

OPIT: A Simple but Effective Method for Sparse Subspace Tracking in High-dimension and Low-sample-size Context

Thanh Trung Le, Karim Abed-Meraim, *Fellow, IEEE*, Nguyen Linh Trung, *Senior Member, IEEE*, and Adel Hafiane, *Member, IEEE*.

Abstract—In recent years, sparse subspace tracking has attracted increasing attention in the signal processing community. In this paper, we propose a new provable effective method called OPIT (which stands for **Online Power Iteration via Thresholding**) for tracking the sparse principal subspace of data streams over time. Particularly, OPIT introduces a new adaptive variant of power iteration with space and computational complexity linear to the data dimension. In addition, a new column-based thresholding operator is developed to regularize the subspace sparsity. Utilizing both advantages of power iteration and thresholding operation, OPIT is capable of tracking the underlying subspace in both the classical regime and high dimensional regime. We also present a theoretical result on its convergence to verify its consistency in high dimensions. Several experiments are carried out on both synthetic and real data to demonstrate the performance of OPIT.

Index Terms—Sparse subspace tracking, data streams, high dimensions, thresholding, power iteration.

I. INTRODUCTION

Subspace tracking (ST) is a fundamental problem in adaptive signal processing with various applications in sensor array processing, wireless communication, image/video processing, and more [1]. It involves the task of tracking a low-dimensional subspace that can effectively represent data streams over time. Specifically, when the underlying subspace can be represented by a basis matrix consisting of sparse vectors, it is referred to as a *sparse subspace*. In such cases, ST becomes sparse subspace tracking (SST), which has gained significant interest in recent applications due to the increasing dimensions of data. By leveraging the advantages of sparse representation, SST enables more efficient modeling and analysis of high-dimensional streaming data.

In the literature, most subspace tracking methods are designed to estimate the underlying subspace from the sample covariance matrix (SCM); see [1]–[4] for good surveys. However, many rigorous theoretical results in random matrix theory

(e.g., [5]–[7]) indicated that the SCM is not a good estimator of the population (actual) covariance matrix in high-dimension, low-sample-size (HDLSS) contexts where datasets are massive in both dimension n and sample size T , and typically $n/T \rightarrow c \in (0, \infty]$. Without further structural knowledge about the data, ST methods turn out to be inconsistent in such a regime. Interestingly, the consistency of covariance estimation can be guaranteed under suitably structured sparsity regularizations, such as [8]–[11]. As a result, several effective methods have been proposed for sparse subspace estimation; see [12]–[15] for examples and [16], [17] for comprehensive surveys. However, in the adaptive (online) setting, there have been only few studies on SST thus far. Moreover, the existing SST methods suffer from certain limitations. Some of these methods are designed specifically for tracking rank-1 subspaces, which restricts their applicability to more general subspace tracking scenarios. Additionally, certain methods only support row sparsity which may not always align with practical situations. Furthermore, these existing methods often demonstrate inconsistencies when confronted with high-dimensional data. For detailed discussions, please refer to section II.

Contributions: In this paper, we introduce a new provable adaptive algorithm called OPIT (OPIT stands for Online Power Iteration via Thresholding) for SST.¹ OPIT overcomes the limitations mentioned earlier and offers several appealing features as compared to state-of-the-art SST algorithms. By leveraging the benefits of power iteration (PI) and thresholding methods, OPIT demonstrates improved performance. One key advantage of OPIT is its efficient use of past observations in a recursive manner while maintaining linear space complexity. As a result, OPIT achieves faster convergence rates and better estimation accuracy compared to existing PI-based SST methods. Additionally, OPIT has the capability to track rank- r subspaces and handle various sparsity scenarios, including row-sparse, elementwise-sparse, and local region-sparse subspaces. Notably, OPIT accurately tracks the sparse subspace in both classical and HDLSS regimes thanks to a newly introduced thresholding operator. Our theoretical analysis confirms the guaranteed convergence of OPIT with this thresholding operation under mild conditions. Moreover, OPIT is flexible and adaptable for different scenarios. For instance, its procedure can be adjusted to handle multiple incoming data streams or incorporate a forgetting factor, allowing the discounting of distant observations and enhancing OPIT’s tracking ability in

Thanh Trung Le is with the VNU University of Engineering and Technology, Hanoi, Vietnam and the University of Orléans, INSA CVL, PRISME, France. Email: thanhletrung@vnu.edu.vn.

Karim Abed-Meraim is with the University of Orléans, INSA CVL, PRISME, France, and the Academic Institute of France. Email: karim.abed-meraim@univ-orleans.fr

Nguyen Linh Trung (corresponding author) is with the VNU University of Engineering and Technology, Vietnam. Email: linhtrung@vnu.edu.vn.

Adel Hafiane is with the University of Orléans, INSA CVL, PRISME, France. Email: adel.hafiane@insa-cvl.fr.

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

¹A short part of this work has been presented at ICASSP 2022 [18].

dynamic environments. Additionally, by recasting its update rule into a column-wise update and employing the deflation transformation, we introduce a computationally efficient variant of OPIT called OPITd. This variant has lower complexity in terms of computation and memory storage, making it particularly suitable for tracking high-dimensional and large-scale data streams.

Paper Organization: The rest of the paper is organized as follows. Section II discusses the related works. Section III formulates the SST problem. Section IV presents the proposed OPIT algorithm and its variant OPITd, while Section V establishes its convergence analysis. Section VI provides several experiments to demonstrate the performance of the proposed algorithms as compared to state-of-the-art algorithms. Section VII concludes the paper. For easy reference, Table I summarizes frequently used acronyms and notations in this paper.

TABLE I: Acronyms and notational conventions

Acronyms	
ST	Subspace tracking
SST	Sparse ST
SCM	Sample covariance matrix
PCA	Principal component analysis
HDLSS	High-dimension, low-sample-size
PAST	Projection approximation subspace tracking
PI	Power iteration
API	Approximated PI
DPM	Data projection method
RLS	Recursive least-squares
SNR	Signal to noise ratio
Notations	
$x, \mathbf{x}, \mathbf{X}, \& \mathcal{X}$ or \mathbb{X}	scalar, vector, matrix, and set/subset/support
x_i or $\mathbf{x}(i)$	i -th entry of \mathbf{x}
$x_{i,j}$ or $\mathbf{X}(i,j)$	(i,j) -th entry of \mathbf{X}
$\mathbf{X}(i,:), \mathbf{X}(:,j)$	i -th row and j -th column of \mathbf{X}
$\mathbf{X}^\top, \mathbf{X}^{-1}, \mathbf{X}^\#$	transpose, inverse, and pseudo-inverse of \mathbf{X}
$\lambda_{\max}(\mathbf{X}), \lambda_{\min}(\mathbf{X})$	largest and smallest singular values of \mathbf{X}
$\ \cdot\ _p, \ \cdot\ _F$	ℓ_p -norm and Frobenius norm
$\ \mathbf{X}\ _0$	number of non-zero elements in \mathbf{X}
$\kappa(\mathbf{X})$	condition number of \mathbf{X} equal to $\frac{\lambda_{\max}(\mathbf{X})}{\lambda_{\min}(\mathbf{X})}$
$\lfloor x \rfloor$	integer closest to x
$\max\{x, y\}, \min\{x, y\}$	maximum and minimum of x and y
$(\cdot)_\perp$	orthogonal (perpendicular) complement
$\mathbb{E}\{\cdot\}$	expectation operator
\mathbf{I}_m	identity matrix of size m
$\mathcal{N}(\mu, \sigma^2)$	normal distribution of mean μ and variance σ^2
$\mathcal{N}(\mu, \Sigma)$	multivariate normal distribution of mean vector μ and covariance matrix Σ
$\theta(\mathbf{X}, \mathbf{Y})$	canonical angle (the largest principal angle) between two subspaces $\text{span}(\mathbf{X})$ and $\text{span}(\mathbf{Y})$

II. RELATED WORKS

In the literature, some online algorithms have been introduced for sparse subspace tracking [4]. A few of them are based on a two-stage approach in which one first utilizes a standard ST algorithm to estimate the underlying subspace and then seeks a sparse basis of the estimation under some sparsity criteria. Particularly in [19]–[21], several variants of the well-known Projection Approximation Subspace Tracking (PAST) [22] and Fast Approximated Power Iteration (FAPI) [23] were proposed to track the sparse principal subspace. Another good approach is to regularize the objective

function that aims to account for the sparse basis. In [24], the authors modified the objective function of PAST by adding a ℓ_1 -norm regularization term on the subspace basis matrix and then proposed a new robust variant of PAST called ℓ_1 -PAST to optimize it. Similar to ℓ_1 -PAST, the authors in [25] also introduced another adaptive algorithm using ℓ_1 -norm minimization called SPCAur (Sparse PCA) for sparse subspace tracking. SPCAur adopts the stochastic gradient descent on Grassmann manifolds, and it is capable of tracking the underlying sparse subspace from incomplete observations. In [26], a Bayesian-based algorithm called Online Variational Bayes Subspace Learning (OVBSL) was proposed to deal with the sparsity constraint on the subspace basis matrix. An advantage of OVBSL is that it is fully automated, i.e., no finetuning parameter is required. However, these algorithms are only effective in the classical regime where the sample size is much larger than the dimension, i.e., $n/T \rightarrow 0$ asymptotically.

