)}{\lambda^-}} \geq \frac{5 - 0.06}{1 + 0.06} = 4.67 > \hat{g}^+\end{aligned}$$

This shows that the first cluster is correctly recovered. Proceeding in the same manner,

$$\frac{\hat{\lambda}_{k_{i-1}+1}}{\hat{\lambda}_{k_i}} \leq \frac{g^+ + \frac{(q_2+q)}{\lambda^-}}{1 - \frac{(q_2+q)}{\lambda^-}} \leq \frac{3 + 0.06}{1 - 0.06} = \hat{g}^+.$$

and

$$\frac{\hat{\lambda}_{k_{i-1}+1}}{\hat{\lambda}_{k_i+1}} \geq \frac{\frac{1}{\chi^+} - \frac{(q_2+q)}{\lambda^-}}{1 + \frac{(q_2+q)}{\lambda^-}} = \frac{5 - 0.06}{1 + 0.06} = 4.67 > \hat{g}^+.$$

Recall that the clustering algorithm excludes all eigenvalues below $0.25\hat{\lambda}^-_{\text{train}}$. Recall also that $\mathbf{\Sigma}_{(j)}$ has rank rank $r_j = k_{\vartheta_j}$. Thus from the upper and lower bounds on $\hat{\lambda}_i$ given above and using Lemma 4.5.20, we can also conclude that,

$$\hat{\lambda}_{k_{\vartheta_j}} \geq \lambda_{k_{\vartheta_j}} - q_2 - q \geq \lambda^- - 0.06\lambda^- > 0.75\lambda^- > 0.25\hat{\lambda}^-_{\text{train}}$$

and

$$\hat{\lambda}_{k_{\vartheta_j}+1} \leq \lambda_{k_{\vartheta_j}+1} + q_2 + q \leq 0 + 0.06\lambda^- < 0.25\hat{\lambda}^-_{\text{train}}$$

Thus, the algorithm also stops at the correct place. We have shown that all of the clusters will be recovered exactly and no extra clusters will be formed (algorithm stops at the correct place). Thus,

$$\Pr\left(\text{CLUSTER}_j^a \mid \Gamma_{j,K}^a\right) \geq p_{\text{cluster}} := 1 - p_{\text{cl}} - p_{\tilde{\boldsymbol{l}}\boldsymbol{e}} - p_{\tilde{\boldsymbol{e}}\boldsymbol{e}}$$

for $a = u_j$ or $a = u_j + 1$. This proves the lemma. □

Proof of Lemma 4.5.31. Since we condition on the event $\Gamma_{j,k-1}^{\hat{u}_j}$ and $\Gamma_{j,k-1}^{\hat{u}_j} \subseteq \text{CLUSTER}_j^{\hat{u}_j}$, the clusters are correctly recovered, i.e. $\hat{\mathcal{G}}_{j,k} = \mathcal{G}_{j,k}$. This lemma then follows by combining Lemma 4.5.33 with the bounds from Lemmas 4.5.39, 4.5.39, 4.5.41 and finally using Lemma C.1.11. □

4.6 Proof Of The Addition Azuma Lemmas

4.6.1 A general decomposition used in all the proofs

A general decomposition will be developed here. We will use this in all the proofs that follow. Consider an interval $\mathcal{J}_u$ and let t_0 denote the first time instant of this interval. Let $X \equiv X_{(t_0-1)/\alpha} = \{\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots \boldsymbol{\nu}_{t_0-1}, \{\mathcal{T}_t\}_{t=1,2,\ldots t_{\max}}\}$. Let $\boldsymbol{M}_t$ and $\boldsymbol{N}_t$ be matrices that are deterministic given X. Consider bounding

$$\frac{1}{\alpha} \sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{N}_t \boldsymbol{\ell}_t \boldsymbol{\ell}_t{}' \boldsymbol{M}_t$$

conditioned on X for $X \in \Gamma$.

From our model, notice that

$$\boldsymbol{\ell}_t = b^{t-t_0+1} \boldsymbol{\ell}_{t_0-1} + \sum_{\tau=t_0}^{t} b^{t-\tau} \boldsymbol{\nu}_\tau$$

Thus,

$$\begin{aligned}
&\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{N}_t \boldsymbol{\ell}_t \boldsymbol{\ell}_t' \boldsymbol{M}_t \\
=&\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{N}_t \left(b^{t-t_0+1}\boldsymbol{\ell}_{t_0-1} + \sum_{\tau=t_0}^{t} b^{t-\tau}\boldsymbol{\nu}_\tau\right)\left(b^{t-t_0+1}\boldsymbol{\ell}_{t_0-1} + \sum_{\tilde{\tau}=t_0}^{t} b^{t-\tilde{\tau}}\boldsymbol{\nu}_{\tilde{\tau}}\right)' \boldsymbol{M}_t \\
:=&\text{term1} + \text{term2} + \text{term3}
\end{aligned}$$

where

$$\begin{aligned}
\text{term1} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} b^{2(t-t_0)+2}\boldsymbol{N}_t(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}')\boldsymbol{M}_t \\
\text{term3} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-t_0-\tau+1}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\ell}_{t_0-1}' + \boldsymbol{\ell}_{t_0-1}\boldsymbol{\nu}_\tau')\boldsymbol{M}_t \\
\text{term2} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{N}_t\left(\sum_{\tau=t_0}^{t} b^{t-\tau}\boldsymbol{\nu}_\tau\right)\left(\sum_{\tilde{\tau}=t_0}^{t} b^{t-\tilde{\tau}}\boldsymbol{\nu}_{\tilde{\tau}}\right)'\boldsymbol{M}_t \\
&:= \text{term21} + \text{term22} + \text{term23 where} \\
\text{term21} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau')\boldsymbol{M}_t \\
\text{term22} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t}\sum_{\tilde{\tau}=t_0}^{\tau-1} b^{2t-\tau-\tilde{\tau}}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}_{\tilde{\tau}}')\boldsymbol{M}_t \\
\text{term23} &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t}\sum_{\tilde{\tau}=\tau+1}^{t} b^{2t-\tau-\tilde{\tau}}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}_{\tilde{\tau}}')\boldsymbol{M}_t
\end{aligned}$$

We will show that term22, term23, term3 are close to zero whp, and that term1 can be bounded by a very small value (proportional to $1/\alpha$). The only non-trivial term is term21 and we will show how to (i) bound its spectral norm whp and, (ii) when $\boldsymbol{N}_t = \boldsymbol{M}_t'$ (so that this term is symmetric), how to also bound its minimum eigenvalue whp. For all terms, except term1 (which is a constant when conditioning on X), we will use the matrix Azuma inequalities (given in Appendix C.1). We first show how to bound the

near-zero terms. Consider term22. By Lemma C.1.8 (exchange order of double sum),

$$\begin{aligned}\text{term22} &= \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0}^{\tau-1} b^{2t-\tau-\tilde{\tau}}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}})\boldsymbol{M}_t \\ &:= \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\boldsymbol{Z}_\tau\end{aligned}$$

To apply matrix Azuma (Lemma C.1.14), we need to bound $\|\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\|_2$ and $\|\boldsymbol{Z}_\tau\|$ conditioned on X. Now,

$$\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X] = \sum_{t=\tau}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0}^{\tau-1} b^{2t-\tau-\tilde{\tau}}\boldsymbol{N}_t\mathbb{E}[\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\boldsymbol{M}_t$$

Consider $\mathbb{E}[\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]$. Notice that here $\tilde{\tau}\leq\tau-1$. Thus, this is a case a of $\mathbb{E}[WY|Z]$ where W is independent of $\{Y,Z\}$ with $W\equiv\boldsymbol{\nu}_\tau$, $Y\equiv\boldsymbol{\nu}_{\tilde{\tau}}$ and $Z\equiv\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\}$. This is true because $\boldsymbol{Z}_\tau$ is a function of $\boldsymbol{\nu}_{t_0},\boldsymbol{\nu}_{t_0+1},\ldots,\boldsymbol{\nu}_\tau$ and thus $\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\} = f(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots\boldsymbol{\nu}_{\tau-1},\{\mathcal{T}_{\tilde{t}}\}_{\tilde{t}=1,2,\ldots,t_{\max}})$. So $\{Y,Z\} = g(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots\boldsymbol{\nu}_{\tau-1},\{\mathcal{T}_{\tilde{t}}\}_{\tilde{t}=1,2,\ldots,t_{\max}})$ and this is independent of $\boldsymbol{\nu}_\tau$ (by the independence assumption from the theorem). Thus, by Lemma C.1.10, since $\boldsymbol{\nu}_\tau$ is zero mean,

$$\mathbb{E}[\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X] = \mathbb{E}[\boldsymbol{\nu}_\tau]\mathbb{E}[\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X] = 0$$

Also,

$$\begin{aligned}\|\boldsymbol{Z}_\tau\|_2 &\leq (\max_\tau\sum_{t=\tau}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0}^{\tau-1} b^{2t-\tau-\tilde{\tau}})\max_{t,\tau,\tilde{\tau}}\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}\boldsymbol{M}_t\|_2 \\ &\leq b_{prob,term22} := \frac{b}{(1-b^2)(1-b)}\max_{t,\tau,\tilde{\tau}}\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}\boldsymbol{M}_t\|_2\end{aligned}$$

Thus by Azuma, conditioned on X, $\|\text{term22}\|_2\leq\epsilon$ w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term22})^2}\right)$.

Consider term23. By Lemma C.1.8 (exchange order of double sum),

$$\text{term23} = \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1}\sum_{\tilde{\tau}=\tau+1}^{t} b^{2t-\tau-\tilde{\tau}}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}})\boldsymbol{M}_t$$

This term is not in a form where we can apply the matrix Azuma inequalities to get a useful bound. But we can get it into a nicer form by a simple change of variables. Let

$p = (t_0 + \alpha - 1) - \tau$ and use this to replace τ. Then,

$$\begin{aligned}\text{term23} &= \frac{1}{\alpha}\sum_{p=0}^{\alpha-1}\sum_{t=t_0+\alpha-1-p}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0+\alpha-p}^{t} b^{2t-t0-\alpha+p-\tilde{\tau}+1}\boldsymbol{N}_t(\boldsymbol{\nu}_{t_0+\alpha-1-p}\boldsymbol{\nu}'_{\tilde{\tau}})\boldsymbol{M}_t \\ &:= \frac{1}{\alpha}\sum_{p=0}^{\alpha-1}\boldsymbol{Z}_p\end{aligned}$$

To apply Azuma (Lemma C.1.14), we need to bound $\|\frac{1}{\alpha}\sum_{p=0}^{\alpha-1}\mathbb{E}[\boldsymbol{Z}_p|\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X]\|_2$ and $\|\boldsymbol{Z}_p\|$ conditioned on X. Now,

$$\begin{aligned}&\mathbb{E}[\boldsymbol{Z}_p|\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X] \\ &= \sum_{t=t_0+\alpha-1-p}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0+\alpha-p}^{t} b^{2t-t_0-\alpha+1-\tilde{\tau}}\boldsymbol{N}_t\mathbb{E}[\boldsymbol{\nu}_{t_0+\alpha-1-p}\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X]\boldsymbol{M}_t\end{aligned}$$

Notice that $\boldsymbol{Z}_p$ is a function of $\boldsymbol{\nu}_{t_0+\alpha-p-1}, \boldsymbol{\nu}_{t_0+\alpha-p}, \ldots, \boldsymbol{\nu}_{t_0+\alpha-1}$. Also recall that $X = \{\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots \boldsymbol{\nu}_{t_0-1}\}$. Thus, $\{\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X\} = f(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots \boldsymbol{\nu}_{t_0-1}, \boldsymbol{\nu}_{t_0+\alpha-p}, \boldsymbol{\nu}_{t_0+\alpha-p+1}, \ldots \boldsymbol{\nu}_{t_0+\alpha-1})$. Notice also that $\tilde{\tau} \geq t_0 + \alpha - p$. Thus, the expectation above is again a case of $\mathbb{E}[WY|Z]$ where W is independent of $\{Y, Z\}$ with $W = \boldsymbol{\nu}_{t_0+\alpha-p-1}$, $Y = \boldsymbol{\nu}_{\tilde{\tau}}$ (for a $\tilde{\tau} \geq t_0 + \alpha - p$) and $Z = \{\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X\} = f(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots \boldsymbol{\nu}_{t_0-1}, \boldsymbol{\nu}_{t_0+\alpha-p}, \boldsymbol{\nu}_{t_0+\alpha-p+1}, \ldots, \boldsymbol{\nu}_{t_0+\alpha-1})$. Using, this, by Lemma C.1.10, $\mathbb{E}[\boldsymbol{\nu}_{t_0+\alpha-1-p}\boldsymbol{\nu}'_{\tilde{\tau}}|\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p-1}, X] = 0$. Also,

$$\begin{aligned}\|\boldsymbol{Z}_p\| &\leq (\max_p \sum_{t=t_0+\alpha-1-p}^{t_0+\alpha-1}\sum_{\tilde{\tau}=t_0+\alpha-p}^{t} b^{2t-t_0-\alpha+1-\tilde{\tau}})\max_{t,p,\tilde{\tau}}\|\boldsymbol{N}_t\boldsymbol{\nu}_{t_0+\alpha-1-p}\boldsymbol{\nu}'_{\tilde{\tau}}\boldsymbol{M}_t\|_2 \\ &\leq b_{prob,term23} := \frac{1}{(1-b)^2}\max_{t,\tau,\tilde{\tau}}\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\nu}'_{\tilde{\tau}}\boldsymbol{M}_t\|_2\end{aligned}$$

Thus by Azuma, conditioned on X, $\|\text{term23}\|_2 \leq \epsilon$ w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term23})^2}\right)$.

Consider term3. By Lemma C.1.8 (exchange order of double sum),

$$\begin{aligned}\text{term3} &= \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1} b^{2t-t_0-\tau+1}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\ell}_{t_0-1}{}' + \boldsymbol{\ell}_{t_0-1}\boldsymbol{\nu}'_\tau)\boldsymbol{M}_t \\ &:= \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\boldsymbol{Z}_\tau\end{aligned}$$

To apply Azuma (Lemma C.1.14), we need to bound $\|\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\|_2$ and $\|\boldsymbol{Z}_\tau\|_2$ conditioned on X. We can show that

$$\begin{aligned}&\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\\&=\sum_{t=\tau}^{t_0+\alpha-1}b^{2t-t_0-\tau+1}\boldsymbol{N}_t\mathbb{E}[(\boldsymbol{\nu}_\tau\boldsymbol{\ell}_{t_0-1}{}'+\boldsymbol{\ell}_{t_0-1}\boldsymbol{\nu}_\tau')|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\boldsymbol{M}_t=0.\end{aligned}$$

This follows because $\boldsymbol{Z}_\tau=f(\boldsymbol{\nu}_\tau,X)$ and thus, $\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\}=\tilde{f}(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{\tau-1})$. Also, $\boldsymbol{\ell}_{t_0-1}=g(X)=\tilde{g}(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{t_0-1})$. Thus, this is again a case of $\mathbb{E}[WY|Z]$ with $W=\boldsymbol{\nu}_\tau$, $Y=\boldsymbol{\ell}_{t_0-1}=\tilde{g}(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{t_0-1})$ and $Z=\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\}=\tilde{f}(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{\tau-1})$.

Also,

$$\|\boldsymbol{Z}_\tau\|\le b_{prob,term3}:=\frac{1}{(1-b^2)}\max_{t,\tau}(\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\ell}_{t_0-1}{}'\boldsymbol{M}_t\|_2+\|\boldsymbol{N}_t\boldsymbol{\ell}_{t_0-1}\boldsymbol{\nu}_\tau'\boldsymbol{M}_t\|_2)$$

Thus by Azuma, conditioned on X, $\|\text{term3}\|_2\le\epsilon$ w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term3})^2}\right)$.

Consider term1. Since $\boldsymbol{\ell}_{t_0-1}=f(X)$ and everything else in this term is also a function of X, this term is a constant given X. Thus we can bound it directly. We have

$$\begin{aligned}\|\text{term1}\|_2&\le\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}b^{2(t-t_0)+2}\max_{t\in[t_0,t_0+\alpha-1]}\|\boldsymbol{N}_t(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}')\boldsymbol{M}_t\|_2\\&\le\frac{b^2}{\alpha(1-b^2)}\max_{t\in[t_0,t_0+\alpha-1]}\|\boldsymbol{N}_t(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}')\boldsymbol{M}_t\|_2\\&\le\frac{(r_{\text{new}}\zeta)^2b^2}{(1-b^2)}\max_{t\in[t_0,t_0+\alpha-1]}\|\boldsymbol{N}_t(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}')\boldsymbol{M}_t\|_2:=b_{term1}\end{aligned}\tag{4.20}$$

Consider term21. By Lemma C.1.8 (exchange summation order),

$$\begin{aligned}\text{term21}&=\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1}b^{2t-2\tau}\boldsymbol{N}_t(\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau')\boldsymbol{M}_t\\&:=\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\boldsymbol{Z}_\tau\end{aligned}$$

To obtain an upper bound on its spectral norm using Azuma, we need to upper bound $\|\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\|_2$ and $\|\boldsymbol{Z}_\tau\|_2$. To get a lower bound on its

minimum eigenvalue we need to lower bound $\lambda_{\min}(\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X])$ as well. We have

$$\begin{aligned}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X] &= \sum_{t=\tau}^{t_0+\alpha-1} b^{2t-2\tau}\boldsymbol{N}_t\mathbb{E}[\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau'|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]\boldsymbol{M}_t \\ &= \sum_{t=\tau}^{t_0+\alpha-1} b^{2t-2\tau}\boldsymbol{N}_t\boldsymbol{\Sigma}_\tau\boldsymbol{M}_t\end{aligned}$$

The last row follows because we condition on a function of $\{\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{\tau-1}\}$, $\boldsymbol{\nu}_\tau$ is independent of all these and $\mathbb{E}[\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau'] = \boldsymbol{\Sigma}_\tau$. Then by applying Lemma C.1.8 in reverse order, we get

$$\begin{aligned}\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X] &:= \frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1} b^{2t-2\tau}\boldsymbol{N}_t\boldsymbol{\Sigma}_\tau\boldsymbol{M}_t \\ &= \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{N}_t\boldsymbol{\Sigma}_\tau\boldsymbol{M}_t\end{aligned}$$

Also,

$$\|\boldsymbol{Z}_\tau\|_2 \leq (\max_\tau \sum_{t=\tau}^{t_0+\alpha-1} b^{2t-2\tau})\max_{t,\tau}\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau'\boldsymbol{M}_t\|_2 \leq b_{prob,term21} := \frac{1}{(1-b^2)}\max_{t,\tau}\|\boldsymbol{N}_t\boldsymbol{\nu}_\tau\boldsymbol{\nu}_\tau'\boldsymbol{M}_t\|_2$$

Thus by Azuma (Lemma C.1.14), conditioned on X,

$$\|\text{term21}\|_2 \leq \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{N}_t\boldsymbol{\Sigma}_\tau\boldsymbol{M}_t\|_2 + \epsilon \tag{4.21}$$

w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term21})^2}\right)$.

Let b_{term21} denote the upper bound on the first term in the RHS of (4.21). Then, conditioned on X,

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\boldsymbol{\ell}_t'\boldsymbol{M}_t\|_2 \leq b_{term1} + b_{term21} + 4\epsilon \tag{4.22}$$

with probability obtained from a union bound.

Consider the special case when $\boldsymbol{N}_t' = \boldsymbol{M}_t$. In this case, $\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\boldsymbol{\ell}_t'\boldsymbol{M}_t$ is a symmetric matrix and so is term21. We can lower bound its minimum eigenvalue using

Azuma Lemma C.1.13 to get that, conditioned on X,

$$\lambda_{\min}(\text{term21}) \geq \lambda_{\min}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{N}_t\boldsymbol{\Sigma}_\tau\boldsymbol{N}_t') - \epsilon \tag{4.23}$$

w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term21})^2}\right)$. Let $b_{lower,term21}$ denote the lower bound on the first term in the RHS of (4.23). Then, conditioned on X, we can conclude that

$$\lambda_{\min}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\boldsymbol{\ell}_t'\boldsymbol{M}_t) \geq b_{lower,term21} - 4\epsilon \tag{4.24}$$

with probability obtained from a union bound. We get the above because of the following reason. Since term1 is symmetric positive semi-definite, $\lambda_{\min}(\text{term1}) \geq 0$. Since term3 is symmetric, $\lambda_{\min}(\text{term3}) \geq -\|\text{term3}\| \geq -\epsilon$. Since term2 is also a symmetric matrix in this case, it follows that term22 + term23 = term2 − term21 is a symmetric matrix. Thus $\lambda_{\min}(\text{term22}+\text{term23}) \geq -\|\text{term22}+\text{term23}\| \geq -\|\text{term22}\| - \|\text{term23}\| \geq -2\epsilon$.

In the special case when $\boldsymbol{N}_t' = \boldsymbol{M}_t = \boldsymbol{M}_0$, using Lemma C.1.9, the RHS in (4.23) can be lower bounded by $\frac{1}{1-b^2}(1-\frac{b^2}{\alpha(1-b^2)})\min_{\tau\in[t_0,t_0+\alpha-1]}\lambda_{\min}(\boldsymbol{M}_0'\boldsymbol{\Sigma}_\tau\boldsymbol{M}_0) - \epsilon$.

In the special case when $\boldsymbol{N}_t = \boldsymbol{N}_0$ and $\boldsymbol{M}_t = \boldsymbol{M}_0$, the RHS in (4.21) can be upper bounded by $\frac{1}{1-b^2}\max_{\tau\in[t_0,t_0+\alpha-1]}\|\boldsymbol{N}_0\boldsymbol{\Sigma}_\tau\boldsymbol{M}_0\|_2 + \epsilon$.

In the special case when $\boldsymbol{N}_t = \boldsymbol{\Phi}_0$ and $\boldsymbol{M}_t = \boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$, we can apply Cauchy-Schwartz for matrices followed by Lemma 4.5.22 (support change lemma) to the RHS of (4.21) to get the final upper bound.

In the special case when $\boldsymbol{N}_t' = \boldsymbol{M}_t = \boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$, we can directly apply Lemma 4.5.22 (support change lemma) to the RHS of (4.21) to get the upper bound.

4.6.2 A general decomposition for terms containing w_t

Consider bounding $\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\mathbf{w}_t'\boldsymbol{M}_t$ conditioned on X. Here X contains $\boldsymbol{\nu}_t$'s for all $t \leq t_0-1$ and contains all the $\mathcal{T}_t$'s. Using the independence assumption from the

theorem,

$$\mathbb{E}[\boldsymbol{Z}_t|\boldsymbol{Z}_{t-1},\boldsymbol{Z}_{t-2},\ldots,\boldsymbol{Z}_{t_0},X]=\mathbb{E}[\boldsymbol{\ell}_t|\boldsymbol{Z}_{t-1},\boldsymbol{Z}_{t-2},\ldots,\boldsymbol{Z}_{t_0},X]\mathbb{E}[\mathbf{w}_t']=0$$

This follows by Lemma C.1.10 with $W\equiv\mathbf{w}_t$, $Y\equiv\boldsymbol{\ell}_t=g(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_t)$ and $Z\equiv\{\boldsymbol{Z}_{t-1},\boldsymbol{Z}_{t-2},\ldots,\boldsymbol{Z}_{t_0},X\}=f(\boldsymbol{w}_{t_0},\boldsymbol{w}_{t_0+1},\ldots,\boldsymbol{w}_{t-1},\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{t-1},\mathcal{T}_{\tilde{\tau}},\tilde{\tau}=1,2,\ldots,t_{\max}\}$ and using the fact that $\mathbf{w}_t$ is zero mean. Also,

$$\|\boldsymbol{N}_t\boldsymbol{\ell}_t\mathbf{w}_t'\boldsymbol{M}_t\|_2\leq b_{prob,\boldsymbol{\ell}_t\mathbf{w}_t}:=\max_t\|\boldsymbol{N}_t\boldsymbol{\ell}_t\mathbf{w}_t'\boldsymbol{M}_t\|_2$$

Thus we can conclude by Azuma Lemma C.1.14 that

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\mathbf{w}_t'\boldsymbol{M}_t\|_2\leq\epsilon$$

w.p. at least $1-(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,\boldsymbol{\ell}_t\mathbf{w}_t})^2}\right)$.

Fact 4.6.1. *In situations where it is not practical to assume that $\mathbf{w}_t$ is independent of $\mathcal{T}_t$, the assumption of Remark 4.2.4 can be used. With this, we can proceed as in Sec. 4.6.1 above. There will be only two terms, $term1=\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_tb^{t-t_0+1}\boldsymbol{\ell}_{t_0-1}\mathbf{w}_t'\boldsymbol{M}_t$ and $term2=\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t}\boldsymbol{N}_tb^{t-\tau}\boldsymbol{\nu}_\tau\mathbf{w}_t'$. We can bound term1 as before by $\frac{(r_{new}\zeta)^2b}{1-b}\sqrt{r}\gamma\epsilon_w\|\boldsymbol{N}_t\|_2\|\boldsymbol{M}_t\|_2$. Everywhere where we use this, $\|\boldsymbol{N}_t\|_2\|\boldsymbol{M}_t\|_2\leq1.2^2=1.44$. With this and with using the bounds on ϵ_w and ζ, this is smaller than $0.001r_{new}\zeta\lambda^-=\epsilon$. By Lemma C.1.8, $term2=\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1}\boldsymbol{N}_tb^{t-\tau}\boldsymbol{\nu}_\tau\mathbf{w}_t':=\frac{1}{\alpha}\sum_{\tau=t_0}^{t_0+\alpha-1}\boldsymbol{Z}_\tau$. Notice that $\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\}=f(\boldsymbol{\nu}_0,\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_{\tau-1},\boldsymbol{w}_{t_0},\boldsymbol{w}_{t_0+1},\ldots,\boldsymbol{w}_{t_0+\alpha-1},\mathcal{T}_{\tilde{\tau}},\tilde{\tau}=1,2,\ldots,t_{\max})$ and so $\mathbb{E}[\boldsymbol{Z}_\tau|\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X]$ is an example of of $\mathbb{E}[WY|Z]$ with W independent of $\{Y,Z\}$ if we let $W=\boldsymbol{\nu}_\tau$, $Y=\mathbf{w}_t$ and $Z=\{\boldsymbol{Z}_{t_0},\boldsymbol{Z}_{t_0+1},\ldots,\boldsymbol{Z}_{\tau-1},X\}$. Hence it is equal to zero. Thus, using Azuma Lemma C.1.14 we can bound term2 by ϵ whp. With this, whenever $\|\boldsymbol{N}_t\|_2\|\boldsymbol{M}_t\|_2\leq1.2^2=1.44$, $\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{N}_t\boldsymbol{\ell}_t\mathbf{w}_t'\boldsymbol{M}_t\|_2<2\epsilon$ (instead of ϵ).*

4.6.3 Proofs of the addition Azuma bounds: Lemmas 4.5.34, 4.5.35, and 4.5.36

We remove the subscript j at various places in this and later sections. Thus, for example, $\boldsymbol{\Phi}_{(j),k-1}$ is replaced by $\boldsymbol{\Phi}_{k-1}$ for $k = 1, 2, \ldots K$.

Definition 4.6.2. *Let* $X \equiv X_{k-1} \equiv X_{\hat{u}_j+k-1}$.

