A FAST AND MEMORY-EFFICIENT ALGORITHM FOR ROBUST PCA (MEROP)

Praneeth Narayanamurthy and Namrata Vaswani

{pkurpadn, namrata} @iastate.edu,
Department of Electrical and Computer Engineering,
Iowa State University, Ames, IA

ABSTRACT

Robust PCA (RPCA) is the problem of separating a given data matrix into the sum of a sparse matrix and a low-rank matrix. We propose an algorithm, MERoP, (Memory-Efficient Robust PCA) based on the Recursive Projected Compressed Sensing (ReProCS) framework to solve the RPCA problem. We demonstrate that we can provably recover the low-rank and sparse components, and MERoP enjoys nearly-optimal memory complexity. We also show that the algorithm is nearly-online and fast. We validate our theoretical claims through extensive numerical experiments.

Index Terms— Robust PCA, Online Algorithms

1. INTRODUCTION

Robust PCA (RPCA) is the problem of decomposing a given data matrix into the sum of a low-rank matrix (true data) and a sparse matrix (outliers). The column space of the low-rank matrix then gives the desired principal subspace (PCA solution). In recent years, the RPCA problem has been extensively studied, e.g., [1, 2, 3, 4, 5, 6]. A common application of RPCA is in video analytics in separating video into a slow-changing background image sequence and a foreground image sequence consisting of moving objects or people [7]. In this work, we propose an algorithm called MERoP which is a nearly-online, and fast algorithm to solve the Robust PCA problem under weaker assumptions as compared to existing algorithms. In particular, we show that the running time of our algorithm is nearly the same as computing a vanilla SVD on the data matrix. Furthermore, we also show that the number of samples required to obtain an ε-accurate estimate of the subspace in which the true data lies is nearly-optimal.

Problem Statement. At each time t we observe data vectors $\boldsymbol{y}_t \in \mathbb{R}^n$ that satisfy

$$\boldsymbol{y}_t = \boldsymbol{\ell}_t + \boldsymbol{x}_t \tag{1}$$

where $\boldsymbol{x}_t$ is the sparse outlier vector and $\boldsymbol{\ell}_t$ is the true data vector that lies in a low-dimensional subspace of $\mathbb{R}^n$. To be precise, $\boldsymbol{\ell}_t = \boldsymbol{P}\boldsymbol{a}_t$ where $\boldsymbol{P}$ is an $n \times r$ *basis matrix*[1] with $r \ll n$. Here and below, $'$ denotes matrix transpose and $\|\cdot\|$ refers to the l_2 norm of a vector or the induced l_2 norm of a matrix. We use $\mathcal{T}_t$ to denote the support set of $\boldsymbol{x}_t$ and assume that $|\mathcal{T}_t| \leq s$ for all t. Define the $n \times d$ data matrix $\boldsymbol{Y} := [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots \boldsymbol{y}_d] = \boldsymbol{L} + \boldsymbol{X}$ where $\boldsymbol{L}$, $\boldsymbol{X}$ are similarly defined.

Given an initial subspace estimate, $\hat{\boldsymbol{P}}_{\text{init}}$, the goal is to estimate $\text{span}(\boldsymbol{P})$ within an ε-accuracy, quickly and provably. A by-product of doing this is that the true data vectors $\boldsymbol{\ell}_t$, the sparse outliers $\boldsymbol{x}_t$, and their support sets $\mathcal{T}_t$ can also be tracked on-the-fly. The initial subspace estimate, $\hat{\boldsymbol{P}}_{\text{init}}$, can be computed by applying any static (batch) RPCA technique, e.g., PCP [1] or AltProj [4], to the first t_{train} data frames, $\boldsymbol{Y}_{[1,t_{\text{train}}]}$. Here and below, $[a, b]$ refers to all integers between a and b, inclusive, $[a, b) := [a, b-1]$, and $\boldsymbol{M}_{\mathcal{T}}$ denotes a sub-matrix of $\boldsymbol{M}$ formed by columns indexed by entries in $\mathcal{T}$. For basis matrices $\hat{\boldsymbol{P}}$, $\boldsymbol{P}$, that are used to denote two r-dimensional subspaces, we use

$$\text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) := \left(\sum_{i=1}^{r} \sin^2 \theta_i(\hat{\boldsymbol{P}}, \boldsymbol{P}) \right)^{1/2}$$

to quantify the distance betwen their column spans (subspaces). Here, θ_i is the i-th principal angle, and can be computed as $\sin \theta_i(\hat{\boldsymbol{P}}, \boldsymbol{P}) = \sigma_i((\boldsymbol{I} - \hat{\boldsymbol{P}}\hat{\boldsymbol{P}}')\boldsymbol{P})$. This is popularly known as the chordal distance [8]. We mention here that we can use any metric which quantifies the change in *all principal angles* such at the *Binet-Cauchy* or *Procrustes* distance. Also, we use $\theta_{\max} = \theta_1$ and $\theta_{\min} = \theta_r$ to denote the largest and smallest principal angles respectively.

Contributions. We develop a new provably correct algorithm called MERoP (Memory-Efficient Robust PCA) to solve RPCA. MERoP is inspired by the recently introduced Recursive Projected Compressive Sensing (ReProCS) solution framework [9, 10] which was originally developed to solve the dynamic RPCA problem: this is the time-varying extension of RPCA (the subspace in which the data lies can change with time). In subsequent work [11, 12], we have obtained progressively simpler correctness guarantees for ReProCS for dynamic RPCA. In this work, we extend the ReProCS line of work to demonstrate that, under mild assumptions, the original (static) RPCA problem can also be provably solved in a fast, memory efficient fashion, and highly robust fashion, using similar ideas. We show that, the propsed algorithm, MERoP, (i) has a running time of $\mathcal{O}(ndr \log(1/\epsilon))$, which is the cost of performing a rank-r vanilla SVD on the data matrix; (ii) can tolerate an order wise larger fraction of outlier per row (can tolerate slow moving and occasionally static foreground objects' occlusions in videos) under mild assumptions on the minimum outlier magnitude; and (iii) has nearly optimal-storage complexity: we need $\mathcal{O}(nr \log n \log(1/\epsilon))$ samples to obtain an ϵ-accurate subspace estimate. This is larger than nr, which is the minimum required to even output a subspace estimate, by only logarithmic factors.