Through the lens of machine learning and statistics, SST is generally referred to as the problem of online/streaming sparse PCA. In [27], the authors introduced an Oja's algorithm with Iterative Soft Thresholding (OIST) for online sparse PCA. Its convergence, steady-state, and phase transition were also derived to investigate the use of OIST in high dimensions. OIST is, however, designed only for rank-1 sparse subspaces. In [28], another streaming sparse PCA (SSPCA) algorithm was proposed and could estimate rank- r subspaces. Specifically, this algorithm uses a simple row truncation operator, which sets rows whose scores are smaller than a threshold to zero, for tracking the sparse principal subspace over time. However, this truncation operator is only designed for subspaces with a row-sparse support (i.e., all eigenvectors must share the same sparsity patterns) which may not always be met in practice. Indeed, it turns out to be ineffective for a sparse subspace with another support (e.g., elementwise sparsity). Its performance in terms of estimation accuracy is typically lower than other SST algorithms, see Fig. 5 and Fig. 6 for illustration.

It is worth noting that algorithms in [20], [21], OIST [27], and SSPCA [28] can be viewed as online variants of a classical method for principal subspace estimation, namely power iteration (PI). In the literature, there exist other power-based subspace trackers, and they can be broadly categorized into the following classes, Oja-types [29], [30], Natural Power (NP)-types [31], [32], Data Projection Method (DPM)-types [33], [34], and Approximated PI (API)-types [23], [35]. Specifically, all of them are designed for tracking the principal subspace of the SCM which is, however, not a good estimator of the true data covariance matrix in high dimensions. Accordingly, they turn out to be inconsistent estimators in the HDLSS regime.

In parallel, recent years have also witnessed considerable research advances on robust ST (RST) which aims to track the underlying subspace in the presence of data corruption [2]–[4]. For example, several RST algorithms were developed to handle sparse outliers, such as Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) [36], Parallel Subspace Estimation and Tracking by Recursive Least Squares (PETRELS)-types [37]–[39], and Recursive Projected Compressive Sensing (ReProCS)-types [40], [41]. To deal

with impulsive noises, three potential approaches are robust statistics [42], [43], adaptive Kalman filtering [44], [45], and weighted RLS [38], [46]. Very recently, α -divergence was specifically exploited to bolster the tracking ability of the well-known PAST and FAPI trackers in noisy and contaminated environments [47], [48]. However, none of them are designed for subspace tracking in the HDLSS context.

III. PROBLEM FORMULATION

Assume that at time t , we collect a data sample $\mathbf{x}_t \in \mathbb{R}^n$ satisfying the standard signal model

$$\mathbf{x}_t = \boldsymbol{\ell}_t + \mathbf{n}_t. \quad (1)$$

Here, $\boldsymbol{\ell}_t \in \mathbb{R}^n$ is a signal living in a low-dimensional subspace spanned by a sparse matrix $\mathbf{A} \in \mathbb{R}^{n \times r}$ with $r < n$, i.e., $\boldsymbol{\ell}_t = \mathbf{A}\mathbf{w}_t$, where $\mathbf{w}_t \in \mathbb{R}^r$ is a weight vector. The vector $\mathbf{n}_t \in \mathbb{R}^n$ represents a random noise, which is independent of \mathbf{w}_t . We assume that both \mathbf{w}_t and \mathbf{n}_t follow Gaussian distributions, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_r)$ and $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_n)$, respectively.²

Sparse subspace tracking problem can be stated as follows:

SST Problem: Given a data stream $\{\mathbf{x}_t\}_{t=1}^T$, we aim to estimate a sparse principle subspace basis \mathbf{A} that approximately spans the signals $\{\boldsymbol{\ell}_t\}_{t=1}^T$.

Generally, the underlying subspace \mathbf{A} can be extracted from an estimated version of the population covariance matrix

$$\mathbf{C} = \mathbb{E}\{\mathbf{x}_t \mathbf{x}_t^\top\} = \sigma_w^2 \mathbf{A} \mathbf{A}^\top + \sigma_n^2 \mathbf{I}_n. \quad (2)$$

Applying eigenvalue decomposition (EVD) on \mathbf{C} yields

$$\mathbf{C} \stackrel{\text{EVD}}{=} \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \boldsymbol{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^\top \\ \mathbf{U}_n^\top \end{bmatrix}. \quad (3)$$

Here, $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal elements are eigenvalues of \mathbf{C} sorted in decreasing order and $\mathbf{U} \in \mathbb{R}^{n \times n}$ contains the corresponding eigenvectors. Accordingly, $\mathbf{U}_s \in \mathbb{R}^{n \times r}$ and $\mathbf{U}_n \in \mathbb{R}^{n \times (n-r)}$ represent the principal subspace and the minor subspace of \mathbf{C} , respectively. The orthogonal projection matrix of the sparse principal subspace is unique (i.e., $\mathbf{U}_s \mathbf{U}_s^\top = \mathbf{A} \mathbf{A}^\#$), so \mathbf{A} can be obtained as $\mathbf{A} = \mathbf{U}_s \mathbf{Q}^*$ with

$$\mathbf{Q}^* = \underset{\mathbf{Q} \in \mathbb{R}^{r \times r}}{\text{argmin}} \|\mathbf{U}_s \mathbf{Q}\|_0 \text{ s.t. } \mathbf{Q} \text{ is full-rank}, \quad (4)$$

where $\|\cdot\|_0$ promotes the sparsity on \mathbf{A} . In several applications, we often emphasize the principal subspace rather than its specific basis, such as dimensionality reduction [49] and array processing [50]. In this work, our main objective is to track the principal (signal) subspace of \mathbf{A} while the sparsifying step (4) is optional.

Most state-of-the-art SST algorithms estimate the principal subspace of $\mathbf{C}_T = 1/T \sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t^\top$ [4]. However, in a high-dimensional regime where $n/T \rightarrow 0$ a.s., \mathbf{C}_T is a poor

estimator of \mathbf{C} . This limitation in an adaptive scheme is not necessarily due to a data shortage but to the time variation which forces us to use a limited window of time instead of all the data. Particularly, it has been shown that \mathbf{C}_T is not a consistent estimate of \mathbf{C} in the HDLSS regime, e.g., [51]–[53]. As a result, most of SST algorithms are ineffective in high dimensions, as illustrated in Fig. 6.

On the other hand, under certain conditions, it is proved in [8], [9], [54] that

$$\|\mathbf{C} - \tau(\mathbf{C}_T, \cdot)\|_2 \rightarrow 0 \text{ a.s. as } T \rightarrow \infty, \quad (5)$$

where $\tau(\mathbf{C}_T, \cdot)$ is an appropriate thresholding operation on \mathbf{C}_T . Thresholding serves as a regularization for covariance matrices, aiming to reduce their variability and simplify their structure by eliminating redundant parameters, and hence, improve the precision and consistency of the covariance estimator. In the HDLSS regime, the sample covariance matrix \mathbf{C}_T may contain unwanted by-products, such as correlations between the signal and noise vectors, even when they are expected to be independent. Therefore, \mathbf{C}_T may not accurately reflect the true underlying covariance structure. By applying thresholding, we can effectively capture the dominant patterns and significant relationships among variables. It is achieved by shrinking or eliminating small and less reliable components in \mathbf{C}_T . By doing so, thresholding can mitigate the undesired effects associated with HDLSS, enhancing the accuracy and consistency of the covariance estimator. Thanks to (5), in the next section, we derive a novel adaptive (online) algorithm based on the power iteration and thresholding techniques that are capable of tracking the sparse principal subspace in both the classical regime and the HDLSS regime.

IV. PROPOSED METHODS

In this section, a novel effective algorithm using thresholding is developed for sparse subspace tracking. This algorithm is dubbed as OPIT which stands for Online Power Iteration via Thresholding. We next derive a fast variant of OPIT called OPITd with lower complexity, thanks to the deflation transformation. Some remarks on OPIT and OPITd are discussed in the following subsection.

A. OPIT Algorithm

We first recall the main steps of the standard power iteration (PI) method we primarily leverage in order to develop our OPIT algorithm for computing the dominant eigenvectors of \mathbf{C}_t . At the ℓ -th iteration, PI updates (i) $\mathbf{S}_\ell \leftarrow \mathbf{C}_t \mathbf{U}_{\ell-1}$ and (ii) $\mathbf{U}_\ell \leftarrow \text{QR}(\mathbf{S}_\ell)$, where \mathbf{U}_ℓ is the Q-factor of QR factorization of \mathbf{S}_ℓ . PI starts from an initial matrix $\mathbf{U}_0 \in \mathbb{R}^{n \times r}$ and returns an orthonormal matrix \mathbf{U}_L , where L is the number of iterations [1].

In an adaptive scheme, the iteration step of PI can coincide with the data collection in time. At time t , the sample covariance matrix \mathbf{C}_t can be recursively updated by $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}_t \mathbf{x}_t^\top$ and $\mathbf{C}_t = t^{-1} \mathbf{R}_t$. As streaming data can vary with time, we propose to use a forgetting factor β ($0 < \beta \leq 1$) to discount the impact of old observations exponentially. The

²In an adaptive scheme, the matrix \mathbf{A} may be slowly varying with time, i.e., $\mathbf{A} = \mathbf{A}_t$. The Gaussian distribution assumption is employed for both \mathbf{w}_t and \mathbf{n}_t to facilitate our convergence analysis, as detailed in Section V. Our algorithm is capable of successfully estimating the underlying subspace as well as tracking its variation over time in other settings. For illustrative examples, please refer to Section VI.

