

MOSES: A Streaming Algorithm for Linear Dimensionality Reduction

Armin Eftekhari, Raphael A. Hauser, Andreas Grammenos



Abstract—This paper introduces Memory-limited Online Subspace Estimation Scheme (MOSES) for both estimating the principal components of streaming data and reducing its dimension. More specifically, in various applications such as sensor networks, the data vectors are presented sequentially to a user who has limited storage and processing time available. Applied to such problems, MOSES can provide a running estimate of leading principal components of the data that has arrived so far and also reduce its dimension.

MOSES generalises the popular incremental Singular Value Decomposition (iSVD) to handle thin blocks of data, rather than just vectors. This minor generalisation in part allows us to complement MOSES with a comprehensive statistical analysis, thus providing the first theoretically-sound variant of iSVD, which has been lacking despite the empirical success of this method. This generalisation also enables us to concretely interpret MOSES as an approximate solver for the underlying non-convex optimisation program. We find that MOSES consistently surpasses the state of the art in our numerical experiments with both synthetic and real-world datasets, while being computationally inexpensive.

Keywords— Principal component analysis, Linear dimensionality reduction, Subspace identification, Streaming algorithms, Non-convex optimisation.

1 INTRODUCTION

Linear models are pervasive in data and computational sciences and, in particular, Principal Component Analysis (PCA) is an indispensable tool for detecting linear structure in collected data [1], [2], [3], [4], [5], [6], [7]. Principal components are the directions that preserve most of the “energy” of a dataset and can be used for linear dimensionality reduction, among other things. In turn, successful dimensionality reduction is at the heart of statistical learning and serves to tackle the “curse of dimensionality” [8].

AE is with the Institute of Electrical Engineering at the École Polytechnique Fédérale de Lausanne. RAH is with the Mathematical Institute at the University of Oxford. AG is with the Department of Computer Science at the University of Cambridge. RAH and AG are also affiliated with the Alan Turing Institute in London. Emails: armin.eftekhari@epfl.ch, hauser@maths.ox.ac.uk, ag926@cl.cam.ac.uk.

In this work, we are interested in both computing the principal components *and* reducing the dimension of data that is presented sequentially to a user. Due to hardware limitations, the user can only store small amounts of data, which in turn would severely limit the available processing time for each incoming data vector.

For example, consider a network of battery-powered and cheap sensors that must relay their measurements to a central node on a daily basis. Each sensor has a small storage and does not have the power to relay all the raw data to the central node. One solution is then for each sensor to reduce the dimension of its data to make transmission to the central node possible. Even if each sensor had unlimited storage, the frequent daily updates scheduled by the central node would force each sensor to reduce the dimension of its data “on the go” before transmitting it to the central node. A number of similar problems are listed in [9].

Motivated by such scenarios, we are interested in developing a *streaming* algorithm for linear dimensionality reduction, namely, an algorithm with minimal storage and computational requirements. As more and more data vectors arrive, this algorithm would keep a running estimate of the principal components of the data *and* project the available data onto this estimate to reduce its dimension. As discussed in Section 2, this is equivalent to designing a streaming algorithm for truncated Singular Value Decomposition (SVD).

Indeed, incremental SVD (iSVD) is a successful streaming algorithm that updates its estimate of the truncated SVD of the data matrix with every new incoming vector [10], [11], [12], [13], [14]. However, to the best of our knowledge and despite its popularity and empirical success, iSVD lacks comprehensive statistical guarantees. In fact, [15] only very recently provided stochastic analysis for two of the variants of iSVD in [16], [17]. More specifically, in [15] the authors studied how well the output of iSVD approximates the

leading principal component of data, in expectation. Crucially, [15] does *not* offer any guarantees for dimensionality reduction; see Section 4 for more details on iSVD and review of the prior art.

Contributions: In this paper, to address the shortcomings of iSVD, we propose Memory-limited Online Subspace Estimation Scheme (MOSES) for streaming dimensionality reduction. MOSES generalises iSVD to update its estimate with every incoming *thin* block of data, rather than with every incoming vector. This small generalisation is in part what enables us to complement MOSES with a comprehensive statistical analysis, thus providing (to the best of our knowledge) the first theoretically-sound variant of iSVD.

Indeed, Theorem 1 below considers the important case where the incoming data vectors are drawn from a zero-mean normal distribution. This stochastic setup is a powerful generalisation of the popular *spiked covariance* model common in statistical signal processing [18]. Theorem 1 states that MOSES nearly matches the performance of “offline” truncated SVD (which has unlimited memory and computing resources), provided that the corresponding covariance matrix is well-conditioned and has a small residual.

Moreover, we concretely interpret MOSES as an approximate solver for the underlying non-convex optimisation program. We also find that MOSES consistently surpasses the state of the art in our numerical experiments with both synthetic and real-world datasets, while being computationally inexpensive.

2 INTRODUCING MOSES

Consider a sequence of vectors $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$, presented to us sequentially, and let

$$\mathbf{Y}_T := \begin{bmatrix} y_1 & y_2 & \cdots & y_T \end{bmatrix} \in \mathbb{R}^{n \times T}, \quad (1)$$

for short. We conveniently assume throughout that \mathbf{Y}_T is *centred*, namely, the entries of each row of \mathbf{Y}_T sum up to zero. For an integer $r \leq \text{rank}(\mathbf{Y}_T)$, let us partition the SVD of \mathbf{Y}_T into two orthogonal components as

$$\begin{aligned} \mathbf{Y}_T &\stackrel{\text{SVD}}{=} \mathbf{S}_T \mathbf{\Gamma}_T \mathbf{Q}_T^* \quad (\mathbf{S}_T \in \text{O}(n), \mathbf{Q}_T \in \text{O}(T)) \\ &= \mathbf{S}_{T,r} \mathbf{\Gamma}_{T,r} \mathbf{Q}_{T,r}^* + \mathbf{S}_{T,r+} \mathbf{\Gamma}_{T,r+} \mathbf{Q}_{T,r+}^* \\ &=: \mathbf{Y}_{T,r} + \mathbf{Y}_{T,r+}, \end{aligned} \quad (2)$$

where $\text{O}(n)$ is the orthogonal group, containing all $n \times n$ matrices with orthonormal columns, and $\mathbf{\Gamma}_T \in \mathbb{R}^{n \times T}$ above contains the singular values in non-increasing order. Moreover, $\mathbf{S}_{T,r} \in \mathbb{R}^{n \times r}$ contains leading r principal components of \mathbf{Y}_T , namely, the singular vectors corresponding to largest r singular values of \mathbf{Y}_T [19], [20].

Given $\mathbf{S}_{T,r}$, we can reduce the dimension of data² from n to r by projecting \mathbf{Y}_T onto the column span of $\mathbf{S}_{T,r}$, that is,

$$\mathbf{S}_{T,r}^* \cdot \mathbf{Y}_T = \mathbf{\Gamma}_{T,r} \mathbf{Q}_{T,r}^* \in \mathbb{R}^{r \times T}. \quad (\text{see (2)}) \quad (3)$$

Above, the projected data matrix $\mathbf{S}_{T,r}^* \mathbf{Y}_T \in \mathbb{R}^{r \times T}$ again has T data vectors (namely, columns) but these vectors are embedded in (often much smaller) \mathbb{R}^r rather than \mathbb{R}^n . Note that $\mathbf{Y}_{T,r}$ in (2) is a rank- r truncation of \mathbf{Y}_T , which we denote with $\mathbf{Y}_{T,r} = \text{SVD}_r(\mathbf{Y}_T)$. That is, $\mathbf{Y}_{T,r}$ is a best rank- r approximation of \mathbf{Y}_T with the corresponding residual

$$\begin{aligned} \|\mathbf{Y}_T - \mathbf{Y}_{T,r}\|_F^2 &= \min_{\text{rank}(\mathbf{X})=r} \|\mathbf{Y}_T - \mathbf{X}\|_F^2 \\ &= \|\mathbf{Y}_{T,r+}\|_F^2 = \sum_{i \geq r+1} \sigma_i^2(\mathbf{Y}_T) \\ &=: \rho_r^2(\mathbf{Y}_T), \end{aligned} \quad (4)$$

where $\sigma_1(\mathbf{Y}_T) \geq \sigma_2(\mathbf{Y}_T) \geq \cdots$ are the singular values of \mathbf{Y}_T . We also observe that

$$\begin{aligned} \mathbf{Y}_{T,r} &= \text{SVD}_r(\mathbf{Y}_T) \\ &= \underbrace{\mathbf{S}_{T,r}}_{\text{PCs}} \cdot \underbrace{\mathbf{\Gamma}_{T,r} \mathbf{Q}_{T,r}^*}_{\text{projected data}}. \quad (\text{see (2,3)}) \end{aligned} \quad (5)$$

That is, rank- r truncation of \mathbf{Y}_T encapsulates both leading r principal components of \mathbf{Y}_T , namely $\mathbf{S}_{T,r}$, and the projected data matrix $\mathbf{S}_{T,r}^* \mathbf{Y}_T = \mathbf{\Gamma}_{T,r} \mathbf{Q}_{T,r}^*$. In other words, computing a rank- r truncation of the data matrix both yields its principal components and reduces the dimension of data at once.

We are in this work interested in developing a streaming algorithm to compute $\mathbf{Y}_{T,r} = \text{SVD}_r(\mathbf{Y}_T)$, namely, a rank- r truncation of the data matrix \mathbf{Y}_T . More specifically, to compute $\mathbf{Y}_{T,r}$, we are only allowed one pass through the columns of \mathbf{Y}_T and have access to a limited amount of storage, namely, $\mathcal{O}(n)$ bits.

For a block size b , our strategy is to iteratively group every b incoming vectors into an $n \times b$ block and then update a rank- r estimate of the data received so far. We assume throughout that $r \leq b \leq T$ and in fact often take the block size as $b = \mathcal{O}(r)$, where \mathcal{O} is the standard Big-O notation. It is convenient to assume that the number of blocks $K := T/b$ is an integer. We call this simple algorithm MOSES for Memory-limited Online Subspace Estimation Scheme, presented in an accessible fashion in Algorithm 1.

The output of MOSES after K iterations is

$$\hat{\mathbf{Y}}_{Kb,r} = \hat{\mathbf{Y}}_{T,r},$$

which contains both an estimate of leading r principal components of \mathbf{Y}_T and also the projection of \mathbf{Y}_T onto this estimate, as discussed below. A computationally efficient implementation of MOSES is given in Algo-

Algorithm 2, which explicitly maintains both the estimates of principal components and the projected data. As also discussed below, the storage and computational requirements of Algorithm 2 are nearly minimal.

Discussion: MOSES maintains a rank- r estimate of the data received so far, and updates its estimate in every iteration to account for the new incoming block of data. More specifically, note that the final output of MOSES, namely $\hat{\mathbf{Y}}_{T,r} \in \mathbb{R}^{n \times T}$, is at most rank- r , and let

$$\hat{\mathbf{Y}}_{T,r} \stackrel{\text{tSVD}}{=} \hat{\mathbf{S}}_{T,r} \hat{\mathbf{\Gamma}}_{T,r} \hat{\mathbf{Q}}_{T,r}^* \quad (\hat{\mathbf{S}}_{T,r} \in \text{St}(n, r), \hat{\mathbf{Q}}_{T,r} \in \text{St}(T, r)) \quad (6)$$

be its thin SVD. Above, $\text{St}(n, r)$ is the Stiefel manifold, containing all $n \times r$ matrices with orthonormal columns, and the diagonal matrix $\hat{\mathbf{\Gamma}}_{T,r} \in \mathbb{R}^{r \times r}$ contains the singular values in nonincreasing order. Then, $\hat{\mathbf{S}}_{T,r} \in \mathbb{R}^{n \times r}$ is MOSES's estimate of leading principal components of the data matrix \mathbf{Y}_T , and

$$\hat{\mathbf{S}}_{T,r}^* \hat{\mathbf{Y}}_{T,r} = \hat{\mathbf{\Gamma}}_{T,r} \hat{\mathbf{Q}}_{T,r} \in \mathbb{R}^{r \times T}$$

is the projection of $\hat{\mathbf{Y}}_{T,r}$ onto this estimate. That is, $\hat{\mathbf{S}}_{T,r}^* \hat{\mathbf{Y}}_{T,r}$ is the MOSES's estimate of the projected data matrix.

Origins: iSVD is a streaming algorithm that updates its estimate of (truncated) SVD of the data matrix with every new incoming vector [10], [11], [12], [13], [14]. Despite its popularity and empirical success, iSVD lacks comprehensive statistical guarantees, as detailed in Section 4.

MOSES generalises iSVD to update its estimate with every incoming block of data, rather than every data vector. As detailed later in Section 4, this minor extension in part enables us to complement MOSES with a comprehensive statistical analysis, summarised in Theorem 1 below. In this sense, MOSES might be interpreted as a variant of iSVD that is both successful in practice and theoretically grounded.

Working with data blocks also allows us to concretely interpret MOSES as an approximate solver for the underlying non-convex program, as detailed in Section 3 of the supplementary material.

Storage and computational requirements: The efficient implementation of MOSES in Algorithm 2 is based on the ideas from iSVD and it is straightforward to verify that Algorithms 1 and 2 are indeed equivalent; at iteration k , the relation between the output of Algorithm 1 ($\hat{\mathbf{Y}}_{kb,r}$) and the output of Algorithm 2 ($\hat{\mathbf{S}}_{kb,r}$, $\hat{\mathbf{\Gamma}}_{kb,r}$, $\hat{\mathbf{Q}}_{kb,r}$) is

$$\hat{\mathbf{Y}}_{kb,r} \stackrel{\text{tSVD}}{=} \hat{\mathbf{S}}_{kb,r} \hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^*$$

More specifically, $\hat{\mathbf{S}}_{kb,r} \in \text{St}(n, r)$ is MOSES's estimate of leading r principal components of $\mathbf{Y}_{kb} \in$

$\mathbb{R}^{n \times kb}$, where we recall that \mathbf{Y}_{kb} is the data received up to iteration k . Moreover,

$$\hat{\mathbf{S}}_{kb,r}^* \hat{\mathbf{Y}}_{kb,r} = \hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^* \in \mathbb{R}^{r \times kb}$$

is the projection of $\hat{\mathbf{Y}}_{kb,r}$ onto this estimate, namely, $\hat{\mathbf{S}}_{kb,r}^* \hat{\mathbf{Y}}_{kb,r}$ is MOSES's estimate of the projected data matrix so far. In words, the efficient implementation of MOSES in Algorithm 2 explicitly maintains estimates of both principal components and the projected data, at every iteration.

As detailed in Section 2 of the supplementary material, Algorithm 2 requires $O(r(n + kr))$ bits of memory at iteration k . This is optimal, as it is impossible to store a rank- r matrix of size $n \times kb$ with fewer bits when $b = O(r)$. On the other hand, Algorithm 2 performs $O(r^2(n + kb)) = O(r^2(n + kr))$ flops in iteration k . The dependence of both storage and computational complexity on k is because MOSES maintains both an estimate of principal components ($\hat{\mathbf{S}}_{kb,r}$) and an estimate of the projected data ($\hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^*$). To maximise the efficiency, one might optionally "flush out" the projected data after every n/b iterations, as described in the last step in Algorithm 2.

Algorithm 1 MOSES: A streaming algorithm for linear dimensionality reduction (accessible version)

Input: Sequence of vectors $\{y_t\}_{t \geq 1} \subset \mathbb{R}^n$, rank r , and block size $b \geq r$.

Output: Sequence $\{\hat{\mathbf{Y}}_{kb,r}\}_{k \geq 1}$, where $\hat{\mathbf{Y}}_{kb,r} \in \mathbb{R}^{n \times kb}$ for every $k \geq 1$.

Body:

- 1) Set $\hat{\mathbf{Y}}_{0,r} \leftarrow \{\}$.
 - 2) For $k \geq 1$, repeat
 - a) Form $\mathbf{y}_k \in \mathbb{R}^{n \times b}$ by concatenating $\{y_t\}_{t=(k-1)b+1}^{kb}$.
 - b) Set $\hat{\mathbf{Y}}_{kb,r} = \text{SVD}_r([\hat{\mathbf{Y}}_{(k-1)b,r}, \mathbf{y}_k])$, where $\text{SVD}_r(\cdot)$ returns a rank- r truncated SVD of its argument.
-

3 PERFORMANCE OF MOSES

In this section, we study the performance of MOSES in a stochastic setup. Consider the probability space $(\mathbb{R}^n, \mathcal{B}, \mu)$, where \mathcal{B} is the Borel σ -algebra and μ is an *unknown* probability measure with zero mean, namely, $\int_{\mathbb{R}^n} y \mu(dy) = 0$. Informally, we are interested in finding an r -dimensional subspace \mathcal{U} that captures most of the mass of μ . That is, with y drawn from this probability space, we are interested in finding an r -

dimensional subspace \mathcal{U} that minimises the *population risk*. To be specific, we wish to solve

$$\begin{aligned} & \min_{\mathcal{U} \in \mathcal{G}(n,r)} \mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\mathcal{U}} y\|_2^2 \\ & = \min_{\mathcal{U} \in \mathcal{G}(n,r)} \int_{\mathbb{R}^n} \|y - \mathbf{P}_{\mathcal{U}} y\|_F^2 \mu(dy) =: \rho_r^2(\mu), \end{aligned} \quad (7)$$

where the Grassmanian $\mathcal{G}(n,r)$ is the set of all r -dimensional subspaces in \mathbb{R}^n , and $\rho_r(\mu)$ denotes the corresponding residual.