Fact 4.6.3. *Let* $\boldsymbol{D}_{new,k-1} := \boldsymbol{\Phi}_{k-1}P_{new}$ *and* $\boldsymbol{D}_{*,k-1} := \boldsymbol{\Phi}_{k-1}P_{*}$. *Recall that* $\boldsymbol{D}_{new} = \boldsymbol{D}_{new,0} = \boldsymbol{\Phi}_0 P_{new}$. *When* $X_{\hat{u}_j+k-1} \in \Gamma^{a}_{j,k-1}$ *for* $a = u_j$ *or* $a = u_j + 1$,

1. $\|\boldsymbol{D}_{*,k-1}\|_2 \leq \zeta^{+}_{j,*}$ *for* $k = 1, \ldots, K$ *(this follows using Fact 4.5.24).*
2. $\|\boldsymbol{D}_{new,k-1}\|_2 \leq \zeta^{+}_{new,k-1}$ *for* $k = 1, \ldots, K+1$ *(by definition of* $\Gamma^{\hat{u}_j}_{j,k-1}$*).*
3. *Recall that* $\zeta^{+}_{new,0} = 1$.
4. $\lambda_{\min}(\boldsymbol{R}_{new}\boldsymbol{R}_{new}{}') \geq 1-(\zeta^{+}_{*})^2$ *(this follows because* $\|\hat{\mathbf{P}}_{*}{}'\mathbf{P}_{new}\|_2 = \|\hat{\mathbf{P}}_{*}{}'(\mathbf{I}-\mathbf{P}_{*}\mathbf{P}_{*}{}')\mathbf{P}_{new}\|_2 \leq \zeta_{*}$*)*
5. $\boldsymbol{E}_{new}{}'\boldsymbol{D}_{new} = \boldsymbol{E}_{new}{}'\boldsymbol{E}_{new}\boldsymbol{R}_{new} = \boldsymbol{R}_{new}$ *and* $\boldsymbol{E}_{new,\perp}{}'\boldsymbol{D}_{new} = \mathbf{0}$.
6. $\|[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \leq \phi^{+}$ *(using Lemma 4.5.25)*
7. $\mathbf{e}_t$ *satisfies (4.11) with probability one (using Lemma 4.5.25).*

Proof of Lemma 4.5.34. In this proof all probabilistic statements are conditioned on $X_{\hat{u}_j+k-1}$ for $X_{\hat{u}_j+k-1} \in \Gamma^{\hat{u}_j}_{j,k-1}$ for $\hat{u}_j = u_j$ or $u_j + 1$. We need a lower bound on the minimum eigenvalue of $\boldsymbol{A}_u$ for $u = \hat{u}_j + k$ for $k = 1, 2, \ldots, K$ and $\hat{u}_j = u_j$ or $u_j + 1$. For $u = \hat{u}_j + k$, recall that

$$\boldsymbol{A}_u := \frac{1}{\alpha}\sum_{t \in \mathcal{J}_u} \boldsymbol{E}_{\text{new}}{}'\boldsymbol{\Phi}_0\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new}}$$

Let t_0 be the first time instant of $\mathcal{J}_{\hat{u}_j+k}$. We proceed as in Section 4.6.1 with $\boldsymbol{N}'_t = \boldsymbol{M}_t = \boldsymbol{\Phi}_0 \boldsymbol{E}_{j,\text{new}}$. Thus,

$$b_{prob,term2} := \max(b_{prob,term21}, b_{prob,term22}, b_{prob,term23}) \leq \frac{1}{(1-b)^2}(r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

$$b_{prob,term3} \leq \frac{1}{(1-b)^2}\frac{(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})}{1-b}(r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})$$

$$\leq \frac{1}{(1-b)^3}(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Use b_{prob} to denote an upper bound on $\max(b_{prob,term2}, b_{prob,term3})$. Then

$$b_{prob} = \frac{1}{(1-b)^3}(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Using (4.24), (4.23) and Lemma C.1.9,

$$\lambda_{\min}(\boldsymbol{A}_u) \geq \lambda_{\min}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{E}_{\text{new}}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new}}) - 4\epsilon$$

$$\geq \frac{1}{1-b^2}(1 - \frac{b^2}{\alpha(1-b^2)}) \min_{\tau\in[t_0,t_0+\alpha-1]} \lambda_{\min}(\boldsymbol{E}_{\text{new}}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new}}) - 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$. Using Fact 4.6.3, and Ostrowski's theorem, we get

$$\lambda_{\min}(\boldsymbol{E}_{\text{new}}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new}}) \geq \lambda_{\min}(\boldsymbol{R}_{\text{new}}\boldsymbol{\Lambda}_{\tau,\text{new}}\boldsymbol{R}'_{\text{new}}) \geq \lambda_{\min}(\boldsymbol{R}_{\text{new}}\boldsymbol{R}'_{\text{new}})\lambda_{\min}(\boldsymbol{\Lambda}_{\tau,\text{new}})$$

$$\geq (1-(\zeta_*^+)^2)\lambda_{\text{new}}^-$$

Thus, using $1/\alpha \leq (r_{\text{new}}\zeta)^2$,

$$\lambda_{\min}(\boldsymbol{A}_u) \geq (1 - \frac{b^2}{\alpha(1-b^2)})(1-(\zeta_*^+)^2)\frac{\lambda_{\text{new}}^-}{1-b^2} - 4\epsilon$$

$$\geq b_{\boldsymbol{A}} := \frac{1}{1-b^2}\left((1-(\zeta_*^+)^2)\lambda_{\text{new}}^- - (r_{\text{new}}\zeta)^2\frac{b^2}{1-b^2}(1-(\zeta_*^+)^2)\lambda_{\text{new}}^-\right) - 4\epsilon$$

w.p. at least $1 - p_{\boldsymbol{A}}$ with $p_{\boldsymbol{A}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$. □

Proof of Lemma 4.5.35. In this proof all probabilistic statements are conditioned on $X_{\hat{u}_j+k-1}$ for $X_{\hat{u}_j+k-1} \in \Gamma_{j,k-1}^{\hat{u}_j}$ for $\hat{u}_j = u_j$ or u_j+1. We need to upper bound the

maximum eigenvalue of

$$\boldsymbol{A}_{u,\perp} := \frac{1}{\alpha}\sum_{t\in\mathcal{J}_u} \boldsymbol{E}_{\text{new},\perp}{}'\boldsymbol{\Phi}_0\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp}.$$

Let t_0 be the first time instant of $\mathcal{J}_{\hat{u}_j+k}$. We proceed as in Section 4.6.1 with $\boldsymbol{N}'_t = \boldsymbol{M}_t = \boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp}$. Thus,

$$b_{prob,term2} = \max(b_{prob,term21}, b_{prob,term22}, b_{prob,term23}) = \frac{1}{(1-b)^2}(r\zeta)^2 r\gamma^2$$
$$b_{prob,term3} \leq \frac{1}{(1-b)^3}(2r\zeta\sqrt{r}\gamma)^2.$$

Use b_{prob} to denote the upper bound on $\max(b_{prob,term2}, b_{prob,term3})$. Then

$$b_{prob} = \frac{1}{(1-b)^3}(2r\zeta\sqrt{r}\gamma)^2$$

Using (4.22), (4.21) and (4.20)

$$\begin{aligned} b_{term1} &= \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \max_{t\in[t_0,t_0+\alpha-1]} \lambda_{\max}(\boldsymbol{E}_{\text{new},\perp}{}'\boldsymbol{\Phi}_0(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}')\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp}) \\ &\leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)}\frac{(r\gamma^2)}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} \end{aligned}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\begin{aligned} \lambda_{\max}(\boldsymbol{A}_{u,\perp}) &\leq \lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{E}_{\text{new},\perp}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp}) + b_{term1} + 4\epsilon \\ &\leq \frac{1}{1-b^2}\max_{\tau\in[t_0,t_0+\alpha-1]} \lambda_{\max}(\boldsymbol{E}_{\text{new},\perp}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp})) + b_{term1} + 4\epsilon \end{aligned}$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$

Using Fact 4.6.3, $\lambda_{\max}(\boldsymbol{E}_{\text{new},\perp}{}'\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_0\boldsymbol{E}_{\text{new},\perp})) \leq (r\zeta)^2\lambda^+$. Thus,

$$\lambda_{\max}(\boldsymbol{A}_{u,\perp}) \leq \frac{1}{1-b^2}(r\zeta)^2\lambda^+ + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon \leq b_{\boldsymbol{A},\perp}$$

w.p. at least $1 - p_{\boldsymbol{A},\perp}$ with $p_{\boldsymbol{A},\perp} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$ □

Proof of Lemma 4.5.36. In this proof all probabilistic statements are conditioned on $X_{\hat{u}_j+k-1}$ for $X_{\hat{u}_j+k-1} \in \Gamma_{j,k-1}^{\hat{u}_j}$ for $\hat{u}_j = u_j$ or $u_j + 1$. Using the expression for $\boldsymbol{\mathcal{H}}_u$ given in Definition 4.5.15, and noting that for a basis matrix $\boldsymbol{E}$, $\boldsymbol{E}\boldsymbol{E}' + \boldsymbol{E}_\perp \boldsymbol{E}_\perp{}' = \mathcal{I}$ we get that

$$\boldsymbol{\mathcal{H}}_u = \frac{1}{\alpha} \sum_{t \in \mathcal{J}_u} \Big(\boldsymbol{\Phi}_0 \boldsymbol{e}_t \boldsymbol{e}_t{}' \boldsymbol{\Phi}_0 - (\boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \boldsymbol{e}_t{}' \boldsymbol{\Phi}_0 + \boldsymbol{\Phi}_0 \boldsymbol{e}_t \boldsymbol{\ell}_t{}' \boldsymbol{\Phi}_0) + (\boldsymbol{F}_t + \boldsymbol{F}_t{}') \Big)$$

where

$$\boldsymbol{F}_t = \boldsymbol{E}_{\text{new},\perp} \boldsymbol{E}_{\text{new},\perp}{}' \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \boldsymbol{\ell}_t' \boldsymbol{\Phi}_0 \boldsymbol{E}_{\text{new}} \boldsymbol{E}_{\text{new}}{}'.$$

Thus,

$$\|\boldsymbol{\mathcal{H}}_u\|_2 \leq 2 \left\| \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \boldsymbol{e}_t{}' \right\|_2 + \left\| \frac{1}{\alpha} \sum_t \boldsymbol{e}_t \boldsymbol{e}_t{}' \right\|_2 + 2 \left\| \frac{1}{\alpha} \sum_t \boldsymbol{F}_t \right\|_2 \tag{4.25}$$

Next we obtain high probability bounds on each of the three terms on the right hand side of (4.25) using the Azuma corollaries.

The $\boldsymbol{\ell}_t \mathbf{e}_t'$ term. Consider the first term. Using Fact 4.6.3 and the expression for $\mathbf{e}_t$ from (4.11),

$$\begin{aligned}
\frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \boldsymbol{e}_t{}' &= \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t (\boldsymbol{\ell}_t + \mathbf{w}_t)' \boldsymbol{\Phi}_{k-1} \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}' - \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \mathbf{w}_t' \\
&:= \text{term} + \text{termw}, \quad \text{where} \\
\text{term} &:= \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \boldsymbol{\ell}_t' \boldsymbol{\Phi}_{k-1} \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}', \\
\text{termw} &:= \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \mathbf{w}_t' \boldsymbol{\Phi}_{k-1} \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}' - \frac{1}{\alpha} \sum_t \boldsymbol{\Phi}_0 \boldsymbol{\ell}_t \mathbf{w}_t'
\end{aligned}$$

Here we use termw to refer to the sum of all terms containing $\mathbf{w}_t$.

By following the approach of Section 4.6.2, under the given conditioning,

$$\|\text{termw}\|_2 \leq 2\epsilon$$

w.p. at least $1 - 2 \cdot (2n) \exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$ where

$$b_{prob,termw} = (\phi^+) \frac{(2r\gamma\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})\epsilon_w}{1-b}$$

We proceed as in Section 4.6.1 for term. In this case, $\boldsymbol{N}_t = \boldsymbol{\Phi}_0$ and $\boldsymbol{M}_t = \boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$. Thus

$$b_{prob,term2} = \max(b_{prob,term21}, b_{prob,term22}, b_{prob,term23}) \leq \frac{1}{(1-b)^2}\phi^+(\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

$$b_{prob,term3} \leq \frac{1}{(1-b)^3}\phi^+(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Use b_{prob} to denote the upper bound on $\max(b_{prob,term2}, b_{prob,term3})$. Then

$$b_{prob} = \frac{1}{(1-b)^3}\phi^+(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Using (4.20), (4.22) and (4.21),

$$b_{term1} = \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \max_t \|\boldsymbol{\Phi}_0 \boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}'\boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\|_2 \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)}\frac{(r\gamma^2)}{(1-b)^2}$$
$$\leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\|\text{term}\|_2 \leq \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\|_2 + b_{term1} + 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$.

First consider the $k=1$ case. In this case, $\boldsymbol{\Phi}_{k-1} = \boldsymbol{\Phi}_0$. By Lemma 4.5.23, under the given conditioning, $\|\mathbf{P}_{\text{new}}'\boldsymbol{\Phi}_0\mathbf{I}_{\mathcal{T}_t}\|_2 = \|\mathbf{I}_{\mathcal{T}_t}'\boldsymbol{\Phi}_0\mathbf{P}_{\text{new}}\|_2 \leq \kappa_{s,\text{new}}^+ = 0.0215$. Using this and Fact 4.6.3,

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\|_2 \leq \frac{1}{1-b^2}\phi^+((\zeta_*^+)^2\lambda^+ + \kappa_{s,\text{new}}^+\lambda_{\text{new}}^+)$$

and so for $k=1$,

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{\Phi}_0\boldsymbol{\ell}_t\boldsymbol{e}_t'\|_2 \leq \frac{1}{1-b^2}\phi^+((\zeta_*^+)^2\lambda^+ + \kappa_{s,\text{new}}^+\lambda_{\text{new}}^+) + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$$

w.p. at least $1 - p_{\boldsymbol{\ell}\mathbf{e}}$ with $p_{\boldsymbol{\ell}\mathbf{e}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right) + 2\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$.

For $k > 1$, we cannot use Lemma 4.5.23. Thus, we follow a different approach - we use Lemma C.1.6 (Cauchy-Schwartz for sums of matrices) followed by Lemma 4.5.22 (support change lemma). Let $\boldsymbol{X}_t := \sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}$ and $\boldsymbol{Y}_t := \mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}{}'$. Then by Lemma C.1.6 (Cauchy-Schwartz),

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}{}'\|_2$$
$$\leq \sqrt{\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{X}_t\boldsymbol{X}_t')\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{Y}_t\boldsymbol{Y}_t')}$$

Now,

$$\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{X}_t\boldsymbol{X}_t') \leq \max_t\|\boldsymbol{X}_t\|^2 \leq \left(\sum_{\tau=t_0}^{t} b^{2t-2\tau}\max_\tau\|\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}\|_2\right)^2$$
$$\leq \left(\frac{1}{1-b^2}((\zeta_*^+)^2\lambda^+ + \zeta_{\text{new},k-1}^+\lambda_{\text{new}}^+)\right)^2$$

By Lemma 4.5.22 (support change lemma)

$$\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{Y}_t\boldsymbol{Y}_t') \leq \rho^2 h^+(\phi^+)^2,$$

Thus,

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Phi}_0\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}{}'\|_2$$
$$\leq \sqrt{\rho^2 h^+(\phi^+)^2}\left(\frac{1}{1-b^2}((\zeta_*^+)^2\lambda^+ + \zeta_{\text{new},k-1}^+\lambda_{\text{new}}^+)\right)$$

and so for $k > 1$,

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{\Phi}_0\boldsymbol{\ell}_t\boldsymbol{e}_t{}'\|_2 \leq \sqrt{\rho^2 h^+(\phi^+)^2}\frac{1}{1-b^2}\left((\zeta_*^+)^2\lambda^+ + \zeta_{\text{new},k-1}^+\lambda_{\text{new}}^+\right) + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$$

with probability at least $1 - p_{\boldsymbol{\ell e}}$, where

$$p_{\boldsymbol{\ell e}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right) + 2\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right).$$

The $\mathbf{e}_t\mathbf{e}_t'$ term. Consider the second term. Using Fact 4.6.3 and the expression for $\mathbf{e}_t$ from (4.11),

$$\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t' = \text{term} + \text{termw}, \text{ where}$$

$$\text{term} := \frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_{k-1}(\boldsymbol{\ell}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$$

$$\begin{aligned}\text{termw} := &\frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_{k-1}(-\mathbf{w}_t\mathbf{w}_t' - \boldsymbol{\ell}_t\mathbf{w}_t') + \mathbf{w}_t\mathbf{w}_t' + \\ &\frac{1}{\alpha}\sum_t(-\mathbf{w}_t\mathbf{w}_t' - \mathbf{w}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}' + \\ &\frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_{k-1}(\boldsymbol{\ell}_t\mathbf{w}_t' + \mathbf{w}_t\mathbf{w}_t' + \mathbf{w}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\end{aligned}$$

Consider the $\mathbf{w}_t\mathbf{w}_t'$ part of termw. Let $\boldsymbol{N}_t = \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_{k-1}$. Using Lemma 4.5.22 (support change lemma), the bound on ϵ_w^2 and Lemma C.1.6 (Cauchy-Schwartz),

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\mathbf{w}_t\mathbf{w}_t'\|_2 \le \sqrt{\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\boldsymbol{N}_t'\|_2\|\frac{1}{\alpha}\sum_t \mathbf{w}_t\mathbf{w}_t'\mathbf{w}_t\mathbf{w}_t'\|_2} \le \sqrt{\rho^2 h^+(\phi^+)^2}\epsilon_w^2$$

Using Lemma 4.5.22 (support change lemma), we have

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\mathbf{w}_t\mathbf{w}_t'\boldsymbol{N}_t'\|_2 \le \rho^2 h^+(\phi^+)^2\epsilon_w^2$$

The $\boldsymbol{\ell}_t\mathbf{w}_t'$ in termw can be bounded by ϵ using the approach of Section 4.6.2. Thus, using the bound on ϵ_w^2 from the theorem,

$$\|\text{termw}\|_2 \le (1 + 2\sqrt{\rho^2 h^+}\phi^+ + 2\rho^2 h^+(\phi^+)^2)(0.03\zeta\lambda^-) + 4\epsilon \le 2(\phi^+)^2(0.03\zeta\lambda^-) + 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$. Here

$$b_{prob,termw} = (\phi^+)^2\frac{(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})\epsilon_w}{1-b}.$$

For term, we proceed as in Section 4.6.1 with $\boldsymbol{N}_t' = \boldsymbol{M}_t = \mathbf{\Phi}_{k-1}\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$. Thus,

$$b_{prob,term2} = \max(b_{prob,term21}, b_{prob,term22}, b_{prob,term23}) \le \frac{1}{(1-b)^2}(\phi^+)^2(\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

$$b_{prob,term3} \le \frac{1}{(1-b)^3}(\phi^+)^2(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Use b_{prob} to denote the upper bound on $\max(b_{prob,term2}, b_{prob,term3})$. Then

$$b_{prob} = \frac{1}{(1-b)^3}(\phi^+)^2(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Using (4.20), (4.22), (4.21), we get

$$b_{term1} \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \frac{r\gamma^2}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\|\text{term}\|_2 \leq \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \mathbf{I}_{\mathcal{T}_t}\left(\sum_{\tau=t_0}^{t} b^{2t-2\tau}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K\mathbf{\Sigma}_\tau\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\right)\mathbf{I}_{\mathcal{T}_t}'\|_2$$
$$+ b_{term1} + 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$. By using Lemma 4.5.22 (support change lemma) and Lemma 4.5.23 for $k=1$ and by using only Lemma 4.5.22 (support change lemma) for $k>1$, we get

$$\|\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t'\|_2 \leq b_{\mathbf{ee}}$$

w.p. at least $1 - p_{\mathbf{ee}}$ with $p_{\mathbf{ee}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right) + 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$.

The $\boldsymbol{F}_t$ term. Consider the smallest term, $\left\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\right\|_2 = \|\boldsymbol{E}_{\text{new},\perp}\boldsymbol{E}_{\text{new},\perp}'\mathbf{\Phi}_0\boldsymbol{\ell}_t\boldsymbol{\ell}_t'\mathbf{\Phi}_0\boldsymbol{E}_{\text{new}}\boldsymbol{E}_{\text{new}}'\|_2$. We again proceed as in Section 4.6.1. In this case $\boldsymbol{N}_t = \boldsymbol{E}_{\text{new},\perp}\boldsymbol{E}_{\text{new},\perp}'\mathbf{\Phi}_0$ and $\boldsymbol{M}_t = \mathbf{\Phi}_0\boldsymbol{E}_{\text{new}}\boldsymbol{E}_{\text{new}}'$. Thus,

$$b_{prob,term2} \leq \frac{1}{(1-b)^2}(\zeta_*^+\sqrt{r}\gamma)(2\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})$$
$$b_{prob,term3} \leq \frac{1}{(1-b)^3}(\zeta_*^+\sqrt{r}\gamma)(2\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})$$

and so

$$b_{prob} = \frac{1}{(1-b)^3}(\zeta_*^+\sqrt{r}\gamma)(2\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})$$
$$\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\|_2 \leq \frac{1}{1-b^2}(\zeta_*^+)^2\lambda^+ + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

w.p. at least $1 - p_{\boldsymbol{F}}$ with $p_{\boldsymbol{F}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$. Combining the bounds on the three terms on the RHS of (4.25) we get the final result of this lemma. □

4.7 Proof Of Deletion Azuma Lemmas - Lemma 4.5.38 And Lemmas 4.5.39, 4.5.40, 4.5.41

4.7.1 Proof of Lemma 4.5.38

Proof of Lemma 4.5.38. In this proof, all of the probabilistic statements are conditioned on $X_{\hat{u}_j+K}$ for $X_{\hat{u}_j+K} \in \Gamma_{j,K}^{\hat{u}_j}$ for $\hat{u}_j = u_j$ or $u_j + 1$.

Let $t_0 := \hat{t}_{cl}$. Using Fact 4.5.24, under the given conditioning, for all $t \in [t_0, t_0+\alpha-1]$,

$$\mathbb{E}[\boldsymbol{\nu}_t \boldsymbol{\nu}_t'] = \boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{(j)} := \mathbf{P}_{(j)} \boldsymbol{\Lambda}_{(j)} \mathbf{P}_{(j)}{}' \tag{4.26}$$

We need to bound $f = \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \boldsymbol{\ell}_t \boldsymbol{\ell}_t{}' - \frac{1}{1-b^2}\boldsymbol{\Sigma}_{(j)}\|_2$. Let

$$\epsilon := \frac{1}{1-b^2} 0.001 r_{\text{new}} \zeta \lambda^-$$

To do this we can proceed as in Section 4.6.1 with $\boldsymbol{N}_t = \boldsymbol{M}_t' = \mathcal{I}$ but with one change. We include the constant term $-\frac{1}{1-b^2}\boldsymbol{\Sigma}_{(j)}$ in term21. Thus,

$$b_{prob,term2} \leq \frac{2}{(1-b)^2}(\sqrt{r}\gamma)^2$$
$$b_{prob,term3} \leq \frac{1}{(1-b)^3}(\sqrt{r}\gamma)^2$$

Let

$$f_{term21} := \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Sigma}_\tau - \frac{1}{1-b^2}\boldsymbol{\Sigma}_{(j)}\|_2.$$

Using (4.20), (4.22), (4.21), we get

$$b_{term1} = \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \max_t \|\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}'\|_2 \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \frac{(r\gamma^2)}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta) b^2 \lambda^-}{(1-b^2)(1-b)^2}$$

and

$$f \leq f_{term21} + b_{term1} + 4\epsilon$$

w.p. at least $1 - 3\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term2})^2}\right) - (2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term3})^2}\right)$.

Since $\mathbf{\Sigma}_\tau = \mathbf{\Sigma}_{(j)} := \mathbf{P}_{(j)}\mathbf{\Lambda}_{(j)}\mathbf{P}_{(j)}'$ for this interval, using Lemma C.1.9 and using the bound on $1/\alpha$ from Fact 4.5.24,

$$\begin{aligned} f_{term21} &= \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\mathbf{\Sigma}_{(j)} - \frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}\|_2 \\ &= \|\frac{1}{1-b^2}(1-\frac{1}{\alpha}\frac{b^2(1-b^{2\alpha})}{1-b^2})\mathbf{\Sigma}_{(j)} - \frac{1}{1-b^2}\mathbf{\Sigma}_{(j)}\|_2 \\ &\leq \frac{1}{\alpha}\frac{b^2}{(1-b^2)^2}\|\mathbf{\Sigma}_{(j)}\|_2 \leq (r_{\text{new}}\zeta)^2\frac{b^2}{(1-b^2)^2}\lambda^+ \leq (r_{\text{new}}\zeta)\frac{b^2}{(1-b^2)^2}0.05\lambda^- \end{aligned}$$

Thus,

$$f \leq (r_{\text{new}}\zeta)\frac{b^2}{(1-b^2)^2}0.05\lambda^- + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon \leq q_2$$

w.p. at least $1-p_{\text{cl}}$ where $p_{\text{cl}} := 4\cdot(2n)\exp\left(-\frac{\alpha\epsilon^2(1-b)^6}{32\cdot 4r^2\gamma^4}\right)$. □

4.7.2 Definitions and preliminaries for proofs of Lemmas 4.5.39, 4.5.40, 4.5.41

Definition 4.7.1. *Define*

1. $\boldsymbol{G}_{j,1,\text{det}} := [.]$ *and for* $k = 2, 3, \ldots \vartheta$, $\boldsymbol{G}_{j,k,\text{det}} := [\boldsymbol{G}_{j,1}\ \boldsymbol{G}_{j,2}\ \ldots\ \boldsymbol{G}_{j,k-1}]$.
2. *Define* $\boldsymbol{G}_{j,k,\text{undet}} := [\boldsymbol{G}_{j,k+1}\ \boldsymbol{G}_{j,k+2}\ \ldots\ \boldsymbol{G}_{j,\vartheta_j}]$, $\boldsymbol{G}_{j,k,\text{cur}} := \boldsymbol{G}_{j,k}$;
3. *Define* $\hat{\boldsymbol{G}}_{j,1,\text{det}} = [.]$ *and* $\hat{\boldsymbol{G}}_{j,k,\text{det}} := [\hat{\boldsymbol{G}}_{j,1}\ \hat{\boldsymbol{G}}_{j,2}\ \ldots\ \hat{\boldsymbol{G}}_{j,k-1}]$
4. $\mathbf{\Psi}_{j,k} := (\mathbf{I} - \hat{\mathbf{G}}_{j,k,\text{det}}\hat{\mathbf{G}}_{j,k,\text{det}}')$; *thus* $\mathbf{\Psi}_{j,1} = \mathcal{I}$
5. $\boldsymbol{D}_{j,k,\text{cur}} := \mathbf{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{cur}}$, $\boldsymbol{D}_{j,k,\text{det}} := \mathbf{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{det}}$, $\boldsymbol{D}_{j,k,\text{undet}} := \mathbf{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{undet}}$;

Definition 4.7.2.

1. *Let* $\boldsymbol{D}_{j,k,\text{cur}} := \mathbf{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{cur}} \overset{\text{QR}}{=} \boldsymbol{E}_{j,k,\text{cur}}\boldsymbol{R}_{j,k,\text{cur}}$ *denote its reduced QR decomposition. So* $\boldsymbol{E}_{j,k,\text{cur}}$ *is a basis matrix, and* $\boldsymbol{R}_{j,k,\text{cur}}$ *is upper triangular. Let* $\boldsymbol{E}_{j,k,\text{cur},\perp}$ *be a basis matrix for the orthogonal complement of* $\text{range}(\boldsymbol{E}_{j,k,\text{cur}})$.

2. *Using* $\boldsymbol{E}_{j,k,\text{cur}}$ *and* $\boldsymbol{E}_{j,k,\text{cur},\perp}$ *, define*

$$\tilde{\boldsymbol{A}}_{j,k} := \frac{1}{\alpha} \sum_{t \in \tilde{\mathcal{I}}_{j,k}} \boldsymbol{E}_{j,k,\text{cur}}{}' \boldsymbol{\Psi}_{j,k} \boldsymbol{\ell}_t \boldsymbol{\ell}_t{}' \boldsymbol{\Psi}_{j,k} \boldsymbol{E}_{j,k,\text{cur}}$$

$$\tilde{\boldsymbol{A}}_{j,k,\perp} := \frac{1}{\alpha} \sum_{t \in \tilde{\mathcal{I}}_{j,k}} \boldsymbol{E}_{j,k,\text{cur},\perp}{}' \boldsymbol{\Psi}_{j,k} \boldsymbol{\ell}_t \boldsymbol{\ell}_t{}' \boldsymbol{\Psi}_{j,k} \boldsymbol{E}_{j,k,\text{cur},\perp}$$

and let

$$\tilde{\boldsymbol{\mathcal{A}}}_{j,k} := \begin{bmatrix} \boldsymbol{E}_{j,k,\text{cur}} \, \boldsymbol{E}_{j,k,\text{cur},\perp} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{A}}_{j,k} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_{j,k,\perp} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{j,k,\text{cur}}{}' \\ \boldsymbol{E}_{j,k,\text{cur},\perp}{}' \end{bmatrix}$$

3. *Define*

$$\tilde{\boldsymbol{\mathcal{H}}}_{j,k} = \frac{1}{\alpha} \sum_{t \in \tilde{\mathcal{I}}_{j,k}} \boldsymbol{\Psi}_{j,k} \hat{\boldsymbol{\ell}}_t \hat{\boldsymbol{\ell}}_t{}' \boldsymbol{\Psi}_{j,k} - \tilde{\boldsymbol{\mathcal{A}}}_{j,k}$$

From Algorithm 4,

$$\frac{1}{\alpha} \sum_{t \in \tilde{\mathcal{I}}_{j,k}} \boldsymbol{\Psi}_{j,k} \hat{\boldsymbol{\ell}}_t \hat{\boldsymbol{\ell}}_t{}' \boldsymbol{\Psi}_{j,k} \overset{\text{EVD}}{=} \begin{bmatrix} \hat{\mathbf{G}}_{j,k} \, \hat{\mathbf{G}}_{j,k,\perp} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\Lambda}}_t & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{\Lambda}}_{t,\perp} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{G}}_{j,k}{}' \\ \hat{\mathbf{G}}_{j,k,\perp}{}' \end{bmatrix}.$$

Lemma 4.7.3. *[44] When* $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}^a_{j,k-1}$ *with* $a = u_j$ *or* $a = u_j + 1$*,*

1. $\|\boldsymbol{D}_{j,k,\text{det}}\|_2 \leq r\zeta$

2. $\sqrt{1-(r)^2\zeta^2} \leq \sigma_i(\boldsymbol{R}_{j,k,\text{cur}}) = \sigma_i(\boldsymbol{D}_{k,\text{cur}}) \leq 1$

3. $\|\boldsymbol{E}_{j,k,\text{cur}}{}' \boldsymbol{D}_{j,k,\text{undet}}\|_2 \leq \dfrac{(r\zeta)^2}{\sqrt{1-(r)^2\zeta^2}}$

4. $$\boldsymbol{\Psi}_{j,k} \boldsymbol{\Sigma}_t \boldsymbol{\Psi}_{j,k} = [\boldsymbol{D}_{j,k,\text{det}} \; \boldsymbol{D}_{j,k,\text{cur}} \; \boldsymbol{D}_{j,k,\text{undet}}] \begin{bmatrix} \boldsymbol{\Lambda}_{t,\text{det}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_{t,\text{cur}} & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda}_{t,\text{undet}} \end{bmatrix} \begin{bmatrix} \boldsymbol{D}_{j,k,\text{det}} \\ \boldsymbol{D}_{j,k,\text{cur}} \\ \boldsymbol{D}_{j,k,\text{undet}} \end{bmatrix}'$$

with $\lambda_{\max}(\boldsymbol{\Lambda}_{t,\text{det}}) \leq \lambda^+$*,* $\lambda^-_{j,k} \leq \lambda_{\min}(\boldsymbol{\Lambda}_{t,\text{cur}}) \leq \lambda_{\max}(\boldsymbol{\Lambda}_{t,\text{cur}}) \leq \lambda^+_{j,k}$*,* $\lambda_{\max}(\boldsymbol{\Lambda}_{t,\text{undet}}) \leq \lambda^+_{j,k+1}$*.*

5. *Using the first four claims, it is easy to see that*

   (a) $\|\boldsymbol{E}_{j,k,\text{cur},\perp}{}'\boldsymbol{\Psi}_{j,k}\boldsymbol{\Sigma}_t\boldsymbol{\Psi}_{j,k}\boldsymbol{E}_{j,k,\text{cur},\perp}\|_2 \leq (r\zeta)^2\lambda^+ + \lambda^+_{k+1}$ *(when $k=1$, the first term disappears)*

   (b) $\|\boldsymbol{E}_{j,k,\text{cur},\perp}{}'\boldsymbol{\Psi}_{j,k}\boldsymbol{\Sigma}_t\boldsymbol{\Psi}_{j,k}\boldsymbol{E}_{j,k,\text{cur}}\|_2 \leq (r\zeta)^2\lambda^+ + \frac{(r\zeta)^2}{\sqrt{1-(r)^2\zeta^2}}\lambda^+_{k+1}$ *(when $k=1$, the bound equals zero)*

   (c) $\|\boldsymbol{\Psi}_{j,k}\boldsymbol{\Sigma}_t\boldsymbol{\Phi}_{j,K}\|_2 \leq ((r\zeta)\lambda^+ + \lambda^+_k)(r + r_{new})\zeta$

   (d) $\|\boldsymbol{\Phi}_{j,K}\boldsymbol{\Sigma}_t\boldsymbol{\Phi}_{j,K}\|_2 \leq ((r + r_{new})\zeta)^2\lambda^+$

Proof. Consider the first claim. When $k=1$, $\boldsymbol{G}_{k,\text{det}} = [.]$ and hence $\boldsymbol{D}_{j,k,\text{det}} = [.]$. Thus $\|\boldsymbol{D}_{j,k,\text{det}}\|_2 = 0 \leq r\zeta$. For $k > 1$, it follows by applying Lemmas 4.5.31 and 4.5.21 applied for $\tilde{k} = 1, 2, \ldots, k-1$. The next two claims follow using Lemma C.1.1. Notice that $\boldsymbol{D}_{k,\text{cur}} = \boldsymbol{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{cur}}$ where $\boldsymbol{\Psi}_{j,k} = (I - \hat{\boldsymbol{G}}_{j,k,\text{det}}\hat{\boldsymbol{G}}_{j,k,\text{det}}{}')$. Use item 4 of Lemma C.1.1 and the fact that $\boldsymbol{G}_{j,k,\text{det}}{}'\boldsymbol{G}_{j,k,\text{cur}} = \boldsymbol{0}$ to get the second claim. For the third claim, notice that $\boldsymbol{E}_{j,k,\text{cur}}{}'\boldsymbol{D}_{j,k,\text{undet}} = \boldsymbol{R}^{-1}_{j,k,\text{cur}}\boldsymbol{G}_{j,k,\text{cur}}{}'\boldsymbol{\Psi}_{j,k}\boldsymbol{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{undet}}$. Use the previous claim to bound $\|\boldsymbol{R}^{-1}_{j,k,\text{cur}}\|_2$. Use item 3 of Lemma C.1.1 and the facts that $\boldsymbol{G}_{j,k,\text{det}}{}'\boldsymbol{G}_{j,k,\text{cur}} = 0$ and $\boldsymbol{G}_{j,k,\text{det}}{}'\boldsymbol{G}_{k,\text{undet}} = 0$ to bound $\|\boldsymbol{G}_{j,k,\text{cur}}{}'\boldsymbol{\Psi}_{j,k}\|_2$ and $\|\boldsymbol{\Psi}_{j,k}\boldsymbol{G}_{j,k,\text{undet}}\|_2$ respectively. When $k=1$, both the above claims follow even more easily: $\boldsymbol{D}_{k,\text{cur}} = \boldsymbol{G}_{k,\text{cur}}$ and so $\sigma_i(\boldsymbol{D}_{k,\text{cur}}) = 1$ and thus satisfies the given bounds; also, $\boldsymbol{E}_{k,\text{cur}} = \boldsymbol{G}_{k,\text{cur}}$ and $\boldsymbol{D}_{k,\text{undet}} = \boldsymbol{G}_{k,\text{undet}}$ and thus, $\|\boldsymbol{E}_{j,k,\text{cur}}{}'\boldsymbol{D}_{j,k,\text{undet}}\|_2 = 0 \leq \frac{(r\zeta)^2}{\sqrt{1-(r)^2\zeta^2}}$.