2. ALGORITHM AND MAIN RESULT

Algorithm 1 proceeds as follows. A coarse subspace estimate is obtained using PCP [1] or AltProj [4] applied to the first t_{train} frames $\boldsymbol{Y}_{[1,t_{\text{train}}]}$. For $t > t_{\text{train}}$, the algorithm proceeds as follows. At time t, let $\hat{\boldsymbol{P}}_{t-1}$ denote the subspace estimate from $(t-1)$. If this estimate is accurate enough, projecting $\boldsymbol{y}_t := \boldsymbol{x}_t + \boldsymbol{\ell}_t$ onto its orthogonal complement will nullify most of $\boldsymbol{\ell}_t$. We compute $\tilde{\boldsymbol{y}}_t := \boldsymbol{\Psi}\boldsymbol{y}_t$ where

[1] tall matrix with mutually orthonormal columns

Algorithm 1 MERoP and Offline MERoP

Input: $\hat{\boldsymbol{P}}_0$, $\boldsymbol{y}_t$, **Output**: $\hat{\boldsymbol{x}}_t$, $\hat{\boldsymbol{\ell}}_t$, $\hat{\boldsymbol{P}}$
Params: ω_{supp}, K, α, ξ, r, ω_{evals}
$\hat{\boldsymbol{P}}_{(t_{\text{train}})} \leftarrow \hat{\boldsymbol{P}}_0$; $k \leftarrow 1$.
for $t > t_{\text{train}}$ **do**
  $\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$;
  $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_t$.
  $\hat{\boldsymbol{x}}_{t,cs} \leftarrow \arg\min_{\tilde{\boldsymbol{x}}} \|\tilde{\boldsymbol{x}}\|_1$ s.t $\|\tilde{\boldsymbol{y}}_t - \boldsymbol{\Psi}\tilde{\boldsymbol{x}}\| \le \xi$.
  $\hat{\mathcal{T}}_t \leftarrow \{i : |\hat{\boldsymbol{x}}_{t,cs}| > \omega_{supp}\}$.
  $\hat{\boldsymbol{x}}_t \leftarrow \boldsymbol{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t})^{-1}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\tilde{\boldsymbol{y}}_t$.
  $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$.
  if $t = t_{\text{train}} + u\alpha$ for $u = 1, 2, \cdots, K$ **then**
    $\hat{\boldsymbol{P}}_k \leftarrow SVD_r[\hat{\boldsymbol{L}}_{t;\alpha}]$, $k \leftarrow k+1$.
  else
    $\hat{\boldsymbol{P}}_{(t)} \leftarrow \hat{\boldsymbol{P}}_{(t-1)}$
  end if
  if $t = t_{\text{train}} + K\alpha$ **then**
    $\hat{\boldsymbol{P}} \leftarrow \hat{\boldsymbol{P}}_k$
    $\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}\hat{\boldsymbol{P}}'$
  end if
end for
for $t > t_{\text{train}}$ **do**
  $\hat{\boldsymbol{x}}_t \leftarrow \boldsymbol{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t})^{-1}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\boldsymbol{y}_t$
  $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$.
end for

} Offline MERoP.

$\boldsymbol{\Psi} := \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$. Thus, $\tilde{\boldsymbol{y}}_t = \boldsymbol{\Psi}\boldsymbol{x}_t + \boldsymbol{b}_t$ where $\boldsymbol{b}_t := \boldsymbol{\Psi}\boldsymbol{\ell}_t$ and $\|\boldsymbol{b}_t\|$ is small. Recovering $\boldsymbol{x}_t$ from $\tilde{\boldsymbol{y}}_t$ is thus a regular compressive sensing (CS) / sparse recovery problem in small noise [13]. We compute $\hat{\boldsymbol{x}}_{t,cs}$ using l_1 minimization followed by thresholding based support estimation to get $\hat{\mathcal{T}}_t$. A Least Squares (LS) based debiasing step on $\hat{\mathcal{T}}_t$ returns the final $\hat{\boldsymbol{x}}_t$. We then estimate $\boldsymbol{\ell}_t$ as $\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$. The $\hat{\boldsymbol{\ell}}_t$'s are used to update the subspace estimate. This is done using K steps of r-SVD, each done with a new set of α frames of $\hat{\boldsymbol{\ell}}_t$. Here r-SVD means compute the top r left singular vectors of $\hat{\boldsymbol{L}}_{t;\alpha} := [\hat{\boldsymbol{\ell}}_{t-\alpha+1}, \hat{\boldsymbol{\ell}}_{t-\alpha+2}, \ldots, \hat{\boldsymbol{\ell}}_t]$.

Using the result of [14] we show that with high probability, the subspace estimation error decreases exponentially after every α frames, and thus, after K-SVD steps, we obtain an ε-accurate estimate of the subspace. In offline MERoP, we use this to re-estimate all the previous $\hat{\boldsymbol{x}}_t$'s and $\hat{\boldsymbol{\ell}}_t$'s which can also shown to be ε-accurate estimates. The name is a misnomer since after K-SVD steps, Offline-MERoP functions in an online manner and hence we refer to our algorithm as *nearly-online*.

2.1. Assumptions and Main Result

Assumption on principal subspace coefficients $\boldsymbol{a}_t$. We assume that the $\boldsymbol{a}_t$'s are zero mean, mutually independent, and *element-wise bounded* random variables (r.v.) with diagonal covariance matrix $\boldsymbol{\Lambda}$. Since the $\boldsymbol{a}_t$'s are element-wise bounded, there exists an $\eta < \infty$, such that $\max_{j=1,2,\ldots r} \max_t \frac{(\boldsymbol{a}_t)_j^2}{\lambda_j(\boldsymbol{\Lambda})} \le \eta$. For most bounded distributions, η is a little more than one, e.g., if the entries of $\boldsymbol{a}_t$ are zero mean uniform, then $\eta = 3$. This bounded-ness assumption is similar to the right singular vectors' incoherence assumption needed by all the other RPCA solutions [3, 4]. There are minor differences since we impose statistical assumptions on $\boldsymbol{a}_t$'s.

Incoherence left singular vectors of $\boldsymbol{L}$. In order to separate the $\boldsymbol{\ell}_t$'s from the sparse outliers $\boldsymbol{x}_t$, we need to assume that the $\boldsymbol{\ell}_t$'s are themselves not sparse (thus we sometimes refer to this property as "denseness".). This is ensured if we can assume that column vectors of $\boldsymbol{P}$ are dense enough. To quantify this, we define μ as the smallest real number that satisfies

$$\max_{i=1,2,\ldots,n} \|\boldsymbol{I}_i{}'\boldsymbol{P}\| \le \sqrt{\frac{\mu r}{n}} \tag{2}$$

The above assumption is equivalent to left incoherence needed by all existing RPCA solutions [3, 4].

Bound on outlier fractions. Similar to earlier RPCA results, we also need outlier fractions to be bounded. However, we need different bounds on this fraction per-column and per-row. The row bound can be much larger. Let max-outlier-frac-col $:= \max_t |\mathcal{T}_t|/n$ denote the maximum outlier fraction in any column of $\boldsymbol{Y}$. Since MERoP is a nearly-online algorithm, we need the fraction of outliers per row of a sub-matrix of $\boldsymbol{Y}$ with α consecutive columns to be bounded. To quantify this, for a time interval, $\mathcal{J}$, define

$$\gamma(\mathcal{J}) := \max_{i=1,2,\ldots,n} \frac{1}{|\mathcal{J}|} \sum_{t \in \mathcal{J}} \mathbf{1}_{\{i \in \mathcal{T}_t\}}. \tag{3}$$

where $\mathbf{1}_S$ is the indicator function for event S. Thus $\gamma(\mathcal{J})$ is the maximum outlier fraction in any row of the sub-matrix $\boldsymbol{Y}_{\mathcal{J}}$ of $\boldsymbol{Y}$. Let $\mathcal{J}^\alpha$ denote a time interval of duration α. We will bound

$$\text{max-outlier-frac-row} := \max_{\mathcal{J}^\alpha \subseteq [t_{\text{train}}, d]} \gamma(\mathcal{J}^\alpha). \tag{4}$$

Initialization. Assume that the initial data, $\boldsymbol{Y}_{[1,t_{\text{train}}]}$, satisfies PCP(H) [3] or AltProj [4] assumptions: (i) let $\boldsymbol{L}_{\text{init}} \overset{\text{SVD}}{=} \boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{V}'$; $\boldsymbol{P}$ and $\boldsymbol{V}$ satisfy incoherence with parameter μ, and (ii) outlier fractions per row and per column are both upper bounded by $c/(\mu r)$; and, (iii) we run enough iterations of AltProj so that the initial subspace estimate satisfies $\sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \le 0.05/\sqrt{r}$.