Since μ is unknown, we cannot directly solve Program (7), but suppose that instead we have access to the *training samples* $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$, drawn independently from the probability measure μ . Let us form $\mathbf{Y}_T \in \mathbb{R}^{n \times T}$ by concatenating these vectors, similar to (1). In lieu of Program (7), we minimise the *empirical risk*, namely, we solve the optimisation program

$$\begin{aligned} & \min_{\mathcal{U} \in \mathcal{G}(n,r)} \frac{1}{T} \sum_{t=1}^T \|y_t - \mathbf{P}_{\mathcal{U}} y_t\|_2^2 \\ & = \min_{\mathcal{U} \in \mathcal{G}(n,r)} \frac{1}{T} \|\mathbf{Y}_T - \mathbf{P}_{\mathcal{U}} \mathbf{Y}_T\|_F^2. \end{aligned} \quad (\text{see (1)}) \quad (8)$$

Let $\mathcal{S}_{T,r} \in \mathcal{G}(n,r)$ be a minimiser of the above program, with the orthonormal basis $\mathbf{S}_{T,r} \in \text{St}(n,r)$. By the Eckart-Young-Mirsky Theorem [19], [20], $\mathcal{S}_{T,r}$ consists of leading r principal components of \mathbf{Y}_T , namely, it contains leading r left singular vectors of \mathbf{Y}_T . Therefore,

$$\begin{aligned} & \min_{\mathcal{U} \in \mathcal{G}(n,r)} \frac{1}{T} \sum_{t=1}^T \|y_t - \mathbf{P}_{\mathcal{U}} y_t\|_2^2 \\ & = \frac{1}{T} \|\mathbf{Y}_T - \mathbf{P}_{\mathcal{S}_{T,r}} \mathbf{Y}_T\|_F^2 \\ & = \frac{1}{T} \|\mathbf{Y}_T - \mathbf{Y}_{T,r}\|_F^2 \quad (\mathbf{Y}_{T,r} = \text{SVD}_r(\mathbf{Y}_T)) \\ & =: \frac{\rho_r^2(\mathbf{Y}_T)}{T}. \end{aligned} \quad (\text{see (4)}) \quad (9)$$

Given its principal components, we can then reduce the dimension of the data matrix $\mathbf{Y}_T \in \mathbb{R}^{n \times T}$ from n to r by computing $\mathbf{S}_{T,r}^* \mathbf{Y}_T \in \mathbb{R}^{r \times T}$. Note also that the subspace $\mathcal{S}_{T,r}$ is a possibly sub-optimal choice in Program (7), namely,

$$\mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 \geq \rho_r^2(\mu). \quad (\text{see (7)}) \quad (10)$$

But one would hope that $\mathcal{S}_{T,r}$ still nearly minimises Program (7), in the sense that

$$\mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 \approx \rho_r^2(\mu), \quad (11)$$

with high probability over the choice of training data $\{y_t\}_{t=1}^T$. That is, one would hope that the *generalisation error* of Program (9) is small. Above, $\mathbb{E}_{y \sim \mu}$ stands for expectation over y , so that the left-hand side of (11) is still a random variable because of its dependence on

$\mathcal{S}_{T,r}$ and, in turn, on the training data.

If the training data $\{y_t\}_{t=1}^T$ is presented to us sequentially and little storage is available, we cannot hope to directly solve Program (9). In this streaming scenario, we may apply MOSES to obtain the (rank- r) output $\hat{\mathbf{Y}}_{T,r}$. We then set

$$\hat{\mathcal{S}}_{T,r} = \text{span}(\hat{\mathbf{Y}}_{T,r}), \quad (12)$$

with orthonormal basis $\hat{\mathbf{S}}_{T,r} \in \text{St}(n,r)$. Note that $\hat{\mathbf{S}}_{T,r}$ is MOSES' estimate of leading r principal components of the data matrix \mathbf{Y}_T and is possibly suboptimal in the sense that

$$\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F \geq \rho_r(\mathbf{Y}_T). \quad (\text{see (9)}) \quad (13)$$

Again, we would still hope that the output $\hat{\mathbf{Y}}_{T,r}$ of MOSES is a nearly optimal choice in Program (9), in the sense that

$$\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F \approx \rho_r(\mathbf{Y}_T), \quad (14)$$

with high probability over the choice of $\{y_t\}_{t=1}^T$. Moreover, similar to (11), $\hat{\mathcal{S}}_{T,r}$ is again a possibly sub-optimal choice for Program (7), and yet we hope that

$$\mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} y\|_2^2 \approx \rho_r^2(\mu), \quad (15)$$

with high probability over the choice of $\{y_t\}_{t=1}^T$.

To summarise, the key questions are whether (11,14,15) hold. We answer these questions for the important case where μ is a zero-mean Gaussian probability measure with covariance matrix $\Xi \in \mathbb{R}^{n \times n}$. For this choice of μ in (7), it is not difficult to verify that

$$\rho_r^2(\mu) = \sum_{i=r+1}^n \lambda_i(\Xi), \quad (16)$$

where $\lambda_1(\Xi) \geq \lambda_2(\Xi) \geq \dots$ are the eigenvalues of the covariance matrix Ξ . From now on, we use

$$\rho_r = \rho_r(\mu), \quad \lambda_i = \lambda_i(\Xi), \quad i \in [1 : n].$$

For our choice of μ above, one can use standard tools to show that (11) holds when T is sufficiently large, see Section B of the supplementary material [21], [22], [23].

Proposition 1. Suppose that $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$ are drawn independently from a zero-mean Gaussian distribution with the covariance matrix $\Xi \in \mathbb{R}^{n \times n}$ and form $\mathbf{Y}_T \in \mathbb{R}^{n \times T}$ by concatenating these vectors, as in (1). Suppose also that $\mathcal{S}_{T,r} \in \mathcal{G}(n,r)$ is the span of leading r principal components of \mathbf{Y}_T . For $1 \leq \alpha \leq \sqrt{T/\log T}$, it then holds that

$$\frac{\rho_r^2(\mathbf{Y}_T)}{T} \lesssim \alpha \rho_r^2, \quad (17)$$

$$\mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 \lesssim \alpha \rho_r^2 + \alpha(n-r) \lambda_1 \sqrt{\frac{\log T}{T}}, \quad (18)$$

except with a probability of at most $T^{-C\alpha^2}$, see (4). Here, C is a universal constant, the value of which may change in every appearance. Above, \lesssim suppresses some of the universal constants for a more tidy presentation.

In words, (18) states that the generalisation error of Program (9) is small, namely, offline truncated SVD successfully estimates the unknown subspace from training data. Indeed, (11) holds when $\alpha = O(1)$ and T is sufficiently large. As the dimension r of the subspace fit to the data approaches the ambient dimension n , the right-hand of (18) vanishes, see (7).

In the streaming setup, Theorem 1 below states that MOSES approximately solves Program (9), namely, MOSES approximately estimates leading principal components of \mathbf{Y}_T and reduces the dimension of data from n to r with only $O(r(n+T))$ bits of memory, rather than $O(nT)$ bits required for solving Program (9) with offline truncated SVD. Moreover, MOSES approximately solves Program (7). In other words, MOSES satisfies both (14,15). These statements are made concrete below and proved in Section C of the supplementary material.

Theorem 1. (Performance of MOSES) Suppose that $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$ are drawn independently from a zero-mean Gaussian distribution with the covariance matrix $\Xi \in \mathbb{R}^{n \times n}$ and form \mathbf{Y}_T by concatenating these vectors, as in (1). Let us define

$$\kappa_r^2 := \frac{\lambda_1}{\lambda_r}, \quad \rho_r^2 = \sum_{i=r+1}^n \lambda_i, \quad \eta_r := \kappa_r + \sqrt{\frac{2\alpha\rho_r^2}{p^{\frac{1}{3}}\lambda_r}}, \quad (19)$$

where $\lambda_1 \geq \lambda_2 \geq \dots$ are the eigenvalues of Ξ . Let $\hat{\mathcal{S}}_{T,r} = \text{span}(\hat{\mathbf{Y}}_{T,r})$ be the span of the output of MOSES, as in (12). Then, for tuning parameters $1 \leq \alpha \leq \sqrt{T/\log T}$ and $p > 1$, the output $\hat{\mathbf{Y}}_{T,r}$ of MOSES satisfies

$$\frac{\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F^2}{T} \lesssim \frac{\alpha p^{\frac{1}{3}} 4^{p\eta_r^2}}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2 \rho_r^2, r\lambda_1 + \rho_r^2) \cdot \left(\frac{T}{p\eta_r^2 b}\right)^{p\eta_r^2 - 1}, \quad (20)$$

explained in words after this theorem. Moreover, the output subspace $\hat{\mathcal{S}}_{T,r}$ of MOSES satisfies

$$\begin{aligned} & \mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} y\|_2^2 \\ & \lesssim \frac{\alpha p^{\frac{1}{3}} 4^{p\eta_r^2}}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2 \rho_r^2, r\lambda_1 + \rho_r^2) \left(\frac{T}{p\eta_r^2 b}\right)^{p\eta_r^2 - 1} \\ & + \alpha(n-r)\lambda_1 \sqrt{\frac{\log T}{T}}, \end{aligned} \quad (21)$$

explained after this theorem. Both (20,21) hold except with a probability of at most $T^{-C\alpha^2} + e^{-C\alpha r}$ and provided that

$$b \geq \frac{\alpha p^{\frac{1}{3}} r}{(p^{\frac{1}{3}} - 1)^2}, \quad b \geq C\alpha r, \quad T \geq p\eta_r^2 b. \quad (22)$$

The requirement $T \geq p\eta_r^2 b$ above is only for cleaner bounds. A general expression for arbitrary T

is given in the proof.

Discussion of Theorem 1: On the one hand, (20) states that MOSES successfully reduces the dimension of streaming data, namely, (14) holds under certain conditions: Loosely-speaking, (20) states that $\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F^2$ scales with $\rho_r^2 T^{p\eta_r^2} / b^{p\eta_r^2 - 1}$, where $\hat{\mathbf{Y}}_{T,r}$ is the output of MOSES. In contrast with (17), we have

$$\begin{aligned} \|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F^2 & \propto \left(\frac{T}{b}\right)^{p\eta_r^2 - 1} \rho_r^2(\mathbf{Y}_T) \\ & = \left(\frac{T}{b}\right)^{p\eta_r^2 - 1} \|\mathbf{Y}_T - \mathbf{Y}_{T,r}\|_F^2, \quad (\text{see (9)}) \end{aligned} \quad (23)$$

after ignoring the less important terms. In words, applying (offline) truncated SVD to \mathbf{Y}_T outperforms the (streaming) MOSES by a polynomial factor in T/b .

This polynomial factor can be negligible when the covariance matrix Ξ of the Gaussian data distribution is well-conditioned ($\kappa_r = O(1)$) and has a small residual ($\rho_r^2 = O(\lambda_r)$), in which case we will have $\eta_r = O(1)$, see (19). Also setting $p = O(1)$, (23) then reads as

$$\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F^2 \propto \left(\frac{T}{b}\right)^{O(1)} \rho_r^2(\mathbf{Y}_T), \quad (24)$$

namely, (streaming) MOSES is comparable to offline truncated SVD. In particular, when $\text{rank}(\Xi) \leq r$, we have by (16) that $\rho_r = 0$. Consequently, (23) reads as $\hat{\mathbf{Y}}_{T,r} = \mathbf{Y}_{T,r} = \mathbf{Y}_T$. That is, the outputs of offline truncated SVD and MOSES coincide in this case.

The dependence in Theorem 1 on the condition number κ_r and the residual ρ_r is very likely *not* an artifact of the proof techniques; see (19). Indeed, when $\kappa_r \gg 1$, certain directions are less often observed in the incoming data vectors $\{y_t\}_{t=1}^T$, which tilts the estimate of MOSES towards the dominant principal components corresponding to the very large singular values. Moreover, if $\rho_r \gg 1$, there are too many significant principal components, while MOSES can at most “remember” r of them from its previous iteration. In this scenario, fitting an r -dimensional subspace to data is not a good idea in the first place because even the residual $\rho_r(\mathbf{Y}_T)$ of the offline truncated SVD will be large, and we should perhaps increase the dimension r of the subspace fitted to the incoming data.

As the block size b increases, the performance of MOSES approaches that of the offline truncated SVD. In particular, when $b = T$, MOSES reduces to offline truncated SVD, processing all of the data at once. This trend is somewhat imperfectly reflected in (20).

On the other hand, Theorem 1 and specifically (21) state that MOSES approximately estimates the (unknown) underlying subspace, namely, (15) holds under certain conditions. Indeed, for sufficiently large

T , (21) loosely-speaking reads as

$$\begin{aligned} \mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\hat{\mathbf{S}}_{T,r}} y\|_2^2 &\propto \left(\frac{T}{b}\right)^{p\eta_r^2-1} \rho_r^2 \\ &= \left(\frac{T}{b}\right)^{p\eta_r^2-1} \min_{\mathcal{U} \in \mathcal{G}(n,r)} \mathbb{E}_{y \sim \mu} \|y - \mathbf{P}_{\mathcal{U}} y\|_2^2, \end{aligned} \quad (25)$$

see Program (7). That is, the output of MOSES is sub-optimal for Program (7) by a polynomial factor in T , which is negligible if the covariance matrix Ξ of the data distribution μ is well-conditioned and has a small residual, as discussed earlier.

As the closing remark, Section 3 of the supplementary material applies Theorem 1 to the popular spiked covariance model as a special case, outlines the proof technique, and discusses extension to non-Gaussian stochastic models.

4 PRIOR ART

In this paper, we presented MOSES for streaming (linear) dimensionality reduction, an algorithm with (nearly) minimal storage and computational requirements. We can think of MOSES as an online “subspace tracking” algorithm that identifies the linear structure of data as it arrives. Once the data has fully arrived, both principal components and the projected data are already made available by MOSES, ready for any additional processing.

The well-known iSVD is a special case of MOSES. At iteration t and given the (truncated) SVD of \mathbf{Y}_{t-1} , iSVD aims to compute the (truncated) SVD of $\mathbf{Y}_t = [\mathbf{Y}_{t-1} \ y_t] \in \mathbb{R}^{n \times t}$, where $y_t \in \mathbb{R}^n$ is the newly arrived data vector and \mathbf{Y}_{t-1} is the matrix formed by concatenating the previous data vectors [10], [11], [12], [13], [14]. MOSES generalises iSVD to handle data blocks; see Algorithm 1. This minor generalisation in part enables us to complement MOSES (and iSVD) with comprehensive statistical analysis in Theorem 1, which has been lacking despite the popularity and empirical success of iSVD.

In fact, [15] only very recently provided stochastic analysis for two of the variants of iSVD in [16], [17]. The results in [15] hold in expectation and for the special case of $r = 1$, the first leading principal component. Crucially, these results measure the angle $\angle[\mathbf{S}_{T,r}, \hat{\mathbf{S}}_{T,r}]$ between the true leading principal components of the data matrix and those estimated by iSVD.

Such results are therefore inconclusive because they are silent about the dimensionality reduction task of iSVD. Indeed, iSVD can estimate both left and right leading singular vectors of the data matrix, namely, iSVD can estimate both the leading principal components of the data matrix $\hat{\mathbf{S}}_{T,r}$ and reduce the dimension of data by computing $\hat{\mathbf{S}}_{T,r}^* \hat{\mathbf{Y}}_{T,r} \in \mathbb{R}^{r \times T}$,

where $\hat{\mathbf{S}}_{T,r}$ and $\hat{\mathbf{Y}}_{T,r}$ are the final outputs of iSVD.⁶ Unlike [15], Theorem 1 and specifically (20) assesses the quality of both of these tasks and establishes that, under certain conditions, MOSES performs nearly as well as the offline truncated SVD.

GROUSE [24] is another algorithm for streaming PCA, for data with possibly missing entries. GROUSE can be interpreted as projected stochastic gradient descent on the Grassmannian manifold. GROUSE is effectively identical to iSVD when the incoming data is low-rank [24]. In [25] and on the basis of [15], the authors offer theoretical guarantees for GROUSE that again does not account for the quality of dimensionality reduction. Their results hold without any missing data, in expectation, and in a setup similar to the spiked covariance model. An alternative to GROUSE is SNIPE that has stronger theoretical guarantees in the case of missing data [26], [27]. In Section 5, we will numerically compare MOSES with GROUSE.

The closely-related method of Frequent Directions (FD) replaces the hard thresholding of the singular values in iSVD with soft thresholding [28], [29]. Later, robust FD [30] improved the performance of FD and addressed some of its numerical issues. On the algorithmic side, FD keeps an estimate of $\mathbf{S}_{t,r} \mathbf{\Gamma}_{t,r}$, whereas MOSES also calculates the projected data, namely, it keeps an estimate of $\{\mathbf{S}_{t,r}, \mathbf{\Gamma}_{t,r}, \mathbf{Q}_{t,r}\}_t$; see Algorithm 2. In terms of guarantees, the available guarantees for FD control the error incurred in estimating the principal components, whereas Theorem 1 for MOSES also controls the error incurred in the dimensionality reduction step. Indeed,

$$\begin{aligned} &\left\| \mathbf{S}_{T,r} \mathbf{\Gamma}_{T,r}^2 \mathbf{S}_{T,r}^* - \hat{\mathbf{S}}_{T,r} \hat{\mathbf{\Gamma}}_{T,r}^2 \hat{\mathbf{S}}_{T,r}^* \right\|_F \\ &\leq \|\mathbf{Y}_{T,r} - \hat{\mathbf{Y}}_{T,r}\|_F \cdot \left(\|\mathbf{Y}_{T,r}\| + \|\hat{\mathbf{Y}}_{T,r}\| \right), \end{aligned} \quad (26)$$

namely, Theorem 1 is a more general result than those exemplified by Theorem 1.1 in [28] which, except for symmetric matrices, do not convey any information about the row-span. Another key difference is that Theorem 1 is a stochastic result versus the deterministic Theorem 1.1 in [28] and similar results. Indeed, an intermediate step to prove Theorem 1 is the deterministic Lemma 2 in the supplementary material. An important feature of this work is to translate Lemma 2 into a stochastic result in learning theory, of interest to the machine learning and statistical signal processing communities. In Section 5, we will numerically compare MOSES with FD.

As detailed in Section 4 of the supplementary material, MOSES can be adapted to the dynamic case, where the distribution of data changes over time. This is achieved by using a “forgetting factor” in Step b of Algorithm 1. Such an extension is crucial, as there are pathological examples where (static) MOSES and iSVD both fail to follow the changes in the distribution

of data [29]. This important research direction is left for future work.