The fourth claim just uses the definitions and Model 8. □

4.7.3 Proofs of Lemmas 4.5.39, 4.5.40, 4.5.41

We remove the subscript j at various places in this section.

Proof of Lemma 4.5.39. In this proof all probabilistic statements are conditioned on $X_{(\hat{u}_j+K+1)+k-1}$ for all $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}^a_{j,k-1}$ with $a = u_j$ or $a = u_j + 1$. Recall that $\tilde{\boldsymbol{A}}_k := \frac{1}{\alpha}\sum_{t\in\tilde{\mathcal{I}}_{j,k}} \boldsymbol{E}_{k,\text{cur}}{}'\boldsymbol{\Psi}_k\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur}}$. We proceed as in Section 4.6.1. In this case

$\boldsymbol{N}'_t = \boldsymbol{M}_t = \boldsymbol{\Psi}_k \boldsymbol{E}_{k,\text{cur}}$ and t_0 is the first time instant of $\tilde{\mathcal{I}}_{j,k}$. Thus,

$$\lambda_{\min}(\tilde{\boldsymbol{A}}_k) \geq \lambda_{\min}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{E}_{k,\text{cur}}{}'\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur}}) - 4\epsilon$$
$$\geq \frac{1}{1-b^2}(1-\frac{b^2}{\alpha(1-b^2)}) \min_{\tau\in[t_0,t_0+\alpha-1]} \lambda_{\min}(\boldsymbol{E}_{k,\text{cur}}{}'\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur}}) - 4\epsilon$$

with probability at least $1 - 4\cdot(2n)\exp\left(-\frac{\alpha\epsilon^2}{32b_{prob}^2}\right)$, where $b_{prob} = \frac{r\gamma^2}{(1-b)^2}$.

Finally, using Lemma 4.7.3 and Ostrowski's theorem,

$$\lambda_{\min}(\tilde{\boldsymbol{A}}_k) \geq \frac{1}{1-b^2}(1-\frac{b^2}{\alpha(1-b^2)})(1-(r\zeta)^2)\lambda_k^-$$

□

Proof of Lemma 4.5.40. In this proof all probabilistic statements are conditioned on $X_{(\hat{u}_j+K+1)+k-1}$ for all $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}^a_{j,k-1}$ with $a = u_j$ or $a = u_j+1$.

Recall that $\tilde{\boldsymbol{A}}_{k,\perp} := \frac{1}{\alpha}\sum_t \boldsymbol{E}_{k,\text{cur},\perp}{}'\boldsymbol{\Psi}_k\boldsymbol{\ell}_t\boldsymbol{\ell}_t{}'\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}$.

We proceed as in Section 4.6.1. In this case $\boldsymbol{N}'_t = \boldsymbol{M}_t = \boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}$. Thus,

$$b_{term1} = \frac{(r_{\text{new}}\zeta)^2b^2}{(1-b^2)}\max_{t\in[t_0,t_0+\alpha-1]}\lambda_{\max}(\boldsymbol{E}_{k,\text{cur},\perp}{}'\boldsymbol{\Psi}_k(\boldsymbol{\ell}_{t_0-1}\boldsymbol{\ell}_{t_0-1}{}')\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}) \leq \frac{(r_{\text{new}}\zeta)^2b^2}{(1-b^2)}\frac{(r\gamma^2)}{(1-b)^2}$$
$$\leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\lambda_{\max}(\tilde{\boldsymbol{A}}_{k,\perp}) \leq \lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{E}_{k,\text{cur},\perp}{}'\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}) + b_{term1} + 4\epsilon$$
$$\leq \frac{1}{1-b^2}\max_{\tau\in[t_0,t_0+\alpha-1]}\lambda_{\max}(\boldsymbol{E}_{k,\text{cur},\perp}{}'\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}) + b_{term1} + 4\epsilon$$

with probability at least $1 - 4\cdot(2n)\exp\left(-\frac{\alpha\epsilon^2}{32b_{prob}^2}\right)$, where $b_{prob} = \frac{r\gamma^2}{(1-b)^2}$.

Thus, using Lemma 4.7.3,

$$\lambda_{\max}(\tilde{\boldsymbol{A}}_{k,\perp}) \leq \frac{1}{1-b^2}((r\zeta)^2\lambda^+ + \lambda^+_{k+1}) + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

□

Proof of Lemma 4.5.41. In this proof all probabilistic statements are conditioned on $X_{(\hat{u}_j+K+1)+k-1}$ for all $X_{(\hat{u}_j+K+1)+k-1} \in \tilde{\Gamma}^a_{j,k-1}$ with $a = u_j$ or $a = u_j + 1$. Recall that $\boldsymbol{\Psi}_{j,1} = \mathcal{I}$.

In a fashion similar to the proof of Lemma 4.5.36, we can show that

$$\|\tilde{\boldsymbol{\mathcal{H}}}_k\|_2 \leq 2\left\|\frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \boldsymbol{e}_t{}'\right\|_2 + \left\|\frac{1}{\alpha}\sum_t \boldsymbol{e}_t \boldsymbol{e}_t{}'\right\|_2 + 2\left\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\right\|_2 \tag{4.27}$$

where

$$\boldsymbol{F}_t = \boldsymbol{E}_{k,\text{cur}} \boldsymbol{E}_{k,\text{cur}}{}' \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \boldsymbol{\ell}_t{}' \boldsymbol{\Psi}_k \boldsymbol{E}_{k,\text{cur},\perp} \boldsymbol{E}_{k,\text{cur},\perp}{}'.$$

We now bound the three terms above.

Consider $\|\frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \boldsymbol{e}_t{}'\|_2$. Using Lemma 4.5.25, $\mathbf{e}_t$ satisfies (4.11) with probability one under the given conditioning. Thus,

$$\begin{aligned}
\frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \boldsymbol{e}_t{}' &= \frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t (\boldsymbol{\ell}_t + \mathbf{w}_t)' \boldsymbol{\Phi}_K \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}' - \frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \mathbf{w}_t' \\
&= \frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \boldsymbol{\ell}_t' \boldsymbol{\Phi}_K \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}' + \frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \mathbf{w}_t' \boldsymbol{\Phi}_K \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \\
&\quad \mathbf{I}_{\mathcal{T}_t}{}' - \frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k \boldsymbol{\ell}_t \mathbf{w}_t' \\
&:= \text{term} + \text{termw}
\end{aligned}$$

Here termw refers to the terms containing $\mathbf{w}_t$. By following the approach of Section 4.6.2, under given conditions,

$$\|\text{termw}\|_2 \leq 2\epsilon$$

w.p. at least $1 - 2 \cdot (2n) \exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$, where

$$b_{prob,termw} = \frac{\phi^+ \sqrt{r}\gamma\epsilon_w}{1-b}.$$

We proceed as in Section 4.6.1 for term. In this case $\boldsymbol{N}_t = \boldsymbol{N}_0 = \boldsymbol{\Psi}_k$ and $\boldsymbol{M}_t = \boldsymbol{\Phi}_K \mathbf{I}_{\mathcal{T}_t} [(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}{}' (\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1} \mathbf{I}_{\mathcal{T}_t}{}'$. Thus,

$$b_{term1} \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \frac{\phi^+(r\gamma^2)}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta) b^2 \lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\|\text{term}\|_2 \leq \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\|_2 + b_{term1} + 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$, where

$$b_{prob} = \frac{\phi^+ r\gamma^2(r+r_{\text{new}})\zeta}{(1-b)^2}.$$

Let

$$\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}' := \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{X}_t\boldsymbol{Y}_t'$$

where $\boldsymbol{X}_t := \sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_K$ and $\boldsymbol{Y}_t := \mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$. By Lemma 4.5.22 (support change lemma) $\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{Y}_t\boldsymbol{Y}_t') \leq \rho^2 h^+(\phi^+)^2$.

By Lemma 4.7.3 and the fact that $\|\Phi_K P_*\|_2 \leq \zeta_*^+ = r\zeta$ and $\|\Phi_K P_{\text{new}}\|_2 \leq r_{\text{new}}\zeta$,

$$\lambda_{\max}(\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\boldsymbol{X}_t\boldsymbol{X}_t') \leq \max_t\|\boldsymbol{X}_t\|^2 \leq (\frac{1}{1-b^2}(r+r_{\text{new}})\zeta((r\zeta)\lambda^+ + \lambda_k^+))^2$$

Thus, by Cauchy-Schwartz for matrices,

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2t-2\tau}\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}'(\boldsymbol{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\|_2$$
$$\leq \sqrt{\rho^2 h^+(\phi^+)^2}(\frac{1}{1-b^2}(r+r_{\text{new}})\zeta((r\zeta)\lambda^+ + \lambda_k^+))$$

Thus, with probability at least $1 - p_{\tilde{\boldsymbol{le}}}$,

$$\left\|\frac{1}{\alpha}\sum_t \boldsymbol{\Psi}_k\boldsymbol{\ell}_t\boldsymbol{e}_t'\right\|_2 \leq \sqrt{\rho^2 h^+(\phi^+)^2}(\frac{1}{1-b^2}(r+r_{\text{new}})\zeta((r\zeta)\lambda^+ + \lambda_k^+)) + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 6\epsilon$$

where $p_{\tilde{\boldsymbol{le}}} = 2\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right) + 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,term})^2}\right)$.

Next consider the $\mathbf{e}_t\mathbf{e}_t'$ term.

Recall that, under the given conditioning, $\mathbf{e}_t = \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K(\boldsymbol{\ell}_t + \mathbf{w}_t) - \mathbf{w}_t$. Thus,

$$\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t' = \text{term} + \text{termw}, \text{ where}$$

$$\text{term} := \frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K(\boldsymbol{\ell}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$$

$$\begin{aligned}\text{termw} := \frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K(-\mathbf{w}_t\mathbf{w}_t' - \boldsymbol{\ell}_t\mathbf{w}_t') + \mathbf{w}_t\mathbf{w}_t' + \\ \frac{1}{\alpha}\sum_t(-\mathbf{w}_t\mathbf{w}_t' - \mathbf{w}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}' + \\ \frac{1}{\alpha}\sum_t \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K(\boldsymbol{\ell}_t\mathbf{w}_t' + \mathbf{w}_t\mathbf{w}_t' + \mathbf{w}_t\boldsymbol{\ell}_t')\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\end{aligned}$$

For the $\mathbf{w}_t\mathbf{w}_t'$ part of termw, let $\boldsymbol{N}_t = \mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K$. Using Lemma C.1.6 (Cauchy-Schwartz), Lemma 4.5.22 (support change lemma) and the bound on ϵ_w^2, we have

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\mathbf{w}_t\mathbf{w}_t'\|_2 \leq \sqrt{\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\boldsymbol{N}_t'\|_2\|\frac{1}{\alpha}\sum_t \mathbf{w}_t\mathbf{w}_t'\mathbf{w}_t\mathbf{w}_t'\|_2} \leq \sqrt{\rho^2 h^+(\phi^+)^2}\epsilon_w^2$$

Using Lemma 4.5.22 (support change lemma), we have

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{N}_t\mathbf{w}_t\mathbf{w}_t'\mathbf{\Phi}_K\boldsymbol{N}_t'\|_2 \leq \rho^2 h^+(\phi^+)^2\epsilon_w^2$$

The $\boldsymbol{\ell}_t\mathbf{w}_t'$ in termw can be bounded by ϵ using the approach of Section 4.6.2. Thus,

$$\|\text{termw}\|_2 \leq (1 + 2\sqrt{\rho^2 h^+}\phi^+ + 2\rho^2 h^+(\phi^+)^2)(0.03\zeta\lambda^-) + 4\epsilon \leq 2(\phi^+)^2(0.03\zeta\lambda^-)$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$ where

$$b_{prob,termw} = (\phi^+)^2\frac{(2r\gamma\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})\epsilon_w}{1-b}.$$

For term, we proceed as in Section 4.6.1 with $\boldsymbol{N}_t' = \boldsymbol{M}_t = \mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'$. Thus,

$$b_{prob,term2} = \max(b_{prob,term21}, b_{prob,term22}, b_{prob,term23}) \leq \frac{1}{(1-b)^2}(\phi^+)^2(\zeta_*^+\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

$$b_{prob,term3} \leq \frac{1}{(1-b)^3}(\phi^+)^2(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Use b_{prob} to denote the upper bound on $\max(b_{prob,term2}, b_{prob,term3})$. Then

$$b_{prob} = \frac{1}{(1-b)^3}(\phi^+)^2(2r\zeta\sqrt{r}\gamma + \sqrt{r_{\text{new}}}\gamma_{\text{new}})^2$$

Using (4.20), (4.22), (4.21), we get

$$b_{term1} \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \frac{(r\gamma^2)}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\|\text{term}\|_2 \leq \|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \mathbf{I}_{\mathcal{T}_t}\left(\sum_{\tau=t_0}^{t} b^{2t-2\tau}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K\mathbf{\Sigma}_\tau\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\right)\mathbf{I}_{\mathcal{T}_t}'\|_2$$
$$+ b_{term1} + 4\epsilon$$

w.p. at least $1 - 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$. By Lemma 4.5.22 (support change lemma), Lemma 4.7.3, and the fact that $\|\Phi_K P_*\|_2 \leq \zeta_*^+ = r\zeta$ and $\|\Phi_K P_{\text{new}}\|_2 \leq r_{\text{new}}\zeta$,

$$\|\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1} \mathbf{I}_{\mathcal{T}_t}\left(\sum_{\tau=t_0}^{t} b^{2t-2\tau}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\mathbf{I}_{\mathcal{T}_t}'\mathbf{\Phi}_K\mathbf{\Sigma}_\tau\mathbf{\Phi}_K\mathbf{I}_{\mathcal{T}_t}[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\right)\mathbf{I}_{\mathcal{T}_t}'\|_2$$
$$\leq \rho^2 h^+ \ (\phi^+)^2\frac{1}{1-b^2}((r+r_{\text{new}})\zeta)^2\lambda^+$$

Combining all the bounds from above,

$$\|\frac{1}{\alpha}\sum_t \mathbf{e}_t\mathbf{e}_t'\|_2 \leq \rho^2 h^+ \ (\phi^+)^2\frac{1}{1-b^2}((r+r_{\text{new}})\zeta)^2\lambda^+ + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + (\phi^+)^2 2(0.03\zeta\lambda^-) + 8\epsilon$$

w.p. at least $1 - p_{\tilde{\mathbf{e}}\tilde{\mathbf{e}}}$ with $p_{\tilde{\mathbf{e}}\tilde{\mathbf{e}}} := 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right) + 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob,termw})^2}\right)$.

Finally consider $\left\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\right\|_2 = \|\frac{1}{\alpha}\sum_t \boldsymbol{E}_{k,\text{cur}}\boldsymbol{E}_{k,\text{cur}}'\mathbf{\Psi}_k\boldsymbol{\ell}_t\boldsymbol{\ell}_t'\mathbf{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}\boldsymbol{E}_{k,\text{cur},\perp}'\|_2$. We proceed as in Section 4.6.1. Here $\boldsymbol{N}_t = \boldsymbol{E}_{k,\text{cur}}\boldsymbol{E}_{k,\text{cur}}'\mathbf{\Psi}_k$ and $\boldsymbol{M}_t = \mathbf{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}\boldsymbol{E}_{k,\text{cur},\perp}'$. Thus, we get

$$b_{term1} \leq \frac{(r_{\text{new}}\zeta)^2 b^2}{(1-b^2)} \frac{(r\gamma^2)}{(1-b)^2} \leq \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2}$$

(we can get a tighter bound for the above, but do not need it and hence do not pursue it) and

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\|_2 \leq \frac{1}{1-b^2}\max_{\tau\in[t_0,t_0+\alpha-1]}\|\boldsymbol{E}_{k,\text{cur}}\boldsymbol{E}_{k,\text{cur}}'\mathbf{\Psi}_k\mathbf{\Sigma}_\tau\mathbf{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}\boldsymbol{E}_{k,\text{cur},\perp}'\|_2 + b_{term1} + 4\epsilon$$

By Lemma 4.7.3,

$$\|\boldsymbol{E}_{k,\text{cur}}\boldsymbol{E}_{k,\text{cur}}{}'\boldsymbol{\Psi}_k\boldsymbol{\Sigma}_\tau\boldsymbol{\Psi}_k\boldsymbol{E}_{k,\text{cur},\perp}\boldsymbol{E}_{k,\text{cur},\perp}{}'\|_2 \leq (r\zeta)^2\lambda^+ + \frac{(r\zeta)^2}{\sqrt{1-(r\zeta)^2}}\lambda^+_{k+1}$$

Thus,

$$\|\frac{1}{\alpha}\sum_t \boldsymbol{F}_t\|_2 \leq \frac{1}{1-b^2}\left((r\zeta)^2\lambda^+ + \frac{(r\zeta)^2}{\sqrt{1-(r\zeta)^2}}\lambda^+_{k+1}\right) + \frac{0.05(r_{\text{new}}\zeta)b^2\lambda^-}{(1-b^2)(1-b)^2} + 4\epsilon$$

with probability at least $1 - p_{\tilde{\boldsymbol{F}}}$, where $p_{\tilde{\boldsymbol{F}}} = 4\cdot(2n)\exp\left(\frac{-\alpha\epsilon^2}{32(b_{prob})^2}\right)$, with $b_{prob} = \frac{r\gamma^2}{(1-b)^2}$.

Combining the bounds on the three terms above, we get the final result of the lemma.

□

4.8 Automatically Setting Algorithm Parameters And Simulation Experiments

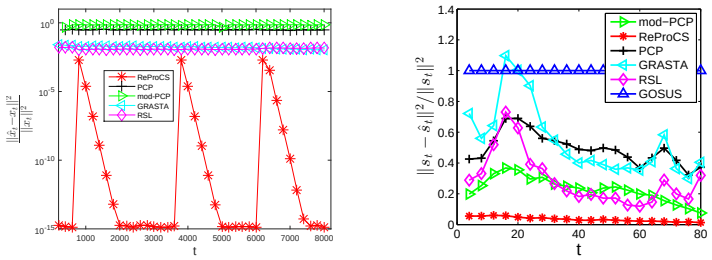
4.8.1 Automatically setting algorithm parameters

The algorithm has five parameters. As explained in [64], one can set $\xi_t = \|\boldsymbol{\Phi}_t\hat{\boldsymbol{\ell}}_{t-1}\|_2$. One can either set $\omega_t = 7\xi_t$ or one can use the average image pixel intensity to set it. In [64], they used $\omega = q\sqrt{\|\mathbf{m}_t\|_2^2/n}$ with $q = 1$ when it was known that $\|\mathbf{x}_t\|_2$ is of the same order as $\|\boldsymbol{\ell}_t\|_2$; and $q = 0.25$ when $\|\mathbf{x}_t\|_2$ was known to be much smaller (the case of foreground moving objects whose intensity is very similar to that of background objects). There is no good heuristic to pick α except that α_{add} should be large enough compared to r_{new} and α_{del} should be large enough compared to r. We used $\alpha = 100$ and $K = 12$ in our experiments. We need K to be large enough so that the new subspace is accurately recovered at the end of K projection-PCA iterations. Thus, one way to set K indirectly is as follows: do projection-PCA for at least $K_{\min}$ times, but after that stop when there is not much difference between $\hat{\mathbf{P}}_{j,\text{new},k}{}'\hat{\boldsymbol{\ell}}_t$ and $\hat{\mathbf{P}}_{j,\text{new},k+1}{}'\hat{\boldsymbol{\ell}}_t$ [44, 64]. This, along with imposing an upper bound on K works well in practice [64]. We can set $\hat{g}^+$ as suggested in [44]; by applying any clustering algorithm from literature, e.g., k-means

clustering or split-and-merge and then finding the maximum condition number of any cluster. This can be applied to the empirical covariance matrix used in the clustering step of cluster-PCA.

4.8.2 Simulated data

Here we used simulated data to compare performance of PCP [32], mod-PCP [86], GRASTA [80], RSL [79] and Automatic ReProCS-cPCA. We generated data as explained in Sec. 4.2, with $n = 256$, $J = 3$, $r_0 = 40$, $t_{\text{train}} = 200$, $t_{\max} = 8200$. We generated ℓ_t as in Model 5 and Model 8 with $r_{j,\text{new}} = 4$, $r_{j,\text{old}} = 4$, $j = 1, 2, 3$, $t_1 = 700$, $t_2 = 3700$, $t_3 = 6200$, $\vartheta = 3$, $b = 0.1$. The subspace $[\boldsymbol{P}_0, \boldsymbol{P}_{t_1,\text{new}}, \boldsymbol{P}_{t_2,\text{new}}, \boldsymbol{P}_{t_3,\text{new}}]$ was generated by orthonormalizing an $n \times (r_0 + r_{1,\text{new}} + r_{2,\text{new}} + r_{3,\text{new}})$ matrix of iid Gaussian entries. The coefficients $a_{t,*} := \boldsymbol{P}_{j,*}^* \nu_t$, were generated as follows. They were divided into three clusters. The coefficeints of the first cluster were iid uniformly distributed over $[-100, 100]$, those of the second cluster were iid uniform over $[-10, 10]$, and those of the third cluster were iid uniform over $[-1, 1]$. We generated $a_{t,\text{new}} := \boldsymbol{P}_{j,\text{new}}^* \nu_t$ iid uniform over $[-1, 1]$ for the first 1700 time units after the subspace change. After that, it was in one of the three intervals. The sparse matrix $\mathbf{S}$ was generated as in Model 4 with $s = 10$, $\rho = 2$. The support of x_t started from the top, and moved down by 5 indices every $\beta = 25$ time instants. Once it reached the bottom, it started from the top again. We set $(x_t)_i \sim \text{Unif}[x_{\min}, 3x_{\min}]$ for all $i \in \mathcal{T}_t$ with $x_{\min} = 20$. We ran Automatic ReProCS-cPCA with $\alpha = 100$, $K = 12$, $\xi = \sqrt{r_{\text{new}}/2}\gamma_{\text{new}}$, $\omega = (x_{\min} - 14\xi)/2$. We used $\hat{\boldsymbol{P}}_0$ for modified-PCP as partial knowledge. We solved PCP and modified-PCP every 200 frames by using the observations of the last 200 frames as the matrix $\mathbf{M}$. In Fig. 4.4, where the averaged sparse part errors over 50 Monte Carlo simulations are shown, we can see Automatic ReProCS-cPCA outperforms all the other algorithms. We can also see jumps in the Automatic ReProCS-cPCA error at the time instants at which there is a subspace change, and then decays exponentially. This is what is seen from the bounds



(a) fully simulated data (b) lake background

Figure 4.4: Average error comparisons for fully simulated data and for the sequence with the lake background and simulated block object

given in Theorem 4.2.8 and Corollary 4.2.11.

4.8.3 Lake background sequence with simulated foreground

The lake background sequence used is the same as the one used in [64]. The background consisted of a video of moving lake waters. The foreground is a simulated moving rectangular object. The sequence is of size $72 \times 90 \times 1500$, and we used the first 1420 frames as training data (after subtracting the empirical mean of the training images). The rest 80 frames (after subtracting the same mean image) served as the background $\mathbf{L}$ for the test data. For the first frame of test data, we generated a rectangular foreground support with upper left vertex $(20, 5 + j_0)$ and lower right vertex $(40 + i_1, 30 + j_0)$, where $j_0 \sim \text{Unif}[0, 30]$ and $i_1 \sim \text{Unif}[0, 5]$, and the foreground moves to the right 1 column each time. Then we stacked each image as a long vector ℓ_t of size 6480×1. For each index i belonging to the support set of foreground x_t, we assign $(x_t)_i = 185 - (\ell_t)_i$. We set $\boldsymbol{M} = \mathbf{L} + \mathbf{S}$. For mod-PCP, ReProCS and GRASTA, we used the approach used in [64] to estimate the initial background subspace (partial knowledge): do SVD on training data and keep the left singular vectors corresponding to 95% energy as the matrix $\boldsymbol{P}_0$. The averaged normalized mean squared error (NMSE) of the sparse part over 50 Monte Carlo realizations is shown in Fig. 4.4. As can be seen, in this case, ReProCS performs the best. In Fig. 4.1, we show the lake with simulated foreground at $t = 20, 40, 60$, and

corresponding foreground and background recovered by different algorithms, and we can see that ReProCS successfully separated foreground and background apart while others did not.

4.9 Conclusions

In this work, we developed and studied the Automatic ReProCS-cPCA algorithm for incremental or recursive or dynamic or "online" robust PCA. Our result needed the following assumptions: accurate initial subspace knowledge and a slow subspace change change assumption on the $\boldsymbol{\ell}_t$'s; the basis vectors for its subspaces are dense (non-sparse) enough; the eigenvalues of the covariance matrix of $\boldsymbol{\ell}_t$'s are clustered for a certain period of time (this would happen if data has variations across different scales); the outlier support sets $\mathcal{T}_t$ have *some* changes over time (as quantified in Model 4 or Model 10); the square of the smallest outlier magnitude is large enough compared to the energy in the unstructured small noise plus the energy in the changed subspace; and the algorithm parameters are appropriately set. Ongoing work includes studying the undersampled measurements' case, i.e., the case $\mathbf{m}_t = A_t\mathbf{x}_t + B_t\boldsymbol{\ell}_t + \mathbf{w}_t$. Besides this, we expect the cluster-PCA algorithm and the proof techniques developed here to apply to various other problems involving PCA with data and noise terms being correlated.

Algorithm 4 Automatic ReProCS-cPCA

Parameters: α, K, ξ, ω, $\hat{g}^+$, **Inputs**: $\boldsymbol{m}_t$, **Output**: $\hat{\mathbf{x}}_t$, $\hat{\boldsymbol{\ell}}_t$, $\hat{\mathbf{P}}_t$, $\hat{t}_{\hat{\jmath}}$, $\hat{r}_{\hat{\jmath},\text{new},k}$, $\hat{G}_{j,k}$
Compute $\hat{\lambda}^-_{\text{train}}$ as the r_0-th eigenvalue of $\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}} \mathbf{m}_t\mathbf{m}_t'$ and $\hat{\mathbf{P}}_{t_{\text{train}}}$ as its top r_0 eigenvectors. Set thresh $= \frac{\hat{\lambda}^-_{\text{train}}}{2}$. Set $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t_{\text{train}}}$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow [.]$, $\hat{\jmath} \leftarrow 0$, phase $\leftarrow$ detect.
For every $t > t_{\text{train}}$, do

1. **Estimate $\mathcal{T}_t$ and $\mathbf{x}_t$:**
   (a) compute $\boldsymbol{\Phi}_t \leftarrow \boldsymbol{I} - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}'$ and $\boldsymbol{y}_t \leftarrow \boldsymbol{\Phi}_t\boldsymbol{m}_t$
   (b) solve $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1$ *s.t.* $\|\boldsymbol{y}_t - \boldsymbol{\Phi}_t\boldsymbol{x}\|_2 \le \xi$ and let $\hat{\boldsymbol{x}}_{t,\text{cs}}$ denote its solution
   (c) compute $\hat{\mathcal{T}}_t = \{i : |(\hat{\boldsymbol{x}}_{t,\text{cs}})_i| > \omega\}$
   (d) LS: compute $\hat{\boldsymbol{x}}_t = \mathbf{I}_{\hat{\mathcal{T}}_t}((\boldsymbol{\Phi}_t)_{\hat{\mathcal{T}}_t})^\dagger \boldsymbol{y}_t$
2. **Estimate $\boldsymbol{\ell}_t$:** $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{m}_t - \hat{\boldsymbol{x}}_t$
3. **Subspace Update:**
   If $t \mod \alpha \neq 0$ then $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t-1,*}$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow \hat{\mathbf{P}}_{t-1,\text{new}}$, $\hat{\mathbf{P}}_t \leftarrow [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$
   If $t \mod \alpha = 0$ then
   if phase = detect **then**
   (a) Set $u = \frac{t}{\alpha}$ and compute $\boldsymbol{\mathcal{D}}_u = (\mathbf{I} - \hat{\mathbf{P}}_{u\alpha-1,*}\hat{\mathbf{P}}_{u\alpha-1,*}')[\hat{\boldsymbol{\ell}}_{(u-1)\alpha+1}, \dots \hat{\boldsymbol{\ell}}_{u\alpha}]$
   (b) $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t-1,*}$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow \hat{\mathbf{P}}_{t-1,\text{new}}$, $\hat{\mathbf{P}}_t \leftarrow [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$
   (c) If $\lambda_{\max}(\frac{1}{\alpha}\boldsymbol{\mathcal{D}}_u\boldsymbol{\mathcal{D}}_u') \ge$ thresh then
      i. phase $\leftarrow$ pPCA, $\hat{\jmath} \leftarrow \hat{\jmath} + 1$, $k \leftarrow 0$, $\hat{t}_{\hat{\jmath}} = t$
   else if phase = pPCA **then**
   (a) Set $u = \frac{t}{\alpha}$ and compute $\boldsymbol{\mathcal{D}}_u = (\mathbf{I} - \hat{\mathbf{P}}_{u\alpha-1,*}\hat{\mathbf{P}}_{u\alpha-1,*}')[\hat{\boldsymbol{\ell}}_{(u-1)\alpha+1}, \dots \hat{\boldsymbol{\ell}}_{u\alpha}]$
   (b) $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow$ eigenvectors $\left(\frac{1}{\alpha}\boldsymbol{\mathcal{D}}_u\boldsymbol{\mathcal{D}}_u', \text{thresh}\right)$, $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t-1,*}$, $\hat{\mathbf{P}}_t \leftarrow [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$
   (c) $k \leftarrow k + 1$, set $\hat{r}_{j,\text{new},k} = \text{rank}(\hat{\mathbf{P}}_{t,\text{new}})$
   (d) If $k == K$, then phase $\leftarrow$ cPCA, reset $k \leftarrow 0$.
   else if phase = cPCA **then**
   (a) cluster PCA (summarized in Algorithm 5);
   (b) set $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_t$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow [.]$,
   (c) phase $\leftarrow$ detect, reset $k \leftarrow 0$
   end-if

eigenvectors$(\boldsymbol{\mathcal{M}}, \text{thresh})$ returns a basis matrix for the span of eigenvectors with eigenvalue above thresh. eigenvectors$(\boldsymbol{\mathcal{M}}, , r)$ returns a basis matrix for the span of the top r eigenvectors.
Offline RPCA: at $t = \hat{t}_j + K\alpha$, for all $t \in [\hat{t}_{j-1} + K\alpha + 1, \hat{t}_j + K\alpha]$, compute

$$\hat{\mathbf{x}}_t^{\text{offline}} \leftarrow \mathbf{I}_{\hat{\mathcal{T}}_t}((\boldsymbol{\Phi}_{\hat{t}_j+K\alpha})_{\hat{\mathcal{T}}_t})^\dagger \boldsymbol{\Phi}_{\hat{t}_j+K\alpha}\boldsymbol{m}_t \text{ and } \hat{\boldsymbol{\ell}}_t^{\text{offline}} \leftarrow \mathbf{m}_t - \mathbf{x}_t$$

Algorithm 5 cluster PCA

1. If $k == 0$, estimate the clusters
   (a) Set $u = \frac{t}{\alpha}$ and compute $\hat{\mathbf{\Sigma}}_{\text{sample}} = \frac{1}{\alpha}\sum_{t=(u-1)\alpha+1}^{u\alpha} \hat{\boldsymbol{\ell}}_t \hat{\boldsymbol{\ell}}_t{}'$. Let $\hat{\lambda}_i$ denote its i-th largest eigenvalue.
   (b) To get the first cluster $\hat{\mathcal{G}}_{j,1}$, we start with the index of the first (largest) eigenvalue and keep adding indices of the smaller eigenvalues to it until $\frac{\hat{\lambda}_1}{\hat{\lambda}_{i+1}} > \hat{g}^+$ but $\frac{\hat{\lambda}_1}{\hat{\lambda}_i} \leq \hat{g}^+$ or until $\hat{\lambda}_{i+1} < 0.25\hat{\lambda}^-_{\text{train}}$. We set $\hat{\mathcal{G}}_{j,1} = \{1, 2, \ldots i\}$.
   For $\hat{\mathcal{G}}_{j,2}$, start with the $(i+1)$-th eigenvalue and repeat the above procedure. Repeat the above for each new cluster and stop when there are no more eigenvalues larger than $0.25\hat{\lambda}^-_{\text{train}}$.
   (c) $k \leftarrow k+1$, $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t-1,*}$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow \hat{\mathbf{P}}_{t-1,\text{new}}$, $\hat{\mathbf{P}}_t \leftarrow [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$
2. If $1 \leq k \leq \vartheta$, estimate the k-th cluster's subspace by cluster PCA
   (a) Set $u = \frac{t}{\alpha}$, set $\hat{\boldsymbol{G}}_{j,0} \leftarrow [.]$.
   - let $\hat{\boldsymbol{G}}_{j,\text{det},k} := [\hat{\boldsymbol{G}}_{j,0}, \hat{\boldsymbol{G}}_{j,1}, \ldots \hat{\boldsymbol{G}}_{j,k-1}]$ and let $\mathbf{\Psi}_k := (\mathbf{I} - \hat{\boldsymbol{G}}_{j,\text{det},k}\hat{\boldsymbol{G}}_{j,\text{det},k}{}')$ (notice that $\mathbf{\Psi}_{j,1} = \mathcal{I}$); compute $\boldsymbol{\mathcal{M}}_{\text{cpca}} = \mathbf{\Psi}_k \left(\frac{1}{\alpha}\sum_{t\in(u-1)\alpha+1}^{u\alpha} \hat{\boldsymbol{\ell}}_t \hat{\boldsymbol{\ell}}_t{}'\right) \mathbf{\Psi}_k$
   - compute $\hat{\boldsymbol{G}}_{j,k} \leftarrow \text{eigenvectors}(\boldsymbol{\mathcal{M}}_{\text{cpca}}, , |\hat{\mathcal{G}}_{j,k}|)$
   (b) $k \leftarrow k+1$, $\hat{\mathbf{P}}_{t,*} \leftarrow \hat{\mathbf{P}}_{t-1,*}$, $\hat{\mathbf{P}}_{t,\text{new}} \leftarrow \hat{\mathbf{P}}_{t-1,\text{new}}$, $\hat{\mathbf{P}}_t \leftarrow [\hat{\mathbf{P}}_{t,*}\ \hat{\mathbf{P}}_{t,\text{new}}]$
3. If $k == \vartheta$, set $\hat{\mathbf{P}}_t \leftarrow [\hat{\boldsymbol{G}}_{j,1} \cdots \hat{\boldsymbol{G}}_{j,\vartheta}]$.