We use λ^+ and λ^- to denote the maximum and minimum eigenvalues of $\boldsymbol{\Lambda}$ and let $f := \lambda^+/\lambda^-$ denote the condition number.

Theorem 2.1 (RPCA). *Pick an $\varepsilon_{\text{dist}} > 0$. For $t \ge t_{\text{train}}$, assume*

1. *assumptions on $\boldsymbol{a}_t$'s hold,*
2. *the initialization assumption given above on $\hat{\boldsymbol{P}}_{init}$ holds,*
3. $0.35\sqrt{\eta\lambda^+} \le x_{\min}/15$,
4. *max-outlier-frac-row* $\le b_0/f^2$ *where* $b_0 = 0.02$, *and max-outlier-frac-col* $\le 0.09/(\mu r)$,
5. *algorithm parameters are set as* $K = c\log(1/\varepsilon_{\text{dist}})$, $\alpha = Cf^2(r\log n)$, $\xi = x_{\min}/15$, $\omega_{supp} = x_{\min}/2$

Then, with probability at least $1 - 10dn^{-10}$, the output of Offline MERoP satisfies $\text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) \le \varepsilon_{\text{dist}}$, $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\| \le (\varepsilon_{\text{dist}}/\sqrt{r})\|\boldsymbol{\ell}_t\|$ *and $\hat{\mathcal{T}}_t = \mathcal{T}_t$ for all t.*

Proof: See Appendix A.

Remark 2.2. *The lower bound on $x_{\min}$ seems counter-intuitive since small magnitude corruptions should not be problematic. With simple changes to the proof of Theorem 2.1, we can show the following more intuitive result. Define $\zeta_k := 0.3^k 0.05$. Let $\mathcal{J}_k := [t_{\text{train}} + (k-1)\alpha, t_{\text{train}} + k\alpha)$. Make the following changes to Theorem 2.1.*

1. *Assume that $\boldsymbol{x}_t$'s and $\boldsymbol{\ell}_t$'s are mutually independent*
2. *Pick* $\varepsilon_{\text{dist}} < \min_t \min_{i \in \mathcal{T}_t} |(\boldsymbol{x}_t)_i|/30$

Table 1: Comparing assumptions, time and memory complexity. For simplicity, we ignore all dependence on condition numbers.

Algorithm	Outlier tolerance, rank of $(\boldsymbol{L})$	Assumptions	Memory, Time,	# params.
PCP(C)[1] (offline)	max-outlier-frac-row $= \mathcal{O}(1)$ max-outlier-frac-col $= \mathcal{O}(1)$ $r \leq \frac{c \min(n,d)}{\log^2 n}$	strong incoh, unif. rand. support,	Mem: $\mathcal{O}(nd)$ Time: $\mathcal{O}(nd^2 \frac{1}{\epsilon})$	zero
AltProj[4], (offline)	max-outlier-frac-row $= O(1/r)$ max-outlier-frac-col $= O(1/r)$		Mem: $\mathcal{O}(nd)$ Time: $\mathcal{O}(ndr^2 \log \frac{1}{\epsilon})$	2
RPCA-GD [5] (offline)	max-outlier-frac-row $= \mathcal{O}(1/r^{3/2})$ max-outlier-frac-col $= \mathcal{O}(1/r^{3/2})$		Mem: $\mathcal{O}(nd)$ Time: $\mathcal{O}(ndr \log \frac{1}{\epsilon})$	5
PG-RMC [6] (offline)	max-outlier-frac-row $= O(1/r)$ max-outlier-frac-col $= \mathcal{O}(1/r)$	$d \approx n$	Mem: $\mathcal{O}(nd)$ Time: $\mathcal{O}(nr^3 \log n \log \frac{1}{\epsilon})$	4
MERoP (this work) (online and offline)	**max-outlier-frac-row** $= \mathcal{O}(1)$ **max-outlier-frac-col** $= \mathcal{O}(1/r)$	$\boldsymbol{a}_t$'s independent, init data: AltProj assu's, outlier mag. lower bounded	**Mem:** $\mathcal{O}(nr \log n \log \frac{1}{\epsilon})$ **Time:** $\mathcal{O}(ndr \log \frac{1}{\epsilon})$	4

Note: The table assumes an $n \times d$ data matrix $\boldsymbol{Y} := \boldsymbol{L} + \boldsymbol{X}$, where $\boldsymbol{L}$ has rank r and the outlier matrix $\boldsymbol{X}$ is sparse. It compares MERoP for solving the original robust PCA problem with other methods for solving the same problem. Thus the subspace that MERoP recovers is also of dimension $r = r$. With only a mild extra assumption on outlier magnitudes, MERoP is able to achieve a significant gain in outlier tolerance per row. We also note that all the algorithms require *left and right* incoherence, and thus we do not list this in the third column.

3. *Use the following to replace Item 3 of Theorem 2.1. For $t \in \mathcal{J}_k$, define the $\widetilde{\mathcal{T}}_t := \{i : |(\boldsymbol{x}_t)_i| > 30\zeta_k\sqrt{\eta\lambda^+}\}$. Define $\tilde{\boldsymbol{x}}_t := \boldsymbol{I}_{\widetilde{\mathcal{T}}_t}\boldsymbol{I}_{\widetilde{\mathcal{T}}_t}{}'\boldsymbol{x}_t$. Let $\boldsymbol{v}_t := \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$. Assume that $\|\boldsymbol{v}_t\| \leq \zeta_k\sqrt{\eta\lambda^+}$. (Thus $\tilde{\boldsymbol{x}}_t$ contains the entries of $\boldsymbol{x}_t$ whose magnitude is larger than $30\zeta_k\sqrt{\eta\lambda^+}$. These are the "real" outliers while $\boldsymbol{v}_t$ contains the smaller magnitude corruptions. Our assumption requires that the small magnitude corruptions are small enough so that $15(\zeta_k + \|\boldsymbol{v}_t\|)$ is smaller than the minimum non-zero entry of $\tilde{\boldsymbol{x}}_t$.*
4. *Assume that the $\widetilde{\mathcal{T}}_t$'s satisfies the max-outlier-frac-row and max-outlier-frac-col bounds (Item 4 of Theorem 2.1).*

Then, with the same probability of success, the output of Algorithm 1 satisfies all the conditions of Theorem 2.1 with $\widetilde{\mathcal{T}}_t$ replaced with $\mathcal{T}_t$.