One might also view MOSES as a stochastic algorithm for PCA. Indeed, note that Program (7) is equivalent to

$$\begin{aligned} \begin{cases} \max_{y \sim \mu} \mathbb{E} \|UU^*y\|_F^2 \\ U^*U = I_r \end{cases} &= \begin{cases} \max_{y \sim \mu} \mathbb{E} \langle UU^*, yy^* \rangle \\ U^*U = I_r \end{cases} \\ &= \begin{cases} \max_{y \sim \mu} \mathbb{E} \langle UU^*, yy^* \rangle \\ U^*U \preceq I_r, \end{cases} \end{aligned} \quad (27)$$

where the maximisation is over the matrix $U \in \mathbb{R}^{n \times r}$. Above, $U^*U \preceq I_r$ is the unit ball with respect to the spectral norm and $A \preceq B$ means that $B - A$ is a positive semi-definite matrix. The last identity above holds because a convex function is always maximised on the boundary of the feasible set. With the Schur complement, we can equivalently write the last program above as

$$\begin{cases} \max_{y \sim \mu} \mathbb{E} \langle UU^*, yy^* \rangle \\ \begin{bmatrix} I_n & U \\ U^* & I_r \end{bmatrix} \succcurlyeq 0 \end{cases} = \begin{cases} \max \langle UU^*, \Xi \rangle \\ \begin{bmatrix} I_n & U \\ U^* & I_r \end{bmatrix} \succcurlyeq 0, \end{cases} \quad (28)$$

where $\Xi = \mathbb{E}[yy^*] \in \mathbb{R}^{n \times n}$ is the covariance matrix of the data distribution μ . Program (28) has a convex (in fact, quadratic) objective function that is *maximised* on a convex (conic) feasible set. We cannot hope to directly compute the gradient of the objective function above (namely, $2\Xi U$) because the distribution of y and hence its covariance matrix Ξ are unknown. Given an iterate \hat{S}_t , one might instead draw a random vector y_{t+1} from the probability measure μ and move along the direction $2y_{t+1}y_{t+1}^*\hat{S}_t$, motivated by the observation that $\mathbb{E}[2y_{t+1}y_{t+1}^*\hat{S}_t] = 2\Xi\hat{S}_t$. This is then followed by back projection onto the feasible set of Program (27). That is,

$$\hat{S}_{t+1} = \mathcal{P} \left(S_t + 2\alpha_{t+1}y_{t+1}y_{t+1}^*\hat{S}_t \right), \quad (29)$$

for an appropriate step size α_{t+1} . Above, $\mathcal{P}(A)$ projects onto the unit spectral norm ball by setting to one all singular values of A that exceed one.

The stochastic projected gradient ascent for PCA, described above, is itself closely related to the so-called *power method* and is at the heart of [31], [32], [33], [34], [35], all lacking a statistical analysis similar to Theorem 1. One notable exception is the power method in [31] which in a sense applies *mini-batch* stochastic projected gradient ascent to solve Program (28), with data blocks (namely, batches) of size $b = \Omega(n)$. There the authors offer statistical guarantees for the spiked covariance model, defined in Section d of the

supplementary material. As before, these guarantees are for the quality of estimated principal components and silent about the quality of projected data, which is addressed in Theorem 1. Note also that, especially when the data dimension n is large, one disadvantage of this approach is its large block size; it takes a long time (namely, $\Omega(n)$ iterations) for the algorithm to update its estimate of the principal components, a big disadvantage in the dynamic case. In this setup, we may think of MOSES as a stochastic algorithm for PCA based on alternative minimisation rather than gradient ascent, as detailed in Section 3 of the supplementary material. Moreover, MOSES updates its estimate frequently, after receiving every $b = O(r)$ data vectors, and also maintains the projected data. In Section 5, we numerically compare MOSES with the power method in [31]. A few closely related works are [36], [37], [38], [37].

In the context of online learning and *regret minimisation*, [39], [35] offer two algorithms, the former of which is not memory optimal and the latter does not have guarantees similar to Theorem 1; see also [40]. A Bayesian approach to PCA is studied in [41], [42] and the *expectation maximisation* algorithm therein could be implemented in an online fashion but without theoretical guarantees.

More generally, MOSES might be interpreted as a deterministic *matrix sketching* algorithm. Common sketching algorithms either randomly sparsify a matrix, randomly combine its rows (columns), or randomly subsample its rows (columns) according to its *leverage scores* [43], [44], [45], [46], [28], [47]. In particular, FD was observed to outperform random sketching algorithms in practice [28]. The relation between streaming algorithms and distributed computing is also perhaps worth pointing out; see Figure 6 of the supplementary material and [48], [49]. Lastly, when the data vectors have missing entries, a closely related problem is low-rank matrix completion [50], [51], [52], [53].

5 EXPERIMENTS

In this section, we investigate the numerical performance of MOSES and compare it against competing algorithms, namely, GROUSE [24], the method of frequent directions (FD) [29], [30], and the power method (PM) [54], all detailed in Section 4. On both synthetic and real-world datasets, we reveal one by one the data vectors $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$ and, for every t , wish to compute a rank- r truncated SVD of $[y_1, \dots, y_t]$, the data arrived so far.

For the tests on synthetic datasets, the vectors $\{y_t\}_{t=1}^T$ are drawn independently from a zero-mean Gaussian distribution with the covariance matrix $\Xi = S\Lambda S^*$, where $S \in O(n)$ is a generic orthonormal

basis obtained by orthogonalising a standard random Gaussian matrix. The entries of the diagonal matrix $\Lambda \in \mathbb{R}^{n \times n}$ (the eigenvalues of the covariance matrix Ξ) are selected according to the power law, namely, $\lambda_i = i^{-\alpha}$, for a positive α . To be more succinct, where possible we will use MATLAB's notation for specifying the value ranges in this section.

To assess the performance of MOSES, let $\mathbf{Y}_t = [y_1, \dots, y_t] \in \mathbb{R}^{n \times t}$ be the data received by time t and let $\hat{\mathbf{Y}}_{t,r}^m$ be the output of MOSES at time t .¹ Then the error incurred by MOSES is

$$\frac{1}{t} \|\mathbf{Y}_t - \hat{\mathbf{Y}}_{t,r}^m\|_F^2, \quad (30)$$

see Theorem 1. Recall from (4) that the above error is always larger than the residual of \mathbf{Y}_t , namely

$$\|\mathbf{Y}_t - \hat{\mathbf{Y}}_{t,r}^m\|_F^2 \geq \|\mathbf{Y}_t - \mathbf{Y}_{t,r}\|_F^2 = \rho_r^2(\mathbf{Y}_t), \quad (31)$$

see (4). Above, $\mathbf{Y}_{t,r} = \text{SVD}_r(\mathbf{Y}_t)$ is a rank- r truncated SVD of \mathbf{Y}_t and $\rho_r^2(\mathbf{Y}_t)$ is the corresponding residual.

Later in this section, we compare MOSES against GROUSE [24], FD [29], PM [31], described in Section 4. In contrast with MOSES, these algorithms only estimate the principal components of the data. More specifically, let $\hat{\mathcal{S}}_{t,r}^g \in \mathcal{G}(n, r)$ be the span of the output of GROUSE, with the outputs of the other algorithms defined similarly. These algorithms then incur the errors

$$\begin{aligned} \frac{1}{t} \|\mathbf{Y}_t - \mathbf{P}_{\hat{\mathcal{S}}_{t,r}^g} \mathbf{Y}_t\|_F^2, \quad \frac{1}{t} \|\mathbf{Y}_t - \mathbf{P}_{\hat{\mathcal{S}}_{t,r}^f} \mathbf{Y}_t\|_F^2, \\ \frac{1}{t} \|\mathbf{Y}_t - \mathbf{P}_{\hat{\mathcal{S}}_{t,r}^p} \mathbf{Y}_t\|_F^2, \end{aligned} \quad (32)$$

respectively. Above, $\mathbf{P}_{\mathcal{A}} \in \mathbb{R}^{n \times n}$ is the orthogonal projection onto the subspace \mathcal{A} . Even though robust FD [30] improves over FD in the quality of matrix sketching, since the subspaces produced by FD and robust FD coincide, there is no need here for computing a separate error for robust FD. We now set out to do various tests and report the results. To ensure the reproducibility of our results, both the accompanying MATLAB code and the datasets used are publicly available.²

Ambient dimension: On a synthetic dataset with $\alpha = 1$ and $T = 2000$, we first test MOSES by varying the ambient dimension as $n \in \{200 : 200 : 1200\}$, and setting the rank and block size to $r = 15$, $b = 2r = 30$. The average error over ten trials is reported in Figure 1a. Note that the error is increasing

in n , which indeed agrees with Theorem 1, as detailed³ in the supplementary material under the discussion of the spiked covariance model in Section d therein.

Block size: On a synthetic dataset with $\alpha = 1$ and $T = 2000$, we test MOSES by setting the ambient dimension and rank to $n = 1200$, $r = 15$, and varying the block size as $b \in \{r : r : 15r\}$. The average error over ten trials is reported in Figure 1b. Note that MOSES is robust against the choice of the block size and that, at the extreme case of $b = T$, error vanishes and MOSES reduces to offline truncated SVD. This is predicted by Theorem 1, as seen in (23).

Rank: On a synthetic dataset with $\alpha = 1$ and $T = 2000$, we test MOSES by setting the ambient dimension and block size to $n = 1200$, $b = 2r$, and varying the rank as $r \in \{5 : 5 : 25\}$. The average error over ten trials is reported in Figure 1c. As expected, the error is decreasing in the dimension r of the subspace that we fit to the data and in fact, at the extreme case of $r = n$, there would be no error at all. This observation is corroborated with Theorem 1 and, in particular, (23).

Comparisons on synthetic datasets: On synthetic datasets with $\alpha \in \{0.01, 0.1, 0.5, 1\}$ and $T = 2000$, we compare MOSES against GROUSE, FD, and PM.³ More specifically, we set the ambient dimension to $n = 200$ and the rank to $r = 10$. For MOSES, the block size was set to $b = 2r$. For GROUSE, we set the step size to 2. For FD and PM, the block size was set to $2r = 20$ and $2n = 400$, respectively, as these values seemed to produced the best results overall for these algorithms. Both GROUSE and PM were initialised randomly, as prescribed in [24], [54], while FD does not require any initialisation. The average errors of all three algorithms over ten trials versus time is shown in Figure 3.

Because of its large blocks size of $O(n)$ [54], PM updates its estimate of the principal components much slower than MOSES, but the two algorithms converge to similar errors. The slow updates of PM is a major problem in a dynamic scenario, where the distribution of data changes over time. We will also see later that MOSES is much faster than PM and performs better on the real datasets we tested.

We next evaluate all these four algorithms on publicly-available sensor network data; we use four different datasets that contain *mote* (sensor node) voltage, humidity, light, and temperature measurements over time [55].

Mote voltage dataset: The first dataset we evaluate has an ambient dimension of $n = 46$ and has $T = 7712$ columns. With $r = 20$ and the rest of the parameters as described in the synthetic comparison above, the

1. Note that MOSES updates its estimate after receiving each block of data, namely after every b data vectors. For the sake of an easier comparison with other algorithms (with different block sizes), we properly “interpolate” the outputs of all algorithms over time.

2. github.com/andylamp/moses

3. The MATLAB code for GROUSE is publicly available at web.eecs.umich.edu/~girasole/grouse. The Python code for FD is available at github.com/edoliberty/frequent-directions/

errors over time for all algorithms is shown in Figure 4a in logarithmic scale. MOSES here outperforms GROUSE, FD, and PM.

Mote humidity dataset: The second dataset evaluated has an ambient dimension of $n = 48$ and has $T = 7712$ columns. This dataset contains the humidity measurements of motes. With $r = 20$ and the rest of the parameters as described in the synthetic comparison above, the errors over time for all algorithms is shown in Figure 4b in logarithmic scale. MOSES again outperforms the other algorithms.

Mote light dataset: The third dataset has an ambient dimension $n = 48$ and has $T = 7712$ columns. This dataset contains the light measurements of the motes. With $r = 20$ and the rest of the parameters as described in the synthetic comparison above, the errors over time for all algorithms is shown in Figure 4c in logarithmic scale. As before, MOSES outperforms the competing algorithms.

Mote temperature dataset: The last real dataset we consider in this instance has an ambient dimension of $n = 56$ and has $T = 7712$ columns. This dataset contains the temperature measurements of the sensor motes and has mostly periodic value changes and infrequent spikes. With $r = 20$ and the rest of the parameters as described in the synthetic comparison above, the errors over time for all algorithms is shown in Figure 4d in logarithmic scale. It is evident that MOSES outperforms the other competing algorithms.

Storage: We also performed memory complexity tests for all mote datasets above. Our experiments showed that often MOSES required the least amount of memory allocation against competing methods. Specifically, MOSES required 189.36 Kb, 92.25Kb, 127.02Kb, and 215.97Kb for the voltage, humidity, light, and temperature datasets, respectively. PM required 572.06Kb, 638.80Kb, 548.63Kb, and 682.59Kb, GROUSE required 2896.87Kb, 2896.45Kb, 2897.45Kb, and 3769.42Kb, and finally FD required 173.86Kb, 281.82Kb, 194.32 Kb, 655.78Kb for the same datasets.

Complexity on synthetic datasets: Let us now turn our attention to the computational efficiency of these four algorithms. On synthetic datasets with $\alpha = 1$ and $T = 10000$, we compare the run-time of MOSES against GROUSE, FD, and PM, with the parameters set as described in the synthetic tests earlier. This simulation was carried out using MATLAB 2018b on a 2012 Mac Pro configured with Dual 6-core Intel Xeon X5690 with 64GB of DDR3 ECC RAM. The average run-time of all three algorithms over five trials and for various choices of rank r is shown in Figure 2. We note that the computational cost of MOSES remains consistently small throughout these simulations, especially for large ambient dimensions and ranks where GROUSE and PM perform poorly regardless of the desired recovery rank r used; see Figure 2c. Interest-

ingly enough, FD performs poorly when attempting a relatively low rank recovery ($r \in \{1, 10\}$) and closely matches MOSES as r increases, which can be attributed to the buffering size of FD.

REFERENCES

- [1] P. van Overschee and B. L. de Moor. *Subspace identification for linear systems: Theory, implementation, applications*. Springer US, 2012.
- [2] B. A. Ardekani, J. Kershaw, K. Kashikura, and I. Kanno. Activation detection in functional MRI using subspace modeling and maximum likelihood estimation. *IEEE Transactions on Medical Imaging*, 18(2):101–114, 1999.
- [3] H. Krim and M. Viberg. Two decades of array signal processing research: The parametric approach. *IEEE Signal processing magazine*, 13(4):67–94, 1996.
- [4] L. Tong and S. Perreau. Multichannel blind identification: From subspace to maximum likelihood methods. *Proceedings of IEEE*, 86:1951–1968, 1998.
- [5] R. Vidal, Y. Ma, and S. Sastry. *Generalized Principal Component Analysis*. Interdisciplinary Applied Mathematics. Springer New York, 2016.
- [6] Raphael H Hauser and Armin Eftekhar. Pca by optimisation of symmetric functions has no spurious local optima. *arXiv preprint arXiv:1805.07459*, 2018.
- [7] Raphael A Hauser, Armin Eftekhar, and Heinrich F Matzinger. Pca by determinant optimization has no spurious local optima. *arXiv preprint arXiv:1803.04049*, 2018.
- [8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.
- [9] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 704–711. IEEE, 2010.
- [10] James R Bunch, Christopher P Nielsen, and Danny C Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978.
- [11] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.
- [12] Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. *ECCV 2002*, pages 707–720, 2002.
- [13] Pierre Comon and Gene H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.
- [14] Yongmin Li. On incremental and robust subspace learning. *Pattern recognition*, 37(7):1509–1518, 2004.
- [15] A. Balsubramani, S. Dasgupta, and Y. Freund. The fast convergence of incremental pca. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.
- [16] TP Krasulina. The method of stochastic approximation for the determination of the least eigenvalue of a symmetrical matrix. *USSR Computational Mathematics and Mathematical Physics*, 9(6):189–195, 1969.
- [17] E. Oja. *Subspace methods of pattern recognition*. Electronic & electrical engineering research studies. Research Studies Press, 1983.
- [18] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.
- [19] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.
- [20] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford*, pages 1156–1159, 1966.
- [21] Armin Eftekhar, Ping Li, Michael B Wakin, and Rachel A Ward. Learning the differential correlation matrix of a smooth function from point samples. *arXiv preprint arXiv:1612.06339*, 2016.

- [22] Raphael Hauser, Raul Kangro, Jüri Lember, and Heinrich Matzinger. Quantifying the estimation error of principal components. *arXiv preprint arXiv:1710.10124*, 2017.
- [23] Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012.
- [24] L. Balzano and S. J. Wright. On GROUSE and incremental SVD. In *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–4. IEEE, 2013.
- [25] Dejiao Zhang and Laura Balzano. Global convergence of a grassmannian gradient descent algorithm for subspace estimation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, page 1460. $\mu\epsilon$ 1468, 2016.
- [26] Armin Eftekhari, Gregory Ongie, Laura Balzano, and Michael B Wakin. Streaming principal component analysis from incomplete data. *arXiv preprint arXiv:1612.00904*, 2018.
- [27] A. Eftekhari, L. Balzano, and M. B. Wakin. What to expect when you are expecting on the Grassmannian. *arXiv preprint arXiv:1611.07216*, 2016.
- [28] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.
- [29] Amey Desai, Mina Ghashami, and Jeff M Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.
- [30] Luo Luo, Cheng Chen, Zhihua Zhang, Wu-Jun Li, and Tong Zhang. Robust frequent directions with application in online learning. *arXiv preprint arXiv:1705.05067*, 2017.
- [31] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming pca. In *Advances in Neural Information Processing Systems*, pages 2886–2894, 2013.
- [32] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.
- [33] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.
- [34] Kwang In Kim, Matthias O Franz, and Bernhard Scholkopf. Iterative kernel principal component analysis for image modeling. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1351–1366, 2005.
- [35] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton)*, 2012 50th Annual Allerton Conference on, pages 861–868. IEEE, 2012.
- [36] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems*, pages 2861–2869, 2014.
- [37] Christopher De Sa, Kunle Olukotun, and Christopher Ré. Global convergence of stochastic gradient descent for some non-convex matrix problems. *arXiv preprint arXiv:1411.1134*, 2014.
- [38] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja’s algorithm. In *Conference on Learning Theory*, pages 1147–1164, 2016.
- [39] Manfred K Warmuth and Dima Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(Oct):2287–2320, 2008.
- [40] Christos Boutsidis, Dan Garber, Zohar Karnin, and Edo Liberty. Online principal components analysis. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 887–901. Society for Industrial and Applied Mathematics, 2015.
- [41] Sam T Roweis. Em algorithms for pca and spca. In *Advances in neural information processing systems*, pages 626–632, 1998.
- [42] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
- [43] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. In *Advances in Neural Information Processing Systems*, pages 1225–1234, 2017.
- [44] Jiawei Chiu and Laurent Demanet. Sublinear randomized algorithms for skeleton decompositions. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1361–1383, 2013.
- [45] Farhad Pourkamali-Anaraki and Stephen Becker. Randomized clustered nystrom for large-scale kernel machines. *arXiv preprint arXiv:1612.06470*, 2016.
- [46] Alex Gittens and Michael W Mahoney. Revisiting the nystrom method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016.
- [47] Anna C Gilbert, Jae Young Park, and Michael B Wakin. Sketched svd: Recovering spectral features from compressive measurements. *arXiv preprint arXiv:1211.0361*, 2012.
- [48] MA Iwen and BW Ong. A distributed and incremental svd algorithm for agglomerative data analysis on large networks. *SIAM Journal on Matrix Analysis and Applications*, 37(4):1699–1718, 2016.
- [49] Ahmed Sameh, Bernard Philippe, Dani Mezher, and Michael W Berry. Parallel algorithms for the singular value decomposition. In *Handbook of parallel computing and statistics*, pages 133–180. Chapman and Hall/CRC, 2005.
- [50] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.
- [51] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.
- [52] Armin Eftekhari, Dehui Yang, and Michael B Wakin. Weighted matrix completion and recovery with prior subspace information. *IEEE Transactions on Information Theory*, 2018.
- [53] A. Eftekhari, M. B. Wakin, and R. A. Ward. MC²: A two-phase algorithm for leveraged matrix completion. *arXiv preprint arXiv:1609.01795*, 2016.
- [54] I. Mitliagkas, C. Caramanis, and P. Jain. Streaming PCA with many missing entries. *Preprint*, 2014.
- [55] Amol Deshpande, Carlos Guestrin, Samuel R Madden, Joseph M Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 588–599. VLDB Endowment, 2004.
- [56] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. C. Eldar and G. Kutyniok, editors, *Compressed Sensing: Theory and Applications*, pages 95–110. Cambridge University Press, 2012.
- [57] Mark Rudelson, Roman Vershynin, et al. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.
- [58] P. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.
- [59] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Classics in Mathematics. Springer Berlin Heidelberg, 2013.