CHAPTER 5. CONCLUSION AND FUTURE WORK

In this work, we studied the sparse recovery problem in the presence of different noise and obtained results for different cases.

In Chapter 2, we obtained performance guarantees for recursive noisy modified-CS which has been shown in earlier work to be a practically useful algorithm [14, 99, 29]. We show that, under mild assumptions – a lower bound on either the initial nonzero magnitude or on the magnitude increase rate, or an upper bound on the maximum number of nonzero entries with magnitude below a certain threshold; mild RIP conditions (which imply conditions on the required number of measurements); appropriately set algorithm parameters; and a special start condition – the support and signal recovery error of modified-CS and its improvement, modified-CS-add-LS-del can be bounded by time-invariant and small values.

The special start condition is a possible limitation of our analysis. This can be removed in various ways. If some prior knowledge about signal support is available, that can be used at $t = 0$ as suggested and demonstrated in [14]. Or, one can solve a batch problem (multiple measurement vector (MMV) problem) for the first set of k frames. If we let $\mathcal{N} = \cup_{t=1}^{k} \mathcal{N}_t$, then we have an MMV problem with row support $\mathcal{N}$ that can be solved using mixed norm minimization [100], simultaneous-OMP [101, 102], compressive MUSIC [103], iterative MUSIC [104], block sparsity approaches [105] or M-SBL (Sparse Bayesian Learning) [106]. In this case one could adopt guarantees for the chosen batch method for the initialization. In Chapter 2, we used a deterministic set of assumptions on signal change. Notice however that one can assume any probabilistic model that

ensures that $a_{j,t} \geq a_{\min}$ and $r_{j,\tau}$ is anything larger than $r_{\min}(d_0)$ for for the first d_0 frames after a new addition; and at later times, $r_{j,\tau}$ can be anything between zero and infinity. Similarly, any probabilistic model for coefficient decrease that ensures removal within at most b frames after decrease begins will suffice. We can fix d_0 to be any integer between zero and $d_{\min}$ and our result will then hold for that particular value of d_0.

Other ongoing and future work includes designing and analyzing better support prediction techniques rather than just using the previous support estimate as the prediction for the current support. Some initial ideas are presented in [107].

In Chapter 3, we studied the following problem. Suppose that we have a partial estimate of the column space of the low rank matrix $\mathbf{L}$. How can we use this information to improve the PCP solution? We proposed a simple modification of PCP, called *modified-PCP*, that allows us to use this knowledge. We derived its correctness result that allows us to argue that, when the available subspace knowledge is accurate enough, modified-PCP requires significantly weaker incoherence assumptions on the low-rank matrix than PCP. We also obtained a useful corollary (Corollary 3.2.1) for the online or recursive robust PCA problem. Extensive simulation experiments and some experiments for a real application further illustrate these claims. Ongoing work includes studying the error stability of modified-PCP for online robust PCA. Future work will include developing a fast and recursive algorithm for solving modified-PCP and using the resulting algorithm for various practical applications. Two applications that will be explored are (a) video layering, e.g. using the BMC dataset of [108], and (b) recommendation system design in the presence of outliers and missing data. For getting a recursive algorithm, we will explore the use of ideas similar to those introduced in Feng et al's recent work on developing a recursive algorithm that asymptotically approximates the PCP solution [66].

In Chapter 4, we studied the problem of recursively separate low rank and sparse component apart in the presence of bounded noise. We develop and study an algorithm

based on the ReProCS idea introduced and studied in [44, 84, 85]. We call it Automatic ReProCS with cluster PCA (ReProCS-cPCA). This is a significantly improved ReProCS algorithm compared to what was studied in previous work. It is able to automatically detect subspace changes within a short delay; is able to correctly estimate the number of directions added or deleted; and is also able to correctly estimate the clusters of eigenvalues along the existing directions. The algorithms studied in [44, 84] could not do any of this. Moreover it is able to accurately estimate both the newly added subspace as well as the newly deleted subspace. The latter is done by re-estimating the current subspace using an approach called cluster PCA (cPCA). The basic idea of cPCA was introduced in [44], but the current work uses that idea to develop an automatic algorithm. The cPCA step ensures that the estimated subspace dimension does not keep increasing with time. The algorithms studied in [84, 85] did not do this. Finally, unlike past works, the current algorithm also returns more accurate offline estimates. We also derive a correctness result for the proposed algorithm under relatively mild assumptions. (1) First, we obtain a result for the case where the $\boldsymbol{\ell}_t$'s can be correlated over time (follow an autoregressive (AR) model) where as the result of [84, 85] needed mutual independence of the $\boldsymbol{\ell}_t$'s. This models mostly static backgrounds in which changes are only due to independent variations at each time, e.g., light flickers. However, a large class of background image sequences change due to factors that are correlated over time, e.g., moving waters. This can be better modeled using an AR model. (2) Second, with one extra assumption – that the eigenvalues of the covariance matrix of $\boldsymbol{\ell}_t$ are clustered for a period of time after the previous subspace change has stabilized – we are able to remove another significant limitation of [84, 85].

APPENDIX A. PROOF OF THE LEMMAS IN CHAPTER 2

A.0.1 Proof of Lemma 2.2.7

We provide the proof here for the sake of completion and for ease of review. This will be removed later. In this proof, we use $\mathcal{T}, \Delta, \mathcal{N}$ instead of $\mathcal{T}_t, \Delta_t, \mathcal{N}_t$ respectively for simplicity. Let $h := \hat{\mathbf{x}}_{modcs} - \mathbf{x}$. We adapt the approach of [12] to bound the reconstruction error, $\|\mathbf{h}\| := \|\hat{\mathbf{x}}_{modcs} - \mathbf{x}\|$. A similar result was obtained in [28]. Let Δ_1 denote the set of indices of $\mathbf{h}$ with the $|\Delta|$ largest values outside of $\mathcal{T} \cup \Delta$, let Δ_2 denote the indices of the next $|\Delta|$ largest values and so on. Then using the same approach as that of [12], i.e., $\|\mathbf{h}_{\Delta_j}\| \leq \frac{1}{\sqrt{\Delta}} \|\mathbf{h}_{\Delta_{j-1}}\|_1$,

$$\|\mathbf{h}_{(\mathcal{T} \cup \Delta \cup \Delta_1)^c}\| \leq \sum_{j \geq 2} \|\mathbf{h}_{\Delta_j}\| \leq \frac{1}{\sqrt{|\Delta|}} \|\mathbf{h}_{(\mathcal{T} \cup \Delta)^c}\|_1 \tag{A.1}$$

Since $\hat{\mathbf{x}}_{modcs} = \mathbf{x} + \mathbf{h}$ is the minimizer of (2.2) and since both $\mathbf{x}$ and $\hat{\mathbf{x}}_{modcs}$ are feasible; and since $\mathbf{x}$ is supported on $\mathcal{N} \subseteq \mathcal{T} \cup \Delta$,

$$\begin{aligned} \|\mathbf{x}_\Delta\|_1 = \|\mathbf{x}_{\mathcal{T}^c}\|_1 &\geq \|(\mathbf{x} + \mathbf{h})_{\mathcal{T}^c}\|_1 \\ &\geq \|\mathbf{x}_\Delta\|_1 - \|\mathbf{h}_\Delta\|_1 + \|\mathbf{h}_{(\mathcal{T} \cup \Delta)^c}\|_1 \end{aligned} \tag{A.2}$$

Thus,

$$\|\mathbf{h}_{(\mathcal{T} \cup \Delta)^c}\|_1 \leq \|\mathbf{h}_\Delta\|_1 \tag{A.3}$$

Combining this with (A.1), and using $\frac{\|\mathbf{h}_\Delta\|_1}{\sqrt{|\Delta|}} \leq \|\mathbf{h}_\Delta\|$, we get

$$\|\mathbf{h}_{(\mathcal{T} \cup \Delta \cup \Delta_1)^c}\| \leq \sum_{j \geq 2} \|\mathbf{h}_{\Delta_j}\| \leq \|\mathbf{h}_\Delta\| \tag{A.4}$$

Next, since both $\mathbf{x}$ and $\hat{\mathbf{x}}_{modcs}$ are feasible,

$$\begin{aligned}\|\mathbf{Ah}\| &= \|\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}}_{modcs})\| \\ &\leq \|\mathbf{y}-\mathbf{Ax}\| + \|\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{modcs}\| \leq 2\epsilon \end{aligned} \tag{A.5}$$

In this proof, let

$$\delta \triangleq \delta_{|\mathcal{T}|+3|\Delta|} \tag{A.6}$$

Now, we upper bound $\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|$. By $\delta_{|\mathcal{T}|+2|\Delta|} \leq \delta$, we have

$$(1-\delta)\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|^2 \leq \|\mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|^2 \tag{A.7}$$

To bound the RHS of the above, notice that $\mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1} = \mathbf{Ah} - \sum_{j\geq 2}\mathbf{Ah}_{\Delta_j}$ and so

$$\|\mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|^2 = \langle \mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}, \mathbf{Ah}\rangle - \sum_{j\geq 2}\langle \mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}, \mathbf{Ah}_{\Delta_j}\rangle$$

Using (A.5) and the definition of δ_S given in (2.8) and $\delta_{|\mathcal{T}|+2|\Delta|} \leq \delta$,

$$|\langle \mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}, \mathbf{Ah}\rangle| \leq 2\epsilon\sqrt{1+\delta}\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\| \tag{A.8}$$

Using the definition of θ_{S_1,S_2} given in (2.9); equation (A.4); and the fact that $\|\mathbf{h}_{\mathcal{T}}\| + \|\mathbf{h}_{\Delta\cup\Delta_1}\| \leq \sqrt{2}\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|$, we get the following. Using $\theta_{|\mathcal{T}|,|\Delta|} \leq \delta_{|\mathcal{T}|+|\Delta|} \leq \delta_{|\mathcal{T}|+3|\Delta|}, \theta_{2|\Delta|,|\Delta|} \leq \delta_{3|\Delta|} \leq \delta_{|\mathcal{T}|+3|\Delta|}$ [10],

$$\begin{aligned}&|\sum_{j\geq 2}\langle \mathbf{Ah}_{\mathcal{T}\cup\Delta\cup\Delta_1}, \mathbf{Ah}_{\Delta_j}\rangle| \\ &\leq \theta_{|\mathcal{T}|+2|\Delta|,|\Delta|}\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|\sum_{j\geq 2}\|\mathbf{h}_{\Delta_j}\| \\ &\leq \delta\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|\ \|\mathbf{h}_{\Delta}\| \end{aligned} \tag{A.9}$$

Combining the last six equations above, using $\|\mathbf{h}_{\Delta}\| \leq \|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\|$, we can simplify the above to get

$$\begin{aligned}\|\mathbf{h}\| &\leq 2\|\mathbf{h}_{\mathcal{T}\cup\Delta\cup\Delta_1}\| \leq \frac{4\sqrt{1+\delta}}{1-2\delta}\epsilon \\ &\leq \frac{4\sqrt{1+\delta}}{1-2\delta}\epsilon \end{aligned} \tag{A.10}$$

Clearly, all of the above discussion holds only if the RHS is positive which is true only if $2\delta_{|\mathcal{T}|+3|\Delta|} < 1$. Thus, we can get Lemma 2.2.7.

A.0.2 Proof of Theorem 2.3.2

We prove the first two claims by induction. Using condition 4 of the theorem, the claim holds for $t = 0$. This proves the base case. For the induction step, assume that the claims hold at $t-1$, i.e. $|\tilde{\Delta}_{e,t-1}| = 0$, $|\tilde{\mathcal{T}}_{t-1}| \leq S$, and $|\tilde{\Delta}_{t-1}| \leq S_a$, so $|\mathcal{T}_t| \leq S$. At t, there are at most S_a new support, so $|\Delta_t| \leq |\tilde{\Delta}_{t-1}| + S_a \leq 2S_a$; there are at most S_a removed support at time t, so $|\Delta_{e,t}| \leq |\tilde{\Delta}_{t-1}| + S_a = S_a$. Thus the second claim holds.

Next we bound $|\tilde{\Delta}_t|$, $|\tilde{\Delta}_{e,t}|$, $|\tilde{\mathcal{T}}_t|$. Consider the support estimation step. Since condition 1 of the theorem holds, we can apply Lemma 2.2.7 with $S_{\mathcal{T}_t} = S$, $S_{\Delta_t} = 2S_a$. This gives $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.5\epsilon$. Using Proposition 2.3.1, this, along with conditions 2 and 3 implies that all elements of $\mathcal{N}_t \setminus \mathcal{B}_t$ will get detected and all zero elements will get deleted, i.e., there will be no false detections. Thus, $|\tilde{\Delta}_t| \leq |\mathcal{B}_t| \leq S_a$ and $|\tilde{\Delta}_{e,t}| = 0$ and so $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}| \leq S$. Thus the first claim holds.

The third claim follows using the second claim and Lemma 2.2.7.

A.0.3 Proof of Theorem 2.3.3

We prove the first three claims of the theorem by induction. Using condition 4 of the theorem, the claim holds for $t = 0$. This proves the base case. For the induction step, assume that the claim holds at $t-1$, i.e. $|\tilde{\Delta}_{e,t-1}| = 0$, $|\mathcal{T}_{t-1}| \leq S$, and $|\tilde{\Delta}_{t-1}| \leq S_a$. Using this, we prove the first three claims holds at t.

The bounding of $|\mathcal{T}_t|, |\Delta_t|, |\Delta_{e,t}|$ is exactly as in the proof of Theorem 2.3.2.

Consider the detection step. There are at most f false detects (from condition 1a) and thus $|\tilde{\Delta}_{e,\text{add},t}| \leq |\Delta_{e,t}| + f \leq S_a + f$. Thus $|\mathcal{T}_{\text{add},t}| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,\text{add},t}| \leq S + S_a + f$. So the third claim holds.

Next, consider $|\Delta_{\text{add},t}|$. Applying Lemma 2.2.7 with condition 2, i.e., $\delta_{|\mathcal{T}_t|+3|\Delta_t|} \leq \delta_{S+6S_a} \leq 0.207$, we have $\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon$. Thus, all elements of $\{i : |(\mathbf{x}_t)_i| > \alpha_{\text{add}} + 7.50\epsilon\}$ will definitely get detected at time t and so $\Delta_{\text{add},t} \subseteq \{i : |(\mathbf{x}_t)_i| \leq \alpha_{\text{add}} + 7.50\epsilon\}$. Since condition 3 holds, $\{i : |(\mathbf{x}_t)_i| \leq \alpha_{\text{add}} + 7.50\epsilon\} \subseteq \mathcal{B}_t$, and so $|\Delta_{\text{add},t}| \leq |\mathcal{B}_t| \leq S_a$.

Consider the deletion step. As $\Delta_{\text{add},t} \subseteq \mathcal{B}_t$, and $|(\mathbf{x}_t)_i| \leq \alpha_{\text{add}} + 7.50\epsilon$ for $i \in \Delta_{\text{add},t}$, we have $\|(\mathbf{x}_t)_{\Delta_{\text{add},t}}\| \leq \sqrt{S_a}(\alpha_{\text{add}} + 7.50\epsilon)$. Applying Lemma 2.2.9 with condition 2, i.e., $\delta_{|\mathcal{T}_{\text{add},t}|+|\Delta_{\text{add},t}|} = \delta_{S+2S_a+f} \leq 0.207$, we have $\|(\mathbf{x}_t - \mathbf{x}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\| \leq 1.12\epsilon + 0.261\sqrt{S_a}(\alpha_{\text{add}} + 7.50\epsilon)$. Thus, using these facts and condition 1b, all elements of $\tilde{\Delta}_{e,\text{add},t}$ will get deleted and elements of $\{i : |(\mathbf{x}_t)_i| > 2\alpha_{\text{del}}\}$ will not be deleted. Thus $|\tilde{\Delta}_{e,t}| = 0$, and since condition 3 holds, $\tilde{\Delta}_t \subseteq \{i : |(\mathbf{x}_t)_i| \leq 2\alpha_{\text{del}}\} \subseteq \mathcal{B}_t$, i.e., $|\tilde{\Delta}_t| \leq S_a$. Thus $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}| \leq S$. So the first claim holds.

The fourth claim follows using the previous claims and Lemma 2.2.7. The fifth claim follows using previous claims, Lemma 2.2.9.

A.0.4 Proof of Theorem 2.4.3

We prove the first claim by induction. Using condition 4 of the theorem, the claim holds for $t = 0$. This proves the base case. For the induction step, assume that the claim holds at $t-1$, i.e. $|\tilde{\Delta}_{e,t-1}| = 0$, $|\tilde{\mathcal{T}}_{t-1}| \leq S$, and $\tilde{\Delta}_{t-1} \subseteq \mathcal{S}_{t-1}(d_0)$ so that $|\tilde{\Delta}_{t-1}| \leq 2(d_0 - 1)S_a$. Using this we prove that the claim holds at t. In the proof, we use the following facts often: (a) $\mathcal{R}_t \subseteq \mathcal{N}_{t-1}$ and $\mathcal{A}_t \subseteq \mathcal{N}_{t-1}^c$, (b) $\mathcal{N}_t = \mathcal{N}_{t-1} \cup \mathcal{A}_t \setminus \mathcal{R}_t$, and (c) if two sets B, C are disjoint, then, $D \cup C \setminus B := (D \cup C) \setminus B = (D \cap B^c) \cup C$ for any set D.

We first bound $|\mathcal{T}_t|$, $|\Delta_{e,t}|$, $|\Delta_t|$. Since $\mathcal{T}_t = \tilde{\mathcal{T}}_{t-1} = \hat{\mathcal{N}}_{t-1}$, so $|\mathcal{T}_t| \leq S$. Also, $\Delta_{e,t} = \hat{\mathcal{N}}_{t-1} \setminus \mathcal{N}_t = \hat{\mathcal{N}}_{t-1} \cap [(\mathcal{N}_{t-1}^c \cap \mathcal{A}_t^c) \cup \mathcal{R}_t] \subseteq \tilde{\Delta}_{e,t-1} \cup \mathcal{R}_t = \mathcal{R}_t$. The last equality follows since $|\tilde{\Delta}_{e,t-1}| = 0$. Thus $|\Delta_{e,t}| \leq |\mathcal{R}_t| = S_a$.

Consider $|\Delta_t|$. Notice that $\Delta_t = \mathcal{N}_t \setminus \hat{\mathcal{N}}_{t-1} = (\mathcal{N}_{t-1} \cap \hat{\mathcal{N}}_{t-1}^c \cap \mathcal{R}_t^c) \cup (\mathcal{A}_t \cap \hat{\mathcal{N}}_{t-1}^c) = (\tilde{\Delta}_{t-1} \cap \mathcal{R}_t^c) \cup (\mathcal{A}_t \cap \hat{\mathcal{N}}_{t-1}^c) \subseteq (\mathcal{S}_{t-1}(d_0) \cap \mathcal{R}_t^c) \cup \mathcal{A}_t = \mathcal{S}_{t-1}(d_0) \cup \mathcal{A}_t \setminus \mathcal{R}_t$. Here we used $\tilde{\Delta}_{t-1} \subseteq \mathcal{S}_{t-1}(d_0)$. When $d_0 \geq 2, \mathcal{R}_t \subseteq \mathcal{S}_{t-1}(d_0)$ and $\mathcal{A}_t$ is disjoint with $\mathcal{S}_{t-1}(d_0)$. Thus $|\Delta_t| \leq |\mathcal{S}_{t-1}(d_0)| + |\mathcal{A}_t| - |\mathcal{R}_t| = 2(d_0 - 1)S_a + S_a - S_a$. When $d_0 = 1, \mathcal{S}_{t-1}(d_0) = \emptyset$, and $\mathcal{A}_t$ is disjoint with $\mathcal{R}_t$. Thus $|\Delta_t| \leq |\mathcal{A}_t \setminus \mathcal{R}_t| = |\mathcal{A}_t| = S_a$. Thus, $|\Delta_t| \leq k_1 S_a$.

Next we bound $|\tilde{\Delta}_t|$, $|\tilde{\Delta}_{e,t}|$, $|\tilde{\mathcal{T}}_t|$. Consider the support estimation step. Apply the first claim of Lemma 2.4.2 with $S_{\mathcal{N}} = S$, $S_{\Delta e} = S_a$, $S_\Delta = k_1 S_a$, and $b_1 = d_0 r$. Since

conditions 2 and 3 of the theorem hold, all elements of $\mathcal{N}_t$ with magnitude equal to or greater than $d_0 r$ will get detected. Thus, $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$. Apply the second claim of the lemma. Since conditions 2 and 1 hold, all zero elements will get deleted and there will be no false detections, i.e. $|\tilde{\Delta}_{e,t}| = 0$. Finally, $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}| \leq S + 0$.

The second claim for time t follows using the first claim for time $t-1$ and the arguments from the paras above. The third claim follows using the second claim and Lemma 2.2.7.

A.0.5 Proof of Theorem 2.4.8

We prove the first claim of the theorem by induction. Using condition 4 of the theorem, the claim holds for $t = 0$. This proves the base case. For the induction step, assume that the claim holds at $t-1$, i.e. $|\tilde{\Delta}_{e,t-1}| = 0$, $|\mathcal{T}_{t-1}| \leq S$, and $\tilde{\Delta}_{t-1} \subseteq \mathcal{S}_{t-1}(d_0)$ so that $|\tilde{\Delta}_{t-1}| \leq 2(d_0 - 1)S_a$. Using this, we prove that the claim holds at t. We will use the following facts often: (a) $\mathcal{R}_t \subseteq \mathcal{N}_{t-1}$, (b) $\mathcal{A}_t \subseteq \mathcal{N}_{t-1}^c$, (c) $\mathcal{N}_t = \mathcal{N}_{t-1} \cup \mathcal{A}_t \setminus \mathcal{R}_t$, and (d) if two sets B, C are disjoint, then, $D \cup C \setminus B := (D \cup C) \setminus B = (D \cap B^c) \cup C$ for any set D.

The bounding of $|\mathcal{T}_t|, |\Delta_t|, |\Delta_{e,t}|$ is exactly as in the proof of Theorem 2.4.3. Since $\mathcal{T}_t = \tilde{\mathcal{T}}_{t-1}$, so $|\mathcal{T}_t| \leq S$. Also, $\Delta_{e,t} = \hat{\mathcal{N}}_{t-1} \setminus \mathcal{N}_t = \hat{\mathcal{N}}_{t-1} \cap [(\mathcal{N}_{t-1}^c \cap \mathcal{A}_t^c) \cup \mathcal{R}_t] \subseteq \tilde{\Delta}_{e,t-1} \cup \mathcal{R}_t = \mathcal{R}_t$. Thus $|\Delta_{e,t}| \leq |\mathcal{R}_t| = S_a$. Finally, $\Delta_t = \mathcal{N}_t \setminus \hat{\mathcal{N}}_{t-1} = (\tilde{\Delta}_{t-1} \cap \mathcal{R}_t^c) \cup (\mathcal{A}_t \cap \hat{\mathcal{N}}_{t-1}^c) \subseteq (\mathcal{S}_{t-1}(d_0) \cap \mathcal{R}_t^c) \cup \mathcal{A}_t$. Thus,

$$\Delta_t \subseteq \mathcal{S}_{t-1}(d_0) \cup \mathcal{A}_t \setminus \mathcal{R}_t \tag{A.11}$$

When $d_0 \geq 2, \mathcal{R}_t \subseteq \mathcal{S}_{t-1}(d_0)$ and $\mathcal{A}_t$ is disjoint with $\mathcal{S}_{t-1}(d_0)$, so $|\Delta_t| \leq |\mathcal{S}_{t-1}(d_0)| + |\mathcal{A}_t| - |\mathcal{R}_t| = 2(d_0 - 1)S_a + S_a - S_a$. When $d_0 = 1, \mathcal{S}_{t-1}(d_0) = \emptyset$, and $\mathcal{A}_t$ is disjoint with $\mathcal{R}_t$, so $|\Delta_t| \leq |\mathcal{A}_t \setminus \mathcal{R}_t| = |\mathcal{A}_t| = S_a$. Thus, $|\Delta_t| \leq k_1 S_a$.

Consider the detection step. There are at most f false detects (from condition 1a) and thus $|\tilde{\Delta}_{e,\text{add},t}| \leq |\Delta_{e,t}| + f \leq S_a + f$. Thus $|\mathcal{T}_{\text{add},t}| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,\text{add},t}| \leq S + S_a + f$.

Next, consider $|\Delta_{\text{add},t}|$. Notice that

$$\begin{aligned}\Delta_t &\subseteq \mathcal{S}_{t-1}(d_0) \cup \mathcal{A}_t \setminus \mathcal{R}_t \\ &\subseteq \mathcal{S}_t(d_0) \cup \mathcal{I}_t(d_0) \setminus \mathcal{D}_t(d_0 - 1). \end{aligned} \tag{A.12}$$

The first $\subseteq$ is from (A.11), the second one follows by using (2.11) for $j = d_0$. Now, apply Lemma 2.4.6 with $S_{\mathcal{N}_t} = S$, $S_{\Delta_{e,t}} = S_a$, $S_{\Delta_t} = k_1 S_a$, and with $b_1 = d_0 r$. Using (A.12), $\{i \in \Delta_t : |(\mathbf{x}_t)_i| \geq b_1\} = \Delta_t \cap \mathcal{I}_t(d_0)$. Since conditions 2 and 3 hold, by Lemma 2.4.6, all elements of $\{i \in \Delta_t : |(\mathbf{x}_t)_i| \geq b_1\}$ will definitely get detected at time t. Thus $\Delta_{\text{add},t} \subseteq \Delta_t \setminus \{i \in \Delta_t : |(\mathbf{x}_t)_i| \geq b_1\} \subseteq \Delta_t \setminus \mathcal{I}_t(d_0)$. But from (A.12), $\Delta_t \setminus \mathcal{I}_t(d_0) \subseteq \mathcal{S}_t(d_0) \setminus \mathcal{D}_t(d_0 - 1)$. Since when $d_0 \geq 2$, $\mathcal{D}_t(d_0 - 1) \subseteq \mathcal{S}_t(d_0)$, then $|\Delta_{\text{add},t}| \leq |\mathcal{S}_t(d_0)| - |\mathcal{D}_t(d_0 - 1)| = 2(d_0 - 1)S_a - S_a$; when $d_0 = 1, \mathcal{D}_t(d_0 - 1) = \mathcal{S}_t(d_0) = \emptyset$, then $|\Delta_{\text{add},t}| = 0$. Thus, $|\Delta_{\text{add},t}| \leq k_2 S_a$

Consider the deletion step. Apply Lemma 2.4.7 with $S_{\mathcal{T}_{\text{add},t}} = S$, $S_{\Delta_{\text{add},t}} = k_1 S_a$. Since condition 2b holds, $\delta_{S+S_a+f} < 1/2$ holds. Since $\Delta_{\text{add},t} \subseteq \mathcal{S}_t(d_0) \setminus \mathcal{D}_t(d_0 - 1)$, $\Delta_{\text{add},t}$ contains only $2S_a$ elements of magnitude $\{r, 2r, \cdots, (d_0 - 2)r\}$ and S_a elements of magnitude $(d_0 - 1)r$. Thus, $\|(\mathbf{x}_t)_{\Delta_{\text{add},t}}\| \leq k_3 \sqrt{S_a} r$. Using these facts and condition 1b, by Lemma 2.4.7, all elements of $\tilde{\Delta}_{e,\text{add},t}$ will get deleted. Thus $|\tilde{\Delta}_{e,t}| = 0$. Thus $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}| \leq S$.

To bound $|\tilde{\Delta}_t|$, apply Lemma 2.4.7 with $S_{\mathcal{T}_{\text{add},t}} = S + S_a + f$, $S_{\Delta_{\text{add},t}} = k_2 S_a$, $b_1 = d_0 r$. By Lemma 2.4.7, to ensure that all elements of $\{i \in \mathcal{T}_{\text{add},t} : |(\mathbf{x}_t)_i| \geq b_1\}$ do not get falsely deleted, we need $\delta_{S_0+S_a+f} < 1/2$ and $d_0 r > \alpha_{\text{del}} + \frac{\zeta_L}{\sqrt{S_a}}(\sqrt{2}\epsilon + 2\theta_{S_0+S_a+f,k_2 S_a} k_3 \sqrt{S_a} r)$. From condition 1b, $\alpha_{\text{del}} = \sqrt{\frac{2}{S_a}}\zeta_L \epsilon + 2k_3 \theta_{S+S_a+f,k_2 S_a} \zeta_L r$. Thus, we need $\delta_{S_0+S_a+f} < 1/2$ and $d_0 r > 2(\sqrt{\frac{2}{S_a}}\zeta_L \epsilon + 2k_3 \theta_{S+S_a+f,k_2 S_a} \zeta_L r)$. $\delta_{S_0+S_a+f} < 1/2$ holds since condition 2b holds. The second one holds since condition 2c and $r \geq G_2$ of condition 3 hold. Thus, we can ensure that all elements of $\{i \in \mathcal{T}_{\text{add},t} : |(\mathbf{x}_t)_i| \geq b_1\}$, i.e. all elements of $\mathcal{T}_{\text{add},t}$ with magnitude greater than or equal to $b_1 = d_0 r$ do not get falsely deleted. But nothing can be said about the elements smaller than $d_0 r$ (in the worst case all of them may get falsely

deleted). Thus, $\tilde{\Delta}_t \subseteq \mathcal{S}_t(d_0)$ and so $|\tilde{\Delta}_t| \leq 2(d_0-1)S_a$. This finishes the proof of the first claim. To prove the second and third claims for any $t > 0$: use the first claim for $t-1$ and the arguments from the paragraphs above to show that the second and third claim hold for t. The fourth claim follows using the previous claims and Lemma 2.2.7. The fifth claim follows using previous claims, Lemma 2.2.9 and a bound on $\|(\mathbf{x}_t)_{\tilde{\Delta}_t}\|_2$. It is easy to see that $\|(\mathbf{x}_t)_{\tilde{\Delta}_t}\|_2 \leq k_3\sqrt{S_a}r$.