3. DISCUSSION

There is a vast body of literature on analyzing RPCA. The first papers to provide guarantees for RPCA were [1, 2] which studied a convex relaxation to recover the "sparsest" $\boldsymbol{X}$ and the "least-rank" $\boldsymbol{L}$ using the l_1 norm, and the nuclear norm, respectively. Although these ideas offer an elegant theory, they are slow in practice: to obtain an ϵ-accurate solution, the time complexity is $\mathcal{O}(nd^2/\epsilon)$. Since then, there has been a rich development of faster and more efficient algorithms to attack this problem. In particular, AltProj [4] was one of the first algorithms to study a provable non-convex focmulation of RPCA. Their method relied on starting with an initial guess of the sparse outliers (which consisted of the "largest entries" of the observed matrix), followed by alternately projecting the "residuals" onto the non-convex sets of sparse, and, low-rank matrices. To circumvent the problem of sensitivity of the SVD to outliers, the algorithm proceeds in stages; incrementing the "target rank" at each stage until a halting criterion is attained. This algorithm was shown to be significantly faster than the convex approaches and enjoyed a run-time of $\mathcal{O}(ndr^2 \log(1/\epsilon))$. Following this, there have been additional improvements to the running time of subsequent algorithms. A recent work RPCA-GD [5] showed that it is indeed possible to design an algorithm that runs in time $\mathcal{O}(ndr \log(1/\epsilon))$. However, this approach requires a tighter bound on the allowed outlier-fractions (see Table 1). A recently proposed algorithm, PG-RMC [6] is the fastest exisiting batch algorithm for the RPCA with a run time of $\mathcal{O}(nr^3 \log n \log(1/\epsilon))$. The reason for the nearly-linear speed is that PG-RMC uniformly-randomly selects a few entries of the data matrix, and thus needs to work with a fewer number of samples. Although this presents a significant improvement in applications where running time is a severe bottleneck, two notable disadvantages of this approach are that (i) one cannot recover the sparse matrix due to the uniform random sampling of the data matrix and (ii) it requires d to be of the same order as n. This is a very a strong requirement in the high-dimensional regime. All other methods including ours just require $d \geq cr \log n$. For a summary, see Table 1.

Solutions to RPCA in the online setting have also received considerable interest. We discuss a few most relevant algorithms and note that this is by no means an exhaustive list. Firstly, there are algorithms based on the GRASTA framework, which also rely on the ReProCS framework [9, 10] in the sense of alternating sparse recovery and subspace update for incoming data; but use stochastic Gradient Descent for the latter [15, 16]. Although this is very fast, the authors do not provide theoretical guarantees and the algorithm performance is not good for large sized or slow moving foregrounds (see next section). Another approach is the "online-reformulation" of PCP [17], based on stochastic optimization techniques. This provies a partial guarantee that requires assumptions on intermediate algorithm estimates (assumes that the intermediate subspace estimates are full rank), and the result is only an asymptotic convergence guarantee. Finally, a streaming algorithm using the ideas of AltProj [4] – but replacing SVD by a block-stochastic power-method was proposed in [18]. The advantage of such an approach is that the algorithm needs only a single pass over the data samples. However, the algorithm only provably works in case of one-dimensional RPCA, i.e., $r = 1$.

To the best of our knowledge, there is no algorithm that simulatenously addresses the following two questions. (A) What is the minimum number of samples needed to guarantee an ϵ-accurate solution in some metric? (B) Is it possible to obtain such an estimate

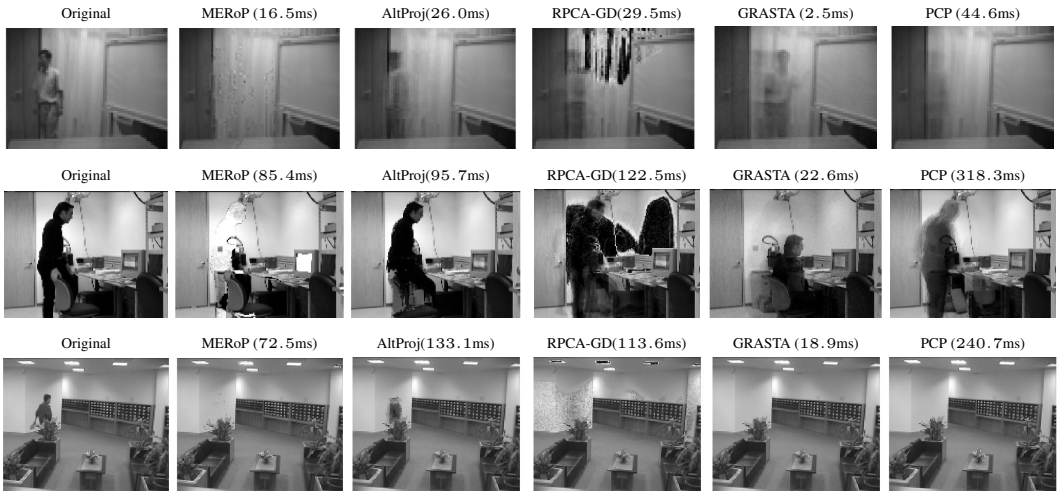

Fig. 1: Background recovery. For MR and SL sequences (first two rows), only MERoP background does not contain the person or even his shadow. All others do. Also MERoP is faster than all except GRASTA. For LB, MERoP is as good as PCP and GRASTA, while others fail. Time taken per frame is shown in parentheses.

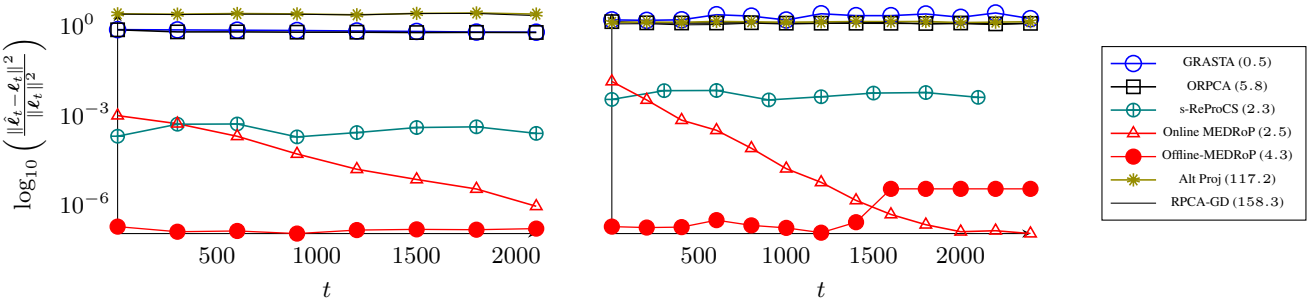

Fig. 2: Relative error in recovering $\boldsymbol{\ell}_t$'s. Left: Moving object model on $\mathcal{T}_t$. Right: Bernoulli model on $\mathcal{T}_t$. Time taken per frame in milliseconds (ms) for the Bernoulli model is shown in parentheses in the legend. The errors are plotted every $k\alpha - 1$ time-frames. Observe the nearly-exponential decay of the subspace error with time.

whilst using nearly optimal memory? Our proposed approach does both.