Algorithm 2 MOSES: A streaming algorithm for linear dimensionality reduction (efficient version)

Input: Sequence of vectors $\{y_t\}_t \subset \mathbb{R}^n$ and block size b .

Output: Sequence $\{\hat{S}_{kb,r}, \hat{\Gamma}_{kb,r}, \hat{Q}_{kb,r}\}_k$.

Body:

1) For $k = 1$,

- a) Form $y_1 \in \mathbb{R}^{n \times b}$ by concatenating $\{y_t\}_{t=1}^b$.
- b) Set

$$[\hat{S}_{b,r}, \hat{\Gamma}_{b,r}, \hat{Q}_{b,r}] = \text{SVD}_r(y_1),$$

where $\hat{S}_{b,r} \in \mathbb{R}^{n \times r}$ and $\hat{Q}_{b,r} \in \mathbb{R}^{b \times r}$ have orthonormal columns, and the diagonal matrix $\hat{\Gamma}_{b,r} \in \mathbb{R}^{r \times r}$ contains leading r singular values.

2) For $k \geq 2$, repeat

- a) Form $y_k \in \mathbb{R}^{n \times b}$ by concatenating $\{y_t\}_{t=(k-1)b+1}^{kb}$.
- b) Set

$$\dot{q}_k = \hat{S}_{(k-1)b,r}^* y_k \in \mathbb{R}^{r \times b}, \quad \hat{z}_k = y_k - \hat{S}_{(k-1)b,r} \dot{q}_k \in \mathbb{R}^{n \times b}.$$

- c) Let $[\hat{s}_k, v_k] = \text{QR}(\hat{z}_k)$ be the QR decomposition of \hat{z}_k , where $\hat{s}_k \in \mathbb{R}^{n \times b}$ has orthonormal columns and $v_k \in \mathbb{R}^{b \times b}$.
- d) Let

$$[u_k, \hat{\Gamma}_{kb,r}, \hat{q}_k] = \text{SVD}_r \left(\begin{bmatrix} \hat{\Gamma}_{(k-1)b,r} & \dot{q}_k \\ \mathbf{0}_{b \times r} & v_k \end{bmatrix} \right), \quad (33)$$

where $u_k, \hat{q}_k \in \mathbb{R}^{(r+b) \times r}$ have orthonormal columns and the diagonal matrix $\hat{\Gamma}_{kb,r} \in \mathbb{R}^{r \times r}$ contains leading r singular values in nonincreasing order.

e) Let

$$\hat{S}_{kb,r} = \begin{bmatrix} \hat{S}_{(k-1)b,r} & \hat{s}_k \end{bmatrix} u_k.$$

- f) **(optional)** If the number of rows of $\hat{Q}_{(k-1)b,r}$ exceeds n and $\hat{Q}_{(k-1)b,r}$ is not needed any more, it is optional in order to improve efficiency to set $\hat{Q}_{kb,r} = \hat{q}_k$.
- g) Otherwise, set

$$\hat{Q}_{kb,r} = \begin{bmatrix} \hat{Q}_{(k-1)b,r} & 0 \\ 0 & I_b \end{bmatrix} \hat{q}_k. \quad (34)$$

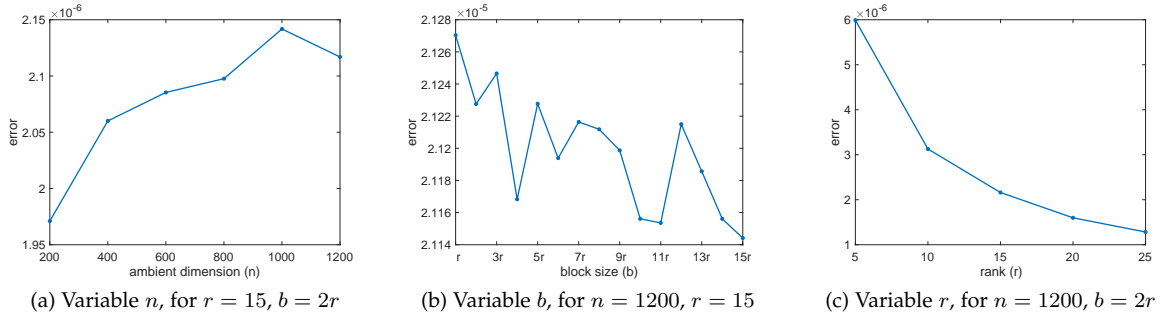


Figure 1: Performance of MOSES on synthetic datasets, see Section 5 for the details.

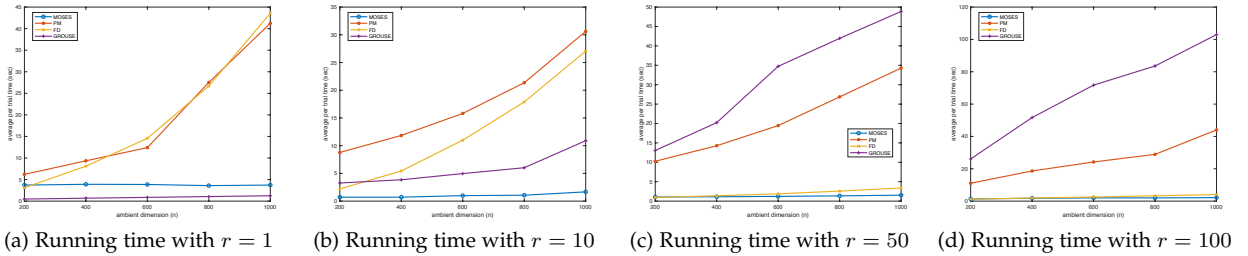


Figure 2: Computational complexity of all algorithms on synthetic datasets, see Section 5 for the details.

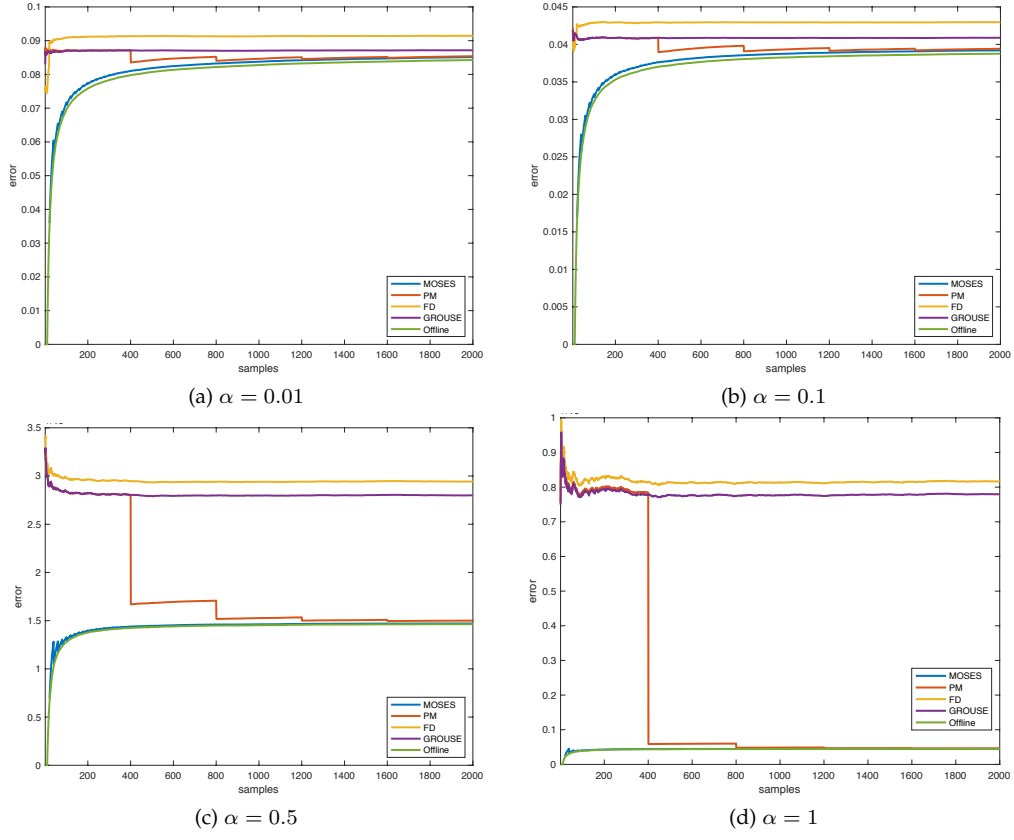


Figure 3: Comparisons on synthetic datasets, see Section 5 for the details.

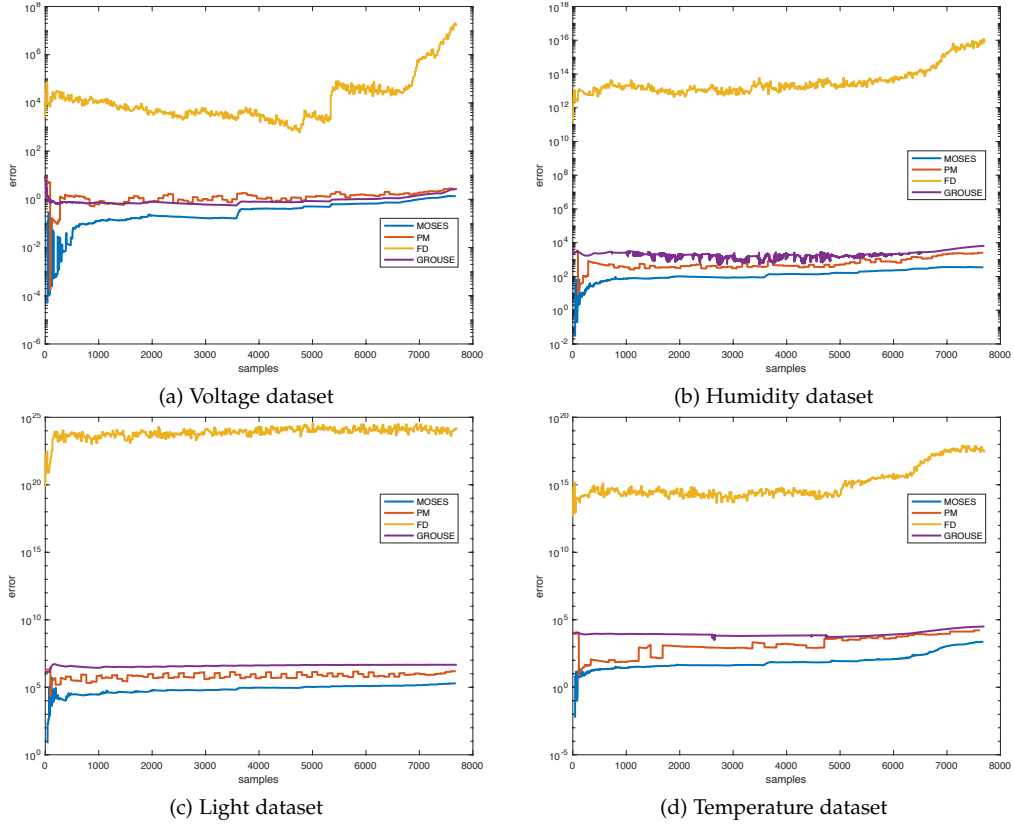


Figure 4: Comparisons on real-world datasets, see Section 5 for the details.

Supplementary Material

13

A OVERVIEW

The supplementary material contains more details about MOSES, an optimisation interpretation of MOSES, the proofs of the main results, as well as the acknowledgements.

B STORAGE AND COMPUTATIONAL REQUIREMENTS OF MOSES

The efficient implementation of MOSES in Algorithm 2 is based on the ideas from incremental SVD and it is straightforward to verify that Algorithms 1 and 2 are indeed equivalent; at iteration k , the relation between the output of Algorithm 1 ($\hat{\mathbf{Y}}_{kb,r}$) and the output of Algorithm 2 ($\hat{\mathbf{S}}_{kb,r}, \hat{\mathbf{\Gamma}}_{kb,r}, \hat{\mathbf{Q}}_{kb,r}$) is

$$\hat{\mathbf{Y}}_{kb,r} \stackrel{\text{SVD}}{=} \hat{\mathbf{S}}_{kb,r} \hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^*,$$

where the right-hand side above is the SVD of $\hat{\mathbf{Y}}_{kb,r}$. More specifically, $\hat{\mathbf{S}}_{kb,r} \in \mathbb{R}^{n \times r}$ has orthonormal columns and is the MOSES's estimate of leading r principal components of $\mathbf{Y}_{kb} \in \mathbb{R}^{n \times kb}$, where we recall that \mathbf{Y}_{kb} is the data received so far. Moreover,

$$\hat{\mathbf{S}}_{kb,r}^* \hat{\mathbf{Y}}_{kb,r} = \hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^* \in \mathbb{R}^{r \times kb}$$

is the projection of $\hat{\mathbf{Y}}_{kb,r}$ onto this estimate, namely $\hat{\mathbf{S}}_{kb,r}^* \hat{\mathbf{Y}}_{kb,r}$ is MOSES's estimate of the projected data matrix so far. In words, the efficient implementation of MOSES in Algorithm 2 explicitly maintains estimates of both principal components and the projected data, at every iteration.

Let us now evaluate the storage and computational requirements of MOSES. At the start of iteration k , Algorithm 2 stores the matrices

$$\hat{\mathbf{S}}_{(k-1)b,r} \in \mathbb{R}^{n \times r}, \quad \hat{\mathbf{\Gamma}}_{(k-1)b,r} \in \mathbb{R}^{r \times r}, \quad \hat{\mathbf{Q}}_{(k-1)b,r} \in \mathbb{R}^{(k-1)b \times r},$$

and after that also receives and stores the incoming block $\mathbf{y}_k \in \mathbb{R}^{n \times b}$. This requires $O(r(n + (k-1)b + 1)) + O(bn)$ bits of memory, because $\hat{\mathbf{\Gamma}}_{(k-1)b,r}$ is diagonal. Assuming that $b = O(r)$, Algorithm 2 therefore requires $O(r(n + kr))$ bits of memory at iteration k . Note that this is optimal, as it is impossible to store a rank- r matrix of size $n \times kb$ with fewer bits when $b = O(r)$.

It is also easy to verify that Algorithm 2 performs $O(r^2(n + kb)) = O(r^2(n + kr))$ flops in iteration k . The dependence of both storage and computational complexity on k is due to the fact that MOSES maintains both an estimate of principal components in $\hat{\mathbf{S}}_{kb,r}$ and an estimate of the projected data in $\hat{\mathbf{\Gamma}}_{kb,r} \hat{\mathbf{Q}}_{kb,r}^*$. To maximise the efficiency, one might optionally “flush out” the projected data after every n/b iterations, as described in the last step in Algorithm 2.