Algorithm 1: OPIT

INPUT: $\{\mathbf{x}_t\}_{t=1}^T$, $\mathbf{x}_t \in \mathbb{R}^n$, target rank r , a forgetting factor $0 < \beta \leq 1$, window of length $W \geq 1$, and a thresholding factor k :

$$k = \begin{cases} \lfloor (1 - \omega_{\text{sparse}})n \rfloor & \text{if } \omega_{\text{sparse}} \text{ is given,} \\ \lfloor 10r \log n \rfloor & \text{if } \omega_{\text{sparse}} \text{ is unknown,} \end{cases}$$

where ω_{sparse} is the sparsity level of the sparse basis (i.e., the percentage of zero-valued elements in the basis matrix).

INITIAL: Any $\mathbf{U}_0 \in \mathbb{R}^{n \times r}$, $\mathbf{S}_0 = \mathbf{0}_{n \times r}$, $\mathbf{E}_0 = \mathbf{0}_{r \times r}$

MAIN PROGRAM:

PROCEDURE	Complexity
for $t = 1, 2, \dots, T/W$ do	-
$\mathbf{X}_t = [\mathbf{x}_{(t-1)W+1}, \dots, \mathbf{x}_{tW}]$	
$\mathbf{Z}_t = \mathbf{U}_{t-1}^\top \mathbf{X}_t$	$\mathcal{O}(Wnr)$
$\mathbf{S}_t = \beta \mathbf{S}_{t-1} \mathbf{E}_{t-1} + \mathbf{X}_t \mathbf{Z}_t^\top$	$\mathcal{O}(nr^2 + Wnr)$
$\hat{\mathbf{S}}_t = \tau(\mathbf{S}_t, k)$ // thresholding	$\mathcal{O}(nr + rk \log k)$
$\mathbf{U}_t = \begin{cases} \text{QR}(\hat{\mathbf{S}}_t) \\ \hat{\mathbf{S}}_t / \ \hat{\mathbf{S}}_t\ _2 \end{cases}$	$\mathcal{O}(nr^2)$
$\mathbf{E}_t = \mathbf{U}_{t-1}^\top \mathbf{U}_t$	$\mathcal{O}(nr^2)$
end for	
OUTPUT: $\mathbf{U}_t \in \mathbb{R}^{n \times r}$	

// THRESHOLDING $\hat{\mathbf{S}}_t = \tau(\mathbf{S}_t, k)$

PROCEDURE
for $i = 1, 2, \dots, r$ do
$\mathbf{s}_i = \mathbf{S}_t(:, i)$
Find the set $\mathcal{T}_i \subset [1, 2, \dots, n]$ containing indices of k strongest (absolute value) elements of \mathbf{s}_i
Form $\hat{\mathbf{S}}_t(:, i) = \hat{\mathbf{s}}_i$, where $\hat{\mathbf{s}}_i(j) = \begin{cases} \mathbf{s}_i(j) & \text{if } j \in \mathcal{T}_i \\ 0 & \text{if } j \notin \mathcal{T}_i \end{cases}$
end for
OUTPUT: $\hat{\mathbf{S}}_t \in \mathbb{R}^{n \times r}$

underlying subspace \mathbf{U}_t is then derived from spectral analysis of \mathbf{R}_t which is updated continuously by

$$\mathbf{R}_t = \beta \mathbf{R}_{t-1} + \mathbf{x}_t \mathbf{x}_t^\top. \quad (6)$$

Together with the fact that $\text{QR}(\mathbf{R}_t \mathbf{U}_{t-1}) = \text{QR}(\mathbf{C}_t \mathbf{U}_{t-1})$, we can rewrite the first step of PI as follows

$$\mathbf{S}_t = \mathbf{R}_t \mathbf{U}_{t-1} = \beta \mathbf{R}_{t-1} \mathbf{U}_{t-1} + \mathbf{x}_t \mathbf{z}_t^\top, \quad (7)$$

where $\mathbf{z}_t = \mathbf{U}_{t-1}^\top \mathbf{x}_t$.

We can utilize the previous subspace as a warm start in the tracking process. Hereby, a key step at each time t is to project \mathbf{U}_t onto the column space of \mathbf{U}_{t-1} , i.e.,

$$\mathbf{U}_t = \mathbf{U}_{t-1} \mathbf{E}_t + \mathbf{U}_{t-1, \perp} \mathbf{F}_t, \quad (8)$$

where $\mathbf{U}_{t-1, \perp}$ is the orthogonal complement of \mathbf{U}_{t-1} ,³ $\mathbf{E}_t = \mathbf{U}_{t-1}^\top \mathbf{U}_t$ and $\mathbf{F}_t = \mathbf{U}_{t-1, \perp}^\top \mathbf{U}_t$ are coefficient matrices. Specifically, the first term of (8) represents the ‘‘old’’ information in \mathbf{U}_t , while the second one is its distinctive new information. Substituting \mathbf{U}_{t-1} according to (8) (one time-step delayed) into (7) results in

$$\mathbf{S}_t = \beta \mathbf{S}_{t-1} \mathbf{E}_{t-1} + \beta \mathbf{R}_{t-1} \mathbf{U}_{t-2, \perp} \mathbf{F}_{t-1} + \mathbf{x}_t \mathbf{z}_t^\top. \quad (9)$$

The complement of projecting \mathbf{x}_t into the subspace \mathbf{U}_{t-1} at time t can be given by

$$\mathbf{y}_t = (\mathbf{I} - \mathbf{U}_{t-1} \mathbf{U}_{t-1}^\top) \mathbf{x}_t = \mathbf{x}_t - \mathbf{U}_{t-1} \mathbf{z}_t. \quad (10)$$

³The columns of $\mathbf{U}_{t-1, \perp}$ constitute an orthonormal basis for the orthogonal complement of the column span of \mathbf{U}_{t-1} .

Here, \mathbf{y}_t is orthogonal to the column space of \mathbf{U}_{t-1} . For short, we denote $\Delta \mathbf{U}_{t-1} = \mathbf{U}_{t-2, \perp} \mathbf{F}_{t-1}$. Based on (10), we obtain another expression of $\Delta \mathbf{U}_{t-1}$ as follows

$$\Delta \mathbf{U}_{t-1} = \bar{\mathbf{y}}_{t-1} \mathbf{h}_{t-1}^\top \quad \text{where } \mathbf{h}_{t-1} = \mathbf{U}_{t-1}^\top \bar{\mathbf{y}}_{t-1}, \quad (11)$$

where $\bar{\mathbf{y}}_{t-1} = \mathbf{y}_{t-1} / \|\mathbf{y}_{t-1}\|_2$. See Section A in our supplementary document for its derivation. Under the assumption that the underlying subspace is fixed or slowly varying with time (i.e., $\mathbf{U}_{t-2} \mathbf{U}_{t-2}^\top \simeq \mathbf{U}_{t-1} \mathbf{U}_{t-1}^\top$), $\bar{\mathbf{y}}_{t-1}$ is nearly orthogonal to the subspace spanned by \mathbf{U}_{t-1} . In other words, angles between $\bar{\mathbf{y}}_{t-1}$ and columns of \mathbf{U}_{t-1} are very close to $\pi/2$, and hence, the norm of \mathbf{h}_{t-1} in (11) is very small. Therefore, $\Delta \mathbf{U}_{t-1}$ and $\mathbf{R}_{t-1} \Delta \mathbf{U}_{t-1}$ are negligible and can be ignored during the tracking process without any major performance degradation. It stems from the fact that the presence of a small perturbation does not really affect the performance of power methods [55]. Accordingly, a good approximation to (9) can be given by

$$\mathbf{S}_t \simeq \beta \mathbf{S}_{t-1} \mathbf{E}_{t-1} + \mathbf{x}_t \mathbf{z}_t^\top. \quad (12)$$

In this work, we utilize the thresholding operator $\tau(\cdot, \cdot)$ on the update (12) in the following manner

$$\hat{\mathbf{S}}_t \triangleq \tau(\mathbf{S}_t, k), \quad (13)$$

where the thresholding factor k can be determined as in Algorithm 1. Here, $\hat{\mathbf{S}}_t$ is computed from \mathbf{S}_t by keeping the k strongest (absolute value) elements in each column of \mathbf{S}_t and setting the remaining ones to zero. Then, the second step of PI is replaced with

$$\mathbf{U}_t = \begin{cases} \text{QR}(\hat{\mathbf{S}}_t) & \text{if orthonormalization,} \\ \hat{\mathbf{S}}_t / \|\hat{\mathbf{S}}_t\|_2 & \text{if normalization.} \end{cases} \quad (14)$$

By definition, $\tau(\cdot, \cdot)$ in (13) is a very versatile operator that can be applied for several sparsity scenarios, e.g., row-sparse, elementwise-sparse, and local region-sparse. Accordingly, it makes OPIT adaptable and flexible in many tracking cases. More importantly, with this thresholding operation, we are able to establish the consistency of OPIT in high dimensions which is currently challenging the state-of-the-art subspace tracking methods, see Section V for details.

The OPIT algorithm introduces the window parameter W . Here, the inclusion of W is useful in some applications where we often collect multiple data samples instead of a single sample at each time t . The main steps of OPIT are summarized in Algorithm 1.

Complexity: For convenience of analysis, we suppose the window length $W = 1$. Most of the steps in OPIT require a computational complexity of $\mathcal{O}(nr^2)$ except the thresholding operator which costs $\mathcal{O}(nr + rk \log k)$ operations. Thus, the overall computational complexity of OPIT is $\mathcal{O}(\max\{nr, k \log k\}r)$. In terms of memory storage, OPIT operates in a recursive manner and does not need to go back over past observations. Rather, it effectively utilizes the information from previous data without the need to revisit them. Hence, the proposed algorithm requires a space of nr elements for saving the estimate \mathbf{U}_t , while two buffer matrices \mathbf{S}_t and \mathbf{E}_t need only $nr + r^2$ elements in total. In conclusion, the space complexity of OPIT is linear in the data dimension n .