4. EMPIRICAL EVALUATION

In this section we present the results of numerical experiments to compare the performance of MERoP with existing state-of-the-art algorithms on synthetic data and in the task of Foreground-Background separation in real videos. All experiments are performed on a Desktop Computer with Intel® Xeon E3-1240 8-core CPU @ 3.50GHz and 32GB RAM.

Synthetic Data. We perform an experiment on synthetic data to demonstrate the superiority of MEDRoP over existing algorithms. We generate the data as follows. $\boldsymbol{P}_0$ is generated by ortho-normalizing the columns of an $n \times r$ i.i.d standard normal matrix. We used $n = 1000$, $r = 30$, $d = 3000$. For the low-rank matrix $\boldsymbol{L}$ we generate the coefficients $\boldsymbol{a}_t \in \mathbb{R}^r$ according to $(\boldsymbol{a}_t)_i \overset{i.i.d}{\sim} Unif[-q_i, q_i]$ where $q_i = \sqrt{f} - \sqrt{f}i/2r$ for $i = 1, 2, \cdots, r-1$ and $q_r = 1$. thus the condition number is f and we selected $f = 50$. We used the first $t_{\text{train}} = 300$ frames as the training part, where we generated a smaller fraction of outliers. For the moving object model (see Appendix [12, Model G.24]) with parameters $s/n = 0.01$, $b_0 = 0.01$ and for $t > t_{\text{train}}$ we used $s/n = 0.05$ and $b_0 = 0.3$. For the Bernoulli model we set $\rho = 0.01$ for the first t_{train} frames and $\rho = 0.3$ for the subsequent frames. The sparse outlier magnitudes are generated uniformly at random from the interval $[x_{\min}, x_{\max}]$ with $x_{\min} = 10$ and $x_{\max} = 20$ in both experiments. The results are averaged over 50 independent trials. The results are shown in Fig. 2.

We initialized MERoP[2] and s-ReProCS [12] using AltProj [4] applied to $\boldsymbol{Y}_{[1,t_{\text{train}}]}$. The smaller outlier fraction helped achieve $\sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}_0) \approx 10^{-3}$. For the batch methods used in the comparisons – PCP, AltProj and RPCA-GD, we implement the algorithms on $\boldsymbol{Y}_{[1,t]}$. Further, we set the regularization parameter for PCP $1/\sqrt{n}$ in accordance with [1]. The other known parameters, r for Alt-Proj, outlier-fraction for RPCA-GD, are set using the true values. For online methods we implement ORPCA by [17] and GRASTA by [19]. The regularization parameter for ORPCA was set as with $\lambda_1 = 1/\sqrt{n}$ and $\lambda_2 = 1/\sqrt{d}$ according to [17].

Video Experiments. We also illustrate the efficacy of MERoP on real videos in this section. In particular, we implement several algorithms on three datasets, `MR` (Meeting Room), `SL` (Switch Light) and `LB` (Lobby) which are a few commonly accepted benchmark datasets in background separation. We show one recovered background frame for each video in Fig. 1. All algorithms used $r = 40$ and default parameters in their code. MERoP used $\alpha = 60$, $K = 3$, $\xi_t = \|\boldsymbol{\Psi}\hat{\boldsymbol{\ell}}_{t-1}\|$, $\omega_{supp} = \|\boldsymbol{y}_t\|/\sqrt{n}$, $\omega_{evals} = 0.011\lambda^-$. ORPCA failed completely, gave a black background. Hence it is not shown. Time taken in milliseconds per frame is shown above each image.

[2]All the codes can be found at https://github.com/praneethmurthy/reprocs

5. REFERENCES

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

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

[3] D. Hsu, S.M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Trans. Info. Th.*, Nov. 2011.

[4] P. Netrapalli, U N Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Neural Info. Proc. Sys. (NIPS)*, 2014.

[5] Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis, "Fast algorithms for robust pca via gradient descent," in *Neural Info. Proc. Sys. (NIPS)*, 2016.

[6] Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain, "Nearly-optimal robust matrix completion," *arXiv preprint arXiv:1606.07315*, 2016.

[7] Florian Seidel, Clemens Hage, and Martin Kleinsteuber, "pROST: a smoothed\ell _p-norm robust online subspace tracking method for background subtraction in video," *Machine vision and applications*, vol. 25, no. 5, pp. 1227–1240, 2014.

[8] Ke Ye and Lek-Heng Lim, "Schubert varieties and distances between subspaces of different dimensions," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1176–1197, 2016.

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

[10] 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.*, pp. 5007–5039, August 2014.

[11] J. Zhan, B. Lois, H. Guo, and N. Vaswani, "Online (and Offline) Robust PCA: Novel Algorithms and Performance Guarantees," in *Intnl. Conf. Artif. Intell. and Stat. (AISTATS)*, 2016.

[12] P. Narayanamurthy and N. Vaswani, "New Results for Provable Dynamic Robust PCA," *arXiv:1705.08948*, 2017.

[13] 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.

[14] N. Vaswani and P. Narayanamurthy, "PCA in Data-Dependent Noise: Nearly Optimal Finite Sample Guarantees," *arXiv:1702.03070*, 2017.

[15] Laura Balzano, Robert Nowak, and Benjamin Recht, "Online identification and tracking of subspaces from highly incomplete information," *arXiv:1012.1086v3*, 2010.

[16] 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.

[17] J. Feng, H. Xu, and S. Yan, "Online robust pca via stochastic optimization," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013.

[18] UN Niranjan and Yang Shi, "Streaming robust pca," 2016.

[19] Laura Balzano, Robert Nowak, and Benjamin Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 704–711.

A. PROOF OF MAIN RESULT

In this section we provide the proof of Theorem 2.1. To prove this, we first state two main Lemmas which are used in the proof. We prove the Lemmas after proving the Theorem. We use the following definitions in our proof

1. $\Delta_{\text{init}} := 0.05$, $g_k = (0.3)^k \Delta_{\text{init}}$
2. $\Gamma_0 = \sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \le \Delta_{\text{init}}/\sqrt{r}$
3. $\Gamma_k = \{\Gamma_{k-1} \cap \sin\theta_{\max}(\hat{\boldsymbol{P}}_k, \boldsymbol{P}) \le g_k)\}$

Lemma A.3 (MERoP First Subspace Update)**.** *Under the conditions of Theorem 2.1, conditioned on* Γ_0

1. *For all* $t \in [t_{\text{train}}, t_{\text{train}} + \alpha)$*, the error* $\boldsymbol{e}_t = \hat{\boldsymbol{x}}_t - \boldsymbol{x}_t = \boldsymbol{\ell}_t - \hat{\boldsymbol{\ell}}_t$ *satisfies*
$$\boldsymbol{e}_t = \boldsymbol{I}_{\mathcal{T}_t} \left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}' \boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1} \boldsymbol{I}_{\mathcal{T}_t}{}' \boldsymbol{\Psi}_{\mathcal{T}_t} \boldsymbol{\ell}_t, \tag{5}$$
and $\|\boldsymbol{e}_t\| \le 1.2\varepsilon_{\text{dist}}\sqrt{\eta\lambda^+}$.
2. *With probability at least* $1 - 10n^{-10}$ *the subspace estimate* $\hat{\boldsymbol{P}}_1$ *satisfies* $\sin\theta_{\max}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}) \le g_1$*, i.e.,* Γ_1 *holds*