A.0.6 Proof of Theorem 2.5.5

Recall from the signal model that $|\mathcal{N}_t| \leq S$ for all t, and that $|\mathcal{SD}_t| \leq \frac{(b+1)}{2}S_d$. Also $\mathcal{N}_t = \cup_{\tau=t-d_{\min}+1}^{t}\mathcal{A}_\tau \cup \mathcal{L}_t \cup \mathcal{SD}_t$, noting that the first two sets might not be disjoint.

The proof follows using induction. The base case is easy. Assume that the result holds at $t-1$. At t, at most S_a new elements get added to the support, thus $|\Delta_t| \leq |\tilde{\Delta}_{t-1}| + S_a \leq \frac{(b+1)}{2}S_d + d_0S_a + S_a$. Also, since $\mathcal{T}_t = \tilde{\mathcal{T}}_{t-1}$, thus $|\mathcal{T}_t| \leq S$. And $\Delta_{e,t} = \tilde{\Delta}_{e,t-1} \cup R_t$, indicating $|\Delta_{e,t}| \leq |\tilde{\Delta}_{e,t-1}| + |R_t| \leq S_r$. The second condition of the theorem ensures that $\delta_{|\mathcal{T}_t|+3|\Delta_t|} \leq (\sqrt{2}-1)/2$. Thus using Lemma 2.2.7, $||\mathbf{x}_t - \hat{\mathbf{x}}_t|| \leq 7.50\epsilon$.

Consider the support detection step. Consider an $i \notin \mathcal{N}_t$, i.e. $(\mathbf{x}_t)_i = 0$. Since $\alpha = \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon \geq \frac{\zeta_M}{\sqrt{S_a}}||\mathbf{x}_t - \hat{\mathbf{x}}_t|| \geq ||\mathbf{x}_t - \hat{\mathbf{x}}_t||_\infty \geq |(\hat{\mathbf{x}}_t)_i|$, thus i will never get detected into the support estimate. Thus, $|\tilde{\Delta}_{e,t}| = 0$. Thus $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}| \leq S$.

The third condition ensures that any newly added element exceeds $\alpha + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon$ within d_0 time units and any element of $\mathcal{L}_t$ exceeds $\alpha + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon$ as $\ell > \alpha + \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon$. Consider any such element j. This means that $|(\hat{\mathbf{x}}_t)_j| \geq |(\mathbf{x}_t)_j| - |(\mathbf{x}_t - \hat{\mathbf{x}}_t)_j| \geq |(\mathbf{x}_t)_j| - ||\mathbf{x}_t - \hat{\mathbf{x}}_t||_\infty \geq |(\mathbf{x}_t)_j| - \frac{\zeta_M}{\sqrt{S_a}}||\mathbf{x}_t - \hat{\mathbf{x}}_t|| \geq |(\mathbf{x}_t)_j| - \frac{\zeta_M}{\sqrt{S_a}}7.50\epsilon \geq \alpha$. Thus such an element will definitely get detected into the support. This means that the only nonzero elements that are missed are either those that got added in the last d_0 frames or those that are currently decreasing. The maximum number of elements that got added in the last d_0 time units is d_0S_a. The maximum number of decreasing elements at t is less than or equal to $\frac{(b+1)}{2}S_d$. Thus, $|\tilde{\Delta}_t| \leq \frac{(b+1)}{2}S_d + d_0S_a$. This proves the induction step and hence of the theorem.

A.0.7 Proof of Theorem 2.5.9

Proposition A.0.1 (simple facts)**.** *Consider Algorithm 2.*

1. *An $i \in \mathcal{N}_t$ will definitely get detected if $|(\mathbf{x}_t)_i| > \alpha_{add} + \frac{\zeta_M}{\sqrt{S_a}}\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\|$.*
2. *An $i \in \mathcal{N}_t$ will definitely not be deleted if $|(\mathbf{x}_t)_i| > \alpha_{del} + \frac{\zeta_L}{\sqrt{S_a}}\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,add}\|$.*
3. *All $i \in \Delta_{e,t}$ (the zero elements of $\mathcal{T}_t$) will definitely get deleted if $\alpha_{del} \geq \|x - \hat{\mathbf{x}}_{t,add}\|_\infty$.*

Recall from the signal model that $\mathcal{N}_t = \cup_{\tau=t-d_{\min}+1}^{t} \mathcal{A}_\tau \cup \mathcal{L}_t \cup \mathcal{SD}_t$, noting that the first two sets might not be disjoint. By the induction assumption, $|\tilde{\mathcal{T}}_{t-1}| \leq S$. Since $\mathcal{T}_t = \tilde{\mathcal{T}}_{t-1} = \hat{\mathcal{N}}_{t-1}$, thus,

$$|\mathcal{T}_t| \leq S \tag{A.13}$$

Also, by the induction assumption,

$$\tilde{\Delta}_{t-1} \subseteq \mathcal{SD}_{t-1} \cup \mathcal{A}_{t-1} \ldots \mathcal{A}_{t-d_0} \tag{A.14}$$

Recall that $\mathcal{N}_t = \mathcal{N}_{t-1} \cup \mathcal{A}_t \setminus \mathcal{R}_t$. Also, $\mathcal{SD}_{t-1} \subseteq \mathcal{SD}_t \cup \mathcal{R}_t$. Thus, $\mathcal{SD}_{t-1} \cap \mathcal{R}_t^c \subseteq \mathcal{SD}_t$. Thus,

$$\begin{aligned}
\Delta_t = \mathcal{N}_t \cap \hat{\mathcal{N}}_{t-1}^c &= (\mathcal{N}_{t-1} \cap \mathcal{R}_t^c \cap \hat{\mathcal{N}}_{t-1}^c) \cup (\mathcal{A}_t \cap \hat{\mathcal{N}}_t^c) \\
&\subseteq (\tilde{\Delta}_{t-1} \cap \mathcal{R}_t^c) \cup \mathcal{A}_t \\
&\subseteq \mathcal{SD}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0} \cup \mathcal{A}_t
\end{aligned} \tag{A.15}$$

Thus,

$$|\Delta_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a + S_a \tag{A.16}$$

Using the above bounds on $|\mathcal{T}_t|$ and $|\Delta_t|$ and the RIP condition of the theorem, we can apply Lemma 2.2.7 to show that

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_{t,modcs}\| \leq 7.50\epsilon \tag{A.17}$$

Thus, using the Proposition A.0.1 and condition 3, all elements of $\mathcal{A}_{t-d_0}$ are definitely detected in the add step at t, i.e.

$$\mathcal{A}_{t-d_0} \subseteq \hat{\mathcal{A}}_t \tag{A.18}$$

Also since ℓ satisfies condition 3, all elements of $\mathcal{L}_t$ will be detected in the add step at t.

Using (A.18),

$$\begin{aligned}\Delta_{\text{add},t} = \Delta_t \setminus \hat{\mathcal{A}}_t &= \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0} \setminus \hat{\mathcal{A}}_t \\ &\subseteq \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0+1}\end{aligned} \tag{A.19}$$

Thus,

$$|\Delta_{\text{add},t}| \leq \frac{(b+1)}{2} S_a + d_0 S_a \tag{A.20}$$

Also, $\mathcal{T}_{\text{add},t} \subseteq \mathcal{N}_t \cup \Delta_{e,\text{add},t}$ and

$$\Delta_{e,\text{add},t} = \Delta_{e,t} \cup (\hat{\mathcal{A}}_t \setminus \mathcal{N}_t) \subseteq \tilde{\Delta}_{e,t-1} \cup \mathcal{R}_t \cup (\hat{\mathcal{A}}_t \setminus \mathcal{N}_t) \tag{A.21}$$

Thus, $|\Delta_{e,\text{add},t}| \leq S_a + f$ and so

$$|\mathcal{T}_{\text{add},t}| \leq S + |\Delta_{e,\text{add},t}| \leq S + S_a + f \tag{A.22}$$

By Lemma 2.2.9 and condition 2c of the Theorem, we have

$$\begin{aligned}\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})\| &\leq 1.12\epsilon + (1 + 1.261\theta_{|\mathcal{T}_{\text{add},t}|,|\Delta_{\text{add},t}|})\|(\mathbf{x}_t)_{\Delta_{\text{add},t}}\| \\ &\leq 1.12\epsilon + 1.261\|(\mathbf{x}_t)_{\Delta_{\text{add},t}}\|\end{aligned} \tag{A.23}$$

Recall that, by Proposition A.0.1, any element of $\mathbf{x}_{\Delta_{\text{add},t}}$ will have magnitude smaller than $\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon$. By (A.20), we have

$$\begin{aligned}\|\mathbf{x}_{\Delta_{\text{add},t}}\| &\leq \sqrt{|\Delta_{\text{add},t}|(\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon)} \\ &\leq \sqrt{(\frac{(b+1)}{2} S_a + d_0 S_a)(\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon)}\end{aligned} \tag{A.24}$$

Let $h = \sqrt{(\frac{(b+1)}{2} + d_0)(\alpha_{\text{add}} + \frac{\zeta_M}{\sqrt{S_a}} 7.50\epsilon)}$. Combining this with the bound on $|\mathcal{T}_{\text{add},t}|$ and $|\Delta_{\text{add},t}|$ we can bound the LS step error by a time-invariant quantity,

$$\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\| \leq 1.12\epsilon + 1.261h\sqrt{S_a} \tag{A.25}$$

Using Assumption 2.4.5, we have,

$$\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\|_\infty \leq 1.12\frac{\zeta_L}{\sqrt{S_a}}\epsilon + 0.261\zeta_L h \tag{A.26}$$

Using the fact that α_{del} is equal to the RHS of the above equation and proposition fact 3, if $(\mathbf{x}_t)_j = 0$, then $j \in \hat{\mathcal{R}}_t$. Thus,

$$\mathcal{N}_t^c \subseteq \hat{\mathcal{R}}_t \tag{A.27}$$

Next, using (2.18), (A.26), fact 2 of Proposition A.0.1 and the value of α_{del}, we can conclude the following: if $j \in \mathcal{L}_t, j$ will not get falsely deleted; the same is true if $j \in \mathcal{A}_\tau, \tau \leq t - d_0$. Thus,

$$\hat{\mathcal{R}}_t \subseteq \mathcal{N}_t^c \cup \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0+1} \tag{A.28}$$

Recall that $\hat{\mathcal{N}}_t = \hat{\mathcal{N}}_{t-1} \cup \hat{\mathcal{A}}_t \setminus \hat{\mathcal{R}}_t$. Thus

$$\begin{aligned}\tilde{\Delta}_t &= \mathcal{N}_t \setminus \hat{\mathcal{N}}_t = (\mathcal{N}_t \cap \hat{\mathcal{N}}_{t-1}^c \cap \hat{\mathcal{A}}_t^c) \cup (\mathcal{N}_t \cap \hat{\mathcal{R}}_t) \\ &\subseteq (\Delta_t \cap \hat{\mathcal{A}}_t^c) \cup (\mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \ldots \mathcal{A}_{t-d_0+1})\end{aligned} \tag{A.29}$$

Since $\mathcal{A}_{t-d_0} \subset \hat{\mathcal{A}}_t$, using (A.15), we get

$$\Delta_t \cap \hat{\mathcal{A}}_t^c \subseteq \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0+1} \tag{A.30}$$

Thus, using (A.29),

$$\tilde{\Delta}_t \subseteq \mathcal{SD}_t \cup \mathcal{A}_t \cup \mathcal{A}_{t-1} \cdots \cup \mathcal{A}_{t-d_0+1} \tag{A.31}$$

Thus,

$$|\tilde{\Delta}_t| \leq \frac{(b+1)}{2} S_a + d_0 S_a \tag{A.32}$$

Now consider $\tilde{\Delta}_{e,t}$.

$$\begin{aligned}\tilde{\Delta}_{e,t} &= \hat{\mathcal{N}}_t \setminus \mathcal{N}_t \\ &= (\hat{\mathcal{N}}_{t-1} \cap \hat{\mathcal{R}}_t^c \cap \mathcal{N}_t^c) \cup (\hat{\mathcal{A}}_t \cap \hat{\mathcal{R}}_t^c \cap \mathcal{N}_t^c)\end{aligned}$$

As $\mathcal{N}_t^c \subseteq \hat{\mathcal{R}}_t$, we have $\hat{\mathcal{R}}_t^c \subseteq \mathcal{N}_t$. Thus,

$$\tilde{\Delta}_{e,t} = \emptyset \tag{A.33}$$

Thus,

$$|\tilde{\Delta}_{e,t}| = 0 \tag{A.34}$$

Since $|\mathcal{N}_t| \leq S$ and since $|\tilde{\mathcal{T}}_t| \leq |\mathcal{N}_t| + |\tilde{\Delta}_{e,t}|$, thus

$$|\tilde{\mathcal{T}}_t| \leq S \tag{A.35}$$

By condition 2,

$$\begin{aligned}\theta_{|\tilde{\mathcal{T}}_t|,|\tilde{\Delta}_t|} &\leq \theta_{S,\frac{b+1}{2}S_a+d_0S_a+S_a} \\ &\leq \delta_{S+3(\frac{b+1}{2}S_a+d_0S_a+S_a)} \leq 0.207\end{aligned} \tag{A.36}$$

and

$$\delta_{|\tilde{\mathcal{T}}_t|} \leq \delta_S \leq \delta_{S+S_a+f} \leq 0.207$$

Using the same way as getting $\|(\mathbf{x}_t - \hat{\mathbf{x}}_{t,\text{add}})\|$, we have

$$\|(\mathbf{x}_t - \hat{\mathbf{x}}_t)\| \leq 1.12\epsilon + 1.261\|\mathbf{x}_{\tilde{\Delta}_t}\|$$

Also, using Proposition A.0.1, any element of $\mathbf{x}_{\tilde{\Delta}_t}$ will have magnitude smaller than $\alpha_{\text{del}} + 1.12\frac{\zeta_L}{\sqrt{S_a}}\epsilon$. By (A.32), we have

$$\|\mathbf{x}_{t,\tilde{\Delta}_t}\| \leq \sqrt{(\frac{(b+1)}{2}S_a + d_0S_a)(\alpha_{\text{del}} + 1.12\frac{\zeta_L}{\sqrt{S_a}}\epsilon)}$$

Thus, the final claim is proved.

A.0.8 Proof of Remark 2.3.4: necessary and sufficient conditions

Necessity: Consider the noise-free case, i.e. $\epsilon = 0$ and Algorithm 1. We claim that $\delta_{S+S_a,\text{left}} < 1$ at all times $t > 0$ is necessary to ensure exact recovery of all sparse signal sequences with support size at most S, and number of support additions and removals at most S_a. We prove this here. Assume exact recovery at $t-1$. Assume also that the support size at $t-1$ is S, there are S_a new additions and S_a new removals at time t. Thus support size at time t is also S.

Suppose that $\delta_{S+S_a,\text{left}} < 1$ does not hold. This means there is a set, R, of size $S+S_a$ for which $rank((A_t)_R) < S+S_a$. Pick a z so that $z_R \in null((A_t)_R)$ (i.e. $(A_t)_R z_R = 0$) and $z_{R^c} = 0$. Partition R into three sets $R = D \cup D_1 \cup D_2$ s.t. all are disjoint; $|D| = S - S_a$, $|D_1| = S_a = |D_2|$ and $\|z_{D_2}\|_1 \leq \|z_{D_1}\|_1$. Create two sparse vectors x^1 and x^2 supported on $D \cup D_1$ and $D \cup D_2$ respectively as follows. Let $(x^1)_D = z_D/2$, $(x^1)_{D_1} = z_{D_1}$, $(x^1)_{(D\cup D_1)^c} = 0$. Let $(x^2)_D = -z_D/2$, $(x^2)_{D_2} = -z_{D_2}$, $(x^2)_{(D\cup D_2)^c} = 0$. Then both x^1 and x^2 have support size S.

Suppose that the signal at time t is x^1, i.e. $\mathbf{x}_t = x^1$ so that $y_t = A_t x^1$, and suppose that the support (equal to support estimate) from $t-1$ is $T = D \cup \Delta_e$ where Δ_e is a subset of $(D \cup D_1 \cup D_2)^c$ of size S_a. Consider the solution of modified-CS with $\epsilon = 0$. In this case, both x^1 and x^2 are feasible since $A_t(x^1 - x^2) = (A_t)_D z_D/2 + (A_t)_{D_1} z_{D_1} - (A_t)_D(-z_D/2) - (A_t)_{D_2}(-z_{D_2}) = (A_t)_R z_R$. But, $\|(x^1)_{D^c}\| = \|(x^1)_{D_1}\|_1 = \|z_{D_1}\|_1 \geq \|z_{D_2}\|_1 = \|(x^2)_{D^c}\|_1$. Thus, clearly x^1 will not be the unique solution to modified-CS with $\epsilon = 0$. This proves that $\delta_{S+S_a,\text{left}} < 1$ is necessary.

Sufficiency: Assume exact recovery at $t-1$, i.e., $\mathcal{T}_t = \tilde{\mathcal{T}}_{t-1} = \mathcal{N}_{t-1}$, $\Delta_t = \mathcal{N}_t \setminus \mathcal{T}_t = \mathcal{N}_t \setminus \mathcal{N}_{t-1}$, i.e., $|\mathcal{T}_t| \leq S$, $|\Delta_t| \leq S_a$, thus by Lemma 2.2.7 and $\delta_{S+3S_a} < 0.5$, we have $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\| \leq 0$, i.e., $\hat{\mathbf{x}}_t = \mathbf{x}_t$.

A.0.9 Generative model for Signal Model 2:

This model requires that when a new element j gets added to the support, its magnitude keeps increasing at rate $r_{j,t}$ until it reaches large set, and that an element i of the large set starts to decrease at rate $r_{i,t}$ until it reaches 0. The sign is selected as +1 or -1 with equal probability when the element gets added to the support, but remains the same after that. We can choose values for $a_{\min}, d_{\min}, r_{\min}(d_{\min}), S_a, m, b$ during simulation.

Mathematically, it can be described as follows. Let $(\mathbf{x}_t)_j = (M_t)_j(s_t)_j$ where $(M_t)_j$ denotes the magnitude and $(s_t)_j$ denotes the sign of $(\mathbf{x}_t)_j$ at time t. $\mathbf{x}_t$ is a $m \times 1$ vector; $S_0 = [\mu_1 S]$, here μ_1 is a random number between 0.9 and 1.

For $1 \leq t \leq b$, let $S_{a,t} = 0$, $S_{r,t} = 0$, $S_{d,t} = S_a$; For any $t > b$, do the following.

1. Generate

   (a) the new addition set, $\mathcal{A}_t$, of size $S_{a,t} = [\mu_2(\Sigma_{\tau=1}^{t-b} S_{d,\tau} - \Sigma_{\tau=1}^{t-1} S_{a,\tau})]$ (here μ_2 is a random number between 0.9 and 1) uniformly at random from ${\mathcal{N}_{t-1}}^c$,

   (b) the new decreasing set, $\mathcal{B}_t$, of size $S_{d,t} = [\mu_3 S_a]$ (here μ_3 is a random number between 0.5 and 1) uniformly at random from $\mathcal{L}_{t-1}$, and

   (c) the new deleted set, $\mathcal{R}_t$, of size $S_{r,t} = [\mu_4 |\mathcal{SD}_{t-1}|]$ (here μ_4 is a random number between 0.1 and 0.3), as the smallest $S_{r,t}$ elements of $\mathcal{SD}_{t-1}$.

2. Update the coefficients' magnitudes as follows.

$$(M_t)_i = \begin{cases} (M_{t-1})_i + r_{i,t}, & i \in \mathcal{A}_{t-d_{\min}} \cup \mathcal{L}_{t-1} \setminus \mathcal{B}_t, r_{j,t} = \mu_5; \\ (M_{t-1})_i + r_{i,t}, & i \in \cup_{\tau=t-d_{\min}+1}^{t} \mathcal{A}_\tau, r_{i,t} = \mu_6 r_{\min}(d_{\min}); \\ (M_{t-1})_i - r_{i,t}, & i \in \mathcal{SD}_{t-1} \setminus \mathcal{R}_t, r_{i,t} = \mu_7 \frac{\ell}{b}; \\ (M_{t-1})_i - r_{i,t}, & i \in \mathcal{B}_t, r_{i,t} = \mu_8 (M_{i,t-1} - \ell); \\ 0, & i \in \mathcal{N}_t^c. \end{cases}$$

where μ_6, μ_7 and μ_8 are random numbers between 1 and 1.44; μ_5 is a random number larger than $-((M_{t-1})_i - \ell)$.

3. Update the signs as follows.

$$(s_t)_i = \begin{cases} (s_{t-1})_i, & i \in \mathcal{N}_t \setminus \mathcal{A}_t \\ iid(\pm 1), & i \in \mathcal{A}_t \\ 0, & i \in \mathcal{N}_t^c \end{cases} \tag{A.37}$$

where $iid(\pm 1)$ refers to generating the sign as +1 or -1 with equal probability and doing this independently for each element i.

4. Set $(\mathbf{x}_t)_i = (M_t)_i (s_t)_i$ for all i.

5. Update

$$\mathcal{L}_t = \mathcal{A}_{t-d_{\min}} \cup \mathcal{L}_{t-1} \setminus \mathcal{B}_t,$$
$$\mathcal{SD}_t = \mathcal{SD}_{t-1} \cup \mathcal{B}_t \setminus \mathcal{R}_t.$$

APPENDIX B. PROOF OF THE LEMMAS IN CHAPTER 3

B.0.10 Derivation for (1.5)

Recall from Sec 1.1.2 that $r_{\text{new}} = \text{rank}(\mathbf{L}_{\text{new}})$,

$$\mathbf{L}_{\text{new}} = (\mathbf{I} - \mathbf{G}\mathbf{G}^*)\mathbf{L} \overset{\text{SVD}}{=} \mathbf{U}_{\text{new}}\mathbf{\Sigma}_{\text{new}}\mathbf{V}^*_{\text{new}} \tag{B.1}$$

Let $\mathbf{U}_0$ be a basis matrix for $\text{range}(\mathbf{L}) \cap \text{range}(\mathbf{G}) = \text{range}(\mathbf{U}) \cap \text{range}(\mathbf{G})$ with $r_0 = \text{rank}(\mathbf{U}_0)$ Thus, there exist rotation matrices $\mathbf{R}_1, \mathbf{R}_G$ and basis matrices $\mathbf{U}_1, \mathbf{G}_{\text{extra}}$ such that

$$\mathbf{U}\mathbf{R}_1 = [\mathbf{U}_0\ \mathbf{U}_1] \text{ and } \mathbf{G}\mathbf{R}_G = [\mathbf{U}_0\ \mathbf{G}_{\text{extra}}] \tag{B.2}$$

with $\mathbf{G}_{\text{extra}}{}^*\mathbf{U}_1 = 0$.

Clearly, $\text{rank}(\mathbf{U}_1) = r_{\text{new}}$ [1]. Split the $r \times r$ matrix $\mathbf{R}_1$ as $\mathbf{R}_1 = [(\mathbf{R}_1)_0,\ (\mathbf{R}_1)_1]$ so that $(\mathbf{R}_1)_0$ contains the first r_0 columns and $(\mathbf{R}_1)_1$ contains the last r_{new} columns. Thus,

$$\mathbf{L}_{\text{new}} = (\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^*)[\mathbf{U}_0\ \mathbf{U}_1]\mathbf{R}_1^*\mathbf{\Sigma}\mathbf{V}^* = \mathbf{U}_1(\mathbf{R}_1)_1^*\mathbf{\Sigma}\mathbf{V}^*.$$

Let $((\mathbf{R}_1)_1^*\mathbf{\Sigma}\mathbf{V}^*) \overset{\text{SVD}}{=} \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^*$ denote its full SVD. Thus $\mathbf{L}_{\text{new}} = \mathbf{U}_1\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^*$. Comparing with the SVD of $\mathbf{L}_{\text{new}}$ we get that $\mathbf{U}_{\text{new}} = \mathbf{U}_1\mathbf{U}_2$ where $\mathbf{U}_2$ is a $r_{\text{new}} \times r_{\text{new}}$ unitary matrix; $\mathbf{\Sigma}_{\text{new}} = \mathbf{\Sigma}_2$ and $\mathbf{V}_{\text{new}} = \mathbf{V}_2$. Thus,

$$\mathbf{U}\mathbf{R}_1 = [\mathbf{U}_0\ \mathbf{U}_{\text{new}}\mathbf{U}_2^*] = [\mathbf{U}_0\ \mathbf{U}_{\text{new}}]\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}_2^*\end{pmatrix} \tag{B.3}$$

[1] This follows because $(\mathbf{I} - \mathbf{G}\mathbf{G}^*)\mathbf{L} = (\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^*)[\mathbf{U}_0\ \mathbf{U}_1]\mathbf{R}_1^{-1}\mathbf{\Sigma}\mathbf{V}^* = [0\ \mathbf{U}_1]\mathbf{R}_1^*\mathbf{\Sigma}\mathbf{V}^*$. Since $\text{rank}([0\ \mathbf{U}_1]) = \text{rank}(\mathbf{U}_1)$ and all other matrices are full rank r, we get that $\text{rank}(\mathbf{U}_1) = \text{rank}(\mathbf{L}_{\text{new}}) = r_{\text{new}}$. Here we have used Sylvester's inequality on $\mathbf{L}_{\text{new}} = [0\ \mathbf{U}_1](\mathbf{R}_1^*\mathbf{\Sigma}\mathbf{V}^*)$ to get that $\text{rank}(\mathbf{U}_1) + r - r \leq \text{rank}(L_{\text{new}}) = r_{\text{new}} \leq \min(\text{rank}(\mathbf{U}_1), r) = \text{rank}(\mathbf{U}_1)$.

By taking $\mathbf{R}_U = \mathbf{R}_1 \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2^* \end{pmatrix}^{-1} = \mathbf{R}_1 \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & U_2 \end{pmatrix}$, we get

$$\mathbf{U}\mathbf{R}_U = [\mathbf{U}_0 \ \mathbf{U}_{\text{new}}] \text{ and } \mathbf{G}\mathbf{R}_G = [\mathbf{U}_0 \ \mathbf{G}_{\text{extra}}] \tag{B.4}$$

Rearranging, we get (1.5).

B.0.11 Proof of Lemma 3.3.1

First we state and prove the following fact[2].

Proposition B.0.2. *Assume $m_1 < m_2 < n_1 n_2$, we have*

$$\mathbb{P}_{Unif(m_1)}(Success) \geq \mathbb{P}_{Unif(m_2)}(Success).$$

There are a total of $\binom{n_1 n_2}{m_2}$ size-m_2 subsets of the set of indices of an $n_1 \times n_2$ matrix. The probability of any one of them getting selected is $1/\binom{n_1 n_2}{m_2}$ under the Unif(m_2) model. Suppose that the algorithm succeeds for k out of these $\binom{n_1 n_2}{m_2}$ sets. Call these the "good" sets. Then,

$$\mathbb{P}_{\text{Unif}(m_2)}(\text{Success}) = \frac{k}{\binom{n_1 n_2}{m_2}}.$$

By Theorem 2.2 of [32], the algorithm definitely also succeeds for all size-m_1 subsets of these k "good" size-m_2 sets. Let k_1 be the number of such size m_1 subsets. Under the Unif(m_1) model, the probability of any one such set getting selected is $\frac{1}{\binom{n_1 n_2}{m_1}}$. Thus $\mathbb{P}_{\text{Unif}(m_1)}(\text{Success}) = \frac{k_1}{\binom{n_1 n_2}{m_1}}$.

Now we need to lower bound k_1. There are a total of $\binom{n_1 n_2}{m_2}$ size-m_2 sets and each of them has $\binom{m_2}{m_1}$ subsets of size m_1. However, the total number of distinct size-m_1 sets is only $\binom{n_1 n_2}{m_1}$. Because of symmetry, this means that in the collection of all size-m_1 subsets of all size-m_2 sets, a given set is repeated $b = \frac{\binom{n_1 n_2}{m_2}\binom{m_2}{m_1}}{\binom{n_1 n_2}{m_1}}$ times.

In the sub-collection of size-m_1 subsets of the k "good" size-m_2 sets, the number of times a set is repeated is less than or equal to b. Also, the number of entries in

[2]This fact may seem intuitively obvious, however we cannot find a simpler proof for it than the one we give.

this collection (including repeated ones) is $k\binom{m_2}{m_1}$. Thus, the number of distinct size-m_1 subsets of the "good" sets is lower bounded by $\frac{k\binom{m_2}{m_1}}{b}$, i.e. $k_1 \geq \frac{k\binom{m_2}{m_1}}{b}$. Thus,

$$\mathbb{P}_{\text{Unif}(m_1)}(\text{Success}) \geq \frac{k\binom{m_2}{m_1}\binom{n_1n_2}{m_1}}{\binom{n_1n_2}{m_1}\binom{m_2}{m_1}\binom{n_1n_2}{m_2}} = \mathbb{P}_{\text{Unif}(m_2)}(\text{Success}).$$

Proof of Lemma 3.3.1. Denote by Ω_0 the support set. We have

$$\begin{aligned}
&\mathbb{P}_{\text{Ber}(\rho_0)}(\text{Success}) \\
=&\sum_{k=0}^{n_1n_2} \mathbb{P}_{\text{Ber}(\rho_0)}(\text{Success} \,|\, |\Omega_0| = k)\mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| = k) \\
\leq&\sum_{k=0}^{m_0-1} \mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| = k) + \\
&\sum_{k=m_0}^{n_1n_2} \mathbb{P}_{\text{Unif}(k)}(\text{Success})\mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| = k) \\
\leq&\mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| < m_0) + \mathbb{P}_{\text{Unif}(m_0)}(\text{Success}),
\end{aligned}$$

where we have used the fact that for $k \geq m_0$, $\mathbb{P}_{\text{Unif}(k)}(\text{Success}) \leq \mathbb{P}_{\text{Unif}(m_0)}(\text{Success})$ by Proposition B.0.2, and that the conditional distribution of Ω_0 given its cardinality is uniform. Thus,

$$\mathbb{P}_{\text{Unif}(m_0)}(\text{Success}) \geq \mathbb{P}_{\text{Ber}(\rho_0)}(\text{Success}) - \mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| < m_0).$$

Let random matrix $\mathbf{X}^{n_1\times n_2}$ be a matrix whose each entry is i.i.d. Bernoulli distributed as $\mathbb{P}(\mathbf{X}_{ij} = 1) = \rho_0, \mathbb{P}(\mathbf{X}_{ij} = 0) = 1-\rho_0$. Then, under the Bernoulli model, $|\Omega_0| = \sum_{i,j}\mathbf{X}_{ij}$, $\mathbb{E}[\sum_{i,j}\mathbf{X}_{ij}] = \mathbb{E}[|\Omega_0|] = \rho_0 n_1 n_2$, and $0 \leq \mathbf{X}_{ij} \leq 1$. Thus by the Hoeffding inequality, we have

$$\mathbb{P}(\mathbb{E}[\sum_{i,j}\mathbf{X}_{ij}] - \sum_{i,j}\mathbf{X}_{ij} \geq t) \leq \exp(-\frac{2t^2}{n_1n_2}).$$

As $\rho_0 = \frac{m_0}{n_1n_2} + \epsilon_0$, take $t = \epsilon_0 n_1 n_2$, we have

$$\mathbb{P}_{\text{Ber}(\rho_0)}(|\Omega_0| \leq m_0) = \mathbb{P}(\sum_{i,j}\mathbf{X}_{ij} \leq m_0) \leq \exp(-2\epsilon_0^2 n_1 n_2).$$

Thus $\mathbb{P}_{\text{Unif}(m_0)}(\text{Success}) \geq \mathbb{P}_{\text{Ber}(\rho_0)}(\text{Success}) - \exp(-2\epsilon_0^2 n_1 n_2)$. □

B.0.12 Proof of Lemma 3.3.2

Proof. First, we state the theorem used in this proof.