Algorithm 2: OPITd - OPIT with Deflation

INPUT: $\{\mathbf{x}_t\}_{t=1}^T$, $\mathbf{x}_t \in \mathbb{R}^{n \times 1}$, target rank r , a forgetting factor $0 < \beta \leq 1$, and a thresholding factor k

$$k = \begin{cases} \lfloor (1 - \omega_{\text{sparse}})n \rfloor & \text{if } \omega_{\text{sparse}} \text{ is given,} \\ \lfloor 10r \log n \rfloor & \text{if } \omega_{\text{sparse}} \text{ is unknown,} \end{cases}$$

where ω_{sparse} is the sparsity level of the sparse basis (i.e., the percentage of zero-valued elements in the basis matrix).

INITIAL: Any $\mathbf{U}_0 \in \mathbb{R}^{n \times r}$, $\mathbf{S}_0 = \mathbf{0}_{n \times r}$, $\mathbf{e}_0 = \mathbf{1}_{r \times 1}$.

// Denote $\mathbf{u}_{t,j} = \mathbf{U}_t(:, j)$, $\mathbf{s}_{t,j} = \mathbf{S}_t(:, j)$, and $e_{t,j} = e_t(j)$.

MAIN PROGRAM:

PROCEDURE	Complexity
for $t = 1, 2, \dots, T$ do	
for $j = 1, 2, \dots, r$ do	
$z_{t,j} = \mathbf{u}_{t-1,j}^\top \mathbf{x}_t$	$\mathcal{O}(n)$
$\mathbf{s}_{t,j} = \beta e_{t-1,j} \mathbf{s}_{t-1,j} + z_{t,j} \mathbf{x}_t$	$\mathcal{O}(n)$
$\hat{\mathbf{s}}_{t,j} = \tau(\mathbf{s}_{t,j}, k)$ // thresholding	$\mathcal{O}(n + k \log k)$
$\mathbf{u}_{t,j} = \hat{\mathbf{s}}_{t,j} / \ \hat{\mathbf{s}}_{t,j}\ _2$	$\mathcal{O}(n)$
$e_{t,j} = \mathbf{u}_{t-1,j}^\top \mathbf{u}_{t,j}$	$\mathcal{O}(n)$
$\mathbf{x}_t = \mathbf{x}_t - \mathbf{u}_{t,j} \mathbf{u}_{t,j}^\top \mathbf{x}_t$ // deflation	$\mathcal{O}(n)$
end for	
end for	
OUTPUT: $\mathbf{U}_t \in \mathbb{R}^{n \times r}$	

B. OPIT with Deflation

A low cost subspace tracking algorithm with linear complexity of computation $\mathcal{O}(nr)$ is always preferable due to its fast implementation time, especially for real-time applications.⁴ Here, we derive a fast variant of OPIT using deflation called OPITd which can achieve linear complexity while preserving the algorithm's accuracy in most cases.

Our main motivation stems from the fact that if we apply the following projection deflation

$$\tilde{\mathbf{R}}_t = (\mathbf{I} - \mathbf{u}_1 \mathbf{u}_1^\top) \mathbf{R}_t (\mathbf{I} - \mathbf{u}_1 \mathbf{u}_1^\top), \quad (15)$$

where \mathbf{u}_1 is the most dominant eigenvector of \mathbf{R}_t , then the eigenvectors of $\tilde{\mathbf{R}}_t$ are exactly the same as \mathbf{R}_t with eigenvalues $\{0, \lambda_2, \dots, \lambda_{n-1}, \lambda_n\}$. Here, λ_i is the i^{th} strongest eigenvalue of \mathbf{R}_t . It means that the most dominant eigenvector of $\tilde{\mathbf{R}}_t$ is exactly the second most dominant eigenvector of \mathbf{R}_t . As a result, once we estimate \mathbf{u}_1 by using a specific (online) method, the second dominant eigenvector of \mathbf{R}_t can be extracted from $\tilde{\mathbf{R}}_t$ in the same way as \mathbf{u}_1 . Repeating this procedure r times results in r leading eigenvectors of \mathbf{R}_t . Interestingly, in the case even when \mathbf{u}_1 is not a true eigenvector of \mathbf{R}_t , the projection deflation (15) still retains desirable properties (e.g., positive semi-definiteness) that may be lost to other deflation transformations [56]. Accordingly, we employ the projection deflation (15) technique to OPIT in order to reduce its complexity.

To update the j^{th} column $\mathbf{u}_{t,j}$ of \mathbf{U}_t , for $j = 1, 2, \dots, r$, in sequential order, we replace the recursive rule (12) with⁵

$$\mathbf{s}_{t,j} = \beta e_{t-1,j} \mathbf{s}_{t-1,j} + z_{t,j} \mathbf{x}_t, \quad \text{where} \quad (16a)$$

$$z_{t,j} = \mathbf{u}_{t-1,j}^\top \mathbf{x}_t \quad \text{and} \quad e_{t-1,j} = \mathbf{u}_{t-2,j}^\top \mathbf{u}_{t-1,j}, \quad (16b)$$

⁴With respect to computational complexity, subspace tracking algorithms are categorized into three groups: high complexity $\mathcal{O}(n^2r)$ and $\mathcal{O}(n^2)$, moderate complexity $\mathcal{O}(nr^2)$, and low complexity $\mathcal{O}(nr)$. The last group, which is referred to as fast algorithms, is the most important class for online processing [1].

⁵Indeed, the update (16) corresponds to a rank-1 subspace version of (12).

where $\mathbf{s}_{t,j}$, $e_{t-1,j}$, and $z_{t,j}$ are equivalent to \mathbf{S}_t , \mathbf{E}_{t-1} , and \mathbf{z}_t in (12) in the rank-1 subspace setting. Next, the thresholding operation (13) becomes

$$\hat{\mathbf{s}}_{t,j} = \tau(\mathbf{s}_{t,j}, k). \quad (17)$$

Then, the column $\mathbf{u}_{t,j}$ is simply derived from normalizing (17) to unit length as $\mathbf{u}_{t,j} = \hat{\mathbf{s}}_{t,j} / \|\hat{\mathbf{s}}_{t,j}\|_2$. At the end of the column-wise update, we deflate the component $\mathbf{u}_{t,j}$ from \mathbf{x}_t as $\mathbf{x}_t \leftarrow \mathbf{x}_t - \mathbf{u}_{t,j} \mathbf{u}_{t,j}^\top \mathbf{x}_t$ for the estimation of the next component $\mathbf{u}_{t,j+1}$. The main steps of OPITd are summarized in Algorithm 2.

Complexity: The most expensive computation comes from the thresholding operation $\tau(\mathbf{s}_{t,j}, k)$ which requires a cost of $\mathcal{O}(n + k \log k)$. The remaining steps of OPITd require a computational complexity of $\mathcal{O}(n)$ only. Accordingly, OPITd costs a complexity of $\mathcal{O}(r \max\{n, k \log k\})$ for updating the whole matrix \mathbf{U}_t at each time t . In practice, we often set the value of k to $\mathcal{O}(r \log n)$ or $\lfloor (1 - \omega_{\text{sparse}})n \rfloor$ which might be much smaller than n , and thus, the overall complexity of OPITd is approximately linear to nr . OPITd also requires less memory storage than OPIT. Specifically, its space complexity is $2nr + r$ for storing \mathbf{U}_t , $\mathbf{S}_t = [\mathbf{s}_{t,1}, \mathbf{s}_{t,2}, \dots, \mathbf{s}_{t,r}]$ of size $n \times r$ and $\mathbf{e}_t = [e_{t,1}, e_{t,2}, \dots, e_{t,r}]^\top$ of size $r \times 1$ at time t .

C. Discussions

1) *Orthogonality and Sparsity:* First, it is worth noting that both OPIT and OPITd cannot enforce orthogonality and strong sparsity in the estimate at the same time. On the one hand, when the orthonormalization step using QR factorization is applied, OPIT ensures orthogonality but reduces sparsity. While the QR step improves the numerical stability of OPIT, it affects sparsity, particularly when the target rank r is large. In most cases, the Q-factor of the thresholded $\hat{\mathbf{S}}_t$ is a dense (orthogonal) matrix. However, if the columns of \mathbf{S}_t exhibit high sparsity, with mostly non-zero elements appearing in non-overlapping sets within its row support, \mathbf{S}_t tends to be nearly orthogonal, resulting in a sparse Q-factor. This scenario occurs when dealing with high-dimensional data streams characterized by a low rank (i.e., $r \ll n$) and/or an extremely high sparsity level ω_{sparse} .

On the other hand, when the normalization step (e.g., $\mathbf{U}_t = \hat{\mathbf{S}}_t / \|\hat{\mathbf{S}}_t\|_2$) is computed instead of the QR step, OPIT yields a sparse but non-orthogonal matrix \mathbf{U}_t . This operation has a complexity of $\mathcal{O}(nr)$, while the QR step costs $\mathcal{O}(nr^2)$, making it more computationally efficient, particularly when the rank r is reasonably high as compared to the data dimension n . Importantly, with this simple normalization, OPIT achieves excellent subspace estimation accuracy as compared to state-of-the-art SST algorithms; please see Figs. 5 and 6 for examples.