C OPTIMISATION VIEWPOINT

MOSES has a natural interpretation as an approximate solver for the non-convex optimisation program underlying PCA, which serves as its motivation. More specifically, recall that leading r principal components of \mathbf{Y}_T are obtained by solving the non-convex program

$$\min_{\mathcal{U} \in \mathcal{G}(n,r)} \|\mathbf{Y}_T - \mathbf{P}_{\mathcal{U}} \mathbf{Y}_T\|_F^2, \quad (35)$$

where the minimization is over the Grassmannian $\mathcal{G}(n,r)$, the set of all r -dimensional subspaces in \mathbb{R}^n . Above, $\mathbf{P}_{\mathcal{U}} \in \mathbb{R}^{n \times n}$ is the orthogonal projection onto the subspace \mathcal{U} . By construction in Section b, note that

$$\begin{aligned} \mathbf{Y}_T &= \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_T \end{bmatrix} \quad (\text{see (1)}) \\ &= \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_K \end{bmatrix} \in \mathbb{R}^{n \times T}, \end{aligned} \quad (36)$$

where $\{\mathbf{y}_k\}_{k=1}^K$ are the incoming blocks of data. This allows us to rewrite Program (35) as

$$\begin{aligned} \min_{\mathcal{U} \in \mathcal{G}(n,r)} \|\mathbf{Y}_T - \mathbf{P}_{\mathcal{U}} \mathbf{Y}_T\|_F^2 &= \min_{\mathcal{U} \in \mathcal{G}(n,r)} \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{P}_{\mathcal{U}} \mathbf{y}_k\|_F^2 \quad (\text{see (36)}) \\ &= \begin{cases} \min \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_1} \mathbf{y}_k\|_F^2 \\ \mathcal{U}_1 = \mathcal{U}_2 = \cdots = \mathcal{U}_K, \end{cases} \end{aligned} \quad (37)$$

where the last minimisation above is over all identical subspaces $\{\mathcal{U}_k\}_{k=1}^K \subset G(n, r)$. Our strategy is to make a sequence of approximations to the program in the last line above. In the first approximation, we only keep the first summand in the last line of (37). That is, our first approximation reads as

$$\begin{aligned} \left\{ \min \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_k} \mathbf{y}_k\|_F^2 \right. \\ \left. \mathcal{U}_1 = \mathcal{U}_2 = \cdots = \mathcal{U}_K \right\} &\geq \left\{ \min \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_1} \mathbf{y}_1\|_F^2 \right. \\ &\quad \left. \mathcal{U}_1 = \mathcal{U}_2 = \cdots = \mathcal{U}_K \right\} \\ &= \min_{\mathcal{U} \in G(n, r)} \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}} \mathbf{y}_1\|_F^2, \end{aligned} \quad (38)$$

where the second line above follows by setting $\mathcal{U} = \mathcal{U}_1 = \cdots = \mathcal{U}_K$. Let $\hat{\mathcal{S}}_{b,r}$ be a minimiser of the program in the last line above. Note that $\hat{\mathcal{S}}_{b,r}$ simply spans leading r principal components of \mathbf{y}_1 , akin to Program (35). This indeed coincides with the output of MOSES in the first iteration, because

$$\begin{aligned} \hat{\mathbf{Y}}_{b,r} &= \text{SVD}_r(\mathbf{y}_1) \quad (\text{see Algorithm 1}) \\ &= \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1. \quad (\text{similar to the second line of (5)}) \end{aligned} \quad (39)$$

Next consider the next approximation in which we keep two of the summands in the last line of (37), namely

$$\left\{ \min \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_k} \mathbf{y}_k\|_F^2 \right. \\ \left. \mathcal{U}_1 = \mathcal{U}_2 = \cdots = \mathcal{U}_K \right\} \geq \left\{ \min \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_1} \mathbf{y}_1\|_F^2 + \|\mathbf{y}_2 - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_2} \mathbf{y}_2\|_F^2 \right. \\ \left. \mathcal{U}_1 = \mathcal{U}_2 = \cdots = \mathcal{U}_K \right\}, \quad (40)$$

and then we substitute $\mathcal{U}_1 = \hat{\mathcal{S}}_{b,r}$ above to arrive at the new program

$$\begin{aligned} \left\{ \min \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_2} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1\|_F^2 + \|\mathbf{y}_2 - \mathbf{P}_{\mathcal{U}_K} \cdots \mathbf{P}_{\mathcal{U}_2} \mathbf{y}_2\|_F^2 \right. \\ \left. \mathcal{U}_2 = \mathcal{U}_3 = \cdots = \mathcal{U}_K \right\} \\ = \min_{\mathcal{U} \in G(n, r)} \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1\|_F^2 + \|\mathbf{y}_2 - \mathbf{P}_{\mathcal{U}} \mathbf{y}_2\|_F^2, \end{aligned} \quad (41)$$

where the second program above follows by setting $\mathcal{U} = \mathcal{U}_2 = \cdots = \mathcal{U}_K$. We can rewrite the above program as

$$\begin{aligned} &\min_{\mathcal{U} \in G(n, r)} \|\mathbf{y}_1 - \mathbf{P}_{\mathcal{U}} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1\|_F^2 + \|\mathbf{y}_2 - \mathbf{P}_{\mathcal{U}} \mathbf{y}_2\|_F^2 \\ &= \min_{\mathcal{U} \in G(n, r)} \left\| \begin{bmatrix} \mathbf{y}_1 - \mathbf{P}_{\mathcal{U}} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1 & \mathbf{y}_2 - \mathbf{P}_{\mathcal{U}} \mathbf{y}_2 \end{bmatrix} \right\|_F^2 \\ &= \min_{\mathcal{U} \in G(n, r)} \left\| \begin{bmatrix} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}^\perp} \mathbf{y}_1 & \mathbf{0}_{n \times b} \end{bmatrix} + \mathbf{P}_{\mathcal{U}^\perp} \begin{bmatrix} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1 & \mathbf{y}_2 \end{bmatrix} \right\|_F^2 \\ &= \|\mathbf{P}_{\hat{\mathcal{S}}_{b,r}^\perp} \mathbf{y}_1\|_F^2 + \min_{\mathcal{U} \in G(n, r)} \left\| \mathbf{P}_{\mathcal{U}^\perp} \begin{bmatrix} \mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1 & \mathbf{y}_2 \end{bmatrix} \right\|_F^2 \quad (\text{see the text below}) \\ &= \|\mathbf{P}_{\hat{\mathcal{S}}_{b,r}^\perp} \mathbf{y}_1\|_F^2 + \min_{\mathcal{U} \in G(n, r)} \left\| \mathbf{P}_{\mathcal{U}^\perp} \begin{bmatrix} \hat{\mathbf{Y}}_{b,r} & \mathbf{y}_2 \end{bmatrix} \right\|_F^2, \quad (\text{see (39)}) \end{aligned} \quad (42)$$

and let $\hat{\mathcal{S}}_{2b,r}$ be a minimiser of the last program above. Above, \perp shows the orthogonal complement of a subspace. The second to last line above follows because $\hat{\mathcal{S}}_{2b,r}$ is always within the column span of $[\mathbf{P}_{\hat{\mathcal{S}}_{b,r}} \mathbf{y}_1 \quad \mathbf{y}_2]$. Note also that $\hat{\mathcal{S}}_{2b,r}$ is the span of leading r principal components of the matrix $[\hat{\mathbf{Y}}_{b,r} \quad \mathbf{y}_2]$, similar to Program (35). This again coincides with the output of MOSES in the second iteration, because

$$\begin{aligned} \hat{\mathbf{Y}}_{2b,r} &= \text{SVD}_r \left(\begin{bmatrix} \hat{\mathbf{Y}}_{b,r} & \mathbf{y}_2 \end{bmatrix} \right) \quad (\text{see Algorithm 1}) \\ &= \mathbf{P}_{\hat{\mathcal{S}}_{2b,r}} \begin{bmatrix} \hat{\mathbf{Y}}_{b,r} & \mathbf{y}_2 \end{bmatrix}. \quad (\text{similar to the second line of (5)}) \end{aligned} \quad (43)$$

Continuing this procedure precisely produces the iterates of MOSES. Therefore we might interpret MOSES as an optimisation algorithm for solving Program (35) by making a sequence of approximations.

D SPIKED COVARIANCE MODEL AND ADDITIONAL REMARKS

A popular model in the statistics literature is the spiked covariance model, where the data vectors $\{\mathbf{y}_t\}_{t=1}^T$ are drawn from a distribution with a covariance matrix Ξ . Under this model, Ξ is a low-rank perturbation of the

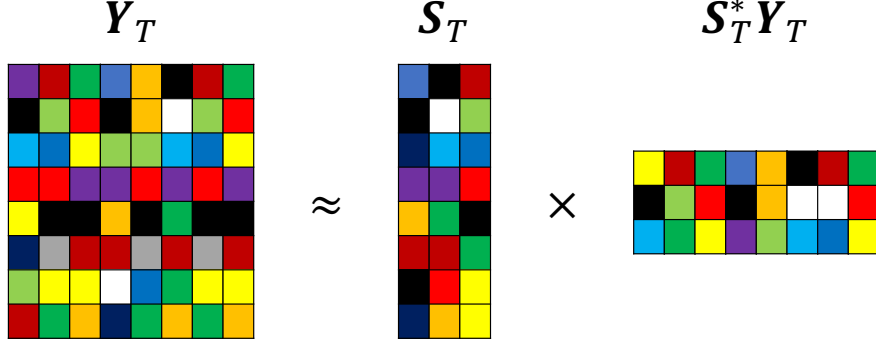


Figure 5: Given a data matrix $\mathbf{Y}_T \in \mathbb{R}^{n \times T}$, truncated SVD finds the best low-dimensional linear model to represent the data: For a typically small integer r , we compute $\mathbf{Y}_{T,r} = \text{SVD}_r(\mathbf{Y}_T) = \mathbf{S}_{T,r} \cdot \mathbf{S}_{T,r}^* \mathbf{Y}_T$, where $\mathbf{S}_{T,r} \in \mathbb{R}^{n \times r}$ contains leading r principal components of \mathbf{Y}_T and $\mathbf{S}_{T,r}^* \mathbf{Y}_T \in \mathbb{R}^{r \times T}$ is the projected data matrix with reduced dimension r (instead of n). This paper presents MOSES, a streaming algorithm for truncated SVD. Put differently, MOSES keeps both a running estimate of the principal components and the projection of data, received so far, onto this estimate.

identity matrix [18], [23], namely $\lambda_1(\Xi) = \dots = \lambda_r(\Xi) = \lambda$ and $\lambda_{r+1}(\Xi) = \dots = \lambda_n(\Xi) = 1$. Proposition 1 in this case reads as

$$\mathbb{E} \|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 \propto (n-r) + (n-r) \lambda \sqrt{\frac{\log T}{T}}, \quad (44)$$

where $\mathcal{S}_{T,r}$ spans leading r principal components of the data matrix \mathbf{Y}_T . In contrast, Theorem 1 roughly speaking states that

$$\mathbb{E} \|y - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} y\|_2^2 \propto (n-r) \left(\frac{T\lambda}{bn} \right)^{\frac{n}{\lambda}} + (n-r) \lambda \sqrt{\frac{\log T}{T}}, \quad (45)$$

where $\hat{\mathcal{S}}_{T,r}$ spans the output of MOSES. When $\lambda \gtrsim n \log(T/b) = n \log K$ in particular, we find that the error bounds in (44,45) are of the same order. That is, under the spiked covariance model, MOSES for streaming truncated SVD matches the performance of “offline” truncated SVD, provided that the underlying distribution has a sufficiently large spectral gap. In practice, (45) is often a conservative bound.

d.0.0.1 Proof strategy.: Starting with (21), the proof of Theorem 1 in Section g of the supplementary material breaks down the error associated with MOSES into two components as

$$\mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} y\|_2 \leq \frac{1}{T} \|\mathbf{Y}_T - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} \mathbf{Y}_T\|_F^2 + \left| \frac{1}{T} \|\mathbf{Y}_T - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} \mathbf{Y}_T\|_F^2 - \mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} y\|_2^2 \right|. \quad (46)$$

That is, we bound the population risk with the empirical risk. We control the empirical risk in the first part of the proof by noting that

$$\begin{aligned} \|\mathbf{Y}_T - \mathbf{P}_{\hat{\mathcal{S}}_{T,r}} \mathbf{Y}_T\|_F &= \|\mathbf{P}_{\hat{\mathcal{S}}_{T,r}^\perp} \mathbf{Y}_T\|_F \\ &= \|\mathbf{P}_{\hat{\mathcal{S}}_{T,r}^\perp} (\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r})\|_F \quad (\text{see (12)}) \\ &\leq \|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F, \end{aligned} \quad (47)$$

where the last line gauges how well the output of MOSES approximates the data matrix \mathbf{Y}_T , see (20). We then bound $\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F$ in two steps: As it is common in these types of arguments, the first step finds a deterministic upper bound for this norm, which is then evaluated for our particular stochastic setup.

- The deterministic bound appears in Lemma 1 and gives an upper bound for $\|\mathbf{Y}_T - \hat{\mathbf{Y}}_{T,r}\|_F$ in terms of the overall “innovation”. Loosely speaking, the innovation $\|\mathbf{P}_{\mathcal{S}_{(k-1)b,r}^\perp} \mathbf{y}_k\|_F$ at iteration k is the part of the new data block \mathbf{y}_k that cannot be described by the leading r principal components of data arrived so far, which span the subspace $\mathcal{S}_{(k-1)b,r}$.
- The stochastic bound is given in Lemma 2 and uses a tight perturbation result.

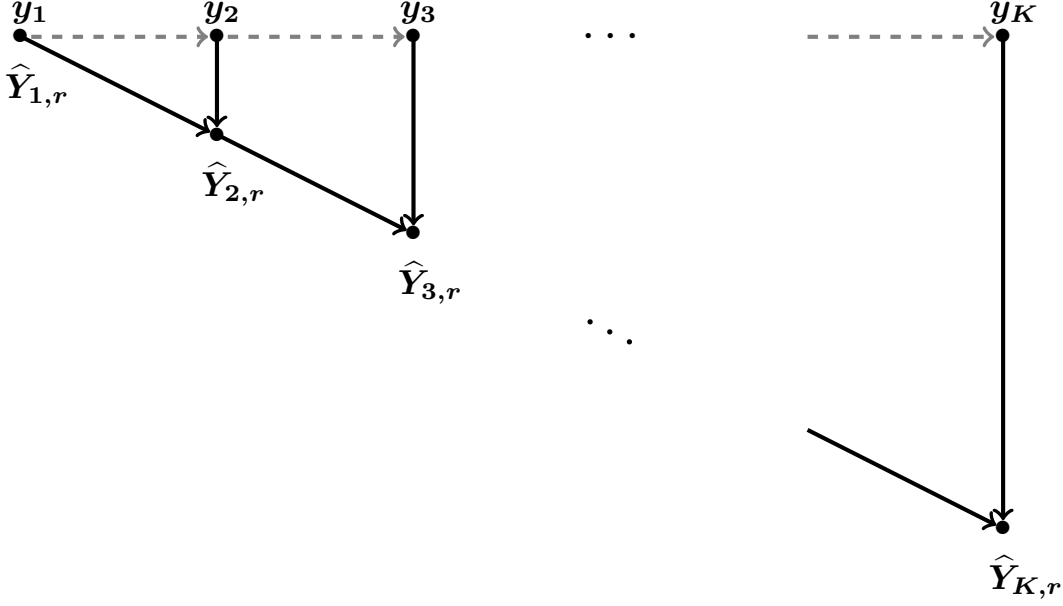


Figure 6: Streaming problems may be interpreted as a special case of distributed computing. Each data block y_k lives on a node of the chain graph and the nodes are combined, from left to right, following the structure of the “cone” tree.

Our argument so far yields an upper bound on the empirical loss $\|Y_T - P_{\hat{S}_{T,r}} Y_T\|_F$ that holds with high probability. In light of (46), it remains to control

$$\begin{aligned} \left| \frac{1}{T} \|Y_T - P_{\hat{S}_{T,r}} Y_T\|_F^2 - \mathbb{E}_y \|y - P_{\hat{S}_{T,r}} y\|_2^2 \right| &= \frac{1}{T} \left| \|Y_T - P_{\hat{S}_{T,r}} Y_T\|_F^2 - \mathbb{E} \|Y_T - P_{\hat{S}_{T,r}} Y_T\|_F^2 \right| \\ &= \frac{1}{T} \left| \|P_{\hat{S}_{T,r}^\perp} Y_T\|_F^2 - \mathbb{E} \|P_{\hat{S}_{T,r}^\perp} Y_T\|_F^2 \right| \end{aligned} \quad (48)$$

with a standard large deviation bound.

d.0.0.2 Other stochastic models.: While our results were restricted to the Gaussian distribution, they extend easily and with minimal change to the larger class of subgaussian distributions. Beyond subgaussian data models, Lemma 1 is the key deterministic result, relating the MOSES error to the overall innovation. One might therefore control the overall innovation, namely the right-hand side of (67) in Lemma 1, for any other stochastic model at hand.

ACKNOWLEDGEMENTS

AE is supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1 and also by the Turing Seed Funding grant SF019. RAH is supported by EPSRC grant EP/N510129/1. AG is supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1 and TU/C/000003. AE is grateful to Chinmay Hedge, Mike Wakin, Jared Tanner, and Mark Davenport for insightful suggestions and valuable feedback. Parts of this project were completed when AE was a Leibniz Fellow at Oberwolfach Research Institute for Mathematics and AE is extremely grateful for their hospitality.

E NOTATION AND TOOLBOX

This section collects the notation and a number of useful results in one place for the convenience of the reader. We will always use bold letters for matrices and calligraphic letters for subspaces, for example matrix \mathbf{A} and subspace \mathcal{S} . In particular, $\mathbf{0}_{a \times b}$ denotes the $a \times b$ matrix of all zeros. For integers $a \leq b$, we use the convention that $[a : b] = \{a, \dots, b\}$. We will also use MATLAB’s matrix notation to represent rows, columns, and blocks of matrices, for example $\mathbf{A}[1 : r, :]$ is the restriction of matrix \mathbf{A} to its first r rows. Throughout, C is an absolute constant, the value of which might change in every appearance.

In the appendices, $\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \dots$ denote the eigenvalues of a symmetric matrix \mathbf{A} and $\sigma_1(\mathbf{B}) \geq \sigma_2(\mathbf{B}) \geq \dots$ denotes the singular values of a matrix \mathbf{B} . Also $\rho_r^2(\mathbf{B}) = \sum_{i \geq r+1} \sigma_i^2(\mathbf{B})$ stands for the residual of matrix \mathbf{B} .