Lemma A.4 (MERoP k-th Subspace Update)**.** *Under the conditions of Theorem 2.1, conditioned on* $\Gamma_{j,k-1}$

1. *For all* $t \in [t_{\text{train}} + (k-1)\alpha, t_{\text{train}} + k\alpha)$*, the error* $\boldsymbol{e}_t = \hat{\boldsymbol{x}}_t - \boldsymbol{x}_t = \boldsymbol{\ell}_t - \hat{\boldsymbol{\ell}}_t$ *satisfies* (8) *and for this interval,* $\|\boldsymbol{e}_t\| \le (0.3)^{k-1} \cdot 1.2\varepsilon_{\text{dist}}\sqrt{\eta\lambda^+}$.
2. *With probability at least* $1 - 10n^{-10}$ *the subspace estimate* $\hat{\boldsymbol{P}}_k$ *satisfies* $\sin\theta_{\max}(\hat{\boldsymbol{P}}_k, \boldsymbol{P}) \le (0.3)^k g_1$*, i.e.,* Γ_k *holds.*

Proof of Theorem 2.1. Notice from the definitions that if we show $\Pr(\Gamma_K|\Gamma_0) \ge 1 - dn^{-10}$ we are done. Also note from the definitions that $\Gamma_K \subseteq \Gamma_{K-1} \subseteq \cdots \subseteq \Gamma_0$ and thus,
$$\begin{aligned}\Pr(\Gamma_K|\Gamma_0) &= \Pr(\Gamma_K, \Gamma_{K-1}, \cdots, \Gamma_1|\Gamma_0) \\ &= \prod_{k=1}^{K} \Pr(\Gamma_k|\Gamma_{k-1}) \overset{(a)}{\ge} (1 - 10n^{-10})^K \\ &\ge 1 - 10dn^{-10}\end{aligned}$$
where (a) used Lemmas B.6 and B.7. ∎

We now prove Lemmas B.6 and B.7. The two crucial ideas that are used to prove Lemmas B.6 and B.7 are (i) Using the idea of [10] to relate the order s-Restricted Isometry Constant of the projection matrix, $\boldsymbol{\Psi}$ to the left-incoherence property as
$$\delta_s(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}') = \max_{|\mathcal{T}|\le s} \|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\|^2 \le s \max_i \|\boldsymbol{I}_i{}'\boldsymbol{P}\|^2 \tag{6}$$
To illustrate briefly, we show that the matrix $\boldsymbol{I} - \hat{\boldsymbol{P}}_{\text{init}}\hat{\boldsymbol{P}}_{\text{init}}{}'$ satisfies the $2s$-RIP using (6) and $\sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \le \Delta_{\text{init}}$, followed by the left-incoherence assumption on $\boldsymbol{P}$ and obtain $\delta_{2s}(\boldsymbol{I} - \hat{\boldsymbol{P}}_{\text{init}}\hat{\boldsymbol{P}}_{\text{init}}{}') \le 0.15$. A similar idea can be used to show that the RIC of all subsequent matrices are small constants. This allows us to get precise bounds on the reconstruction error of $\hat{\boldsymbol{x}}_t$, and subsequently guarantee exact support recovery. (ii) We now estimate (or update) $\hat{\boldsymbol{P}}_k$ using the second crucial idea from [14]. This gives guarantees for PCA in sparse-data-dependent-noise (PCA-SDDN). Once we obtain high probability conditional probabilities, choosing the value of K as given in the Theorem ensures that the estimate after K-SVD steps is an ε-accurate estimate.

Proof of Lemma B.6. We now provide the analysis for the first α frames. For the sparse recovery step, we wish to compute the $2s$-RIP for the matrix $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{\text{init}}\hat{\boldsymbol{P}}_{\text{init}}{}'$. To do this, we first obtain bound on $\max_{|\mathcal{T}|\le 2s} \|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_{\text{init}}\|$ as follows. Consider any set $\mathcal{T}$ such that $|\mathcal{T}| \le 2s$. Then,
$$\begin{aligned}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_{\text{init}}\right\| &\le \left\|\boldsymbol{I}_{\mathcal{T}}{}'(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}')\hat{\boldsymbol{P}}_{\text{init}}\right\| + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\boldsymbol{P}'\hat{\boldsymbol{P}}_{\text{init}}\right\| \\ &\le \sin\theta_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}_{\text{init}}) + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\| \\ &\overset{(a)}{=} \sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\|\end{aligned}$$
where (a) used the fact that for basis matrices $\hat{\boldsymbol{P}}$, $\boldsymbol{P}$ of equal dimensions, $\sin\theta_{\max}(\hat{\boldsymbol{P}}, \boldsymbol{P}) = \sin\theta_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}})$. Using the definition of μ, and the bound on max-outlier-frac-col (condition 4 of Theorem 2.1),
$$\max_{|\mathcal{T}|\le 2s} \|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\|^2 \le 2s \max_i \|\boldsymbol{I}_i{}'\boldsymbol{P}\|^2 \le \frac{2s\mu r}{n} \le 0.09 \tag{7}$$
Additionally, $\text{dist}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \le \Delta_{\text{init}} \implies \sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \le \Delta_{\text{init}}$ and then taking the maximum over all such sets, we get
$$\max_{|\mathcal{T}|\le 2s} \left\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_{\text{init}}\right\| \le \varepsilon + \max_{|\mathcal{T}|\le 2s} \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\| \le \varepsilon + 0.3$$
Finally from (6), it follows that $\delta_{2s}(\boldsymbol{\Psi}) \le 0.35^2 < 0.15$,
$$\left\|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\right\| \le \frac{1}{1 - \delta_s(\boldsymbol{\Psi})} \le \frac{1}{1 - \delta_{2s}(\boldsymbol{\Psi})} \le \frac{1}{1 - 0.15} < 1.2 = \phi^+.$$
This gives
$$\begin{aligned}\|\boldsymbol{\Psi}\boldsymbol{\ell}_t\| &= \left\|(\boldsymbol{I} - \hat{\boldsymbol{P}}_{\text{init}}\hat{\boldsymbol{P}}_{\text{init}}{}')\boldsymbol{P}\boldsymbol{a}_t\right\| \le \sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) \|\boldsymbol{a}_t\| \\ &\le \sin\theta_{\max}(\boldsymbol{P}_{\text{init}}, \boldsymbol{P})\sqrt{\eta r\lambda^+} \overset{(b)}{\le} 0.05\sqrt{\eta\lambda^+} := b_b\end{aligned}$$
where (b) uses $\Delta_{\text{init}} = 0.05$. Thus, under the condition of Theorem 2.1, $b_b < x_{\min}/15$ and ensures exact support recovery as follows. We set $\xi = x_{\min}/15$. Using these facts, and $\delta_{2s}(\boldsymbol{\Psi}) \le 0.12 < 0.15$, [13, Theorem 1.2] implies that
$$\|\hat{\boldsymbol{x}}_{t,cs} - \boldsymbol{x}_t\| \le 7\xi = 7x_{\min}/15$$
Thus,
$$|(\hat{\boldsymbol{x}}_{t,cs} - \boldsymbol{x}_t)_i| \le \|\hat{\boldsymbol{x}}_{t,cs} - \boldsymbol{x}_t\| \le 7x_{\min}/15 < x_{\min}/2$$
We have $\omega_{supp} = x_{\min}/2$. Consider an index $i \in \mathcal{T}_t$. Since $|(\boldsymbol{x}_t)_i| \ge x_{\min}$,
$$x_{\min} - |(\hat{\boldsymbol{x}}_{t,cs})_i| \le |(\boldsymbol{x}_t)_i| - |(\hat{\boldsymbol{x}}_{t,cs})_i| \le |(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t,cs})_i| < \frac{x_{\min}}{2}$$
Thus, $|(\hat{\boldsymbol{x}}_{t,cs})_i| > \frac{x_{\min}}{2} = \omega_{supp}$ which means $i \in \hat{\mathcal{T}}_t$. Hence $\mathcal{T}_t \subseteq \hat{\mathcal{T}}_t$. Next, consider any $j \notin \mathcal{T}_t$. Then, $(\boldsymbol{x}_t)_j = 0$ and so
$$|(\hat{\boldsymbol{x}}_{t,cs})_j| = |(\hat{\boldsymbol{x}}_{t,cs})_j)| - |(\boldsymbol{x}_t)_j| \le |(\hat{\boldsymbol{x}}_{t,cs})_j - (\boldsymbol{x}_t)_j| \le b_b < \frac{x_{\min}}{2}$$
which implies $j \notin \hat{\mathcal{T}}_t$ and $\hat{\mathcal{T}}_t \subseteq \mathcal{T}_t$ implying that $\hat{\mathcal{T}}_t = \mathcal{T}_t$. With $\hat{\mathcal{T}}_t = \mathcal{T}_t$ and since $\mathcal{T}_t$ is the support of $\boldsymbol{x}_t$, $\boldsymbol{x}_t = \boldsymbol{I}_{\mathcal{T}_t}\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{x}_t$, and so
$$\begin{aligned}\hat{\boldsymbol{x}}_t &= \boldsymbol{I}_{\mathcal{T}_t}\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'(\boldsymbol{\Psi}\boldsymbol{\ell}_t + \boldsymbol{\Psi}\boldsymbol{x}_t) \\ &= \boldsymbol{I}_{\mathcal{T}_t}\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}\boldsymbol{\ell}_t + \boldsymbol{x}_t\end{aligned}$$