OPITd promotes sparsity; however, it may lead to a loss of orthogonality among the estimated components. The deflation used in OPITd enables efficient column-wise updates for tracking the underlying subspace and successfully achieving sparse columns of \mathbf{U}_t . This deflation approach has the advantage to estimate the principal components while the matrix \mathbf{U}_t in OPIT can be any basis of the underlying subspace. Consequently, OPITd has benefits in some applications such as data

whitening. By combining the thresholding operation $\tau(\mathbf{s}_{t,j}, k)$ and the column normalization, OPITd directly produces sparse components in the estimate \mathbf{U}_t at each time step. However, the deflation process in OPITd can cause a loss of orthogonality and induces artifacts, which can adversely affect the subsequent estimation of the next principal component [57]. In scenarios where the target rank r is large, both the convergence rate and estimation accuracy of OPITd are less than those of OPIT, as illustrated in Fig. 9(b). In such cases, an effective solution is to re-orthonormalize \mathbf{U}_t after a period of time. This approach not only addresses the issue but also enhances the numerical stability of OPITd.

2) *Parameter Selection*: Now, let's discuss how to choose the value of k . Ideally, this factor should be an $r \times 1$ vector $[k_1, k_2, \dots, k_r]$, where k_j represents the threshold level for the j^{th} column of the ground truth \mathbf{A}_t . The value of k_j should ideally be close to the number of non-zero elements in $\mathbf{A}_t(:, j)$. In the case where the sparsity patterns of \mathbf{A}_t are (nearly) uniformly distributed, we can set $k \simeq k_j \simeq \lfloor (1 - \omega_{\text{sparse}})n \rfloor$ when we have prior knowledge of the sparsity level ω_{sparse} . This threshold remains useful in other scenarios, particularly in the HDLSS regime where the dimension n is significantly larger than the number of non-zero elements in each column of \mathbf{A}_t , i.e., ω_{sparse} is large. Consequently, the ratio k_i/n is small and close to k_j/n , where $i \neq j$, allowing us to assume $k/n \approx k_j/n \approx 1 - \omega_{\text{sparse}}$. Even if this choice does not precisely represent the true threshold level k_j , it still captures the most dominant elements in each column of \mathbf{A}_t . As the presence of a small error (caused by the choice of k) does not significantly affect the estimation accuracy of power methods [55], the performance of OPIT is still guaranteed. If the sparsity level information is not available, we can tune this factor through cross-validation or simply choose $k = \lfloor mr \log n \rfloor$ where m is a positive number (we can choose the value of m in the range $[1, 10]$ in practice).

The former approach is useful for batch sparse subspace estimation and sparse PCA [58]. However, it requires a validation set and can be inefficient for tracking problems as it involves multiple passes over the observations. The latter approach is straightforward and performs reasonably well in practice. It is based on rigorous evidence in [59]–[61] that sparse subspace/PCA algorithms can recover the sparse principal components in polynomial time when the expected number of non-zero elements in each component is at most $\mathcal{O}(\sqrt{T/\log n})$ where T is the number of data samples. As shown later in Section V, if T is on the order of $\mathcal{O}(\epsilon^{-2}n)$ where $\epsilon > 0$ is a predefined accuracy, OPIT is guaranteed to converge. Since $\log n < \sqrt{T/\log n}$ for large n and $T = \mathcal{O}(\epsilon^{-2}n)$, we can choose the thresholding factor $k = \mathcal{O}(\log n)$ to achieve this condition. A natural question arises here is whether the tracking ability of OPIT deteriorates when the number of selected elements is smaller than the actual number of non-zero elements in \mathbf{A}_t (e.g., it might occur due to low sparsity levels). Fortunately, even in such cases, when the number of observations is sufficiently large, OPIT still provides a good estimate of \mathbf{A}_t .

3) *Novelty and Originality*: Compared to state-of-the-art power-based subspace tracking methods, OPIT has several distinctive characteristics. While many power-based subspace trackers (such as, Oja-types, NP-types, and DPM-types) estimate the underlying subspace using the update rule $\mathbf{U}_t = \text{orthnorm}(\mathbf{U}_{t-1} + \eta_t \mathbf{x}_t \mathbf{z}_t^\top)$ with η_t as the step size and $\text{orthnorm}(\cdot)$ denoting an orthonormalization procedure [1], OPIT distinguishes itself by incorporating \mathbf{E}_{t-1} in (12), which greatly bolsters its tracking ability. The matrix \mathbf{E}_{t-1} , comprising the cosines of the principal angles between successive subspaces, acts as feedback during the tracking process. This inclusion improves the adaptation rate and stability of OPIT, particularly in nonstationary environments.

Approximated PI (API)-type subspace trackers, on the other hand, rely on the projection approximation $\mathbf{U}_t \simeq \mathbf{U}_{t-1} \Theta_t$, where Θ_t is nearly orthogonal and close to an identity matrix [23]. These trackers predict the current tracking performance error and use it for estimating the true subspace. Specifically, they follow the update rule $\mathbf{U}_t = \mathbf{U}_{t-1} \Theta_t + \mathbf{y}_t \mathbf{g}_t^\top \Theta_t$, where \mathbf{y}_t represents the complement (error) of projecting \mathbf{x}_t onto \mathbf{U}_{t-1} as defined in (10), \mathbf{g}_t is a gain vector, and $\Theta_t = (\mathbf{I}_r + \|\mathbf{y}_t\|^2 \mathbf{g}_t \mathbf{g}_t^\top)^{-1/2}$. However, API-type trackers can struggle when abrupt changes occur, such as impulsive noises, outliers, or data drift. In such cases, the error \mathbf{y}_t becomes very large and the state transition matrix Θ_t deviates significantly from the ideal, resulting in degraded estimation accuracy and convergence rate. See Section E.1 in our supplementary document for examples. By contrast, OPIT leverages the past tracking performance error (one time step delayed), which is independent of the current error \mathbf{y}_t . This property makes OPIT less sensitive to abrupt changes. Together with the hard-thresholding operator $\tau(\cdot, k)$ in (13), OPIT is distinct from other existing models in both its design and tracking approach. The tracking ability of OPIT is demonstrated through several experiments in Section VI, where the results indicate that OPIT outperforms state-of-the-art subspace trackers, including various power-based methods, in both classical and high-dimensional regimes.

V. CONVERGENCE ANALYSIS

In this section, we provide a convergence analysis for the proposed OPIT algorithm under the assumption that $\mathbf{A}_t = \mathbf{A}$ is unchanged over time and the forgetting factor $\beta = 1$.⁶

We make the following assumptions to facilitate our convergence analysis:

(A1) \mathbf{A} is chosen in the set $\mathcal{U} = \{\mathbf{U} \in \mathbb{U}_{n,r}, \|\mathbf{U}\|_2 = 1, \text{ and } \|\mathbf{U}(:, i)\|_0 = \|\mathbf{U}(:, j)\|_0 = n(1 - \omega_{\text{sparse}}) \forall i, j\}$, where $\mathbb{U}_{n,r}$ denotes the set of $n \times r$ well-conditioned matrices whose

⁶We limit our analysis in this work to a stationary case when $\mathbf{A}_t = \mathbf{A} \forall t$ and $\beta = 1$. Establishing the ϵ -relative-error approximation guarantee for OPIT in nonstationary environments is non-trivial as data samples do not share the same population. Specifically, finding a tight upper bound on the error matrix $\Delta \mathbf{C}_t$ – which plays a key role in establishing the two necessary conditions (21) and (22) as well as Lemmas 1 and 2 – is challenging. Instead of the normal sample covariance matrix (SCM), an exponential weighted variant of the SCM is applied here because of the forgetting factor $\beta < 1$. It would make the theoretical convergence analysis more complicated. We leave this challenge for future work.

condition number is bounded by a small constant close to one, independent of the data dimension n . Here, the parameter ω_{sparse} represents the sparsity level of \mathbf{A} and serves as prior information. In addition, \mathbf{A} is sparse enough in the sense that the number of non-zero entries in each column is at most $\sqrt{n/\log n}$.

(A2) Coefficients $\{\mathbf{w}_t\}_{t \geq 1}$ and noises $\{\mathbf{n}_t\}_{t \geq 1}$ are modeled as zero-mean random Gaussian vectors with covariance matrix $\sigma_w^2 \mathbf{I}_r$ and $\sigma_n^2 \mathbf{I}_n$, respectively and we also assume that $\sigma_w \geq \sigma_n$.

In (A1), the underlying subspace is assumed to be sparse in the sense of *column sparsity* defined by Vu *et al.* in [62].⁷ The set \mathcal{U} covers several supports such as row-sparse, elementwise-sparse, and local region-sparse. Besides, the unit-norm constraint of (A1) is a very mild condition as we can rescale \mathbf{A} by recasting its operator norm into the signal power. Also, (A1) ensures subspace trackers to estimate the sparse subspace with high probability [59]. Meanwhile, (A2) is a common assumption for subspace tracking problems and holds in many situations [39]. Together with (A1), they help prevent the ill-conditioned computation and support the perturbation analysis of the QR decomposition from the thresholding operation.

Denote by $\mathbf{S}_t = \mathbf{U}_{t,\mathcal{F}} \mathbf{R}_{t,\mathcal{F}}$ the QR decomposition of \mathbf{S}_t , where “ \mathcal{F} ” stands for the “full entries” of \mathbf{S}_t without thresholding or setting small entries to zeros. This notation is used to distinguish the QR representation of the original matrix \mathbf{S}_t from that of its thresholded version $\hat{\mathbf{S}}_t$. We provide Lemmas 1 and 2, which serve as crucial components in establishing the convergence of OPIT as stated in Theorem 1.