Let us also recall some of the spectral properties of a standard random Gaussian matrix, namely a matrix populated with independent random Gaussian variables with zero-mean and unit variance. For a standard Gaussian matrix $\mathbf{G} \in \mathbb{R}^{a \times b}$ with $a \geq b$ and for fixed $\alpha \geq 1$, Corollary 5.35 in [56] dictates that

$$\sqrt{a} - \alpha\sqrt{b} \leq \sigma_b(\mathbf{G}) \leq \sigma_1(\mathbf{G}) \leq \sqrt{a} + \alpha\sqrt{b}, \quad (49)$$

except with a probability of at most $e^{-C\alpha^2 b}$. Moreover, for a matrix $\mathbf{\Gamma} \in \mathbb{R}^{a' \times a}$ and $\alpha \geq 1$, an application of the Hensen-Wright inequality [57, Theorem 1.1] yields that

$$\left| \|\mathbf{\Gamma}\mathbf{G}\|_F^2 - \mathbb{E}\|\mathbf{\Gamma}\mathbf{G}\|_F^2 \right| \leq \beta, \quad (50)$$

for $\beta \geq 0$ and except with a probability of at most

$$\exp\left(-\min\left(\frac{\beta^2}{b\|\mathbf{\Gamma}\|^2\|\mathbf{\Gamma}\|_F^2}, \frac{\beta}{\|\mathbf{\Gamma}\|^2}\right)\right),$$

where $\|\cdot\|$ stands for spectral norm. In particular, with the choice $\beta = \alpha^2\|\mathbf{\Gamma}\|_F^2 b$ above and $\alpha \geq 1$, we find that

$$\|\mathbf{\Gamma}\mathbf{G}\|_F^2 \leq (1 + \alpha^2)\|\mathbf{\Gamma}\|_F^2 b \leq 2\alpha^2\|\mathbf{\Gamma}\|_F^2 b, \quad (51)$$

except with a probability of at most

$$\exp(-C\alpha^2 b\|\mathbf{\Gamma}\|_F^2/\|\mathbf{\Gamma}\|^2) \leq \exp(-C\alpha^2 b).$$

In a different regime, with the choice of $\beta = \alpha^2\|\mathbf{\Gamma}\|_F^2\sqrt{b}$ in (50) and $\alpha^2 \leq \sqrt{b}$, we arrive at

$$\left| \|\mathbf{\Gamma}\mathbf{G}\|_F^2 - \mathbb{E}\|\mathbf{\Gamma}\mathbf{G}\|_F^2 \right| = \left| \|\mathbf{\Gamma}\mathbf{G}\|_F^2 - b\|\mathbf{\Gamma}\|_F^2 \right| \leq \alpha^2\|\mathbf{\Gamma}\|_F^2\sqrt{b}, \quad (52)$$

except with a probability of at most

$$\exp(-C\alpha^4\|\mathbf{\Gamma}\|_F^2/\|\mathbf{\Gamma}\|^2) \leq \exp(-C\alpha^4).$$

F PROOF OF PROPOSITION 1

Let

$$\mathbf{\Xi} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^* = \mathbf{S}\mathbf{\Sigma}^2\mathbf{S}^* \in \mathbb{R}^{n \times n} \quad (53)$$

be the eigen-decomposition of the covariance matrix $\mathbf{\Xi}$, where $\mathbf{S} \in \mathbb{R}^{n \times n}$ is an orthonormal matrix and the diagonal matrix $\mathbf{\Lambda} = \mathbf{\Sigma}^2 \in \mathbb{R}^{n \times n}$ contains the eigenvalues of $\mathbf{\Xi}$ in nonincreasing order, namely

$$\mathbf{\Lambda} = \mathbf{\Sigma}^2 = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad \sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_n^2. \quad (54)$$

Throughout, we also make use of the condition number and residual, namely

$$\kappa_r = \frac{\sigma_1}{\sigma_r}, \quad \rho_r^2 = \rho_r^2(\mathbf{\Xi}) = \sum_{i=r+1}^n \sigma_i^2. \quad (\text{see (16)}) \quad (55)$$

Recall that $\{y_t\}_{t=1}^T \subset \mathbb{R}^n$ are the data vectors drawn from the Gaussian measure μ with zero mean and covariance matrix $\mathbf{\Xi}$, and that $\mathbf{Y}_T \in \mathbb{R}^{n \times T}$ is obtained by concatenating $\{y_t\}_{t=1}^T$. It follows that

$$y_t = \mathbf{S}\mathbf{\Sigma}g_t, \quad t \in [1 : T],$$

$$\mathbf{Y}_T = \mathbf{S}\mathbf{\Sigma}\mathbf{G}_T, \quad (56)$$

where $g_t \in \mathbb{R}^n$ and $\mathbf{G}_T \in \mathbb{R}^{n \times T}$ are standard random Gaussian vector and matrix, respectively. That is, g_t and \mathbf{G}_T are populated with independent Gaussian random variables with zero mean and unit variance. With these

preparations, we are now ready to prove Proposition 1. For y drawn from the Gaussian measure μ , note that¹⁸

$$\begin{aligned}
\mathbb{E}_y \|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 &= \mathbb{E}_y \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} y\|_2^2 \\
&= \mathbb{E}_y \langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, yy^* \rangle \\
&= \langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, \Xi \rangle \\
&= \left\langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, \Xi - \frac{\mathbf{Y}_T \mathbf{Y}_T^*}{T} \right\rangle + \frac{1}{T} \langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, \mathbf{Y}_T \mathbf{Y}_T^* \rangle \\
&= \left\langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, \Xi - \frac{\mathbf{Y}_T \mathbf{Y}_T^*}{T} \right\rangle + \frac{1}{T} \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 \\
&= \left\langle \mathbf{P}_{\mathcal{S}_{T,r}^\perp}, \Xi - \frac{\mathbf{Y}_T \mathbf{Y}_T^*}{T} \right\rangle + \frac{\rho_r^2(\mathbf{Y}_T)}{T} \quad (\text{see Program (9)}) \\
&= \frac{1}{T} \left(\mathbb{E} \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 - \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 \right) + \frac{\rho_r^2(\mathbf{Y}_T)}{T}. \quad (\text{see (56)}) \tag{57}
\end{aligned}$$

Let us next control the two components in the last line above. The first component above involves the deviation of random variable $\|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2$ from its expectation. By invoking the Hensen-Wright inequality in Section e and for $\tilde{\alpha}^2 \leq \sqrt{T}$, we write that

$$\begin{aligned}
\mathbb{E} \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 - \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 &= \mathbb{E} \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{S} \Sigma \cdot \mathbf{G}_T\|_F^2 - \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{S} \Sigma \cdot \mathbf{G}_T\|_F^2 \quad (\text{see (56)}) \\
&\leq \tilde{\alpha}^2 \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{S} \Sigma\|_F^2 \sqrt{T} \quad (\text{see (52)}) \\
&\leq \tilde{\alpha}^2 \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{S}\|_F^2 \|\Sigma\|^2 \sqrt{T} \\
&\leq \tilde{\alpha}^2 (n-r) \sigma_1^2 \sqrt{T}, \quad (\text{see (54,55)}) \tag{58}
\end{aligned}$$

except with a probability of at most $e^{-C\tilde{\alpha}^4}$. In particular, for the choice of $\tilde{\alpha}^2 = \alpha^2 \sqrt{\log T}$ with $\alpha^2 \leq \sqrt{T/\log T}$, we find that

$$\mathbb{E} \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 - \|\mathbf{P}_{\mathcal{S}_{T,r}^\perp} \mathbf{Y}_T\|_F^2 \leq \alpha^2 (n-r) \sigma_1^2 \sqrt{T \log T}, \tag{59}$$

except with a probability of $T^{-C\alpha^4}$. We next bound the second term in the last line of (57), namely the residual of \mathbf{Y}_T . Note that

$$\begin{aligned}
\rho_r^2(\mathbf{Y}_T) &= \rho_r^2(\mathbf{S} \Sigma \mathbf{G}_T) \quad (\text{see (56)}) \\
&= \rho_r^2(\Sigma \mathbf{G}_T) \quad (\mathbf{S}^* \mathbf{S} = \mathbf{I}_n) \\
&= \min_{\text{rank}(\mathbf{X})=r} \|\Sigma \mathbf{G}_T - \mathbf{X}\|_F^2. \quad (\text{see (55)}) \tag{60}
\end{aligned}$$

By substituting above the suboptimal choice of

$$\mathbf{X}_o = \begin{bmatrix} \Sigma[1:r, 1:r] \cdot \mathbf{G}_T[1:r, :] \\ \mathbf{0}_{(n-r) \times T} \end{bmatrix}, \tag{61}$$

we find that

$$\begin{aligned}
\rho_r^2(\mathbf{Y}_T) &= \min_{\text{rank}(\mathbf{X})=r} \|\Sigma \mathbf{G}_T - \mathbf{X}\|_F^2 \quad (\text{see (60)}) \\
&\leq \|\Sigma \mathbf{G}_T - \mathbf{X}_o\|_F^2 \\
&= \|\Sigma[r+1:n, r+1:n] \cdot \mathbf{G}_T[r+1:n, :]\|_F^2. \quad (\text{see (61)}) \tag{62}
\end{aligned}$$

Note that $\mathbf{G}_T[r+1:n, :] \in \mathbb{R}^{(n-r) \times T}$ is a standard Gaussian matrix. For $\alpha \geq 1$, an application of the Hensen-Wright inequality in Section e therefore implies that

$$\begin{aligned}
\rho_r^2(\mathbf{Y}_T) &\leq \|\Sigma[r+1:n, r+1:n] \cdot \mathbf{G}_T[r+1:n, :]\|_F^2 \quad (\text{see (62)}) \\
&\leq 2\alpha^2 \|\Sigma[r+1:n, r+1:n]\|_F^2 T \quad (\text{see (51)}) \\
&= 2\alpha^2 \rho_r^2 T, \quad (\text{see (55)}) \tag{63}
\end{aligned}$$

except with a probability of at most $e^{-C\alpha^2 T}$. We now substitute the bounds in (59) and (63) back into (57) to arrive at

$$\mathbb{E}\|y - \mathbf{P}_{\mathcal{S}_{T,r}} y\|_2^2 \leq \alpha^2(n-r)\sigma_1^2 \sqrt{T \log T} + 2\alpha^2 \rho_r^2, \quad (64)$$

when $\alpha^2 \leq \sqrt{T/\log T}$ and except with a probability of at most

$$T^{-C\alpha^4} + e^{-C\alpha^2 T} \leq T^{-C\alpha^4}, \quad \left(\alpha^2 \leq \sqrt{T/\log T} \right)$$

where we have used the abuse of notation in which C is a universal constant that is allowed to change in every appearance. This completes the proof of Proposition 1.

G PROOF OF THEOREM 1

In the rest of this paper, we slightly unburden the notation by using $\mathbf{Y}_k \in \mathbb{R}^{n \times kb}$ to denote \mathbf{Y}_{kb} . For example, we will use $\mathbf{Y}_K \in \mathbb{R}^{n \times T}$ instead of \mathbf{Y}_T because $T = Kb$. We also write $\hat{\mathcal{S}}_{k,r}$ instead of $\hat{\mathcal{S}}_{kb,r}$. As with the proof of Proposition 1, we argue that

$$\begin{aligned} \mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{K,r}} y\|_2^2 &\leq \frac{1}{T} \left(\mathbb{E} \|\mathbf{P}_{\hat{\mathcal{S}}_{K,r}^\perp} \mathbf{Y}_T\|_F^2 - \|\mathbf{P}_{\hat{\mathcal{S}}_{K,r}^\perp} \mathbf{Y}_T\|_F^2 \right) + \frac{1}{T} \|\mathbf{P}_{\hat{\mathcal{S}}_{K,r}^\perp} \mathbf{Y}_K\|_F^2 \quad (\text{similar to (57)}) \\ &\leq \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{1}{T} \|\mathbf{P}_{\hat{\mathcal{S}}_{K,r}^\perp} \mathbf{Y}_K\|_F^2 \quad (\text{see (59)}) \\ &= \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{1}{T} \|\mathbf{P}_{\hat{\mathcal{S}}_{K,r}^\perp} (\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r})\|_F^2 \quad (\text{see (12)}) \\ &\leq \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{1}{T} \|\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r}\|_F^2, \end{aligned} \quad (65)$$

except with a probability of at most $T^{-C\alpha^4}$ and provided that $\alpha^2 \leq \sqrt{T/\log T}$. It therefore remains to control the norm in the last line above. Recall that the output of MOSES, namely $\hat{\mathbf{Y}}_{K,r}$, is intended to approximate a rank- r truncation of \mathbf{Y}_K . We will therefore compare the error $\|\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r}\|_F$ in (65) with the true residual $\rho_r(\mathbf{Y}_K)$. To that end, our analysis consists of a deterministic bound and a stochastic evaluation of this bound. The deterministic bound is as follows, see Section h for the proof.

Lemma 1. *For every $k \in [1 : K]$, let $\mathbf{Y}_{k,r} = \text{SVD}_r(\mathbf{Y}_k) \in \mathbb{R}^{n \times kb}$ be a rank- r truncation of \mathbf{Y}_k and set $\mathcal{S}_{k,r} = \text{span}(\mathbf{Y}_{k,r}) \in \mathcal{G}(n, r)$. For $p > 1$, we also set*

$$\theta_k := 1 + \frac{p^{\frac{1}{3}} \|\mathbf{y}_k\|^2}{\sigma_r(\mathbf{Y}_{k-1})^2}. \quad (66)$$

Then the output of MOSES, namely $\hat{\mathbf{Y}}_{K,r}$, satisfies

$$\|\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r}\|_F^2 \leq \frac{p^{\frac{1}{3}}}{p^{\frac{1}{3}} - 1} \sum_{k=2}^K \left(\prod_{l=k+1}^K \theta_l \right) \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2, \quad (67)$$

where $\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \in \mathbb{R}^{n \times n}$ is the orthogonal projection onto the orthogonal complement of $\mathcal{S}_{k-1,r}$. Above, we use the convention that $\prod_{l=K+1}^K \theta_l = 1$.

In words, (67) gives a deterministic bound on the performance of MOSES. The term $\|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F$ in (67) is in a sense the “innovation” at iteration k , namely the part of the new data block \mathbf{y}_k that cannot be described by the current estimate $\mathcal{S}_{k-1,r}$. The overall innovation in (67) clearly controls the performance of MOSES. In particular, if the data blocks are drawn from the same distribution, this innovation gradually reduces as k increases. For example, if $\{\mathbf{y}_k\}_{k=1}^K$ are drawn from a distribution with a rank- r covariance matrix, then the innovation term vanishes almost surely after finitely many iterations. In contrast, when the underlying covariance matrix is high-rank, the innovation term decays more slowly and never completely disappears even as $k \rightarrow \infty$. We will next evaluate the right-hand side of (67) in a stochastic setup, see Section i for the proof.

Lemma 2. Suppose that $\{y_t\}_{t=1}^T$ are drawn from a zero-mean Gaussian probability measure with the covariance matrix $\Xi \in \mathbb{R}^{n \times n}$. Let $\sigma_1^2 \geq \sigma_2^2 \geq \dots$ be the eigenvalues of Ξ and recall the notation in (55). For $p > 1$, also let

$$\eta_r := \kappa_r + \frac{\sqrt{2}\alpha\rho_r}{p^{\frac{1}{6}}\sigma_r}.$$

For $\alpha \geq 1$, it then holds that

$$\|\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r}\|_F^2 \leq \frac{50p^{\frac{4}{3}}\alpha^2}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2\rho_r^2, r\sigma_1^2 + \rho_r^2) \eta_r^2 b \left(\frac{2K}{p\eta_r^2} + 2\right)^{p\eta_r^2}, \quad (68)$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that

$$b \geq \frac{p^{\frac{1}{3}}\alpha^2 r}{(p^{\frac{1}{6}} - 1)^2}, \quad b \geq C\alpha^2 r.$$

Substituting the right-hand side of (68) back into (65) yields that

$$\begin{aligned} \mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{K,r}} y\|_2^2 &\leq \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{1}{T} \|\mathbf{Y}_K - \hat{\mathbf{Y}}_{K,r}\|_F^2, \quad (\text{see (65)}) \\ &\leq \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{50p^{\frac{4}{3}}\alpha^2}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2\rho_r^2, r\sigma_1^2 + \rho_r^2) \frac{\eta_r^2}{K} \left(\frac{2K}{p\eta_r^2} + 2\right)^{p\eta_r^2}. \end{aligned} \quad (69)$$

In particular, if $K \geq p\eta_r^2$, we may simplify the above bound to read

$$\mathbb{E}_y \|y - \mathbf{P}_{\hat{\mathcal{S}}_{K,r}} y\|_2^2 \leq \alpha^2(n-r)\sigma_1^2 \sqrt{\frac{\log T}{T}} + \frac{50p^{\frac{1}{3}}\alpha^2 4^{p\eta_r^2}}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2\rho_r^2, r\sigma_1^2 + \rho_r^2) \left(\frac{K}{p\eta_r^2}\right)^{p\eta_r^2-1}, \quad (70)$$

which completes the proof of Theorem 1.

H PROOF OF LEMMA 1

Recall that the output of MOSES is the sequence of rank- r matrices $\{\hat{\mathbf{Y}}_k\}_{k=1}^K$. For every $k < K$, it is more convenient in the proof of Lemma 1 to pad both $\mathbf{Y}_k, \hat{\mathbf{Y}}_{k,r} \in \mathbb{R}^{n \times kb}$ with zeros to form the $n \times Kb$ matrices

$$\begin{bmatrix} \mathbf{Y}_k & \mathbf{0}_{n \times (K-k)b} \end{bmatrix}, \quad \begin{bmatrix} \hat{\mathbf{Y}}_{k,r} & \mathbf{0}_{n \times (K-k)b} \end{bmatrix}. \quad (71)$$

We overload the notation $\mathbf{Y}_k, \hat{\mathbf{Y}}_{k,r}$ to show the new $n \times Kb$ matrices in (71). Let

$$\begin{aligned} \hat{\mathcal{S}}_{k,r} &= \text{span}(\hat{\mathbf{Y}}_{k,r}) \in G(n, r), \\ \hat{\mathcal{Q}}_{k,r} &= \text{span}(\hat{\mathbf{Y}}_{k,r}^*) \in G(Kb, r) \end{aligned} \quad (72)$$

denote the (r -dimensional) column and row spaces of the rank- r matrix $\hat{\mathbf{Y}}_{k,r} \in \mathbb{R}^{n \times Kb}$, respectively. Let also $\hat{\mathcal{S}}_{k,r} \in \mathbb{R}^{n \times r}$ and $\hat{\mathcal{Q}}_{k,r} \in \mathbb{R}^{Kb \times r}$ be orthonormal bases for these subspaces. We also let $\mathcal{I}_k \subset \mathbb{R}^{Kb}$ denote the b -dimensional subspace spanned by the coordinates $[(k-1)b+1 : kb]$, namely

$$\mathcal{I}_k = \text{span} \left(\begin{bmatrix} \mathbf{0}_{(k-1)b \times b} \\ \mathbf{I}_b \\ \mathbf{0}_{(K-k)b \times b} \end{bmatrix} \right) \in G(Kb, b), \quad (73)$$

and we use the notation

$$\mathcal{J}_k := \mathcal{I}_1 \oplus \mathcal{I}_2 \cdots \oplus \mathcal{I}_k \in G(Kb, kb), \quad k \in [1 : K], \quad (74)$$

to denote the kb -dimensional subspace that spans the first kb coordinates in \mathbb{R}^{Kb} . The following technical lemma, proved in Section j, gives another way of expressing the output of MOSES, namely $\{\hat{\mathbf{Y}}_{k,r}\}_{k=1}^K$.