Lemma B.0.3. *[67, Theorem 2(10a)] For $n \times n$ matrix $\mathbf{A}$ with entries a_{ij}, let $a_{ij}, i \geq j$ be independent (not necessarily identically distributed) random variables bounded with a common bound K. Assume that for $i \geq j$, the a_{ij} have a common expectation $\mu = 0$ and variance σ^2. Define a_{ij} for $i < j$ by $a_{ij} = a_{ji}$. (The numbers K, μ, σ^2 will be kept fixed as the matrix dimension n will tend to infinity.) For k satisfying $K^2k^6/(4\sigma^2 n) < 1/2$, we have*

$$\mathbb{P}(\max_i(|\lambda_i(\mathbf{A})|) > 2\sigma\sqrt{n} + v) < \sqrt{n}\exp(-\frac{kv}{2\sigma\sqrt{n} + v}).$$

Note that in this theorem, variance is fixed to σ^2, but we have checked that the theorem holds for variance bounded by σ^2, and actually [109, Theorem 4], [110, Theorem 1.4] used or stated similar results.

Let

$$\mathbf{A} := \begin{pmatrix} 0 & \mathbf{E} \\ \mathbf{E}^* & 0 \end{pmatrix} \tag{B.5}$$

Notice that $\mathbf{A}$ is an $(n_1 + n_2) \times (n_1 + n_2)$ symmetric matrix that satisfies requirements of Lemma B.0.3. By Lemma B.0.3 with $K = 1, \mu = 0, \sigma = \sqrt{\rho_s}$ and setting $v = (0.3536 - 2\sqrt{\rho_s})\sqrt{n_1 + n_2}$, and $k = \rho_s^{1/3}(n_1 + n_2)^{1/6}$, we have

$$\begin{aligned} &P(\max_i |\lambda_i(\mathbf{A})| > 0.3536\sqrt{n_1 + n_2}) \\ \leq &\sqrt{n_1 + n_2}\exp(-\frac{\rho_s^{1/3}(n_1 + n_2)^{1/6} \cdot (0.3536 - 2\sqrt{\rho_s})\sqrt{n_1 + n_2}}{0.3536\sqrt{n_1 + n_2}}) \\ \leq &(n_1 + n_2)^{-10} < n_{(1)}^{-10} \end{aligned}$$

In the above, $v > 0$ because $\rho_s < 0.03$ and the second inequality holds because $\frac{(n_1+n_2)^{1/6}}{\log(n_1+n_2)} > \frac{10.5}{\rho_s^{1/3}(1-5.6561\sqrt{\rho_s})}$. Clearly,

$$\|\mathbf{A}\| = \sqrt{\|\mathbf{A}\mathbf{A}^*\|} = \sqrt{\left\|\begin{pmatrix} \mathbf{E}\mathbf{E}^* & 0 \\ 0 & \mathbf{E}^*\mathbf{E} \end{pmatrix}\right\|} = \sqrt{\|\mathbf{E}\mathbf{E}^*\|} = \|\mathbf{E}\| \tag{B.6}$$

Therefore, we have $P(\|\mathbf{E}\| > 0.5\sqrt{n_{(1)}}) < n_{(1)}^{-10}$. □

B.0.13 Implications of Assumption 3.1.2

We summarize here some important implications of Assumption 3.1.2.

Remark B.0.4. *By Assumption 3.1.2(a)(b)(c), we have*

$$\begin{aligned}
\rho_s &\le 1 - 1.5\max\left\{60\rho_r^{1/2}, 11C_{01}\rho_r^{1/2}, 0.11\right\} \\
&\le 1 - 1.5\max\left\{60\rho_r^{1/2}, 11C_{01}\rho_r^{1/2}, \tfrac{11\log^2 n_{(1)}}{n_{(2)}}\right\} \\
&< \left(1 - \frac{1.5\max\{60\rho_r^{1/2}, 11C_{01}\rho_r^{1/2}, \frac{11\log^2 n_{(1)}}{n_{(2)}}\}}{1.5\log n_{(1)}}\right)^{1.5\log n_{(1)}} \\
&< \left(1 - \frac{\max\{60\rho_r^{1/2}, 11C_{01}\rho_r^{1/2}, \frac{11\log^2 n_{(1)}}{n_{(2)}}\}}{\log n_{(1)}}\right)^{1.3\lceil\log n_{(1)}\rceil}
\end{aligned} \tag{B.7}$$

The third inequality holds because $0 < 1.5\max\left\{60\rho_r^{1/2}, 0.11\right\} \le 1.5\max\{60/10^2, 0.11\} < 1$*; and for fixed constant* $b > 1$*,* $(1-x/b)^b > 1-x$ *whenever* $x < 1$*. The fourth inequality holds since* $1.5\log n_{(1)} > 1.3\lceil\log n_{(1)}\rceil$ *for* $n_{(1)} \ge 1024$*.*

Remark B.0.5. *By Assumption 3.1.2(b)(c), we have*

$$\rho_s \le 0.0156 \le 1 - \tfrac{250C_{01}\rho_r}{\log n_{(1)}}. \tag{B.8}$$

This follows since $n_{(1)} \ge \exp(253.9618C_{01}\rho_r)$ *gives* $\frac{250C_{01}\rho_r}{\log n_{(1)}} \le 0.9844$*, and so* $1 - \frac{250C_{01}\rho_r}{\log n_{(1)}} \ge 0.0156$*.*

B.0.14 Proof of Lemma 3.3.8

The proof uses the following three lemmas.

Lemma B.0.6. *[63, Theorem 4.1][32, Theorem 2.6] Suppose* $\Omega_0 \sim Ber(\rho_0)$*. Then there is a numerical constant* C_{01} *such that for all* $\beta > 1$*,*

$$\|\mathbb{P}_\Pi - \rho_0^{-1}\mathbb{P}_\Pi\mathbb{P}_{\Omega_0}\mathbb{P}_\Pi\| \le \epsilon_0, \tag{B.9}$$

with probability at least $1 - 3n_{(1)}^{-\beta}$ *provided that* $\rho_0 \ge C_{01}\,\epsilon_0^{-2}\,\frac{\beta\rho_r}{\log n_{(1)}}$*.*

Lemma B.0.7. *[32, Lemma 3.1] Suppose $\mathcal{Z} \in \Pi$ is a fixed matrix, and $\Omega_0 \sim Ber(\rho_0)$. Then*

$$\|\mathcal{Z} - \rho_0^{-1}\mathbb{P}_\Pi \mathbb{P}_{\Omega_0}\mathcal{Z}\|_\infty \leq \epsilon_0 \|\mathcal{Z}\|_\infty \tag{B.10}$$

with probability at least $1 - 2n_{(1)}^{-11}$, provided that $\rho_0 \geq 60\,\epsilon_0^{-2}\,\frac{\rho_r}{\log n_{(1)}}$.

This is the same as Lemma 3.1 in [32] except that we derive an explicit expression for the lower bound on ρ_0. A proof for this can be found in the Appendix of [69].

Lemma B.0.8. *[63, Theorem 6.3][32, Lemma 3.2] Suppose $\mathcal{Z}$ is fixed, and $\Omega_0 \sim Ber(\rho_0)$. Then there is a constant $C_{03} > 0$ s.t.*

$$\|(\mathbf{I} - \rho_0^{-1}\mathbb{P}_{\Omega_0})\mathcal{Z}\| \leq C_{03}\sqrt{\frac{11 n_{(1)} \log n_{(1)}}{\rho_0}}\|\mathcal{Z}\|_\infty \tag{B.11}$$

with probability at least $1 - n_{(1)}^{-11}$, provided that $\rho_0 \geq \frac{11 \log n_{(1)}}{n_{(2)}}$.

In the following proof, we take

$$\epsilon = (\rho_r)^{1/4} \text{ and } q = 1 - \rho_s^{\frac{1}{1.3\lceil \log n_{(1)} \rceil}} \tag{B.12}$$

Notice from our assumption on ρ_r given in Assumption 3.1.2 that

$$\epsilon \leq (10^{-4})^{1/4} \leq e^{-1}.$$

Let $\mathcal{Z}_j = \mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* - \mathbb{P}_\Pi \mathbf{Y}_j$. Clearly, $\mathcal{Z}_j \in \Pi$. Notice that $\mathbf{Y}_j \in \Omega^\perp$,

$$\mathbf{Y}_j = \mathbf{Y}_{j-1} + q^{-1}\mathbb{P}_{\bar{\Omega}_j}\mathcal{Z}_{j-1}, \text{ and}$$

$$\mathcal{Z}_j = (\mathbb{P}_\Pi - q^{-1}\mathbb{P}_\Pi \mathbb{P}_{\bar{\Omega}_j}\mathbb{P}_\Pi)\mathcal{Z}_{j-1}.$$

Clearly, $\bar{\Omega}_j$ and $\mathcal{Z}_{j-1}$ are independent. Using (B.7) and (B.12), $q \geq \frac{60\sqrt{\rho_r}}{\log n_{(1)}}$. Thus, by Lemma B.0.7

$$\|\mathcal{Z}_j\|_\infty \leq \epsilon^j \|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty, \tag{B.13}$$

w.p. at least $1 - 2jn_{(1)}^{-11}$. By Lemma B.0.6 and (B.7), with probability at least $1 - 3jn_{(1)}^{-11}$,

$$\|\mathcal{Z}_j\|_F \leq \epsilon\|\mathcal{Z}_{j-1}\|_F \leq \epsilon^j \|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_F = \epsilon^j\sqrt{r}. \tag{B.14}$$

Proof of (a)

Proof. As

$$\mathbf{Y}_{j_0} = \sum_{j=1}^{j_0} q^{-1}\mathbb{P}_{\bar{\Omega}_j}\mathcal{Z}_{j-1}, \tag{B.15}$$

and $\mathbb{P}_{\Pi^\perp}\mathcal{Z}_j = 0$, so we have, with probability at least $1 - 3j_0 n_{(1)}^{-11}$,

$$\begin{aligned}
\|\mathbf{W}^L\| =& \|\mathbb{P}_{\Pi^\perp}\mathbf{Y}_{j_0}\| \le \sum_{j=1}^{j_0} \|q^{-1}\mathbb{P}_{\Pi^\perp}\mathbb{P}_{\bar{\Omega}_j}\mathcal{Z}_{j-1}\| \\
=& \sum_{j=1}^{j_0} \|\mathbb{P}_{\Pi^\perp}(q^{-1}\mathbb{P}_{\bar{\Omega}_j}\mathcal{Z}_{j-1} - \mathcal{Z}_{j-1})\| \\
\le& \sum_{j=1}^{j_0} \|q^{-1}\mathbb{P}_{\bar{\Omega}_j}\mathcal{Z}_{j-1} - \mathcal{Z}_{j-1}\| \\
\le& C_{03}\sqrt{\frac{11 n_{(1)}\log n_{(1)}}{q}} \sum_{j=1}^{j_0} \|\mathcal{Z}_{j-1}\|_\infty \\
& \text{(using Lemma B.0.8 and } q \ge \frac{11\log n_{(1)}}{n_{(2)}} \text{ by (B.7))} \\
\le& C_{03}\sqrt{\frac{11 n_{(1)}\log n_{(1)}}{q}} \sum_{j=1}^{j_0} \epsilon^{j-1}\|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty \\
& \text{(using Lemma B.0.7 and } q \ge \frac{60\rho_r^{1/2}}{\log n_{(1)}} \text{ by (B.7))} \\
<& C_{03}(1-\epsilon)^{-1}\sqrt{\frac{11 n_{(1)}\log n_{(1)}}{q}}\|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty \\
\le& C_{03}(1-\epsilon)^{-1}\sqrt{\frac{11\rho_r}{q\log n_{(1)}}} \\
& \text{(using } \|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty \le \sqrt{\frac{\rho_r}{n_{(1)}\log^2 n_{(1)}}} \text{ by (3.3))} \\
\le& \frac{\sqrt{11}C_{03}\rho_r^{1/4}}{\sqrt{60}(1-e^{-1})} \\
& \text{(using } q \ge \frac{60\sqrt{\rho_r}}{\log n_{(1)}} \text{ by (B.7) and } \epsilon \le e^{-1}) \\
\le& \frac{1}{16} \\
& \text{(using } \rho_r \le 7.2483\times 10^{-5} C_{03}^{-4} \text{ by Assu. 3.1.2(a))}
\end{aligned}$$

The fourth step holds with probability at least $1 - j_0 n_{(1)}^{-11}$ by applying Lemma B.0.8 j_0

times; the fifth holds with probability at least $1 - 2j_0 n_{(1)}^{-11}$ by applying Lemma B.0.7 j_0 times for each $\mathcal{Z}_j$ (similar to (B.13)). Since $j_0 = 1.3 \log n_{(1)} < n_{(1)}$ (for $n_{(1)}$ satisfying Assumption 3.1.2), the result follows.

□

Proof of (b)

Proof. Since $\mathbb{P}_\Omega \mathbf{Y}_{j_0} = 0$, we have

$$\mathbb{P}_\Omega(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbb{P}_{\Pi^\perp}\mathbf{Y}_{j_0}) = \mathbb{P}_\Omega(\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* - \mathbb{P}_\Pi \mathbf{Y}_{j_0}) = \mathbb{P}_\Omega(\mathcal{Z}_{j_0}),$$

and by (B.14), (B.12) and (B.7) ($q \geq \frac{11C_{01}\sqrt{\rho_r}}{\log n_{(1)}}$), we have

$$\|\mathbb{P}_\Omega(\mathcal{Z}_{j_0})\|_F \leq \|\mathcal{Z}_{j_0}\|_F \leq \epsilon^{j_0}\sqrt{r} \leq e^{-1.3\log n_{(1)}}\sqrt{r} = \frac{\sqrt{r}}{n_{(1)}^{1.3}}, \tag{B.16}$$

with probability at least $1 - 3j_0 n_{(1)}^{-11}$. Thus, when $\frac{\sqrt{r}}{n_{(1)}^{0.8}} < \frac{1}{4}$, e.g. $n_{(1)} \geq 102$, Lemma 3.3.8(b) holds with probability at least $1 - 3n_{(1)}^{-10}$. □

Proof of (c)

Proof. Recall that $\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^* + \mathbf{W}^L = \mathcal{Z}_{j_0} + \mathbf{Y}_{j_0}$, $\mathbb{P}_{\Omega^\perp}\mathbf{Y}_{j_0} = \mathbf{Y}_{j_0}$. From above,

$$\|\mathcal{Z}_{j_0}\|_\infty \leq \|\mathcal{Z}_{j_0}\|_F \leq \frac{\sqrt{r}}{n_{(1)}^{1.3}} < \frac{\lambda}{8} \tag{B.17}$$

by (B.16) with probability at least $(1 - 3n_{(1)}^{-10})$ when $\frac{\sqrt{r}}{n_{(1)}^{0.8}} < \frac{1}{8}$, e.g. $n_{(1)} \geq 1024$. Thus, we only need to show $\|\mathbf{Y}_{j_0}\|_\infty \leq \frac{11\lambda}{40}$. We have, with probability at least $1 - 2j_0 n_{(1)}^{-11}$,

$$\begin{aligned}
\|\mathbf{Y}_{j_0}\|_\infty &\leq q^{-1}\textstyle\sum_{j=1}^{j_0} \|\mathbb{P}_{\tilde{\Omega}_j}\mathcal{Z}_{j-1}\|_\infty \\
&\leq q^{-1}\textstyle\sum_{j=1}^{j_0} \|\mathcal{Z}_{j-1}\|_\infty \\
&\leq q^{-1}\textstyle\sum_{j=1}^{j_0} \epsilon^{j-1}\|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty \\
&\quad \text{(using Lemma B.0.7 and } q \geq \tfrac{60\rho_r^{1/2}}{\log n_{(1)}} \text{ by (B.7))} \\
&\leq q^{-1}\textstyle\sum_{j=1}^{j_0} \epsilon^{j-1}\sqrt{\frac{\rho_r}{n_{(1)}\log^2 n_{(1)}}} \\
&\quad \text{(using } \|\mathbf{U}_{\text{new}}\mathbf{V}_{\text{new}}^*\|_\infty \leq \sqrt{\tfrac{\rho_r}{n_{(1)}\log^2 n_{(1)}}} \text{ by (3.3))} \\
&\leq \tfrac{\lambda}{60(1-e^{-1})} < \tfrac{11\lambda}{40} \\
&\quad \text{(using } q \geq \tfrac{60\sqrt{\rho_r}}{\log n_{(1)}} \text{ by (B.7) and } \epsilon \leq e^{-1} \text{ by (B.12))}
\end{aligned} \tag{B.18}$$

The third step follows from Lemma B.0.7 with probability at least $1 - 2j_0 n_{(1)}^{-11}$. Thus, Lemma 3.3.8(c) holds with probability at least $1 - 2n_{(1)}^{-10}$.

To sum up, with the assumptions in Lemma 3.3.8, we have (a), (b), (c) of Lemma 3.3.8 hold with probability at least $1 - 11n_{(1)}^{-10}$. □

B.0.15 Proof of Lemma 3.3.9

The proof uses the following lemma.

Lemma B.0.9. *[32, Corollary 2.7] Assume that $\Omega_0 \sim Ber(\rho_0)$, $\mathbf{L}$ satisfies (3.1), (3.2) and (3.3), then there is a numerical constant C_{01} such that for all $\beta > 1$,*

$$\|\mathbb{P}_{\Omega_0}\mathbb{P}_{\Pi}\|^2 \leq \rho_0 + \epsilon_0,$$

with probability at least $1 - 3n_{(1)}^{-\beta}$ provided that $1 - \rho_0 \geq C_{01}\, \epsilon_0^{-2}\, \frac{\beta\rho_r}{\log n_{(1)}}$.

This is a direct corollary of Lemma B.0.6 stated earlier. It follows by replacing Ω by Ω_0^c in Lemma B.0.6.

Proof of (a)

Let $\mathbf{E} := \mathrm{sgn}(\mathbf{S})$. Recall from the assumption in this lemma that $\mathbf{E}$ satisfies the assumptions of Lemma 3.3.2.

By taking $\Omega_0 = \Omega$, $\rho_0 = \rho_s$, $\epsilon_0 = 0.2$, and $\beta = 10$ in Lemma B.0.9, and using (B.8), we get

$$\|\mathbb{P}_{\Omega}\mathbb{P}_{\Pi}\|^2 \leq \sigma := \rho_s + 0.2, \tag{B.19}$$

with probability at least $1 - 3n_{(1)}^{-10}$. Thus, using the bound on ρ_s from (B.8), we get that $\|\mathbb{P}_{\Omega}\mathbb{P}_{\Pi}\|^2 \leq 0.22 < 1/4$.

Proof of (b)

Proof. Note that

$$\mathbf{W}^S = \mathbb{P}_{\Pi^\perp}\lambda\mathbf{E} + \mathbb{P}_{\Pi^\perp}\lambda\sum_{k\geq 1}(\mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^k\mathbf{E}$$
$$:= \mathbb{P}_{\Pi^\perp}\mathbf{W}_0^S + \mathbb{P}_{\Pi^\perp}\mathbf{W}_1^S.$$

By Assumption 3.1.2(b)(e) and Lemma 3.3.2, we have

$$\|\mathbf{E}\| \leq 0.5\sqrt{n_{(1)}}$$

with probability at least $1 - n_{(1)}^{-10}$. Since $\lambda = 1/\sqrt{n_{(1)}}$, we have

$$\|\mathbb{P}_{\Pi^\perp}\mathbf{W}_0^S\| \leq \|\mathbf{W}_0^S\| = \lambda\|\mathbf{E}\| \leq 0.5,$$

with probability at least $1 - n_{(1)}^{-10}$.

Let $\mathcal{R} = \sum_{k\geq 1}(\mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^k$. Let N_1, N_2 denote 1/2-nets for $\mathbf{S}^{n_1-1}, \mathbf{S}^{n_2-1}$ where $\mathbf{S}^{n_1-1}$ is a unit Euclidean sphere in $\mathbb{R}^{n_1}$.A subset N of $\mathbb{R}^{n_1}$ is referred to as a ξ-net, if and only if, for every $\mathbf{y} \in \mathbb{R}^{n_1}$, there is a $\mathbf{y}_1 \in N$ for which $\|\mathbf{y} - \mathbf{y}_1\| \leq \xi$ (here we used the Euclidean distance metric) [68].

By [68, Lemma 5.2], the cardinality of the 1/2-nets N_1 and N_2 is 5^{n_1} and 5^{n_2} respectively.

By [68, Lemma 5.4],

$$\begin{aligned}\|\mathcal{R}(\mathbf{E})\| &= \sup_{x\in\mathbf{S}^{n_2-1}, y\in\mathbf{S}^{n_1-1}} \langle y, \mathcal{R}(\mathbf{E})x\rangle \\ &\leq 4 \sup_{x\in N_2, y\in N_1} \langle y, \mathcal{R}(\mathbf{E})x\rangle. \end{aligned} \tag{B.20}$$

For a fixed pair (y, x) of unit-normed vectors in $N_1 \times N_2$, define the random variable

$$\mathbf{X}(x, y) := \langle y, \mathcal{R}(\mathbf{E})x\rangle = \langle \mathcal{R}(yx^*), \mathbf{E}\rangle.$$

Conditional on $\Omega = \text{supp}(\mathbf{E})$, the signs of $\mathbf{E}$ are i.i.d. symmetric and Hoeffding's inequality gives

$$\mathbb{P}(|\mathbf{X}(x, y)| > t \,|\, \Omega) \leq 2\exp\Big(-\frac{2t^2}{\|\mathcal{R}(yx^*)\|_F^2}\Big).$$

Now since $\|yx^*\|_F = 1$, the matrix $\mathcal{R}(yx^*)$ obeys $\|\mathcal{R}(yx^*)\|_F \leq \|\mathcal{R}\|$ and, therefore,

$$\mathbb{P}\Big(\sup_{x\in N_2, y\in N_1} |\mathbf{X}(x,y)| > t \,|\, \Omega\Big) \leq 2|N_1||N_2| \exp\Big(-\frac{2t^2}{\|\mathcal{R}\|^2}\Big).$$

On the event $\{\|\mathbb{P}_\Omega \mathbb{P}_\Pi\| \leq \sigma\}$,

$$\|\mathcal{R}\| \leq \sum_{k\geq 1} \sigma^{2k} = \frac{\sigma^2}{1-\sigma^2}$$

and, therefore, letting $\gamma = \frac{1-\sigma^2}{2\sigma^2}$, we have,

$$\begin{aligned}
&\mathbb{P}(\lambda\|\mathcal{R}(\mathbf{E})\| > \tfrac{27}{80}) \\
\leq &\mathbb{P}(\lambda\|\mathcal{R}(\mathbf{E})\| > \tfrac{27}{80}, \|\mathbb{P}_\Omega\mathbb{P}_\Pi\| \leq \sigma) + \mathbb{P}(\|\mathbb{P}_\Omega\mathbb{P}_\Pi\| > \sigma) \\
\leq &\mathbb{P}\Big(\sup\nolimits_{x\in N_2, y\in N_1} 4|\mathbf{X}(x,y)| > \tfrac{27\sqrt{n_{(1)}}}{80} \,|\, \|\mathbb{P}_\Omega\mathbb{P}_\Pi\| \leq \sigma\Big) + \\
&\mathbb{P}(\|\mathbb{P}_\Omega\mathbb{P}_\Pi\| > \sigma) \\
\leq &2|N_1||N_2| \exp\Big(-\tfrac{27^2 n_{(1)}\gamma^2}{12800}\Big) + \mathbb{P}(\|\mathbb{P}_\Omega\mathbb{P}_\Pi\| > \sigma) \\
\leq &2 \times 5^{2n_{(1)}} \exp\Big(-\tfrac{27^2 n_{(1)}\gamma^2}{12800}\Big) + 3n_{(1)}^{-10} \\
\leq &2\exp\Big(-n_{(1)}(0.0570\gamma^2 - \log 25)\Big) + 3n_{(1)}^{-10} \\
&(\text{as } \sigma = \rho_s + 0.2 \leq 0.2156, \Rightarrow 0.0570\gamma^2 - \log 25 \geq 2.7773) \\
\leq &5n_{(1)}^{-10} \text{ (when } 2.7773 n_{(1)} \geq 10 \log n_{(1)}, \text{ e.g., } n_{(1)} \geq 10.)
\end{aligned}$$

Thus

$$\|\mathbf{W}^S\| \leq 67/80,$$

with probability at least $1 - 5n_{(1)}^{-10}$. □

Proof of (c)

Proof. Observe that

$$\begin{aligned}
\mathbb{P}_{\Omega^\perp}\mathbf{W}^S &= \lambda\mathbb{P}_{\Omega^\perp}(\mathbf{I} - \mathbb{P}_\Pi)(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\mathbf{E} \\
&= -\lambda\mathbb{P}_{\Omega^\perp}\mathbb{P}_\Pi(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\mathbf{E}
\end{aligned}$$

Let $\mathbf{W}_3^S := \mathbb{P}_{\Omega^\perp}\mathbf{W}^S$. Clearly, for $(i,j) \in \Omega$, $(\mathbf{W}_3^S)_{i,j} = 0$ and for $(i,j) \in \Omega^c$, $(\mathbf{W}_3^S)_{i,j} = (-\lambda\mathbb{P}_\Pi(\mathbb{P}_\Omega - \mathbb{P}_\Omega\mathbb{P}_\Pi\mathbb{P}_\Omega)^{-1}\mathbf{E})_{i,j}$.

For $(i,j) \in \Omega^c$, it can be rewritten as

$$\begin{aligned}(\mathbf{W}_3^S)_{ij} &= \langle \mathbf{e}_i, \mathbf{W}_3^S \mathbf{e}_j \rangle = \langle \mathbf{e}_i \mathbf{e}_j^*, \mathbf{W}_3^S \rangle \\ &= \langle \mathbf{e}_i \mathbf{e}_j^*, -\lambda \mathbb{P}_\Pi \mathbb{P}_\Omega (\mathbb{P}_\Omega - \mathbb{P}_\Omega \mathbb{P}_\Pi \mathbb{P}_\Omega)^{-1} \mathbf{E} \rangle \\ &= \lambda \langle \mathbf{X}(i,j), \mathbf{E} \rangle\end{aligned}$$

where $\mathbf{X}(i,j) := -(\mathbb{P}_\Omega - \mathbb{P}_\Omega \mathbb{P}_\Pi \mathbb{P}_\Omega)^{-1} \mathbb{P}_\Omega \mathbb{P}_\Pi (\mathbf{e}_i \mathbf{e}_j^*)$. Conditional on $\Omega = \mathrm{supp}(\mathbf{E})$, the signs of $\mathbf{E}$ are i.i.d. symmetric, and Hoeffding's inequality gives

$$\mathbb{P}(|(\mathbf{W}_3^S)_{ij}| > t\lambda \,|\, \Omega) \le 2 \exp\Big(-\frac{2t^2}{\|\mathbf{X}(i,j)\|_F^2}\Big),$$

and, thus,

$$\mathbb{P}\Big(\sup_{i,j \in \Omega^c} |(\mathbf{W}_3^S)_{ij}| > t\lambda \,|\, \Omega\Big) \le 2 n_1 n_2 \exp\Big(-\frac{2t^2}{\sup_{i,j} \|\mathbf{X}(i,j)\|_F^2}\Big).$$

Since (3.11) holds, on the event $\{\|\mathbb{P}_\Omega \mathbb{P}_\Pi\| \le \sigma\}$, we have

$$\|\mathbb{P}_\Omega \mathbb{P}_\Pi (\mathbf{e}_i \mathbf{e}_j^*)\|_F \le \|\mathbb{P}_\Omega \mathbb{P}_\Pi\| \|\mathbb{P}_\Pi (\mathbf{e}_i \mathbf{e}_j^*)\|_F \le \sigma \sqrt{2\rho_r / \log^2 n_{(1)}}$$

On the same event, $\|(\mathbb{P}_\Omega - \mathbb{P}_\Omega \mathbb{P}_\Pi \mathbb{P}_\Omega)^{-1}\| \le (1-\sigma^2)^{-1}$ and, therefore,

$$\|\mathbf{X}(i,j)\|_F^2 \le \frac{2\sigma^2}{(1-\sigma^2)^2} \frac{\rho_r}{\log^2 n_{(1)}}.$$

Then unconditionally, letting $\gamma = \frac{(1-\sigma^2)^2}{2\sigma^2}$, we have

$$\begin{aligned}&\mathbb{P}\Big(\|\mathbb{P}_{\Omega^\perp} \mathbf{W}^S\|_\infty > \tfrac{\lambda}{2}\Big) = \mathbb{P}\Big(\|\mathbf{W}_3^S\|_\infty > \tfrac{\lambda}{2}\Big) \\ &\le 2 n_{(1)} n_{(2)} \exp\Big(-\frac{\log^2 n_{(1)} \gamma^2}{4\rho_r}\Big) + \mathbb{P}(\|\mathbb{P}_\Omega \mathbb{P}_\Pi\| \ge \sigma) \\ &\le 2 n_{(1)}^{-\frac{\log n_{(1)} \gamma^2}{4\rho_r} + 2} + 3 n_{(1)}^{-10} \\ &\le 5 n_{(1)}^{-10}\end{aligned}$$

The last bound follows since $\sigma = \rho_s + 0.2 \le 0.2156$ by (B.8) and so $\gamma \ge 9.7798$; and $n_{(1)} \ge \exp(0.5019 \rho_r)$ by Assumption 3.1.2(c).

To sum up, with the assumption in Lemma 3.3.9, we have (a), (b) in Lemma 3.3.9 hold with probability at least $1 - 10 n_{(1)}^{-10}$.

□

APPENDIX C. PROOF OF THE LEMMAS IN CHAPTER 4

C.1 Preliminaries

Lemma C.1.1. *[44, Lemma 2.10] Suppose that* $\mathbf{P}$, $\hat{\mathbf{P}}$ *and* $\mathbf{Q}$ *are three basis matrices. Also,* $\mathbf{P}$ *and* $\hat{\mathbf{P}}$ *are of the same size,* $\mathbf{Q}'\mathbf{P} = \mathbf{0}$ *and* $\|(\mathbf{I} - \hat{\mathbf{P}}\hat{\mathbf{P}}')\mathbf{P}\|_2 = \zeta_*$. *Then,*

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

Weyl's inequality [**?**] (simplified version) states the following

Theorem C.1.2. *Given two Hermitian matrices* $\mathbb{A}$ *and* $\mathbb{H}$,

$$\lambda_i(\mathbb{A}) - \|\mathbb{H}\|_2 \leq \lambda_i(\mathbb{A} + \mathbb{H}) \leq \lambda_i(\mathbb{A}) + \|\mathbb{H}\|_2$$

Davis and Kahan's $\sin\theta$ theorem [95] studies the rotation of eigenvectors by perturbation.