Lemma 1. Denote by $\mathbf{C}_t = t^{-1} \sum_{i=1}^t \mathbf{x}_i \mathbf{x}_i^\top$ the sample covariance matrix, \mathbf{C} the population covariance matrix, and let $\Delta \mathbf{C}_t = \mathbf{C}_t - \mathbf{C}$. We always have

$$\|\mathbf{A}_\perp^\top \mathbf{U}_{t,\mathcal{F}}\|_2 \leq \frac{\sigma_n^2 \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2 + \|\Delta \mathbf{C}_t\|_2}{\left(\left[(\sigma_w^2 + \sigma_n^2) \sqrt{1 - \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2^2} - \|\Delta \mathbf{C}_t\|_2 \right]^2 + \left[\sigma_n^2 \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2 + \|\Delta \mathbf{C}_t\|_2 \right]^2 \right)^{1/2}}. \quad (18)$$

Proof. See Section B in our supplementary document. \square

Lemma 2. The distance between \mathbf{U}_t and $\mathbf{U}_{t,\mathcal{F}}$ is bounded by

$$\begin{aligned} & \|\mathbf{U}_{t,\perp}^\top \mathbf{U}_{t,\mathcal{F}}\|_2 \\ & \leq \frac{\sqrt{r} (\sigma_n^2 \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2 + \|\Delta \mathbf{C}_t\|_2)}{\left((\sigma_w^2 + \sigma_n^2) \sqrt{1 - \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2^2} - (1 + \sqrt{r}(1 + \sqrt{2})) (\sigma_n^2 \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2 + \|\Delta \mathbf{C}_t\|_2) \right)}, \end{aligned} \quad (19)$$

⁷With respect to the concept of subspace sparsity, Vu *et al.* in [62] introduced two notions: *column sparsity* and *row sparsity*. Specifically, a subspace is said to be column sparse if some orthonormal basis contains sparse vectors. Meanwhile, every orthonormal basis of a row sparse subspace must consist of sparse vectors. Accordingly, row sparse subspaces also belong to the class of column sparse subspaces. In this work, OPIT can achieve an ϵ -relative-error approximation guarantee for the class of column sparse subspaces, and thus, its convergence guarantee also holds under the row sparsity.

under the following condition

$$\frac{\sigma_n^2 \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2 + \|\Delta \mathbf{C}_t\|_2}{(\sigma_w^2 + \sigma_n^2) \sqrt{1 - \|\mathbf{A}_\perp^\top \mathbf{U}_{t-1}\|_2^2}} \leq \frac{\sqrt{2} - 1}{\sqrt{r} - 1 + \sqrt{2}}. \quad (20)$$

Proof. See Section C in our supplementary document. \square

Given (A1), (A2), Lemmas 1 and 2, the main result of OPIT’s convergence can be stated by the following theorem:

Theorem 1. Given (A1)-(A2), consider the data model (1) with \mathbf{A} of size $n \times r$, the target rank $r \ll n$, and the true covariance matrix of form $\mathbf{C} = \sigma_w^2 \mathbf{A} \mathbf{A}^\top + \sigma_n^2 \mathbf{I}$. Assume that a block of W data samples is collected at each time t , the forgetting factor $\beta = 1$, and the thresholding factor $k = \lfloor (1 - \omega_{\text{sparse}})n \rfloor$. Let $\epsilon > 0$ be a predefined accuracy, $0 < \delta \ll 1$ be a predefined error probability, and C be a universal positive number. Suppose the initialization matrix \mathbf{U}_0 and the number of blocks of data samples T satisfy the following conditions

$$T \geq \frac{C \log(2/\delta)}{W \epsilon^2} \left(\sqrt{r} + \left(\frac{\sigma_n^2}{\sigma_w^2} + 2 \frac{\sigma_n}{\sigma_w} \right) \sqrt{n} \right)^2, \quad (21)$$

$$\max \{ \sin \theta(\mathbf{A}, \mathbf{U}_0), \epsilon \} \leq \left(\frac{3 - 2\sqrt{2}}{r + 2\sqrt{r}(\sqrt{2} - 1)} \right)^{1/2}. \quad (22)$$

Let \mathbf{U}_t be the weight matrix computed by OPIT with the orthonormalization step using QR factorization at time t . Then, $\forall t \geq T$ and with probability exceeding $1 - \delta$, we have

$$d_t \triangleq \sin \theta(\mathbf{A}, \mathbf{U}_t) \leq \epsilon. \quad (23)$$

Proof. First, we can express $\mathbf{U}_t = \mathbf{U}_{t,\mathcal{F}} \mathbf{W}_1 + \mathbf{U}_{t,\mathcal{F},\perp} \mathbf{W}_2$ where $\mathbf{U}_{t,\mathcal{F},\perp} \in \mathbb{R}^{n \times (n-r)}$ is the orthogonal complement of $\mathbf{U}_{t,\mathcal{F}}$ (i.e., $\mathbf{U}_{t,\mathcal{F}}^\top \mathbf{U}_{t,\mathcal{F},\perp} = \mathbf{0}$), $\mathbf{W}_1 \in \mathbb{R}^{r \times r}$ and $\mathbf{W}_2 \in \mathbb{R}^{(n-r) \times r}$ are coefficient matrices; see Section F in our supplementary document for its derivation. Specifically, it is easy to obtain that $\|\mathbf{W}_1\|_2 = \|\mathbf{U}_{t,\mathcal{F}}^\top \mathbf{U}_t\|_2$ and $\|\mathbf{W}_2\|_2 = \|\mathbf{U}_{t,\mathcal{F},\perp}^\top \mathbf{U}_t\|_2$. Accordingly, we can bound the distance $d_t = \sin \theta(\mathbf{A}, \mathbf{U}_t)$ as follows:

$$\begin{aligned} d_t &= \|\mathbf{A}_\perp^\top \mathbf{U}_t\|_2 = \|\mathbf{A}_\perp^\top (\mathbf{U}_{t,\mathcal{F}} \mathbf{W}_1 + \mathbf{U}_{t,\mathcal{F},\perp} \mathbf{W}_2)\|_2 \\ &\stackrel{(i)}{\leq} \|\mathbf{A}_\perp^\top \mathbf{U}_{t,\mathcal{F}}\|_2 \|\mathbf{W}_1\|_2 + \|\mathbf{A}_\perp^\top \mathbf{U}_{t,\mathcal{F},\perp}\|_2 \|\mathbf{W}_2\|_2 \\ &\stackrel{(ii)}{\leq} \|\mathbf{A}_\perp^\top \mathbf{U}_{t,\mathcal{F}}\|_2 + \|\mathbf{U}_{t,\perp}^\top \mathbf{U}_{t,\mathcal{F}}\|_2. \end{aligned} \quad (24)$$

Here, (i) is due to the triangle and matrix norm inequalities $\|\mathbf{M} + \mathbf{N}\|_2 \leq \|\mathbf{M}\|_2 + \|\mathbf{N}\|_2$ and $\|\mathbf{M}\mathbf{N}\|_2 \leq \|\mathbf{M}\|_2 \|\mathbf{N}\|_2$ for all matrices \mathbf{M} and \mathbf{N} ; and (ii) is due to the following facts: $\|\mathbf{A}_\perp\|_2 = \|\mathbf{U}_t\|_2 = \|\mathbf{U}_{t,\mathcal{F},\perp}\|_2 = 1$, $\|\mathbf{W}_1\|_2 \leq \|\mathbf{U}_{t,\mathcal{F}}^\top \mathbf{U}_t\|_2 \leq 1$, $\|\mathbf{A}_\perp^\top \mathbf{U}_{t,\mathcal{F},\perp}\|_2 \leq \|\mathbf{A}_\perp^\top\|_2 \|\mathbf{U}_{t,\mathcal{F},\perp}\|_2 \leq 1$, and $\|\mathbf{U}_{t,\mathcal{F},\perp}^\top \mathbf{U}_t\|_2 = \|\mathbf{U}_{t,\perp}^\top \mathbf{U}_{t,\mathcal{F}}\|_2$.

The two terms of the right hand side of (24) can be bounded by Lemma 1 and 2, respectively. Next, Lemma 3 indicates an upper bound on $\|\Delta \mathbf{C}_t\|_2$ which plays a crucial role in Lemma 1 and 2 as well as establishing the two conditions (21) and (22) for the convergence of OPIT.

Lemma 3. The error matrix $\Delta \mathbf{C}_t$ is bounded in the operator norm with a probability at least $1 - \delta$:

$$\|\Delta \mathbf{C}_t\|_2 \leq c_\delta \left(\sigma_w^2 \sqrt{\frac{r}{tW}} + (2\sigma_n \sigma_x + \sigma_n^2) \sqrt{\frac{n}{tW}} \right), \quad (25)$$

where $\delta > 0$ is a predefined error probability, and $c_\delta = C\sqrt{\log(2/\delta)}$ with a universal positive number $C > 0$.

Proof. See Section D in our supplementary document. \square

Then, the necessary condition (20) for Lemma 2 is particularly satisfied when (21) is met and the inequality holds

$$\max\{\sin\theta(\mathbf{A}, \mathbf{U}_0), \epsilon\} \leq \sqrt{\frac{\alpha(r, \rho)}{1 - \alpha(r, \rho)}}, \quad \text{where} \quad (26)$$

$$\alpha(r, \rho) = \frac{(3 - 2\sqrt{2})(\sigma_w^2 + \sigma_n^2)^2}{(r + 2\sqrt{r}(\sqrt{2} - 1) + 3 - 2\sqrt{2})(\sigma_n^2 + r^{-1}\rho\sigma_w^2)^2},$$

for any positive number ρ in the range $(0, r]$, please see Sec. D in our supplementary document for details. Clearly, (22) provides a lower bound on $\sqrt{\alpha(r, \rho)/(1 - \alpha(r, \rho))}$.