since $\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi} = \boldsymbol{I}'_{\mathcal{T}_t}\boldsymbol{\Psi}'\boldsymbol{\Psi} = \boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}$. Thus $\boldsymbol{e}_t = \hat{\boldsymbol{x}}_t - \boldsymbol{x}_t$ satisfies,

$$\boldsymbol{e}_t = \boldsymbol{I}_{\mathcal{T}_t}\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}\boldsymbol{\ell}_t$$
$$\|\boldsymbol{e}_t\| \le \left\|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\right\| \left\|\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}\boldsymbol{\ell}_t\right\| \le \phi^+ \left\|\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}\boldsymbol{\ell}_t\right\| \le 1.2b_b$$

Proof of Item 2: Now, updating $\hat{\boldsymbol{P}}_{(t)}$ from the $\hat{\boldsymbol{\ell}}_t$'s is a problem of PCA in sparse data-dependent noise. We use the result of Theorem [14, Remark 4.18]. Recall from above that for $t \in [t_{\text{train}}, t_{\text{train}} + \alpha]$, $\hat{\mathcal{T}}_t = \mathcal{T}_t$, and $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t - \boldsymbol{e}_t$. We estimate the new subspace, $\hat{\boldsymbol{P}}_1$ as the top r eigenvectors of $\frac{1}{\alpha}\sum_{t=t_{\text{train}}}^{t_{\text{train}}+\alpha-1}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t{}'$. In the setting above, $\boldsymbol{y}_t \equiv \hat{\boldsymbol{\ell}}_t$, $\boldsymbol{w}_t \equiv \boldsymbol{e}_t$, $\boldsymbol{\ell}_t \equiv \boldsymbol{\ell}_t$ and $\boldsymbol{M}_{s,t} = -\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'$ and so $\|\boldsymbol{M}_{s,t}\boldsymbol{P}\| = \|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{P}\| \le \phi^+ \sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}) := q_0$. Applying the PCA-SDDN result with $q \equiv q_0$, $b_0 \equiv$ max-outlier-frac-row and setting $\varepsilon_{\text{SE}} = g_1$, observe that we require

$$\sqrt{b_0}q_0 f \le \frac{0.9g_1}{1+g_1}$$

which holds if $\sqrt{b_0}f \le 0.12$ as provided by Theorem 2.1. Thus, with probability at least $1 - 10n^{-10}$, $\sin\theta_{\max}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}) \le g_1$. Consequently $\text{dist}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}) \le \sqrt{r}g_1 \le (\phi^+/4)0.05$. ■

Proof of Lemma B.7. The proof of this lemma has many important differences with respect to Lemma B.6. We first present the proof for $k = 2$ case and subsequently generalize it for an arbitrary k-th SVD step.

(A) $k = 2$

Proof of Item 1: For the sparse recovery step, we need to bound the $2s$-RIC for the matrix $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_1\hat{\boldsymbol{P}}_1{}'$. The approach is to first show that the matrix $\hat{\boldsymbol{P}}_1$ is dense, because $\sin\theta_{\max}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}) \le g_1 = q_0/4$ and by assumption $\boldsymbol{P}$ is dense. Concretely, $\max_{|\mathcal{T}|\le 2s}\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_1\|$ can be bounded as follows. Consider any set $\mathcal{T}$ such that $|\mathcal{T}| \le 2s$. Then,

$$\begin{aligned}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_1\right\| &\le \left\|\boldsymbol{I}_{\mathcal{T}}{}'(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}')\hat{\boldsymbol{P}}_1\right\| + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\boldsymbol{P}'\hat{\boldsymbol{P}}_1\right\| \\ &\le \sin\theta_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}_1) + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\| \\ &= \sin\theta_{\max}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}) + \left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\|\end{aligned}$$

Using (9),

$$\max_{|\mathcal{T}|\le 2s}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_1\right\| \le q_0/4 + \max_{|\mathcal{T}|\le 2s}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\| \le q_0/4 + 0.3$$

Finally, from using the assumptions of Theorem 2.1, it follows that $q_0 \le 0.06$ and subsequently $\delta_{2s}(\boldsymbol{\Psi}) \le 0.315^2 < 0.15$. From, this

$$\left\|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\right\| \le \frac{1}{1-\delta_s(\boldsymbol{\Psi})} \le \frac{1}{1-\delta_{2s}(\boldsymbol{\Psi})} \le 1.2 = \phi^+.$$

We also have that

$$\begin{aligned}\|\boldsymbol{\Psi}\boldsymbol{\ell}_t\| &= \left\|(\boldsymbol{I} - \hat{\boldsymbol{P}}_1\hat{\boldsymbol{P}}_1{}')\boldsymbol{P}\boldsymbol{a}_t\right\| \le \sin\theta_{\max}(\hat{\boldsymbol{P}}_1, \boldsymbol{P})\|\boldsymbol{a}_t\| \\ &\le (q_0/4)\sqrt{\eta r\lambda^+} \overset{(a)}{\le} (\phi^+/4)\sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}))\sqrt{\eta r\lambda^+} \\ &\overset{(b)}{\le} (\phi^+/4)(0.05)\sqrt{\eta\lambda^+} := 0.3b_b\end{aligned}$$

where (a) follows from using Lemma B.6 and (b) follows from the assumption of Theorem 2.1. Furthermore $0.3b_b < b_b < x_{\min}/15$ ensures exact support recovery exactly as in Lemma B.6.