Theorem C.1.3 ($\sin\theta$ theorem [95])**.** *Given two Hermitian matrices* $\mathbb{A}$ *and* $\mathbb{H}$ *and suppose that* $\mathbb{A}$ *satisfies*

$$\mathbb{A} = \begin{bmatrix} \mathbf{E}\,\mathbf{E}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{E}' \\ \mathbf{E}_\perp{}' \end{bmatrix}$$

where $[\mathbf{E}\ \mathbf{E}_\perp]$ *is an orthonormal matrix. Suppose that* $\mathbb{A}+\mathbb{H}$ *can be decomposed as*

$$\mathbb{A}+\mathbb{H}=\begin{bmatrix}\mathbf{F}\,\mathbf{F}_\perp\end{bmatrix}\begin{bmatrix}\Lambda & 0\\ 0 & \Lambda_\perp\end{bmatrix}\begin{bmatrix}\mathbf{F}'\\ \mathbf{F}_\perp{}'\end{bmatrix}$$

where $[\mathbf{F}\ \mathbf{F}_\perp]$ *is another orthonormal matrix and is such that rank*$(\mathbf{F})$ = *rank*$(\mathbf{E})$. *Let* $\mathcal{R}:=(\mathbb{A}+\mathbb{H})\mathbf{E}-\mathbb{A}\mathbf{E}=\mathbb{H}\mathbf{E}$. *If* $\lambda_{\min}(\mathbf{A})>\lambda_{\max}(\Lambda_\perp)$, *then*

$$\|(\mathbf{I}-\mathbf{F}\mathbf{F}')\mathbf{E}\|_2\leq\frac{\|\mathcal{R}\|_2}{\lambda_{\min}(\mathbf{A})-\lambda_{\max}(\Lambda_\perp)}\leq\frac{\|\mathbb{H}\|_2}{\lambda_{\min}(\mathbf{A})-\lambda_{\max}(\Lambda_\perp)}.$$

Remark C.1.4. *In the above theorem, let* r = *rank*$(\mathbf{F})$. *If the decomposition of* $\mathbb{A}+\mathbb{H}$ *is obtained by EVD, then* $\lambda_{\max}(\Lambda_\perp)=\lambda_{r+1}(\mathbb{A}+\mathbb{H})\leq\lambda_{r+1}(\mathbb{A})+\|\mathbb{H}\|_2$. *The inequality follows using Weyl. Moreover, if* $\lambda_{\min}(\mathbf{A})>\lambda_{\max}(\mathbf{A}_\perp)$, *then* $\lambda_{r+1}(\mathbb{A})=\lambda_{\max}(\mathbf{A}_\perp)$. *Thus a useful corollary of the above result is the following. If* $\lambda_{\min}(\mathbf{A})-\lambda_{\max}(\mathbf{A}_\perp)-\|\mathbb{H}\|_2>0$, *then*

$$\|(\mathbf{I}-\mathbf{F}\mathbf{F}')\mathbf{E}\|_2\leq\frac{\|\mathbb{H}\|_2}{\lambda_{\min}(\mathbf{A})-\lambda_{\max}(\mathbf{A}_\perp)-\|\mathbb{H}\|_2}.$$

Lemma C.1.5 (Cauchy-Schwarz for a sum of vectors)**.** *For vectors* $\mathbf{x}_t$ *and* $\mathbf{y}_t$,

$$\left(\sum_{t=1}^{\alpha}\mathbf{x}_t{}'\mathbf{y}_t\right)^2\leq\left(\sum_t\|\mathbf{x}_t\|_2^2\right)\left(\sum_t\|\mathbf{y}_t\|_2^2\right)$$

Lemma C.1.6 (Cauchy-Schwarz for a sum of matrices)**.** *For matrices* $\mathbf{X}_t$ *and* $\mathbf{Y}_t$,

$$\left\|\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{X}_t\mathbf{Y}_t{}'\right\|_2^2\leq\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{X}_t\mathbf{X}_t{}'\right)\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t\mathbf{Y}_t{}'\right)$$

Proof of Lemma C.1.6.

$$\begin{aligned}
\left\|\sum_{t=1}^{\alpha} \mathbf{X}_t \mathbf{Y}_t{}'\right\|_2^2 &= \max_{\substack{\|\mathbf{x}\|=1 \\ \|\mathbf{y}\|=1}} \left|\mathbf{x}'\left(\sum_t \mathbf{X}_t \mathbf{Y}_t{}'\right)\mathbf{y}\right|^2 \\
&= \max_{\substack{\|\mathbf{x}\|=1 \\ \|\mathbf{y}\|=1}} \left|\sum_{t=1}^{\alpha} (\mathbf{X}_t{}'\mathbf{x})'(\mathbf{Y}_t{}'\mathbf{y})\right|^2 \\
&\leq \max_{\substack{\|\mathbf{x}\|=1 \\ \|\mathbf{y}\|=1}} \left(\sum_{t=1}^{\alpha} \|\mathbf{X}_t{}'\mathbf{x}\|_2^2\right)\left(\sum_{t=1}^{\alpha} \|\mathbf{Y}_t{}'\mathbf{y}\|_2^2\right) \\
&= \max_{\|\mathbf{x}\|=1} \mathbf{x}' \sum_{t=1}^{\alpha} \mathbf{X}_t \mathbf{X}_t{}' \, \mathbf{x} \cdot \max_{\|\mathbf{y}\|=1} \mathbf{y}' \sum_{t=1}^{\alpha} \mathbf{Y}_t \mathbf{Y}_t{}' \, \mathbf{y} \\
&= \lambda_{\max}\left(\sum_{t=1}^{\alpha} \mathbf{X}_t \mathbf{X}_t{}'\right) \lambda_{\max}\left(\sum_{t=1}^{\alpha} \mathbf{Y}_t \mathbf{Y}_t{}'\right)
\end{aligned}$$

The inequality is by Lemma C.1.5. The penultimate line is because $\|\mathbf{x}\|_2^2 = \mathbf{x}'\mathbf{x}$. Multiplying both sides by $\left(\frac{1}{\alpha}\right)^2$ gives the desired result. □

Lemma C.1.7 (Exchanging the order of a double sum)**.**

$$\sum_{t=0}^{\alpha-1} \sum_{\tau=0}^{t} f_{t,\tau} = \sum_{\tau=0}^{\alpha-1} \sum_{t=\tau}^{\alpha-1} f_{t,\tau}$$

Proof. Define [statement] to be the Boolean value of statement

$$\begin{aligned}
\sum_{t=0}^{\alpha-1} \sum_{\tau=0}^{t} f_{t,\tau} &= \sum_{t,\tau} [0 \leq \tau \leq t][0 \leq t \leq \alpha - 1] f_{t,\tau} \\
&= \sum_{t,\tau} [0 \leq \tau \leq t \leq \alpha - 1] f_{t,\tau} \\
&= \sum_{t,\tau} [0 \leq \tau \leq \alpha - 1][\tau \leq t \leq \alpha - 1] f_{t,\tau} \\
&= \sum_{\tau=0}^{\alpha-1} \sum_{t=\tau}^{\alpha-1} f_{t,\tau}
\end{aligned}$$

□

The following lemma follows in an exactly analogous fashion.

Lemma C.1.8 (Exchanging the order of a double sum)**.**

$$\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} f_{t,\tau} = \sum_{\tau=t_0}^{t_0+\alpha-1}\sum_{t=\tau}^{t_0+\alpha-1} f_{t,\tau}$$

Lemma C.1.9 (A summation used very often)**.** *We have*

$$\frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2(t-\tau)} = \frac{1}{1-b^2}(1-\frac{1}{\alpha}\frac{b^2(1-b^{2\alpha})}{1-b^2})$$

Thus

$$\frac{1}{1-b^2}(1-\frac{1}{\alpha}\frac{b^2}{1-b^2}) \leq \frac{1}{\alpha}\sum_{t=t_0}^{t_0+\alpha-1}\sum_{\tau=t_0}^{t} b^{2(t-\tau)} \leq \frac{1}{1-b^2}$$

Proof: $\sum_{\tau=t_0}^{t} b^{2(t-\tau)} = \frac{1}{1-b^2}(1-b^{2(t-t_0+1)})$. And $\sum_{t=t_0}^{t_0+\alpha-1}\frac{1}{1-b^2}(1-b^{2(t-t_0+1)}) = \frac{1}{1-b^2}(\alpha-\frac{b^2(1-b^{2\alpha})}{1-b^2})$

Lemma C.1.10. *Let X, Y, and Z be random variables. Assume that X is independent of $\{Y, Z\}$. Then*

$$\mathbb{E}[XY|Z] = \mathbb{E}[X]\mathbb{E}[Y|Z]$$

Proof. By the chain rule, $f_{X,Y|Z}(x,y|z) = f_{X|Y,Z}(x|y,z)f_{Y|Z}(y|z)$. Because X is independent of both Y and Z, $f_{X|Y,Z}(x|y,z) = f_X(x)$. □

Lemma C.1.11. *For an event $\mathcal{E}$ and random variable X, $\mathbb{P}(\mathcal{E}|X) \geq p$ for all $X \in \mathcal{C}$ implies that $\mathbb{P}(\mathcal{E}|X \in \mathcal{C}) \geq p$.*

Theorem C.1.12 (Matrix Azuma)**.** *[96, Theorem 7.1] Consider a finite adapted sequence $\mathbf{Z}_t$, $t = 1, 2, \ldots \alpha$, of $n \times n$ Hermitian matrices, and a fixed sequence $\mathbf{A}_t$ of Hermitian matrices that satisfy*

$$\mathbb{E}[\mathbf{Z}_t|\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{t-1}] = \mathbf{0} \quad \text{and} \quad {\mathbf{Z}_t}^2 \preceq {\mathbf{A}_t}^2 \quad \text{with probability 1.}$$

Define the variance parameter

$$\sigma^2 := \left\| \sum_t {\mathbf{A}_t}^2 \right\|_2.$$

Then, for all $\epsilon > 0$,

$$\Pr\left(\lambda_{\max}\left(\sum_t \mathbf{Z}_t\right) \geq \epsilon\right) \leq n \exp\left(\frac{-\epsilon^2}{8\sigma^2}\right)$$

The following corollary extends the above result to the case where the conditional expectation is not zero and when we also condition on another random variable.

Corollary C.1.13 (Matrix Azuma conditioned on another random variable for a nonzero mean Hermitian matrix)**.** *Consider an α-length sequence $\{\mathbf{Z}_t\}_{t=1,2,\ldots,\alpha}$ of random Hermitian matrices of size $n \times n$ and a random variable X that we condition on. Assume that, for all $X \in \mathcal{C}$, (i) $\Pr(b_1\mathbf{I} \preceq \mathbf{Z}_t \preceq b_2\mathbf{I}|X) = 1$, for $1 \leq t \leq \alpha$ and (ii) $b_3\mathbf{I} \preceq \frac{1}{\alpha}\sum_{t=1}^{\alpha} \mathbb{E}[\mathbf{Z}_t|\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{t-1}, X] \preceq b_4\mathbf{I}$. Then for all $\epsilon > 0$,*

$$\Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\right) \leq b_4 + \epsilon \middle| X\right) \geq 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right)$$
$$\Pr\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\right) \geq b_3 - \epsilon \middle| X\right) \geq 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right)$$

Proof. At certain places, where the meaning is clear, we use $\mathbb{E}_{t-1}[\mathbf{Z}_t|X]$ to refer to $\mathbb{E}[\mathbf{Z}_t|\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{t-1}, X]$

1. Let $\mathbf{Y}_t := \mathbf{Z}_t - \mathbb{E}_{t-1}(\mathbf{Z}_t|X)$. Clearly $\mathbb{E}_{t-1}(\mathbf{Y}_t|X) = \mathbf{0}$. Since for all $X \in \mathcal{C}$, $\Pr(b_1\mathbf{I} \preceq \mathbf{Z}_t \preceq b_2\mathbf{I}|X) = 1$ and since for an Hermitian matrix, $\lambda_{\max}(.)$ is a convex function, and $\lambda_{\min}(.)$ is a concave function, $b_1\mathbf{I} \preceq \mathbb{E}_{t-1}(\mathbf{Z}_t|X) \preceq b_2\mathbf{I}$ for all $X \in \mathcal{C}$. Therefore, $\Pr({\mathbf{Y}_t}^2 \preceq (b_2-b_1)^2\mathbf{I}|X) = 1$ for all $X \in \mathcal{C}$. Thus, for Theorem C.1.12, $\sigma^2 = \|\sum_{t=1}^{\alpha}(b_2-b_1)^2\mathbf{I}\|_2 = \alpha(b_2-b_1)^2$. For any $X \in \mathcal{C}$, applying Theorem C.1.12 for $\{\mathbf{Y}_t\}_{t=1,\ldots,\alpha}$ conditioned on X, we get that, for any $\epsilon > 0$,

$$\Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t\right) \leq \epsilon \middle| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

By Weyl's inequality, $\lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t) = \lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}(\mathbf{Z}_t - \mathbb{E}_{t-1}(\mathbf{Z}_t|X)) \geq \lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t) + \lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha} -\mathbb{E}_{t-1}(\mathbf{Z}_t|X))$.

Since $\lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha} -\mathbb{E}_{t-1}(\mathbf{Z}_t|X)) = -\lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbb{E}_{t-1}(\mathbf{Z}_t|X)) \geq -b_4$, thus $\lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t) \geq \lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t) - b_4$. Therefore,

$$\Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\right) \leq b_4 + \epsilon \middle| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

2. Now let $\mathbf{Y}_t = \mathbb{E}_{t-1}(\mathbf{Z}_t|X) - \mathbf{Z}_t$. As before, $\mathbb{E}_{t-1}(\mathbf{Y}_t|X) = 0$ and conditioned on any $X \in \mathcal{C}$, $\mathbf{P}(\mathbf{Y}_t^2 \preceq (b_2 - b_1)^2\mathbf{I}|X) = 1$. As before, applying Theorem C.1.12, we get that for any $\epsilon > 0$,

$$\Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t\right) \leq \epsilon \middle| X\right) > 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

By Weyl's inequality, $\lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Y}_t) = \lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}(\mathbb{E}_{t-1}(\mathbf{Z}_t|X) - \mathbf{Z}_t)) \geq \lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbb{E}_{t-1}(\mathbf{Z}_t|X)) + \lambda_{\max}(\frac{1}{\alpha}\sum_{t=1}^{\alpha} -\mathbf{Z}_t) = \lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbb{E}_{t-1}(\mathbf{Z}_t|X)) - \lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t) \geq b_3 - \lambda_{\min}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t)$. Therefore, for any $\epsilon > 0$,

$$\Pr\left(\lambda_{\min}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\right) \geq b_3 - \epsilon \middle| X\right) \geq 1 - n\exp\left(\frac{-\alpha\epsilon^2}{8(b_2-b_1)^2}\right) \text{ for all } X \in \mathcal{C}$$

□

We can further extend this to the case of a matrix which is not necessarily Hermitian.

Corollary C.1.14 (Matrix Azuma conditioned on another random variable for an arbitrary matrix)**.** *Consider an α-length adapted sequence $\{\mathbf{Z}_t\}_{t=1,2,\ldots,\alpha}$ of random matrices of size $n_1 \times n_2$ and a random variable X that we condition on. Assume that, for all $X \in \mathcal{C}$, (i) $\Pr(\|\mathbf{Z}_t\|_2 \leq b_1|X) = 1$ and (ii) $\|\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbb{E}[\mathbf{Z}_t|\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{t-1}, X]\|_2 \leq b_2$. Then, for all $\epsilon > 0$,*

$$\Pr\left(\left\|\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\right\|_2 \leq b_2 + \epsilon \middle| X\right) \geq 1 - (n_1+n_2)\exp\left(\frac{-\alpha\epsilon^2}{8(2b_1)^2}\right)$$

Proof. At certain places, where the meaning is clear, we use $\mathbb{E}_{t-1}[\mathbf{Z}_t|X]$ to refer to $\mathbb{E}[\mathbf{Z}_t|\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{t-1}, X]$

Define the dilation of an $n_1 \times n_2$ matrix $\mathbf{M}$ as $\text{dilation}(\mathbf{M}) := \begin{bmatrix} \mathbf{0} & \mathbf{M}' \\ \mathbf{M} & \mathbf{0} \end{bmatrix}$. Notice that this is an $(n_1+n_2) \times (n_1+n_2)$ Hermitian matrix [96] . As shown in [96, equation 2.12],

$$\lambda_{\max}\big(\text{dilation}(\mathbf{M})\big) = \|\text{dilation}(\mathbf{M})\|_2 = \|\mathbf{M}\|_2 \tag{C.1}$$

Thus, the corollary assumptions imply that $\mathbf{P}(\|\text{ dilation}(\mathbf{Z}_t)\|_2 \leq b_1|X) = 1$ for all $X \in \mathcal{C}$. By (C.1) and the definition of dilation,

$$\frac{1}{\alpha}\sum_t \mathbb{E}_{t-1}[\text{dilation}(\mathbf{Z}_t)|X] = \text{dilation}\left(\frac{1}{\alpha}\sum_t \mathbb{E}_{t-1}[\mathbf{Z}_t|X]\right) \preceq b_2\mathbf{I}$$

Thus, applying Corollary C.1.13 to the sequence $\{\text{dilation}(\mathbf{Z}_t)\}_{t=1,\ldots,\alpha}$, we get that,

$$\Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\text{dilation}(\mathbf{Z}_t)\right) \leq b_2 + \epsilon \middle| X\right) \geq 1 - (n_1+n_2)\exp\left(\frac{-\alpha\epsilon^2}{32b_1^2}\right) \text{ for all } X \in \mathcal{C}$$

Using (C.1), $\lambda_{\max}\big(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\text{dilation}(\mathbf{Z}_t)\big) = \lambda_{\max}\big(\text{dilation}(\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t)\big) = \|\frac{1}{\alpha}\sum_{t=1}^{\alpha}\mathbf{Z}_t\|_2$ gives the final result. □

C.2 Proof of Lemma 4.5.20 (Initial Subspace Is Accurately Recovered)

Proof of Lemma 4.5.20. Define $\mathbb{M} := \frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\mathbf{m}_t\mathbf{m}_t'$, $\mathbb{A} := \frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\boldsymbol{\ell}_t\boldsymbol{\ell}_t'$ and perturb $:= \mathbb{M} - \mathbb{A}$.

Using Theorem C.1.3 (sin theta theorem) followed by Weyl's inequality for $\lambda_{\max}(\Lambda_\perp) = \lambda_{\max}(\mathbb{M})$, if $\lambda_{r_0}(\mathbb{A}) - \lambda_{r_0+1}(\mathbb{A}) - \|\text{perturb}\| > 0$, then

$$\text{dif}(\hat{\mathbf{P}}_{\text{train}}, \mathbf{P}_{\text{train}}) \leq \frac{\|\text{perturb}\|_2}{\lambda_{r_0}(\mathbb{A}) - \lambda_{r_0+1}(\mathbb{A}) - \|\text{perturb}\|_2} \tag{C.2}$$

We will use Azuma to lower and upper bound $\lambda_{r_0}(\mathbb{A})$, to upper bound $\lambda_{r_0+1}(\mathbb{A})$ and to upper bound $\|perturb\|_2$. Let

$$\epsilon = \frac{1}{1-b^2}0.001 r_{\text{new}}\zeta\lambda^-$$

To get the first three bounds, we need to bound $\lambda_{\max}(\mathbb{A} - \frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\boldsymbol{\Sigma}_\tau)$ and then use Weyl's inequality. Now $\mathbb{A} = \frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t}\sum_{\tilde{\tau}=0}^{t} b^{2t-\tau-\tilde{\tau}}\boldsymbol{\nu}_\tau\boldsymbol{\nu}_{\tilde{\tau}}'$. We proceed as in Section 4.6.1 but with the difference that we include $-\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\boldsymbol{\Sigma}_\tau$ into term21. Another difference is that $t_0 = 1$ and so $\boldsymbol{\ell}_{t_0-1} = 0$ (and so term1 = 0 and

term3 $= 0$). Thus we get

$$-3\epsilon \leq \lambda_{\max}(\mathbb{A} - \frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\mathbf{\Sigma}_\tau) \leq 3\epsilon$$

with probability $1 - 3\cdot(2n)\exp\left(\frac{-t_{\text{train}}\epsilon^2(1-b^2)^2(1-b)^2}{32(2r\gamma^2)^2}\right)$. Thus, with the above probability, using Weyl's inequality and Lemma C.1.9,

$$\lambda_{r_0}(\mathbb{A}) \geq \lambda_{r_0}(\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\mathbf{\Sigma}_\tau) - 3\epsilon \geq \frac{1}{1-b^2}(1 - \frac{b^2}{t_{\text{train}}(1-b^2)})\lambda^- - 3\epsilon$$

$$\lambda_{r_0}(\mathbb{A}) \leq \lambda_{r_0}(\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\mathbf{\Sigma}_\tau) + 3\epsilon \leq \frac{1}{1-b^2}\lambda^- + 3\epsilon$$

$$\lambda_{r_0+1}(\mathbb{A}) \leq \lambda_{r_0+1}(\frac{1}{t_{\text{train}}}\sum_{t=1}^{t_{\text{train}}}\sum_{\tau=0}^{t} b^{2(t-\tau)}\mathbf{\Sigma}_\tau) + 3\epsilon = 0 + 3\epsilon$$

(the above follows because $\mathbf{\Sigma}_\tau$ has rank r_0 for all $t \leq t_{\text{train}}$). Next consider $\|\text{perturb}\|_2$. It is easy to see that

$$\|\text{perturb}\|_2 \leq 2\|\frac{1}{t_{\text{train}}}\sum_t \boldsymbol{\ell}_t\mathbf{w}_t'\| + \|\frac{1}{t_{\text{train}}}\sum_t \mathbf{w}_t\mathbf{w}_t'\|_2$$

Proceeding as in Section 4.6.2 for the first term and using the deterministic bound of $0.03r_{\text{new}}\zeta\lambda^-$ for the second term, we get

$$\|\text{perturb}\|_2 \leq 0.03r_{\text{new}}\zeta\lambda^- + 2\epsilon$$

with probability $1 - (2n)\exp\left(\frac{-t_{\text{train}}\epsilon^2(1-b)^2}{32\cdot r\gamma^2\epsilon_w^2}\right)$. Using the above bounds and Weyl's inequality, we can conclude that

$$\hat{\lambda}^-_{\text{train}} := \lambda_{r_0}(\mathbb{M}) \leq \lambda_{r_0}(\mathbb{A}) + \|\text{perturb}\|_2 \leq \frac{1}{1-b^2}\lambda^- + 0.08r\zeta\lambda^-$$

$$\hat{\lambda}^-_{\text{train}} := \lambda_{r_0}(\mathbb{M}) \geq \lambda_{r_0}(\mathbb{A}) - \|\text{perturb}\|_2 \geq \frac{1}{1-b^2}(1 - \frac{b^2}{t_{\text{train}}(1-b^2)})\lambda^- - 0.08r\zeta\lambda^-$$

w.p. at least $1 - 3\cdot(2n)\exp\left(\frac{-t_{\text{train}}\epsilon^2(1-b^2)^2(1-b)^2}{32(2r\gamma^2)^2}\right) - (2n)\exp\left(\frac{-t_{\text{train}}\epsilon^2(1-b)^2}{32r\gamma^2\epsilon_w^2}\right) \geq 1 - 4\cdot(2n)\cdot\exp\left(\frac{-t_{\text{train}}\epsilon^2(1-b^2)^2(1-b)^2}{32(2r\gamma^2)^2}\right) \geq 1 - n^{-10}$. The last inequality follows because $t_{\text{train}} \geq \frac{128(r\gamma^2)^2}{(1-b)^2(0.001r_{\text{new}}\zeta\lambda^-)^2}(11\log n + \log 8)$.

Thus, using the fact that $1/t_{\text{train}} < (r\zeta)^2$, $(1 - \frac{b^2}{t_{\text{train}}(1-b^2)}) \geq (1 - \frac{(r\zeta)^2b^2}{(1-b^2)})$ and so we get: with probability at least $1 - n^{-10}$,

a) $\hat{\lambda}^-_{\text{train}} \leq \frac{1}{1-b^2}\lambda^- + 0.08r\zeta\lambda^- < 1.2\frac{\lambda^-}{1-b^2}$

b) $\hat{\lambda}^-_{\text{train}} \geq \frac{1}{1-b^2}(1 - \frac{(r\zeta)^2b^2}{(1-b^2)})\lambda^- - 0.08r\zeta\lambda^- \geq 0.8\frac{\lambda^-}{1-b^2}$

c) and

$$\text{dif}(\hat{\mathbf{P}}_{\text{train}}, \mathbf{P}_{\text{train}}) \leq \frac{0.03r_{\text{new}}\zeta\lambda^- + 2\epsilon}{\frac{1}{1-b^2}(1 - \frac{b^2}{t_{\text{train}}(1-b^2)})\lambda^- - 0.08r\zeta\lambda^-} \leq 0.031r_{\text{new}}\zeta \leq r_0\zeta$$

□

C.3 Proof of Lemma 4.5.21 (Bounds On $\zeta^+_{j,\mathbf{new},k}$ And $\tilde{\zeta}^+_{j,k}$)

Proof of Lemma 4.5.21. Proof of item 1 of the lemma: This follows directly from the bounds for $b_{\mathbf{A}}$, $b_{\mathbf{A},\perp}$, $b_{\mathbb{H},k}$ in Fact 4.5.37, and by using Lemma 4.5.20.

Proof of item 2 of the lemma: Recall that $\zeta^+_{j,\text{new},k} := \dfrac{b_{\mathbb{H},k}}{b_{\mathbf{A}} - b_{\mathbf{A},\perp} - b_{\mathbb{H},k}}$ with the terms on the RHS defined in Lemmas 4.5.34, 4.5.35, 4.5.36. The proof approach is similar to that of [44, Lemma 6.1] and almost exactly the same as that of [85, Lemma 6.14]. The proof is as follows. With the bound in Fact 4.5.37, and since $\zeta^+_{\text{new},k}$ is an increasing function of $b_{\mathbf{A},\perp}$ and $b_{\mathbb{H},k}$, and a decreasing function of $b_{\mathbf{A}}$, we have

$$\zeta^+_{\text{new},1} \leq \frac{0.156 + 0.1r_{\text{new}}\zeta}{0.9999 - 0.005r_{\text{new}}\zeta - (0.156 + 0.12r_{\text{new}}\zeta)} < 0.19 \quad \text{because } r_{\text{new}}\zeta \leq 10^{-4}.$$

For $k \geq 2$, we have

$$\zeta^+_{\text{new},k} \leq \frac{0.073\zeta^+_{\text{new},k-1} + 0.1r_{\text{new}}\zeta}{0.9999 - 0.005r_{\text{new}}\zeta - (0.073\zeta^+_{\text{new},k-1} + 0.12r_{\text{new}}\zeta)}$$

Clearly $\zeta^+_{\text{new},k}$ is an increasing function of $\zeta^+_{\text{new},k-1}$. Also $\zeta^+_{j,\text{new},1} \leq 0.19 \leq \zeta^+_{j,\text{new},0} = 1$. Thus, one can use induction to show that $\zeta^+_{j,\text{new},k} \leq \zeta^+_{j,\text{new},k-1} \leq 0.19$. Using the bound $r_{\text{new}}\zeta \leq 10^{-4}$, we can get $\zeta^+_{\text{new},k} \leq 0.1\zeta^+_{\text{new},k-1} + 0.15r_{\text{new}}\zeta$.

Proof of item 3 of the lemma: Recall that

$$\tilde{\zeta}^+_k := \frac{b_{\tilde{\mathbb{H}},k}}{b_{\tilde{\mathbf{A}},k} - b_{\tilde{\mathbf{A}},k,\perp} - b_{\tilde{\mathbb{H}},k}}.$$

Substituting in the bounds for $b_{\tilde{\mathbb{H}},k}$, $b_{\tilde{\mathbf{A}},k}$, and $b_{\tilde{\mathbf{A}},k,\perp}$ in Fact 4.5.42 gives

$$\tilde{\zeta}_k^+ \leq \frac{0.072(r+r_{\text{new}})\zeta + 0.19 r_{\text{new}}\zeta}{0.9999 - (0.2 + 0.265 r_{\text{new}}\zeta + 0.072(r+r_{\text{new}})\zeta)} \leq 0.09(r+r_{\text{new}})\zeta + 0.119 r_{\text{new}}\zeta$$
$$< 0.15(r+r_{\text{new}})\zeta$$

where we assume $r_{\text{new}} < r$ to get the last inequality.

Using the theorem's assumption $r_{j,k} := |\mathcal{G}_{j,k}| \geq 0.15(r+r_{\text{new}})$, the claim follows. □

C.4 Proof of Lemma 4.5.25 (Compressed Sensing Lemma)

This proof's approach is similar to that of [44, Lemma 6.4]. The proof uses the denseness assumption and subspace error bounds $\zeta_{j,*} \leq \zeta_{j,*}^+$ and $\zeta_{j,\text{new},k-1} \leq \zeta_{j,\text{new},k-1}^+$, that hold when $X_{\hat{u}_j+k-1} \in \Gamma_{j,k-1}^{\hat{u}_j}$ for $\hat{u}_j = u_j$ or $\hat{u}_j = u_j + 1$, to obtain bounds on the restricted isometry constant (RIC) of the sparse recovery matrix $\mathbf{\Phi}_t$ and the sparse recovery error $\|\mathbf{b}_t\|_2$. Applying the noisy compressed sensing (CS) result from [12] and the assumed bounds on ζ and γ, the lemma follows.

Lemma C.4.1 (Bounding the RIC of $\mathbf{\Phi}_t$ [44, Lemma 6.6], [85])**.** *Recall that* $\zeta_{j,*} := \|(\mathbf{I} - \hat{\mathbf{P}}_{(j),*}\hat{\mathbf{P}}_{(j),*}{}')\mathbf{P}_{(j),*}\|_2$.

1. *Suppose that a basis matrix* $\mathbf{P}$ *can be split as* $\mathbf{P} = [\mathbf{P}_1\ \mathbf{P}_2]$ *where* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are also basis matrices. Then* $\kappa_s^2(\mathbf{P}) = \max_{\mathcal{T}:|\mathcal{T}|\leq s} \|\mathbf{I}_{\mathcal{T}}{}'\mathbf{P}\|_2^2 \leq \kappa_s^2(\mathbf{P}_1) + \kappa_s^2(\mathbf{P}_2)$.
2. $\kappa_s^2(\hat{\mathbf{P}}_{(j),*}) \leq (\kappa_{s,*})^2 + 2\zeta_*$ *for all* j
3. $\kappa_s(\hat{\mathbf{P}}_{(j),\text{new},k}) \leq \kappa_{s,\text{new}} + \zeta_{j,new,k} + \zeta_{j,*}$ *for all* j *and* k.
4. *For* $t \in [(u_{j-1}+K)\alpha + 1, (\hat{u}_j+1)\alpha)$, $\delta_s(\mathbf{\Phi}_t) = \kappa_s^2(\hat{\mathbf{P}}_{(j),*}) \leq (\kappa_{s,*})^2 + 2\zeta_{j,*}$.
5. *For* $k = 1, \ldots, K-1$, *for* $t \in [(\hat{u}_j+k)\alpha+1, (\hat{u}_j+k+1)\alpha]$ $\delta_s(\mathbf{\Phi}_t) = \kappa_s^2([\hat{\mathbf{P}}_{(j),*}\ \hat{\mathbf{P}}_{(j),\text{new},k}]) \leq \kappa_s^2(\hat{\mathbf{P}}_{(j),*}) + \kappa_s^2(\hat{\mathbf{P}}_{(j),\text{new},k}) \leq (\kappa_{s,*})^2 + 2\zeta_{j,*} + (\kappa_{s,\text{new}} + \zeta_{j,new,k} + \zeta_{j,*})^2$.