Lemma 3. For every $k \in [1 : K]$, it holds that

$$\hat{\mathbf{Y}}_{k,r} = \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}}, \quad (75)$$

or equivalently

$$\hat{\mathbf{Y}}_{k-1,r} + \mathbf{Y}_k \mathbf{P}_{\mathcal{I}_k} = \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_k}, \quad (76)$$

where

$$\tilde{\mathcal{Q}}_k := \hat{\mathcal{Q}}_{k-1,r} \oplus \mathcal{I}_k \subset \mathbb{R}^{Kb} \quad (77)$$

is the direct sum of the two subspaces $\hat{\mathcal{Q}}_{k-1,r}$ and \mathcal{I}_k . In particular, the update rule in Algorithm 1 can be written as

$$\mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} = \text{SVD}_r \left(\mathbf{Y}_K \mathbf{P}_{\tilde{\mathcal{Q}}_k} \right), \quad k \in [2 : K]. \quad (78)$$

Lastly we have the inclusion

$$\hat{\mathcal{Q}}_{k,r} \subset \tilde{\mathcal{Q}}_k \subset \mathcal{J}_k \in G(Kb, kb). \quad (79)$$

In particular, (75) and (79) together imply that

$$\hat{\mathbf{Y}}_{k,r} = \mathbf{Y}_K \mathbf{P}_{\mathcal{J}_k} \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} = \mathbf{Y}_k \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}},$$

that is, only \mathbf{Y}_k (containing the first kb data vectors) contributes to the formation of $\hat{\mathbf{Y}}_{k,r}$, the output of algorithm at iteration k , which was to be expected of course. Recall that $\hat{\mathbf{Y}}_{k,r}$ is intended to approximate $\mathbf{Y}_{k,r} = \text{SVD}_r(\mathbf{Y}_k)$. In light of Lemma 3, let us now derive a simple recursive expression for the residual $\mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r}$. For every $k \in [2 : K]$, it holds that

$$\begin{aligned} \mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r} &= \mathbf{Y}_K \mathbf{P}_{\mathcal{J}_k} - \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} \quad (\text{see (74) and (75)}) \\ &= \mathbf{Y}_K \mathbf{P}_{\mathcal{J}_{k-1}} + \mathbf{Y}_K \mathbf{P}_{\mathcal{I}_k} - \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} \quad (\text{see (74)}) \\ &= \mathbf{Y}_{k-1} + \mathbf{Y}_K \mathbf{P}_{\mathcal{I}_k} - \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} \quad (\text{see (74)}) \\ &= \mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r} + \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k-1,r}} + \mathbf{Y}_K \mathbf{P}_{\mathcal{I}_k} - \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} \quad (\text{see (75)}) \\ &= (\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_K (\mathbf{P}_{\hat{\mathcal{Q}}_{k-1,r}} + \mathbf{P}_{\mathcal{I}_k}) - \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} \\ &= (\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_K (\mathbf{P}_{\tilde{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}}). \quad (\text{see (77)}) \end{aligned} \quad (80)$$

Interestingly, the two terms in the last line of (80) are orthogonal, as proved by induction in Section k.

Lemma 4. For every $k \in [2 : K]$, it holds that

$$\left\langle \mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}, \mathbf{Y}_K (\mathbf{P}_{\tilde{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}}) \right\rangle = 0. \quad (81)$$

For fixed $k \in [2 : K]$, Lemma 4 immediately implies that

$$\begin{aligned} \|\mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r}\|_F^2 &= \left\| (\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_K (\mathbf{P}_{\tilde{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}}) \right\|_F^2 \quad (\text{see (80)}) \\ &= \|\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}\|_F^2 + \|\mathbf{Y}_K (\mathbf{P}_{\tilde{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}})\|_F^2 \quad (\text{see Lemma 4}) \\ &= \|\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}\|_F^2 + \rho_r \left(\|\hat{\mathbf{Y}}_{k-1,r} + \mathbf{Y}_k \mathbf{P}_{\mathcal{I}_k}\|_F^2 \right). \quad (\text{see (78) and (76)}) \end{aligned} \quad (82)$$

Recalling from (72) that $\hat{\mathcal{S}}_{k-1,r} = \text{span}(\hat{\mathbf{Y}}_{k-1,r})$, we bound the above expression by writing that

$$\begin{aligned} \|\mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r}\|_F^2 &= \|\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}\|_F^2 + \rho_r \left(\|\hat{\mathbf{Y}}_{k-1,r} + \mathbf{Y}_k \mathbf{P}_{\mathcal{I}_k}\|_F^2 \right) \\ &\leq \|\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}\|_F^2 + \left\| \mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \left(\hat{\mathbf{Y}}_{k-1,r} + \mathbf{Y}_k \mathbf{P}_{\mathcal{I}_k} \right) \right\|_F^2 \\ &= \|\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}\|_F^2 + \|\mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2, \quad (\text{see (72)}) \end{aligned} \quad (83)$$

where the second line follows from the sub-optimality of the choice of subspace $\hat{\mathcal{S}}_{k-1,r}$. Let us focus on the last norm above. For every k , let $\mathbf{Y}_{k,r} = \text{SVD}_r(\mathbf{Y}_k)$ be a rank- r truncation of \mathbf{Y}_k with the column span $\mathcal{S}_{k,r} = \text{span}(\mathbf{Y}_{k,r})$. We now write that

$$\begin{aligned} \|\mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{y}_k\|_F &\leq \|\mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{P}_{\mathcal{S}_{k-1,r}} \mathbf{y}_k\|_F + \|\mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F \quad (\text{triangle inequality}) \\ &\leq \|\mathbf{P}_{\hat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{P}_{\mathcal{S}_{k-1,r}}\|_F \cdot \|\mathbf{y}_k\|_F + \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F. \end{aligned} \quad (84)$$

The first norm in the last line above gauges the principal angles between the two r -dimensional subspaces $\hat{\mathcal{S}}_{k-1,r}$ and $\mathcal{S}_{k-1,r}$. We can bound this norm with a standard perturbation result, for example see [26, Lemma 6] or [58]. More specifically, we may imagine that \mathbf{Y}_{k-1} is a perturbed copy of $\mathbf{Y}_{k-1,r}$. Then the angle between

$\mathcal{S}_{k-1,r} = \text{span}(\mathbf{Y}_{k-1,r})$ and $\widehat{\mathcal{S}}_{k-1,r} = \text{span}(\widehat{\mathbf{Y}}_{k-1,r})$ is controlled by the amount of perturbation, namely with the choice of $\mathbf{A} = \widehat{\mathbf{Y}}_{k-1,r}$, $\mathbf{B} = \mathbf{Y}_{k-1}$, $\mathbf{B}_r = \mathbf{Y}_{k-1,r}$ in [26, Lemma 6], we find that

$$\|\mathbf{P}_{\widehat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{P}_{\mathcal{S}_{k-1,r}}\|_F \leq \frac{\|\mathbf{Y}_{k-1} - \widehat{\mathbf{Y}}_{k-1,r}\|_F}{\sigma_r(\mathbf{Y}_{k-1})}. \quad (85)$$

By plugging (85) back into (84), we find that

$$\|\mathbf{P}_{\widehat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{y}_k\| \leq \frac{\|\mathbf{y}_k\|}{\sigma_r(\mathbf{Y}_{k-1})} \cdot \|\mathbf{Y}_{k-1} - \widehat{\mathbf{Y}}_{k-1,r}\|_F + \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{Y}_k\|_F. \quad (86)$$

In turn, for $p > 1$, substituting the above inequality into (83) yields that

$$\begin{aligned} \|\mathbf{Y}_k - \widehat{\mathbf{Y}}_{k,r}\|_F^2 &\leq \|\mathbf{Y}_{k-1} - \widehat{\mathbf{Y}}_{k-1,r}\|_F^2 + \|\mathbf{P}_{\widehat{\mathcal{S}}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2 \quad (\text{see (83)}) \\ &\leq \left(1 + \frac{p^{\frac{1}{3}} \|\mathbf{y}_k\|^2}{\sigma_r(\mathbf{Y}_{k-1})^2}\right) \|\mathbf{Y}_{k-1} - \widehat{\mathbf{Y}}_{k-1,r}\|_F^2 + \frac{p^{\frac{1}{3}}}{p^{\frac{1}{3}} - 1} \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2 \quad (\text{see (86)}) \\ &=: \theta_k \|\mathbf{Y}_{k-1} - \widehat{\mathbf{Y}}_{k-1,r}\|_F^2 + \frac{p^{\frac{1}{3}}}{p^{\frac{1}{3}} - 1} \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2. \end{aligned} \quad (87)$$

where we used the inequality $(a_1 + a_2)^2 \leq qa_1^2 + \frac{qa_2^2}{q-1}$ for scalars a_1, a_2 and $q > 1$, with the choice of $q = p^{\frac{1}{3}}$. By unfolding the recursion in (87), we arrive at

$$\|\mathbf{Y}_K - \widehat{\mathbf{Y}}_{K,r}\|_F^2 \leq \frac{p^{\frac{1}{3}}}{p^{\frac{1}{3}} - 1} \sum_{k=2}^K \left(\prod_{l=k+1}^K \theta_l \right) \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2, \quad (88)$$

which completes the proof of Lemma 1.

I PROOF OF LEMMA 2

Recall that $\mathbf{y}_k \in \mathbb{R}^{n \times b}$, $\mathbf{Y}_k \in \mathbb{R}^{n \times kb}$ denote the k th block and the concatenation of the first k blocks of data, respectively. Since the data vectors are independently drawn from a zero-mean Gaussian probability measure with covariance matrix Ξ , it follows from (53,54) that

$$\begin{aligned} \mathbf{y}_k &= \mathbf{S} \Sigma \mathbf{g}_k, \\ \mathbf{Y}_k &= \mathbf{S} \Sigma \mathbf{G}_k, \end{aligned} \quad (89)$$

for every $k \in [1 : K]$, where $\mathbf{g}_k \in \mathbb{R}^{n \times b}$ and $\mathbf{G}_k \in \mathbb{R}^{n \times kb}$ are standard random Gaussian matrices. For fixed $k \in [2 : K]$, let us now study each of the random quantities on the right-hand side of (67). The following results are proved in Appendices l and m, respectively.

Lemma 5. (Bound on $\|\mathbf{y}_k\|$) For $\alpha \geq 1$, $p > 1$, and fixed $k \in [1 : K]$, it holds that

$$\|\mathbf{y}_k\| \leq p^{\frac{1}{6}} (\sigma_1 + \sqrt{2\alpha} p^{-\frac{1}{6}} \rho_r) \sqrt{b}, \quad (90)$$

except with a probability of at most $e^{-C\alpha^2 b}$ and provided that

$$b \geq \frac{\alpha^2 r}{(p^{\frac{1}{6}} - 1)^2}. \quad (91)$$

Lemma 6. (Bound on $\sigma_r(\mathbf{Y}_k)$) For $\alpha \geq 1$, $p > 1$, and fixed $k \in [1 : K]$, it holds that

$$\sigma_r(\mathbf{Y}_k) \geq p^{-\frac{1}{6}} \sigma_r \sqrt{kb}, \quad (92)$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that

$$b \geq \frac{\alpha^2 r}{(1 - p^{-\frac{1}{6}})^2}. \quad (93)$$

By combining Lemmas 5 and 6, we find for fixed $k \in [2 : K]$ that

$$\begin{aligned}
\theta_k &= 1 + \frac{p^{\frac{1}{3}} \|\mathbf{y}_k\|^2}{\sigma_r(\mathbf{Y}_{k-1})^2} \quad (\text{see (66)}) \\
&\leq 1 + \frac{p(\sigma_1 + \sqrt{2}\alpha p^{-\frac{1}{6}}\rho_r)^2 b}{\sigma_r^2(k-1)b} \quad (\text{see Lemmas 5 and 6}) \\
&=: 1 + \frac{p\eta_r^2}{k-1},
\end{aligned} \tag{94}$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that (93) holds. In particular, it follows that

$$\begin{aligned}
\prod_{l=k+1}^K \theta_l &\leq \prod_{l=k+1}^K \left(1 + \frac{p\eta_r^2}{l-1}\right) \quad (\text{see (94)}) \\
&\leq \frac{(K-1+p\eta_r^2)^{K-1+p\eta_r^2}}{(K-1)^{K-1}} \cdot \frac{(k-1)^{k-1}}{(k-1+p\eta_r^2)^{k-1+p\eta_r^2}} \quad (\text{see below}) \\
&= \left(1 + \frac{p\eta_r^2}{K-1}\right)^{K-1} \left(1 + \frac{p\eta_r^2}{k-1}\right)^{-k+1} \left(\frac{K-1+p\eta_r^2}{k-1+p\eta_r^2}\right)^{p\eta_r^2},
\end{aligned} \tag{95}$$

holds for every $k \in [2 : K]$ and except with a probability of at most $Ke^{-C\alpha r}$, where the failure probability follows from an application of the union bound. The second line above is obtained by bounding the logarithm of the product in that line with the corresponding integral. More specifically, it holds that

$$\begin{aligned}
&\log \left(\prod_{l=k+1}^K \left(1 + \frac{p\eta_r^2}{l-1}\right) \right) \\
&= \sum_{l=k}^{K-1} \log \left(1 + \frac{p\eta_r^2}{l}\right) \\
&\leq \int_{k-1}^{K-1} \log \left(1 + \frac{p\eta_r^2}{x}\right) dx \\
&= (K-1+p\eta_r^2) \log(K-1+p\eta_r^2) - (K-1) \log(K-1) \\
&\quad - (k-1+p\eta_r^2) \log(k-1+p\eta_r^2) + (k-1) \log(k-1),
\end{aligned} \tag{96}$$

where the third line above follows because the integrand is decreasing in x . Let us further simplify (95). Note that $K \geq k \geq 2$ and that $p\eta_r^2 \geq 1$ by its definition in (94). Consequently, using the relation $2 \leq (1 + 1/x)^x \leq e$ for $x \geq 1$, we can write that

$$2 \leq \left(1 + \frac{p\eta_r^2}{k-1}\right)^{\frac{k-1}{p\eta_r^2}} \leq e, \quad 2 \leq \left(1 + \frac{p\eta_r^2}{K-1}\right)^{\frac{K-1}{p\eta_r^2}} \leq e. \tag{97}$$

In turn, (97) allows us to simplify (95) as follows:

$$\begin{aligned}
\prod_{l=k+1}^K \theta_l &\leq \left(1 + \frac{p\eta_r^2}{K-1}\right)^{K-1} \left(1 + \frac{p\eta_r^2}{k-1}\right)^{-k+1} \left(\frac{K-1+p\eta_r^2}{k-1+p\eta_r^2}\right)^{p\eta_r^2} \quad (\text{see (95)}) \\
&\leq \left(\frac{e}{2}\right)^{p\eta_r^2} \left(\frac{K-1+p\eta_r^2}{k-1+p\eta_r^2}\right)^{p\eta_r^2}. \quad (\text{see (97)})
\end{aligned} \tag{98}$$

Next we control the random variable $\|\mathbf{P}_{\mathcal{S}_{k-1}^\perp} \mathbf{y}_k\|_F$ in (67) with the following result, proved in Section n.

Lemma 7. (Bound on the Innovation) For $\alpha \geq 1$ and fixed $k \in [2 : K]$, it holds that

$$\|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F \leq 5\alpha \min(\kappa_r \rho_r, \sqrt{r}\sigma_1 + \rho_r) \sqrt{b}, \tag{99}$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that $b \geq C\alpha^2 r$.

By combining Lemma 7 and (98), we finally find a stochastic bound for the right-hand side of (67). More

specifically, it holds that

$$\begin{aligned}
& \|Y_K - \hat{Y}_{K,r}\|_F^2 \\
& \leq \frac{p^{\frac{1}{3}}}{p^{\frac{1}{3}} - 1} \sum_{k=2}^K \left(\prod_{l=k+1}^K \theta_l \right) \|P_{\mathcal{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F^2 \quad (\text{see (67)}) \\
& \leq \frac{50p^{\frac{1}{3}}\alpha^2}{p^{\frac{1}{3}} - 1} \min(\kappa_r^2 \rho_r^2, r\sigma_1^2 + \rho_r^2) b \cdot \left(\frac{e}{2}\right)^{p\eta_r^2} (K-1 + p\eta_r^2)^{p\eta_r^2} \sum_{k=2}^K (k-1 + p\eta_r^2)^{-p\eta_r^2} \quad (\text{see (98) and Lemma 7}) \\
& \leq \frac{50p^{\frac{1}{3}}\alpha^2}{p^{\frac{1}{3}} - 1} \min(\kappa_r^2 \rho_r^2, r\sigma_1^2 + \rho_r^2) b \cdot \left(\frac{e}{2}\right)^{p\eta_r^2} (K-1 + p\eta_r^2)^{p\eta_r^2} \int_{p\eta_r^2}^\infty x^{-p\eta_r^2} dx \\
& = \frac{50p^{\frac{1}{3}}\alpha^2}{p^{\frac{1}{3}} - 1} \min(\kappa_r^2 \rho_r^2, r\sigma_1^2 + \rho_r^2) b \cdot \left(\frac{e}{2}\right)^{p\eta_r^2} (K-1 + p\eta_r^2)^{p\eta_r^2} \cdot \frac{(p\eta_r^2)^{-p\eta_r^2+1}}{p\eta_r^2 - 1} \\
& \leq \frac{50p^{\frac{1}{3}}\alpha^2}{p^{\frac{1}{3}} - 1} \min(\kappa_r^2 \rho_r^2, r\sigma_1^2 + \rho_r^2) b \left(\frac{2K}{p\eta_r^2} + 2\right)^{p\eta_r^2} \frac{p\eta_r^2}{p\eta_r^2 - 1} \\
& \leq \frac{50p^{\frac{4}{3}}\alpha^2}{(p^{\frac{1}{3}} - 1)^2} \cdot \min(\kappa_r^2 \rho_r^2, r\sigma_1^2 + \rho_r^2) \eta_r^2 b \left(\frac{2K}{p\eta_r^2} + 2\right)^{p\eta_r^2}, \quad (p, \eta_r \geq 1)
\end{aligned} \tag{100}$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that

$$b \geq \frac{p^{\frac{1}{3}}\alpha^2 r}{(p^{\frac{1}{6}} - 1)^2}, \quad b \geq C\alpha^2 r.$$

This completes the proof of Lemma 2.

J PROOF OF LEMMA 3

The proof is by induction. For $k = 1$, it holds that

$$\begin{aligned}
\hat{Y}_{1,r} &= \text{SVD}_r(\mathbf{Y}_1) \quad (\text{see Algorithm 1}) \\
&= \mathbf{Y}_1 P_{\hat{\mathcal{Q}}_{1,r}} \quad (\text{see (72)}) \\
&= \mathbf{Y}_K \mathbf{P}_{\mathcal{I}_1} \mathbf{P}_{\hat{\mathcal{Q}}_{1,r}} \\
&= \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{1,r}}, \quad (\hat{\mathcal{Q}}_{1,r} \subseteq \mathcal{I}_1)
\end{aligned} \tag{101}$$

which proves the base of induction. Next suppose that (75-79) hold for $[2 : k]$ with $k < K$. We now show that (75-79) hold also for $k + 1$. We can then write that

$$\begin{aligned}
\hat{Y}_{k+1,r} &= \text{SVD}_r \left(\hat{Y}_{k,r} + \begin{bmatrix} \mathbf{0}_{n \times kb} & \mathbf{y}_{k+1} & \mathbf{0}_{n \times (K-k-1)b} \end{bmatrix} \right) \quad (\text{see Algorithm 1}) \\
&= \text{SVD}_r \left(\mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}} + \mathbf{Y}_K \mathbf{P}_{\mathcal{I}_{k+1}} \right) \quad (\text{assumption of induction}) \\
&= \text{SVD}_r \left(\mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k+1,r}} \right), \quad (\text{see (77)})
\end{aligned} \tag{102}$$

which completes the proof of Lemma 3.