Proof of Item 2: Again, updating $\hat{\boldsymbol{P}}_{(t)}$ using $\hat{\boldsymbol{\ell}}_t$'s is a problem of PCA in sparse data-dependent noise. We use the result of [14, Remark 4.18]. Recall from *proof of item 1* that for $t \in [t_{\text{train}} + \alpha, t_{\text{train}} + 2\alpha]$, $\hat{\mathcal{T}}_t = \mathcal{T}_t$, and $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t - \boldsymbol{e}_t$. We estimate the new subspace, $\hat{\boldsymbol{P}}_2$ as the top r eigenvectors of $\frac{1}{\alpha}\sum_{t=t_{\text{train}}+\alpha}^{t_{\text{train}}+2\alpha-1}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t{}'$. In the setting above, $\boldsymbol{y}_t \equiv \hat{\boldsymbol{\ell}}_t$, $\boldsymbol{w}_t \equiv \boldsymbol{e}_t$, $\boldsymbol{\ell}_t \equiv \boldsymbol{\ell}_t$, and, $\boldsymbol{M}_{s,t} = -\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'$ and so $\|\boldsymbol{M}_{s,t}\boldsymbol{P}\| = \|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{P}\| \le (\phi^+/4)q_0 := q_1$. Now, applying the PCA-SDDN result with $q \equiv q_1$, $b_0 \equiv$ max-outlier-frac-row, and setting $\varepsilon_{\text{SE}} = g_2 = q_1/4$, observe that we require

$$\sqrt{b_0}q_1 f \le \frac{0.9g_2}{1+g_2}$$

which holds if $\sqrt{b_0}f \le 0.12$. Thus, with probability at least $1 - 10n^{-10}$, $\sin\theta_{\max}(\hat{\boldsymbol{P}}_2, \boldsymbol{P}) \le g_2 = (\phi^+/4)g_1$. In other words, with probability at least $1 - 10n^{-10}$, conditioned on Γ_1, Γ_2 holds. Furthermore, as a direct corollary it also follows that $\text{dist}(\hat{\boldsymbol{P}}_2, \boldsymbol{P}) \le \sqrt{r}g_2 \le (\phi^+/4)^2 0.05$.

(B) General k

Proof of Item 1: Now consider the interval $[t_{\text{train}} + (k-1)\alpha, t_{\text{train}} + k\alpha]$. Using the same idea as for the $k = 2$ case, we have that for the k-th interval, $q_{k-1} = (\phi^+/4)^{k-1}q_0$ and $\varepsilon_{\text{SE}} = g_k$. From this it is easy to see that

$$\begin{aligned}\delta_{2s}(\boldsymbol{\Psi}) &\le \left(\max_{|\mathcal{T}|\le 2s}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\hat{\boldsymbol{P}}_{k-1}\right\|\right)^2 \\ &\le (\sin\theta_{\max}(\hat{\boldsymbol{P}}_{k-1}, \boldsymbol{P}) + \max_{|\mathcal{T}|\le 2s}\left\|\boldsymbol{I}_{\mathcal{T}}{}'\boldsymbol{P}\right\|)^2 \\ &\overset{(a)}{\le} (\sin\theta_{\max}(\hat{\boldsymbol{P}}_{k-1}, \boldsymbol{P}) + 0.3)^2 \\ &\le ((\phi^+/4)^{k-1}0.05 + 0.3)^2 < 0.15\end{aligned}$$

where (a) follows from (9). Using the approach Lemma B.6,

$$\begin{aligned}\|\boldsymbol{\Psi}\boldsymbol{\ell}_t\| &\le \sin\theta_{\max}(\hat{\boldsymbol{P}}_{k-1}, \boldsymbol{P})\|\boldsymbol{a}_t\| \le (\phi^+/4)^{k-1}\sin\theta_{\max}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P})\sqrt{\eta r\lambda^+} \\ &\le (\phi^+/4)^{k-1}0.05\sqrt{\eta\lambda^+} := (\phi^+/4)^{k-1}b_b\end{aligned}$$

Proof of Item 2: Again, updating $\hat{\boldsymbol{P}}_{(t)}$ from $\hat{\boldsymbol{\ell}}_t$'s is a problem of PCA-SDDN. From *proof of Item 1* for $t \in [t_{\text{train}} + (k-1)\alpha, t_{\text{train}} + k\alpha]$, $\hat{\mathcal{T}}_t = \mathcal{T}_t$, and $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t - \boldsymbol{e}_t$. We update the subspace, $\hat{\boldsymbol{P}}_k$ as the top r eigenvectors of $\frac{1}{\alpha}\sum_{t=t_{\text{train}}+(k-1)\alpha}^{t_{\text{train}}+k\alpha-1}\hat{\boldsymbol{\ell}}_t\hat{\boldsymbol{\ell}}_t{}'$. In the setting above $\boldsymbol{y}_t \equiv \hat{\boldsymbol{\ell}}_t$, $\boldsymbol{w}_t \equiv \boldsymbol{e}_t$, $\boldsymbol{\ell}_t \equiv \boldsymbol{\ell}_t$, and $\boldsymbol{M}_{s,t} = -\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'$, and so $\|\boldsymbol{M}_{s,t}\boldsymbol{P}\| = \|\left(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t}\right)^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{P}\| \le (\phi^+/4)^{k-1}q_0 := q_{k-1}$. Now applying the PCA-SDDN result with $q \equiv q_{k-1}$, $b_0 \equiv$ max-outlier-frac-row, and setting $\varepsilon_{\text{SE}} = q_{k-1}/4 = g_k$, observe that we require

$$\sqrt{b_0}q_{k-1}f \le \frac{0.9g_k}{1+g_k}$$

which holds if $\sqrt{b_0}f \le 0.12$ as provided by Theorem 2.1. Thus, with probability at least $1 - 10n^{-10}$, $\sin\theta_{\max}(\hat{\boldsymbol{P}}_k, \boldsymbol{P}) \le g_k = (\phi^+/4)^{k-1}g_1$. In other words, with probability at least $1 - 10n^{-10}$, conditioned on Γ_{k-1}, Γ_k holds. Furthermore, as a direct corollary it follows that $\text{dist}(\hat{\boldsymbol{P}}_k, \boldsymbol{P}) \le \sqrt{r}g_k \le (\phi^+/4)^k 0.05$. ■