Corollary C.4.2.

1. *Conditioned on* $\Gamma_{j-1,\text{end}}$, *for* $t \in [t_j, (\hat{u}_j+1)\alpha]$, $\delta_s(\mathbf{\Phi}_t) \le \delta_{2s}(\mathbf{\Phi}_t) \le (\kappa_{2s,*})^2 + 2\zeta_{j,*}^+ < 0.1 < 0.1479$, *and* $\|[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \le \frac{1}{1-\delta_s(\mathbf{\Phi}_t)} < 1.2 := \phi^+$.

2. *For* $k = 2, \dots, K$ *and* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$, *conditioned on* $\Gamma_{j,k-1}^{\hat{u}_j}$, *for* $t \in [(\hat{u}_j + k - 1)\alpha + 1, (\hat{u}_j + k)\alpha]$, $\delta_s(\mathbf{\Phi}_t) \le \delta_{2s}(\mathbf{\Phi}_t) \le (\kappa_{2s,*})^2 + 2\zeta_{j,*}^+ + (\kappa_{2s,\text{new}} + \zeta_{j,new,k-1}^+ + \zeta_{j,*}^+)^2 < 0.1479$, *and* $\|[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \le \frac{1}{1-\delta_s(\mathbf{\Phi}_t)} < 1.2 := \phi^+$.

3. *For* $\hat{u}_j = u_j$ *or* $\hat{u}_j = u_j + 1$, *conditioned on* $\Gamma_{j,K}^{\hat{u}_j}$, *for* $t \in [(\hat{u}_j + K)\alpha + 1, t_{j+1} - 1]$, $\delta_s(\mathbf{\Phi}_t) \le \delta_{2s}(\mathbf{\Phi}_t) \le (\kappa_{2s,*})^2 + 2\zeta_{j,*}^+ < 0.1 < 0.1479$, *and* $\|[(\mathbf{\Phi}_t)_{\mathcal{T}_t}'(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^{-1}\|_2 \le \frac{1}{1-\delta_s(\mathbf{\Phi}_t)} < 1.2 := \phi^+$.

Proof. This follows using Lemma C.4.1, the definitions of $\Gamma_{j-1,\text{end}}$ and $\Gamma_{j,k}^{\hat{u}_j}$, and Fact 4.5.24. □

Proof of Lemma 4.5.25. We will prove claim 2). The others are done in the same way. By Fact 4.5.24, $\Gamma_{j,k-1}^{\hat{u}_j}$ implies that $\zeta_{j,*} \le \zeta_{j,*}^+$ and $\zeta_{j,\text{new},k-1} \le \zeta_{j,\text{new},k-1}^+$.

a) For $t \in [(\hat{u}_j + k - 1)\alpha + 1, (\hat{u}_j + k)\alpha]$, $\mathbf{b}_t := (\mathbf{I} - \hat{\mathbf{P}}_{t-1}\hat{\mathbf{P}}_{t-1}')(\boldsymbol{\ell}_t + \mathbf{w}_t)$. Thus, using Fact 4.5.26,

$$\|\mathbf{b}_t\|_2 \le \xi_{\text{cor}} = \xi$$

b) By Corollary C.4.2, $\delta_{2s}(\mathbf{\Phi}_t) < 0.15 < \sqrt{2} - 1$. Given $|\mathcal{T}_t| \le s$, $\|\mathbf{b}_t\|_2 \le \xi$, by [12, Theorem 1.1], the CS error satisfies

$$\|\hat{\mathbf{x}}_{t,\text{cs}} - \mathbf{x}_t\|_2 \le \frac{4\sqrt{1+\delta_{2s}(\mathbf{\Phi}_t)}}{1-(\sqrt{2}+1)\delta_{2s}(\mathbf{\Phi}_t)}\xi < 7\xi.$$

c) Using the above, $\|\hat{\mathbf{x}}_{t,\text{cs}} - \mathbf{x}_t\|_\infty \le 7\xi$. Since $\min_{i\in\mathcal{T}_t} |(\mathbf{x}_t)_i| \ge x_{\min}$ and $(\mathbf{x}_t)_{\mathcal{T}_t^c} = 0$, $\min_{i\in\mathcal{T}_t} |(\hat{\mathbf{x}}_{t,\text{cs}})_i| \ge x_{\min} - 7\xi$ and $\max_{i\in\bar{\mathcal{T}}_t} |(\hat{\mathbf{x}}_{t,\text{cs}})_i| \le 7\xi$. If $\omega \le x_{\min} - 7\xi$, then $\hat{\mathcal{T}}_t \supseteq \mathcal{T}_t$. On the other hand, if $\omega \ge 7\xi$, then $\hat{\mathcal{T}}_t \subseteq \mathcal{T}_t$. Since ω satisfies $7\xi \le \omega \le x_{\min} - 7\xi$, the support of $\mathbf{x}_t$ is exactly recovered, i.e. $\hat{\mathcal{T}}_t = \mathcal{T}_t$.

d) Given $\hat{\mathcal{T}}_t = \mathcal{T}_t$, the least squares estimate of $\mathbf{x}_t$ satisfies $(\hat{\mathbf{x}}_t)_{\mathcal{T}_t} = [(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^\dagger \mathbf{y}_t = [(\mathbf{\Phi}_t)_{\mathcal{T}_t}]^\dagger (\mathbf{\Phi}_t \mathbf{x}_t + \mathbf{\Phi}_t \boldsymbol{\ell}_t + \mathbf{\Phi}_t \mathbf{w}_t)$ and $(\hat{\mathbf{x}}_t)_{\bar{\mathcal{T}}_t} = \mathbf{0}$. Also, $(\mathbf{\Phi}_t)_{\mathcal{T}_t}{}' \mathbf{\Phi}_t = \mathbf{I}_{\mathcal{T}_t}{}' \mathbf{\Phi}_t$ (this follows since $(\mathbf{\Phi}_t)_{\mathcal{T}_t} = \mathbf{\Phi}_t \mathbf{I}_{\mathcal{T}_t}$ and $\mathbf{\Phi}_t{}' \mathbf{\Phi}_t = \mathbf{\Phi}_t$). Using this, the error $\mathbf{e}_t := \hat{\mathbf{x}}_t - \mathbf{x}_t$ satisfies (4.11).

e) Using Fact 4.5.26 we get the bound on $\|\mathbf{e}_t\|_2$.

□

C.5 Proof of Lemmas 4.5.27, 4.5.28, 4.5.29

Proof of Lemma 4.5.27. This proof is similar to that of Lemma 6.16 of [85].

Notice that $\Pr(\text{NODETS}_j^{\hat{u}_j} \mid \tilde{\Gamma}_{j,\vartheta}^{\hat{u}_j}) = \Pr\left(\lambda_{\max}\left(\frac{1}{\alpha}\mathcal{D}_u \mathcal{D}_u{}'\right) < \text{thresh for all } u \in [\hat{u}_j + K + (\vartheta+1) + 1, u_{j+1} - 1] \mid \tilde{\Gamma}_{j,\vartheta}^{\hat{u}_j}\right)$ for $\hat{u}_j = u_j$ or $\hat{u}_j = u_j + 1$.

Recall that $\Gamma_{j,\text{end}} := \left(\tilde{\Gamma}_{j,\vartheta}^{u_j} \cap \text{NODETS}_j^{u_j}\right) \cup \left(\tilde{\Gamma}_{j,\vartheta}^{u_j+1} \cap \text{NODETS}_j^{u_j+1}\right)$. Recall from Fact 4.5.24 that $\Gamma_{j,\text{end}}$ implies that $\text{dif}(\hat{\mathbf{P}}_{(j),*}, \mathbf{P}_{(j),*}) \leq r\zeta$.

Also, for $u \in [\hat{u}_j + K + (\vartheta+1) + 1, u_{j+1} - 1]$, $\hat{\mathbf{P}}_{u\alpha-1,*} = \hat{\mathbf{P}}_{(j+1),*}$ and for all $t \in \mathcal{J}_u$ for these u's, $\boldsymbol{\nu}_t = \mathbf{P}_{(j)} \mathbf{a}_t = \mathbf{P}_{(j+1),*} \mathbf{a}_t$.

Using Lemma 4.5.25, under the given conditioning, $\|\mathbf{e}_t\|_2 \leq \frac{\phi^+}{1-b}(2\zeta_{j,*}^+ \sqrt{r}\gamma + 2\epsilon_w)$ for times $t \in \mathcal{J}_u$ for all these u's. Therefore,

$$
\begin{aligned}
\lambda_{\max}\left(\frac{1}{\alpha}\mathcal{D}_u \mathcal{D}_u{}'\right) &= \lambda_{\max}\left(\frac{1}{\alpha}\sum_{t \in \mathcal{J}_u} (\mathbf{I} - \hat{\mathbf{P}}_{u\alpha-1,*}\hat{\mathbf{P}}_{u\alpha-1,*}{}')\hat{\boldsymbol{\ell}}_t \hat{\boldsymbol{\ell}}_t{}' (\mathbf{I} - \hat{\mathbf{P}}_{u\alpha-1,*}\hat{\mathbf{P}}_{u\alpha-1,*}{}')\right) \\
&= \lambda_{\max}\left(\frac{1}{\alpha}\sum_{t \in \mathcal{J}_u} (\mathbf{I} - \hat{\mathbf{P}}_{(j+1),*}\hat{\mathbf{P}}_{(j+1),*}{}')(\boldsymbol{\ell}_t - \mathbf{e}_t)(\boldsymbol{\ell}_t - \mathbf{e}_t)' (\mathbf{I} - \hat{\mathbf{P}}_{(j+1),*}\hat{\mathbf{P}}_{(j+1),*}{}')\right) \\
&\leq \frac{(2\zeta_{j,*}^+)^2 r\gamma^2}{(1-b)^2} + 2\phi^+(2\zeta_{j,*}^+\sqrt{r}\gamma + 2\epsilon_w)\frac{2\zeta_{j,*}^+\sqrt{r}\gamma}{(1-b)^2} + \frac{(\phi^+(2\zeta_{j,*}^+\sqrt{r}\gamma + 2\epsilon_w))^2}{(1-b)^2} \\
&\leq \frac{0.05\zeta\lambda^-}{0.81} + \frac{2.4 \cdot (0.05 + \sqrt{0.006})\zeta\lambda^-}{0.81} + \frac{1.44 \cdot (\sqrt{0.05} + \sqrt{0.03})^2 \zeta\lambda^-}{0.81} \\
&< 0.5\hat{\lambda}_{\text{train}}^- = \text{thresh}
\end{aligned}
$$

The first inequality uses the bound on $\|(\mathbf{I} - \hat{\mathbf{P}}_{(j+1),*}\hat{\mathbf{P}}_{(j+1),*}{}')\boldsymbol{\ell}_t\|_2 = \|\mathbf{\Phi}_{(j+1),0}\boldsymbol{\ell}_t\|_2$ from Fact 4.5.26 and the bound on $\|\mathbf{e}_t\|_2$ from Lemma 4.5.25. The second inequality uses

the bound on ϵ_w from Model 7 and the Theorem; the bound $\zeta \leq \frac{0.05\lambda^-}{(r+r_{\text{new}})^3\gamma^2}$ from the Theorem and the lower bound on $\hat{\lambda}^-_{\text{train}}$ from Lemma 4.5.20. □

Proof of Lemma 4.5.28. This proof is similar to that of the corresponding lemma from [85]. We will prove that $\Pr\left(\text{DET}^{u_j+1} \mid X_{u_j}\right) > p_{\text{det},1}$ for all $X_{u_j} \in \Gamma_{j-1,\text{end}}$. In particular, this will imply that $\Pr(\text{DET}^{u_j+1} \mid X_{u_j}) > p_{\text{det},1}$ for all $X_{u_j} \in \Gamma_{j-1,\text{end}} \cap \overline{\text{DET}^{u_j}}$ and so, by Lemma C.1.11, we can conclude that $\Pr(\text{DET}^{u_j+1} \mid \Gamma_{j-1,\text{end}}, \overline{\text{DET}^{u_j}}) > p_{\text{det},1}$.

The following claim is a direct corollary of Lemmas 4.5.34 and 4.5.36. It follows exactly as the proof of these lemmas for the $k = 1$ case but with using $u = u_j + 1$ instead of $u = \hat{u}_j + 1$.

$$\Pr\left(\lambda_{\min}\left(\mathbf{A}_{u_j+1}\right) \geq b_{\mathbf{A}} \;\middle|\; X_{u_j}\right) \geq 1 - p_{\mathbf{A}},$$
$$\Pr\left(\|\mathbb{H}_{u_j+1}\|_2 \leq b_{\mathbb{H},1} \;\middle|\; X_{u_j}\right) \geq 1 - p_{\mathbb{H}}$$

for all $X_{u_j} \in \Gamma_{j-1,\text{end}}$. By Lemma 4.5.21, $b_{\mathbf{A}} - b_{\mathbb{H},1} \geq \text{thresh}$.

From the algorithm, notice that, $\mathbb{M}_u = \frac{1}{\alpha}\mathcal{D}_u\mathcal{D}_u{}'$. Thus,

$$\Pr\left(\text{DET}^{u_j+1} \mid X_{u_j}\right) = \Pr\left(\lambda_{\max}(\mathbf{M}_{u_j+1}) > \text{thresh} \mid X_{u_j}\right)$$

By Weyl's inequality and the above,

$$\begin{aligned}\lambda_{\max}(\mathbb{M}_{u_j+1}) &\geq \lambda_{\max}(\mathbf{A}_{u_j+1}) - \|\mathbb{H}_{u_j+1}\|_2 \\ &\geq \lambda_{\min}(\mathbf{A}_{u_j+1}) - \|\mathbb{H}_{u_j+1}\|_2 \\ &\geq b_{\mathbf{A}} - b_{\mathbb{H},1} \geq \text{thresh}\end{aligned}$$

with probability at least $1 - p_{\mathbf{A}} - p_{\mathbb{H}} = p_{\text{det},1}$, whenever $X_{u_j} \in \Gamma_{j-1,\text{end}}$. Thus the result follows. □

Proof of Lemma 4.5.29 (p-PCA lemma). This proof is similar to that of the corresponding lemma from [85]. To prove this lemma we need to show two things. First, conditioned on $\Gamma^{\hat{u}_j}_{j,k-1}$, the k^{th} estimate of the number of new directions is correct. That is: $\hat{r}_{j,\text{new},k} = r_{j,\text{new}}$. Second, we must show $\zeta_{j,\text{new},k} \leq \zeta^+_{j,\text{new},k}$, again conditioned on $\Gamma^{\hat{u}_j}_{j,k-1}$.

Notice that $\hat{r}_{j,\text{new},k} = \text{rank}(\hat{\mathbf{P}}_{(j),\text{new},k})$. To show that $\text{rank}(\hat{\mathbf{P}}_{(j),\text{new},k}) = r_{j,\text{new}}$, we need to show that for $u = \hat{u}_j + k$, $k = 1, \ldots, K$, $\lambda_{r_{j,\text{new}}}(\mathbf{M}_u) > \text{thresh}$ and $\lambda_{r_{j,\text{new}}+1}(\mathbf{M}_u) < \text{thresh}$. Observe that, $\mathbb{M}_u = \mathbb{A}_u + \mathbb{H}_u$. By Lemma 4.5.21, Lemmas 4.5.34 and 4.5.35 followed by Lemma C.1.11, $\lambda_{\min}(\mathbf{A}_u) \geq b_{\mathbf{A}} > b_{\mathbf{A},\perp} \geq \lambda_{\max}(\mathbf{A}_{u,\perp})$ with probability at least $1 - p_{\mathbf{A}} - p_{\mathbf{A},\perp}$ under the given conditioning. Since $\mathbf{A}_u$ is of size $r_{j,\text{new}} \times r_{j,\text{new}}$, this means that $\lambda_{r_{j,\text{new}}}(\mathbb{A}_u) = \lambda_{\min}(\mathbf{A}_u)$ and $\lambda_{r_{j,\text{new}}+1}(\mathbb{A}_u) = \lambda_{\max}(\mathbf{A}_{u,\perp})$. Using these facts, Weyl's inequality, Lemmas 4.5.34, 4.5.35 and 4.5.36, and the bounds from Lemma 4.5.21, we can conclude that with probability at least p_{ppca}, under the given conditioning,

$$\begin{aligned}\lambda_{r_{j,\text{new}}}(\mathbb{M}_u) &\geq \lambda_{r_{j,\text{new}}}(\mathbb{A}_u) - \|\mathbb{H}_u\|_2 \\ &= \lambda_{\min}(\mathbf{A}_u) - \|\mathbb{H}_u\|_2 \geq b_{\mathbf{A}} - b_{\mathbb{H},k} \geq \text{thresh}\end{aligned}$$

and

$$\begin{aligned}\lambda_{r_{j,\text{new}}+1}(\mathbb{M}_u) &\leq \lambda_{r_{j,\text{new}}+1}(\mathbb{A}_u) + \|\mathbb{H}_u\|_2 \\ &= \lambda_{\max}(\mathbf{A}_{u,\perp}) + \|\mathbb{H}_u\|_2 \leq b_{\mathbf{A},\perp} + b_{\mathbb{H},k} < \text{thresh}\end{aligned}$$

Therefore $\text{rank}(\hat{\mathbf{P}}_{(j),\text{new},k}) = r_{j,\text{new}}$ with probability greater than p_{ppca} under the given conditioning.

To show that $\zeta_{j,\text{new},k} \leq \zeta^{+}_{j,\text{new},k}$, we use Lemmas 4.5.34, 4.5.35, and 4.5.36. Using $\text{rank}(\hat{\mathbf{P}}_{(j),\text{new},k}) = r_{j,\text{new}}$ and applying Lemma 4.5.33 with these bounds; using $\lambda^{-}_{\text{new}} \geq \lambda^{-}$; and finally using Lemma C.1.11 gives the desired result. □

C.6 Proof of Theorem 4.2.3

The proof follows with the following re-definitions. Redefine $\Gamma_{j,\text{end}}$ as

$$\Gamma_{j,\text{end}} := \left(\Gamma^{u_j}_{j,K} \cap \text{NODETS}^{u_j}_{j}\right) \cup \left(\Gamma^{u_j+1}_{j,K} \cap \text{NODETS}^{u_j+1}_{j}\right).$$

We get Corollary 4.5.32 then by just combining Lemmas 4.5.27, 4.5.28, 4.5.29. The theorem follows using the lower bound on α, using Fact 4.5.24 and Lemma 4.5.25.

The proof of the Lemmas 4.5.27, 4.5.28, 4.5.29 follows using the following redefinitions. Re-define

1. $\hat{\mathbf{P}}_{(j+1),*} := \hat{\mathbf{P}}_{\hat{t}_j+K\alpha}$. Thus, given all subspace change times are correctly detected, $\hat{\mathbf{P}}_{(j+1),*} = [\hat{\mathbf{P}}_{(j),*}, \hat{\mathbf{P}}_{(j),\text{new},K}]$. Thus, $\Gamma^a_{j,\text{end}}$ implies $\zeta_{j+1,*} \leq \zeta_{j,*} + \zeta_{j,\text{new},K}$.

2. $\zeta^+_{j,*} := (r_0 + (j-1)r_{\text{new}})\zeta$ and $\zeta^+_{j,\text{add}} := (r_0 + jr_{\text{new}})\zeta$. Thus, $\zeta_{j+1,*} \leq \zeta_{j,*} + \zeta_{j,\text{new},K} \leq \zeta^+_{j+1,*}$.

BIBLIOGRAPHY

[1] N. Vaswani, "LS-CS-residual (LS-CS): compressive sensing on least squares residual," *IEEE Trans. Sig. Proc.*, vol. 58(8), pp. 4108–4120, August 2010.

[2] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Proc.*, vol. 41(12), pp. 3397 – 3415, Dec 1993.

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

[4] I.F. Gorodnitsky and B.D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Trans. Sig. Proc.*, vol. 45, no. 3, pp. 600–616, 1997.

[5] G Harikumar and Y Bresler, "A new algorithm for computing sparse solutions to linear inverse problems," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on.* IEEE, 1996, vol. 3, pp. 1331–1334.

[6] Ping Feng and Yoram Bresler, "Spectrum-blind minimum-rate sampling and reconstruction of multiband signals," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on.* IEEE, 1996, vol. 3, pp. 1688–1691.

[7] D. Donoho, "Compressed sensing," *IEEE Trans. Info. Th.*, vol. 52(4), pp. 1289–1306, April 2006.

[8] 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.

[9] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ell-1 norm solution is also the sparsest solution," *Comm. Pure and App. Math.*, vol. 59(6), pp. 797–829, June 2006.

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

[11] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59(8), pp. 1207–1223, August 2006.

[12] 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.

[13] N. Vaswani, "Kalman filtered compressed sensing," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2008.

[14] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Processing*, vol. 58(9), pp. 4595–4607, September 2010.

[15] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Trans. Sig. Proc.*, pp. 2405–2409, April 2010.

[16] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Asilomar Conf. on Sig. Sys. Comp.*, 2010.

[17] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE J. Sel. Topics Sig. Proc., Special Issue on Adaptive Sparse Representation of Data and Applications in Signal and Image Processing*, vol. 5, no. 5, pp. 912–926, Sept 2011.

[18] A. Charles, M. S. Asif, J. Romberg, and C. Rozell, "Sparsity penalties in dynamical system estimation," in *Conf. Info. Sciences and Systems*, 2011.

[19] J. Kim, O. K. Lee, and J. C. Ye, "Dynamic sparse support tracking with multiple measurement vectors using compressive music," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2012, pp. 2717–2720.

[20] C. Qiu and N. Vaswani, "Recursive sparse recovery in large but correlated noise," in *Allerton Conf. on Communication, Control, and Computing*, 2011.

[21] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Allerton Conference on Communication, Control, and Computing*. IEEE, 2010, pp. 591–598.

[22] "Rice compressive sensing resources," in *http://www-dsp.rice.edu/cs*.

[23] I. Carron, "Nuit blanche," in *http://nuit-blanche.blogspot.com/*.

[24] A. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Weighted ℓ_1 minimization for sparse recovery with prior information," *IEEE Trans. Sig. Proc.*, 2011.

[25] C. J. Miosso, R. von Borries, M. Argez, L. Valazquez, C. Quintero, and C. Potes, "Compressive sensing reconstruction with prior information by iteratively reweighted least-squares," *IEEE Trans. Sig. Proc.*, vol. 57 (6), pp. 2424–2431, June 2009.

[26] Michael P Friedlander, Hassan Mansour, Rayan Saab, and Oezguer Yilmaz, "Recovering compressively sampled signals using partial support information," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 1122–1134, 2012.

[27] Luis Nunes Vicente Afonso S. Bandeira, Katya Scheinberg, "On partial sparse recovery," *arXiv preprint arXiv:1304.2809*, 2013.

[28] L. Jacques, "A short note on compressed sensing with partially known signal support," *ArXiv preprint 0908.0660*, 2009.

[29] W. Lu and N. Vaswani, "Regularized modified bpdn for noisy sparse reconstruction with partial erroneous support and signal value knowledge," *IEEE Trans. Sig. Proc.*, vol. 60, no. 1, pp. 182–196, 2012.

[30] N. Vaswani, "Stability (over time) of modified-CS and ls-CS for recursive causal sparse reconstruction," *Allerton Conference on Communication, Control, and Computing*, 2010.

[31] J. Wright and Y. Ma, "Dense error correction via ℓ_1-minimization," *IEEE Trans. Info. Th.*, vol. 56, no. 7, pp. 3540–3560, July 2010.

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

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

[34] A. Ganesh, J. Wright, X. Li, E. J Candes, and Y. Ma, "Dense error correction for low-rank matrices via principal component pursuit," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 1513–1517.

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

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

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

[38] Arvind Ganesh, Kerui Min, John Wright, and Yi Ma, "Principal component pursuit with reduced linear measurements," *arXiv:1202.6445*.

[39] D. Hsu, S. M Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *Information Theory, IEEE Transactions on*, vol. 57, no. 11, pp. 7221–7234, 2011.

[40] John Wright, Arvind Ganesh, Kerui Min, and Yi Ma, "Compressive principal component pursuit," *arXiv:1202.4596*.

[41] Min Tao and Xiaoming 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.

[42] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inform. Theory*, vol. 59(7), pp. 4324–4337, 2013.

[43] Alekh Agarwal, Sahand Negahban, and Martin J Wainwright, "Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions," *The Annals of Statistics*, vol. 40, no. 2, pp. 1171–1197, 2012.

[44] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," *IEEE Trans. Info. Th.*, vol. 60, no. 8, pp. 5007–5039, August 2014.

[45] Zhouchen Lin, Minming Chen, and Yi Ma, "Alternating direction algorithms for l1 problems in compressive sensing," Tech. Rep., University of Illinois at Urbana-Champaign, November 2009.

[46] E. T Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.

[47] J. Cai, E. J Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[48] Ricardo Otazo, Emmanuel Candès, and Daniel K Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[49] D. Needell and J.A. Tropp., "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Anal.*, vol. 26(3), pp. 301–321, May 2009.

[50] W. Lu and N. Vaswani, "Modified bpdn for noisy compressive sensing with partially known support," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2010.

[51] T Tony Cai, Lie Wang, and Guangwu Xu, "Shifting inequality and recovery of sparse signals," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1300–1308, 2010.

[52] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35 (6), pp. 2313–2351, 2007.

[53] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. Info. Th.*, pp. 1030–1051, March 2006.

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

[55] M. S. Asif and J. Romberg, "Dynamic Updating for ℓ_1 minimization," *IEEE J. Selected Topics in Signal Processing*, vol. 4(2), April 2010.

[56] E.J. Candes, M.B Wakin, and S.P. Boyd, "Enhancing sparsity by reweighted l(1) minimization," *Journal of Fourier Analysis and Applications*, vol. 14 (5-6), pp. 877–905, 2008.

[57] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, pp. 462–491, 2010.

[58] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2008.

[59] A. Khajehnejad, W. Xu, S. Avestimehr, and B. Hassibi, "Improved sparse recovery thresholds with two-step reweighted ell-1 minimization," in *ArXiv Preprint arXiv:1004.0402*, 2010.

[60] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Info. Th.*, vol. 55(5), pp. 2230 – 2249, May 2009.

[61] L. Jacques, "A short note on compressed sensing with partially known signal support," *Signal Processing*, 2010.

[62] Ian C. Atkinson, Douglas L. Jones Farzad Kamalabadi, and Thulborn, "Blind estimation for localized low contrast-to-noise ratio bold signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, pp. 350–364, 2008.

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

[64] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Trans. Sig. Proc.*, vol. 62, no. 16, pp. 4284–4297, 2014.

[65] J. Feng, H. Xu, and S. Yan, "Online robust pca via stochastic optimization," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013.

[66] J. Feng, H. Xu, S. Mannor, and S. Yan, "Online pca for contaminated data," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013.

[67] Z. Füredi and J. Komlós, "The eigenvalues of random symmetric matrices," *Combinatorica*, vol. 1, no. 3, pp. 233–241, 1981.

[68] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

[69] J. Zhan and N. Vaswani, "Robust pca with partial subspace knowledge," *arXiv:arXiv:1403.1591*, 2014.

[70] D. Gross, Y. Liu, S. T Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing," *Physical review letters*, vol. 105, no. 15, pp. 150401, 2010.

[71] J. Hiriart-Urruty and C. Lemarchal, "Fundamentals of convex analysis," 2001.

[72] A. S Lewis, "The mathematics of eigenvalue optimization," *Mathematical Programming*, vol. 97, no. 1-2, pp. 155–176, 2003.

[73] G A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra and its Applications*, vol. 170, pp. 33–45, 1992.

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

[75] J. Fessler, "Linear operators and adjoints," *http://web.eecs.umich.edu/~fessler/course/600/l/l06.pdf*, p. 12.

[76] D. Gross, "Recovering low-rank matrices from few coefficients in any basis," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1548–1566, 2011.

[77] J. Xu, V. K Ithapu, L. Mukherjee, J. M Rehg, and V. Singh, "Gosus: Grassmannian online subspace updates with structured-sparsity," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3376–3383.

[78] P. N. Belhumeur, J. P Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

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

[80] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *IEEE Conf. on Comp. Vis. Pat. Rec. (CVPR)*, 2012.

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

[82] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[83] B. Lois and N. Vaswani, "A correctness result for online robust pca," *Submitted to IEEE Transaction on Information Theory*, 2014.

[84] B. Lois and N. Vaswani, "A correctness result for online robust pca," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2015.

[85] B. Lois and N. Vaswani, "Online matrix completion and online robust pca," in *ISIT (submitted to IEEE Transactions on Infomration Theory)*, 2015.

[86] J. Zhan and N. Vaswani, "Robust pca with partial subspace knowledge," *IEEE Trans. Sig. Proc.*, 2015.

[87] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Eur. Conf. on Comp. Vis. (ECCV)*, 2002.

[88] 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.

[89] Morteza Mardani, Gonzalo Mateos, and G Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *J. Sel. Topics in Sig. Proc.*, Feb 2013.

[90] G. Mateos and G. Giannakis, "Robust pca as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Sig. Proc.*, Oct 2012.

[91] Symeon Chouvardas, Yannis Kopsinis, and Sergios Theodoridis, "Robust subspace tracking with missing entries: a set–theoretic approach," *IEEE Trans. Sig. Proc.*, vol. 63, no. 19, pp. 5060–5070, 2015.

[92] Wenjie Song, Jianke Zhu, Yang Li, and Chun Chen, "Image alignment by online robust pca via stochastic gradient descent," *Circuits and Systems for Video Technology, IEEE Transactions on*, 2015.

[93] Vinuthna Vinjamuri, Ranjitha Prasad, and Chandra R Murthy, "Sparse signal recovery in the presence of colored noise and rank-deficient noise covariance matrix: An sbl approach," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3761–3765.

[94] D. Hsu, S.M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Trans. Info. Th.*, Nov. 2011.

[95] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. iii," *SIAM Journal on Numerical Analysis*, vol. 7, pp. 1–46, Mar. 1970.

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

[97] J. Zhan and N. Vaswani, "Robust pca with partial subspace knowledge," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2014.

[98] "Email communication with joel tropp," .

[99] W. Lu, T. Li, I. Atkinson, and N. Vaswani, "Modified-CS-residual for recursive reconstruction of highly undersampled functional MRI sequences," in *IEEE Intl. Conf. Image Proc. (ICIP)*. IEEE, 2011, pp. 2689–2692.

[100] J. A Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[101] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Sig. Proc.*, vol. 54, no. 12, pp. 4634–4643, 2006.

[102] Joel A Tropp, Anna C Gilbert, and Martin J Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[103] Jong Min Kim, Ok Kyun Lee, and Jong Chul Ye, "Compressive music: revisiting the link between compressive sensing and array signal processing," *IEEE Trans. Info. Th.*, vol. 58, no. 1, pp. 278–301, 2012.

[104] Kiryung Lee, Yoram Bresler, and Marius Junge, "Subspace methods for joint sparse recovery," *IEEE Trans. Info. Th.*, vol. 58, no. 6, pp. 3613–3641, 2012.

[105] Yonina C Eldar, Patrick Kuppinger, and Helmut Bölcskei, "Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery," *arXiv preprint arXiv:0906.3173*, 2009.

[106] D. P Wipf and B. D Rao, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Sig. Proc.*, vol. 55, no. 7, pp. 3704–3716, 2007.

[107] C. Qiu and N. Vaswani, "Support predicted modified-cs for recursive robust principal components' pursuit," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2011.

[108] T. Bouwmans and E. Zahzah, "Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.

[109] D. Achlioptas and F. McSherry, "Fast computation of low rank matrix approximations," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM, 2001, pp. 611–618.

[110] V. H Vu, "Spectral norm of random matrices," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. ACM, 2005, pp. 423–430.