Accordingly, Lemma 2 is achieved under the two conditions (21) and (22) while Lemma 1 holds for all t . Now, given Lemma 1, 2, and 3, d_t can be bounded by Lemma 4.

Lemma 4. Let $d_0 = \sin\theta(\mathbf{A}, \mathbf{U}_0)$, $\omega_0 = \max\{d_0, \epsilon\}$, $\gamma > 0$ is any positive number satisfying $\omega_0 \leq \gamma r \sqrt{1 - \omega_0^2}$ and $\rho\gamma < 1$. Suppose that $\omega_0 \leq \sqrt{2}/2$, the two conditions (21) and (22) are met, we obtain

$$d_t \leq \frac{r\sigma_n^2 + \rho\sigma_w^2}{r\xi\sqrt{1 - \omega_0^2}} \max\{d_{t-1}, \epsilon\}, \quad (27)$$

where

$$\xi = 0.5 \max\left\{[(1 + \gamma^2 r^2)\sigma_n^4 + (1 - \rho\gamma)^2\sigma_w^4] + 2(1 + \gamma^2 r^2 - \rho\gamma)\sigma_n^2\sigma_w^2\right\}^{1/2}, (\sigma_n^2 + \sigma_w^2)(1 - \varrho)/\sqrt{r}, \quad (28)$$

with $\varrho = \gamma(1 + \sqrt{r}(1 + \sqrt{2}))(r\sigma_n^2 + \rho\sigma_w^2)(\sigma_n^2 + \sigma_w^2)^{-1}$. Furthermore, $d_t \leq \epsilon$ also holds when t satisfies the condition (21).

Proof. See Section E in our supplementary document. \square

Remark 1. The presence of $1/\epsilon^2$ in the sample bound (21) introduces a slight deviation from the original HDLSS regime in the batch setting. However, this holds more relevance and value for the problem of (sparse) subspace tracking. It provides a lower bound on the number of data samples required to achieve a desired (predefined) estimation accuracy. Moreover, this bound exhibits a close connection to the original HDLSS regime, where both the data dimension n and data size T are huge and comparable. Specifically, from (21), we deduce the following bound

$$\frac{n}{T} \leq \frac{W\epsilon^2}{C\log(2/\delta)} \left(\frac{\sigma_n^2}{\sigma_w^2} + 2\frac{\sigma_n}{\sigma_w}\right)^{-2}. \quad (29)$$

Here, the target rank r can be diminished as $\sqrt{r}/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$ and $r \ll n$. The right-hand side of (29) is greater than 0, a difference from the classical setting where $n/T \rightarrow 0$ as $T \rightarrow \infty$.

Remarkably, the sample bound (21) is independent of the subspace sparsity level ω_{sparse} of \mathbf{A} . In our analysis, this bound is established through the error bound on $\|\Delta\mathbf{C}_t\|_2$, quantifying the closeness of the sample covariance matrix to the true one. Beyond factors such as data dimension, rank, sample size,

and noise, this error bound relies on the spectral norm of the subspace basis \mathbf{A} rather than its sparsity. Therefore, the resulting sample bound (21) is independent of ω_{sparse} and the thresholding factor k . Accordingly, (21) can be applied to any case of subspace sparsity. Also, conditions stated in Theorem 1 are mainly needed for the convergence analysis (sufficient but not necessary conditions). In simulations, OPIT has proven effective even when these conditions are not fully met, see Section VI for details.

VI. EXPERIMENTS

In this section, we conduct several experiments on both synthetic and real data to demonstrate the effectiveness and efficiency of OPIT and its variant OPITd. Their performance are evaluated in comparison with state-of-the-art algorithms. Our simulations are implemented using MATLAB on a laptop of Intel core *i7* and 16GB of RAM. Our codes are also available online at <https://github.com/thanhtbt/sst/> to facilitate replicability and reproducibility.

A. Experiments with Synthetic Data

1) *Experiment Setup:* Following the formulation in section III, data samples $\{\mathbf{x}_t\}_{t \geq 1}$ are generated at random under the standard model:

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{w}_t + \sigma_n \mathbf{n}_t, \quad (30)$$

where $\mathbf{n}_t \in \mathbb{R}^n$ is a noise vector derived from $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, $\sigma_n > 0$ is to control the effect of the noise on algorithm's performance, $\mathbf{w}_t \in \mathbb{R}^r$ is an i.i.d. Gaussian random vector of zero-mean and unit-variance to represent the subspace coefficient. The sparse mixing matrix $\mathbf{A}_t \in \mathbb{R}^{n \times r}$ at time t is simulated as

$$\mathbf{A}_t = \mathbf{\Omega} \otimes (\mathbf{A}_{t-1} + \varsigma \mathbf{N}_t), \quad (31)$$

where \otimes denotes the Hadamard product, $\mathbf{\Omega} \in \mathbb{R}^{n \times r}$ is a Bernoulli random matrix with probability $1 - \omega_{\text{sparse}}$, \mathbf{N}_t is a normalized Gaussian white noise matrix, and $\varsigma > 0$ is the time-varying factor aimed to control the subspace variation with time. At $t = 0$, \mathbf{A}_0 is initialized as a random Gaussian matrix with zero-mean and unit-variance entries.

In order to evaluate the subspace estimation performance, we measure the following distance between two subspaces⁸

$$d_t \triangleq \sin\theta(\mathbf{A}_t, \mathbf{U}_t), \quad (32)$$

where \mathbf{U}_t refers to the estimated subspace at time t .

2) *Effect of the forgetting factor β :* The choice of the forgetting factor β plays an essential role in the tracking ability of OPIT. We investigated its effect by varying its value from 0.1 to 1 and then evaluating the performance of OPIT. Here, the data dimension, the true rank, the number of data samples were set at $n = 50$, $r = 10$, and $T = 1000$, respectively. We fixed

⁸Given two matrices \mathbf{A}_t and \mathbf{U}_t of the same size, we always have $\sin\theta(\mathbf{A}_t, \mathbf{U}_t) = \sin\theta(\text{orth}(\mathbf{A}_t), \text{orth}(\mathbf{U}_t))$ where $\text{orth}(\mathbf{M})$ returns an orthonormal basis for the range of the matrix \mathbf{M} . Let $\underline{\mathbf{A}}_t = \text{orth}(\mathbf{A}_t)$ and $\underline{\mathbf{U}}_t = \text{orth}(\mathbf{U}_t)$. We can compute $d_t = \sin\theta(\mathbf{A}_t, \mathbf{U}_t)$ in (32) as follows: $d_t = \sin\theta(\underline{\mathbf{A}}_t, \underline{\mathbf{U}}_t) = \|\underline{\mathbf{A}}_{t,\perp}^T \underline{\mathbf{U}}_t\|_2 = \|\underline{\mathbf{U}}_{t,\perp}^T \underline{\mathbf{A}}_t\|_2 = \|\underline{\mathbf{A}}_t \underline{\mathbf{A}}_t^T - \underline{\mathbf{U}}_t \underline{\mathbf{U}}_t^T\|_2$ where $(\cdot)_\perp$ denotes the orthogonal complement. In MATLAB, this distance can be easily calculated by using the command $\sin(\text{subspace}(\mathbf{A}_t, \mathbf{U}_t))$.

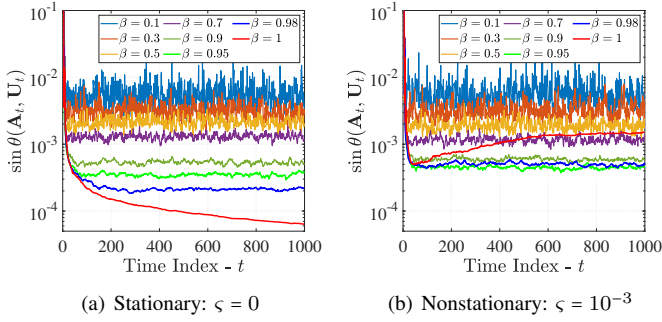


Fig. 1: Effect of the forgetting factor β .

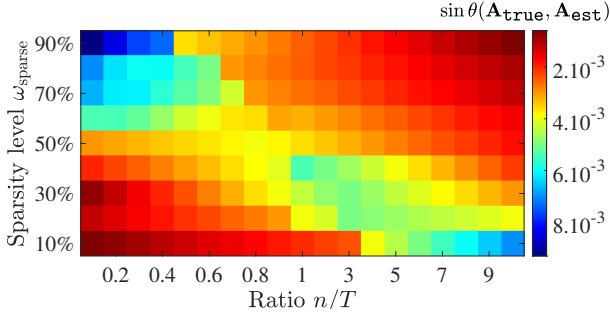


Fig. 2: Effect of the sparsity level ω_{sparse} .

the sparsity level and the noise factor at $\omega_{\text{sparse}} = 50\%$ and $\sigma_n = 10^{-3}$, respectively. Two time-varying levels were considered, namely $\zeta = 0$ (stationary) and $\zeta = 10^{-3}$ (nonstationary). Results are illustrated in Fig. 1. In the stationary environment (Fig. 1(a)), we can see that the higher the value of β is, the better the performance OPIT achieves, and $\beta = 1$ offers the best tracking performance. In the time-varying environment (Fig. 1(b)), $0 \ll \beta < 1$ can provide reasonably high subspace estimation accuracy. When β is close to 0, OPIT can track the underlying subspace over time but its accuracy is low. When $\beta = 1$, OPIT's performance degrades as time passes.