K PROOF OF LEMMA 4

In this proof only, it is convenient to use the notation $\text{rowspan}(\mathbf{A})$ to denote the row span of a matrix \mathbf{A} , namely $\text{rowspan}(\mathbf{A}) = \text{span}(\mathbf{A}^*)$. For every $k \in [1 : K]$, recall from (78) that $\mathbf{Y}_K(\mathbf{P}_{\hat{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}})$ is the residual of rank- r truncation of $\mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_k}$. Consequently,

$$\mathbf{Y}_K(\mathbf{P}_{\hat{\mathcal{Q}}_k} - \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}}) = \mathbf{Y}_K \mathbf{P}_{\hat{\mathcal{Q}}_{k,r}^\perp}, \quad k \in [1 : K], \tag{103}$$

where $\hat{\mathcal{Q}}_{k,r}^\perp$ is the orthogonal complement of $\hat{\mathcal{Q}}_{k,r}$ with respect to $\hat{\mathcal{Q}}_k$, namely

$$\hat{\mathcal{Q}}_k = \hat{\mathcal{Q}}_{k,r} \oplus \hat{\mathcal{Q}}_{k,r}^\perp, \quad \hat{\mathcal{Q}}_{k,r} \perp \hat{\mathcal{Q}}_{k,r}^\perp, \quad k \in [1 : K], \tag{104}$$

in which we conveniently set $\tilde{\mathcal{Q}}_1 = \mathcal{I}_1$, see (73). Using (103), we can rewrite (80) as

$$\begin{aligned} \mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r} &= (\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_k(\mathbf{P}_{\tilde{\mathcal{Q}}_k} - \mathbf{P}_{\tilde{\mathcal{Q}}_{k,r}}) \quad (\text{see (80)}) \\ &= (\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_k \mathbf{P}_{\tilde{\mathcal{Q}}_{k,r}^C}, \quad k \in [2 : K]. \end{aligned} \quad (105)$$

With the preliminaries out of the way, let us rewrite the claim of Lemma 4 as

$$\langle \mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}, \mathbf{Y}_k \mathbf{P}_{\tilde{\mathcal{Q}}_{k,r}^C} \rangle = 0, \quad k \in [2 : K], \quad (106)$$

see (81) and (103). Because $\tilde{\mathcal{Q}}_{k,r}^C \subset \tilde{\mathcal{Q}}_k$ by (104), it suffices to instead prove the stronger claim that

$$\text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) \perp \tilde{\mathcal{Q}}_k, \quad k \in [2 : K]. \quad (107)$$

We next prove (107) by induction. The base of induction, namely $k = 2$, is trivial. Suppose now that (107) holds for $[2 : k]$ with $k < K$. We next show that (107) holds for $k + 1$ as well. Note that

$$\begin{aligned} \text{rowspan}(\mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r}) &= \text{rowspan}((\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) + \mathbf{Y}_k \mathbf{P}_{\tilde{\mathcal{Q}}_{k,r}^C}) \quad (\text{see (105)}) \\ &\subseteq \text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) \oplus \tilde{\mathcal{Q}}_{k,r}^C. \end{aligned} \quad (108)$$

As we next show, both subspaces in the last line above are orthogonal to $\tilde{\mathcal{Q}}_{k+1}$. Indeed, on the one hand,

$$\begin{aligned} &\begin{cases} \text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) \perp \tilde{\mathcal{Q}}_k \supseteq \tilde{\mathcal{Q}}_{k,r}, & (\text{induction hypothesis and (79)}) \\ \text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) \subset \mathcal{J}_{k-1} \perp \mathcal{I}_{k+1}, & (\text{see (79) and (74)}) \end{cases} \\ \implies \text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) &\perp (\tilde{\mathcal{Q}}_{k,r} \oplus \mathcal{I}_{k+1}) = \tilde{\mathcal{Q}}_{k+1}. \quad (\text{see (77)}) \end{aligned} \quad (109)$$

On the other hand,

$$\begin{aligned} &\begin{cases} \tilde{\mathcal{Q}}_{k,r}^C \perp \tilde{\mathcal{Q}}_{k,r}, \\ \tilde{\mathcal{Q}}_{k,r}^C \subset \tilde{\mathcal{Q}}_k \subset \mathcal{J}_k \perp \mathcal{I}_{k+1}, & (\text{see (79) and (74)}) \end{cases} \\ \implies \tilde{\mathcal{Q}}_{k,r}^C &\perp (\tilde{\mathcal{Q}}_{k,r} \oplus \mathcal{I}_{k+1}) = \tilde{\mathcal{Q}}_{k+1}. \quad (\text{see (77)}) \end{aligned} \quad (110)$$

By combining (109) and (110), we conclude that

$$\begin{aligned} \text{rowspan}(\mathbf{Y}_k - \hat{\mathbf{Y}}_{k,r}) &\subseteq \text{rowspan}(\mathbf{Y}_{k-1} - \hat{\mathbf{Y}}_{k-1,r}) \oplus \tilde{\mathcal{Q}}_{k,r}^C \quad (\text{see (108)}) \\ &\perp \tilde{\mathcal{Q}}_{k+1}. \quad (\text{see (109,110)}) \end{aligned} \quad (111)$$

Therefore, (107) holds for every $k \in [2 : K]$ by induction. In particular, this proves Lemma 4.

L PROOF OF LEMMA 5

Note that

$$\begin{aligned} \|\mathbf{y}_k\| &= \|\mathbf{S}\Sigma\mathbf{g}_k\| \quad (\text{see (89)}) \\ &= \|\Sigma\mathbf{g}_k\| \quad (\mathbf{S}^*\mathbf{S} = \mathbf{I}_n) \\ &\leq \|\Sigma[1 : r, 1 : r] \cdot \mathbf{g}_k[1 : r, :]\| + \|\Sigma[r + 1 : n, r + 1 : n] \cdot \mathbf{g}_k[r + 1 : n, :]\| \quad (\text{triangle inequality}) \\ &\leq \sigma_1 \cdot \|\mathbf{g}_k[1 : r, :]\| + \|\Sigma[r + 1 : n, r + 1 : n] \cdot \mathbf{g}_k[r + 1 : n, :]\| \\ &\leq \sigma_1 \cdot \|\mathbf{g}_k[1 : r, :]\| + \|\Sigma[r + 1 : n, r + 1 : n] \cdot \mathbf{g}_k[r + 1 : n, :]\|_F, \end{aligned} \quad (112)$$

where we used MATLAB's matrix notation as usual. Note that both $\mathbf{g}_k[1 : r, :] \in \mathbb{R}^{r \times b}$ and $\mathbf{g}_k[r + 1 : n, :] \in \mathbb{R}^{(n-r) \times b}$ in (112) are standard Gaussian random matrices. For $\alpha \geq 1$ and $p > 1$, invoking the results about the

spectrum of Gaussian random matrices in Section e yields that

$$\begin{aligned}
\|\mathbf{y}_k\| &\leq \sigma_1 \cdot \|\mathbf{g}_k[1:r, :]\| + \|\boldsymbol{\Sigma}[r+1:n, r+1:n] \cdot \mathbf{g}_k[r+1:n, :]\|_F \quad (\text{see (112)}) \\
&\leq \sigma_1(\sqrt{b} + \alpha\sqrt{r}) + \sqrt{2}\alpha\|\boldsymbol{\Sigma}[r+1:n, r+1:n]\|_F\sqrt{b} \quad (\text{see (49,51) and } b \geq r) \\
&= \sigma_1(\sqrt{b} + \alpha\sqrt{r}) + \alpha\rho_r\sqrt{2b} \quad (\text{see (54,55)}) \\
&\leq p^{\frac{1}{6}}\sigma_1\sqrt{b} + \alpha\rho_r\sqrt{2b}, \quad \left(\text{if } b \geq \frac{\alpha^2 r}{(p^{\frac{1}{6}} - 1)^2} \right)
\end{aligned} \tag{113}$$

except with a probability of at most $e^{-C\alpha^2 r} + e^{-C\alpha^2 b} \leq e^{-C\alpha^2 r}$, where this final inequality follows from the assumption that $b \geq r$. This completes the proof of Lemma 5. We remark that a slightly stronger bound can be obtained by using Slepian's inequality for comparing Gaussian processes, see [56, Section 5.3.1] and [59, Section 3.1].

M PROOF OF LEMMA 6

For a matrix $\mathbf{A} \in \mathbb{R}^{n \times kb}$, it follows from the Fisher-Courant representation of the singular values that

$$\sigma_r(\mathbf{A}) \geq \sigma_r(\mathbf{A}[1:r, :]). \tag{114}$$

Alternatively, (114) might be verified using Cauchy's interlacing theorem applied to $\mathbf{A}\mathbf{A}^*$. For a vector $\gamma \in \mathbb{R}^{r \times r}$ and matrix $\mathbf{A} \in \mathbb{R}^{r \times r}$, we also have the useful inequality

$$\sigma_r(\text{diag}(\gamma)\mathbf{A}) \geq \min_{i \in [r]} |\gamma[i]| \cdot \sigma_r(\mathbf{A}), \tag{115}$$

where $\text{diag}(\gamma) \in \mathbb{R}^{r \times r}$ is the diagonal matrix formed from the entries of γ . Using the above inequalities, we may write that

$$\begin{aligned}
\sigma_r(\mathbf{Y}_k) &= \sigma_r(\mathbf{S}\boldsymbol{\Sigma}\mathbf{G}_k) \quad (\text{see (89)}) \\
&= \sigma_r(\boldsymbol{\Sigma}\mathbf{G}_k) \quad (\mathbf{S}^*\mathbf{S} = \mathbf{I}_n) \\
&\geq \sigma_r(\boldsymbol{\Sigma}[1:r, 1:r] \cdot \mathbf{G}_k[1:r, :]) \quad (\text{see (114)}) \\
&\geq \sigma_r \cdot \sigma_r(\mathbf{G}_k[1:r, :]) \quad (\text{see (115,54)})
\end{aligned} \tag{116}$$

Note also that $\mathbf{G}_k[1:r, :] \in \mathbb{R}^{r \times kb}$ above is a standard Gaussian random matrix. Using the spectral properties listed in Section e, we can therefore write that

$$\begin{aligned}
\sigma_r(\mathbf{Y}_k) &\geq \sigma_r \cdot \sigma_r(\mathbf{G}_k[1:r, :]) \quad (\text{see (116)}) \\
&\geq \sigma_r \cdot (\sqrt{kb} - \alpha\sqrt{r}) \quad (\text{see (49) and } b \geq r) \\
&\geq \sigma_r \cdot p^{-\frac{1}{6}}\sqrt{kb}, \quad \left(\text{if } b \geq \frac{\alpha^2 r}{(1 - p^{-\frac{1}{6}})^2} \right)
\end{aligned} \tag{117}$$

except with a probability of at most $e^{-C\alpha^2 r}$. This completes the proof of Lemma 6.

N PROOF OF LEMMA 7

Without loss of generality, we set $\mathbf{S} = \mathbf{I}_n$ in (53) to simplify the presentation, as this renders the contribution of the bottom rows of \mathbf{y}_k to the innovation typically small. We first separate this term via the inequality

$$\begin{aligned}
\|\mathbf{P}_{\mathbf{S}_{k-1,r}^\perp} \mathbf{y}_k\|_F &= \left\| \mathbf{P}_{\mathbf{S}_{k-1,r}^\perp} \begin{bmatrix} \mathbf{y}_k[1:r, :] \\ \mathbf{y}_k[r+1:n, :] \end{bmatrix} \right\|_F \\
&\leq \left\| \mathbf{P}_{\mathbf{S}_{k-1,r}^\perp} \begin{bmatrix} \mathbf{y}_k[1:r, :] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F + \|\mathbf{y}_k[r+1:n, :]\|_F. \quad (\text{triangle inequality})
\end{aligned} \tag{118}$$

To control the last norm above, we simply write that

$$\begin{aligned}
\|\mathbf{y}_k[r+1:n, :]\|_F &= \|\boldsymbol{\Sigma}[r+1:n, r+1:n] \cdot \mathbf{g}_k[r+1:n, :]\|_F \quad (\text{see (89)}) \\
&\leq \alpha\|\boldsymbol{\Sigma}[r+1:n, r+1:n]\|_F\sqrt{2b} \quad (\text{see (51)}) \\
&= \alpha\rho_r\sqrt{2b}, \quad (\text{see (55)})
\end{aligned} \tag{119}$$

except with a probability of at most $e^{-C\alpha^2 b}$. In the second line above, we used the fact that \mathbf{g}_k is a standard Gaussian random matrix. It therefore remains to control the first norm in the last line of (118). Note that

$$\begin{aligned}
\left\| \mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \begin{bmatrix} \mathbf{y}_k[1:r,:] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F &= \left\| \mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \begin{bmatrix} \mathbf{I}_r & \\ & \mathbf{0}_{n-r} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_k[1:r,:] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F \\
&=: \left\| \mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r \cdot \begin{bmatrix} \mathbf{y}_k[1:r,:] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F \\
&\leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \cdot \|\mathbf{y}_k[1:r,:]\| \\
&\leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \cdot \|\Sigma[1:r, 1:r]\| \cdot \|\mathbf{g}_k[1:r,:]\| \quad (\text{see (89)}) \\
&\leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \cdot \sigma_1 \cdot (\sqrt{b} + \alpha\sqrt{r}) \quad (\text{see (54,49)}) \\
&\leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \cdot \sigma_1 \sqrt{2b}, \quad (\text{if } b \geq C\alpha^2 r) \quad (120)
\end{aligned}$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that $b \geq C\alpha^2 r$. The fifth line above again uses the fact that \mathbf{g}_k is a standard Gaussian random matrix. Let us now estimate the norm in the last line above. Recall that $\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \in \mathbb{R}^{n \times n}$ projects onto the span of $\mathbf{Y}_{k-1,r} = \text{SVD}_r(\mathbf{Y}_{k-1})$, namely $\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp}$ projects onto the span of leading r left singular vectors of $\mathbf{Y}_{k-1} = \Sigma \mathbf{G}_{k-1}$, see (89). Because the diagonal entries of $\Sigma \in \mathbb{R}^{n \times n}$ are in nonincreasing order, it is natural to expect that $\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \approx \mathbf{J}_r$. We now formalise this notion using standard results from the perturbation theory. Note that one might think of $\mathbf{Y}_{k-1,r} = \text{SVD}_r(\mathbf{Y}_{k-1})$ as a perturbed copy of \mathbf{Y}_{k-1} . Note also that \mathbf{J}_r is the orthogonal projection onto the subspace

$$\text{span} \left(\begin{bmatrix} \mathbf{Y}_{k-1}[1:r,:] \\ \mathbf{0}_{(n-r) \times (k-1)b} \end{bmatrix} \right),$$

because $\mathbf{Y}_{k-1}[1:r,:]$ is almost surely full-rank. An application of Lemma 6 in [26] with \mathbf{A} as specified inside the parenthesis above and $\mathbf{B} = \mathbf{Y}_{k-1}$ yields that

$$\begin{aligned}
\|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F &\leq \frac{\left\| \mathbf{Y}_{k-1} - \begin{bmatrix} \mathbf{Y}_{k-1}[1:r,:] \\ \mathbf{0}_{(n-r) \times (k-1)b} \end{bmatrix} \right\|_F}{\sigma_r(\mathbf{Y}_{k-1})} \\
&= \frac{\|\mathbf{Y}_{k-1}[r+1:n,:]\|_F}{\sigma_r(\mathbf{Y}_{k-1})} \\
&= \frac{\|\Sigma[r+1:n, r+1:n] \cdot \mathbf{G}_{k-1}[r+1:n,:]\|_F}{\sigma_r(\mathbf{Y}_{k-1})} \quad (\text{see (89)}) \\
&\leq \frac{\alpha \|\Sigma[r+1:n, r+1:n]\|_F \sqrt{2(k-1)b}}{\sigma_r \sqrt{(k-1)b/2}} \quad (\text{see (51) and Lemma 6 with } p=8) \\
&= \frac{2\alpha\rho_r}{\sigma_r}, \quad (\text{see (55)}) \quad (121)
\end{aligned}$$

provided that $b \geq C\alpha^2 r$ and except with a probability of at most $e^{-C\alpha^2 b} + e^{-C\alpha^2 r} \leq e^{-C\alpha^2 r}$, where this last inequality follows from the assumption that $b \geq r$. It also trivially holds that

$$\|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp}\| \cdot \|\mathbf{J}_r\|_F \leq \|\mathbf{J}_r\|_F = \|\mathbf{I}_r\|_F = \sqrt{r},$$

where we used above the definition of \mathbf{J}_r in (120). Therefore, overall we find that

$$\|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \leq \min \left(\frac{2\alpha\rho_r}{\sigma_r}, \sqrt{r} \right). \quad (122)$$

Substituting the above bound back into (120) yields that

$$\begin{aligned}
\left\| \mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \begin{bmatrix} \mathbf{y}_k[1:r,:] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F &\leq \|\mathbf{P}_{\mathcal{S}_{k-1,r}^\perp} \mathbf{J}_r\|_F \cdot \sigma_1 \sqrt{2b} \quad (\text{see (120)}) \\
&\leq \min(\alpha\kappa_r \rho_r, \sigma_1 \sqrt{r}) \sqrt{8b}, \quad (\text{see (122,55)}) \quad (123)
\end{aligned}$$

except with a probability of at most $e^{-C\alpha^2 r}$. Combining (119) and (123) finally controls the innovation as

$$\begin{aligned}
\|P_{S_{k-1,r}^\perp} \mathbf{y}_k\|_F &\leq \left\| P_{S_{k-1,r}^\perp} \begin{bmatrix} \mathbf{y}_k[1:r, :] \\ \mathbf{0}_{(n-r) \times b} \end{bmatrix} \right\|_F + \|\mathbf{y}_k[r+1:n, :]\|_F \quad (\text{see (118)}) \\
&\leq \min(\alpha \kappa_r \rho_r, \sigma_1 \sqrt{r}) \sqrt{8b} + \alpha \rho_r \sqrt{2b} \quad (\text{see (123, 119)}) \\
&\leq 5\alpha \min(\kappa_r \rho_r, \sigma_1 \sqrt{r} + \rho_r) \sqrt{b}, \quad (\alpha, \kappa_r \geq 1) \tag{124}
\end{aligned}$$

except with a probability of at most $e^{-C\alpha^2 r}$ and provided that $b \geq C\alpha^2 r$. This completes the proof of Lemma 7.