3) *Effect of the sparsity level ω_{sparse}* : In order to assess the influence of the sparsity level on the performance of OPIT, we varied the value of ω_{sparse} from 10% to 90% and measured the accuracy of subspace estimation achieved by OPIT under different settings of the ratio n/T . Throughout the experiments, we fixed the true rank $r = 10$, the noise level $\sigma_n = 10^{-3}$, the time-varying factor $\zeta = 0$, and the number of data samples $T = 1000$. The data dimension n was selected from the set $\{100, 200, \dots, 1000, 2000, \dots, 9000, 10000\}$ corresponding to the ratio n/T of $\{0.1, 0.2, \dots, 1, 2, \dots, 9, 10\}$. The experimental results, depicted in Fig. 2, indicate that OPIT consistently achieved good subspace estimation with $\sin \theta(\mathbf{A}_{\text{true}}, \mathbf{A}_{\text{est}}) \leq 10^{-2}$ across all scenarios. Notably, for smaller data dimensions, OPIT yielded the best results when sparsity was low. However, as the dimension n increased, higher sparsity levels led to even better subspace estimation performance by OPIT.

4) *OPIT in Noisy and Dynamic Environments*: To demonstrate the tracking ability of OPIT in nonstationary environments, we varied the value of the noise level σ_n and the time-varying factor ζ among $\{10^{-1}, 10^{-2}, 10^{-3}\}$ and then evaluated its subspace estimation accuracy. Two case studies

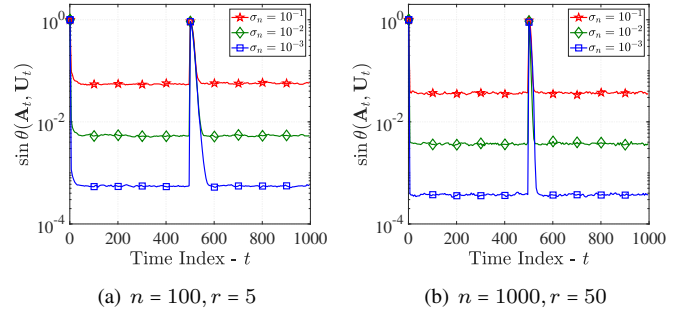


Fig. 3: Effect of the noise level σ_n on performance of OPIT: sparsity level $\omega_{\text{sparse}} = 90\%$, time-varying factor $\zeta = 10^{-4}$, and forgetting factor $\beta = 0.9$.

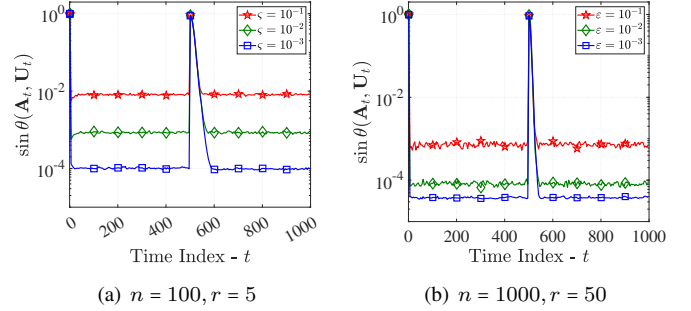


Fig. 4: Effect of the time-varying factor ζ on performance of OPIT: sparsity level $\omega_{\text{sparse}} = 90\%$, noise level $\sigma = 10^{-4}$, and forgetting factor $\beta = 0.9$.

were considered, including the small-scale $\{n = 100, r = 5\}$ and the large-scale $\{n = 1000, r = 50\}$ in which the sparsity level ω_{sparse} was set to 90% and an abrupt change was created at $t = 500$. The forgetting factor β was fixed at 0.9 in both cases. We set the value of the thresholding factor k to $\lceil 10r \log n \rceil$.

Fig. 3 and Fig. 4 illustrate the effect of the noise level σ_n and the time-varying factor ζ on the performance of OPIT, respectively. We can see that the value of σ_n and ζ did not affect the convergence rate of OPIT but its estimation error. Despite the value of σ_n and ζ , OPIT still tracked successfully the underlying sparse subspace even in the presence of a significant change at $t = 500$. The lower σ_n and ζ are, the better subspace estimation accuracy OPIT can achieve. Moreover, these experimental results indicate that the dimension n and rank r had in fact a small impact on how fast OPIT converges in dynamic environments. Specifically, when dealing with the large-scale setting, its convergence rate was faster than that when handling the small-scale one.

5) *OPIT versus Other Subspace Trackers*: Here, performance of OPIT is compared with the state-of-the-art subspace trackers in different scenarios. The considered algorithms are OPAST [63], FAPI [23], LORAF [64], GYAST [65], L1-PAST [24], SS-FAPI [21], SSPCA [28], and Oja [30]. Except SSPCA and Oja, we kept the default hyperparameters for the remaining subspace trackers. SSPCA relies on a sparsity parameter and a window length which were set to $\gamma = \lceil n(1 - \omega_{\text{sparse}}) \rceil$ and $W = \lceil \log n \rceil$, respectively. For Oja, we adopted the learning rate of $\eta_t = 1/t$ in our experiments.

We used 1000 snapshots derived from the model (30) in

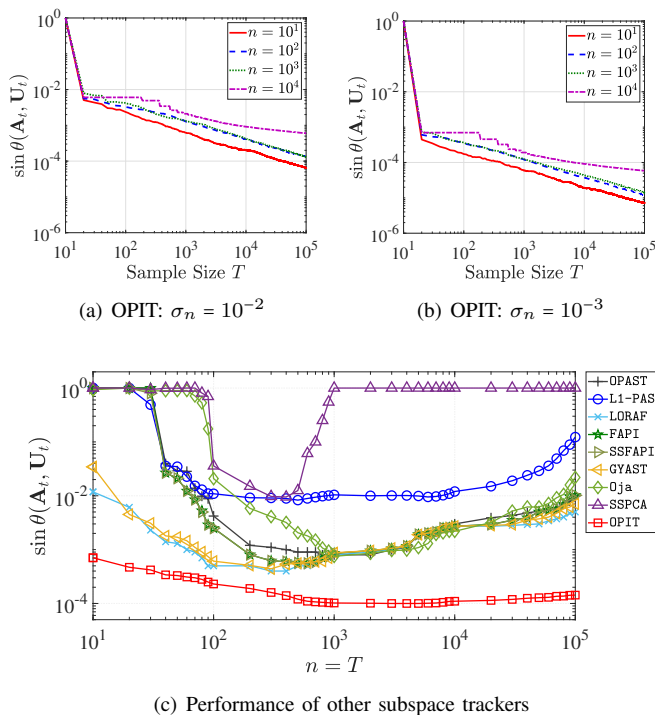


Fig. 7: Effect of the data dimension and sample size $\{n, T\}$.

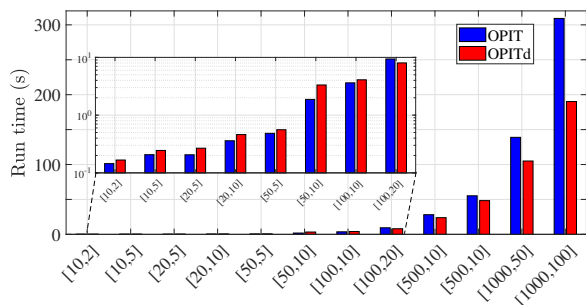


Fig. 8: OPITd versus OPIT: Run time.

of snapshots T , including the sparsity level $\omega_{\text{sparse}} = 90\%$, the noise level $\sigma_n = 10^{-3}$, the time-varying factor $\zeta = 10^{-3}$, and the forgetting factor $\beta = 0.97$. We used 3000 snapshots instead of 1000 for this task. The experimental results in Fig. 8 show that OPITd was faster than OPIT when the dimension n and the target rank r were set to large values ($n \geq 100$ and $r \geq 10$), especially when the dimension n is actually high, e.g., $n = 1000$.

We next investigate the tracking ability of OPITd in time-varying environments with abrupt changes. We reused the experiment setup above and created two abrupt changes at $t = 1000$ and $t = 2000$ to evaluate how fast OPITd converges. The noise level was set at $\sigma_n = 10^{-3}$. The results are illustrated in Fig. 9 and Fig. 10. When the underlying model was of low rank, OPITd had almost the same performance to OPIT, see Fig. 9(a). When the target rank r was large, OPITd did not work well, probably because the projection deflation might lead to a cumulative error between successive estimates. However, if the value of r is not too large, OPITd could track successfully the underlying subspace over time when the

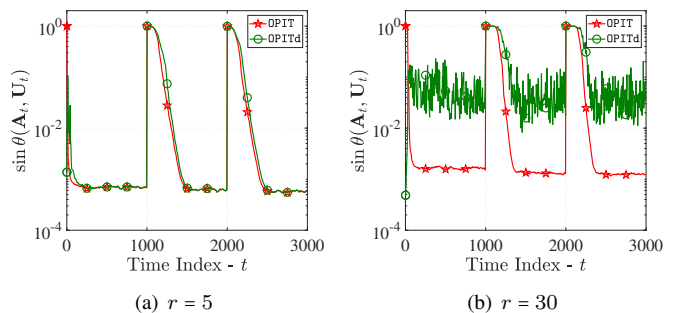


Fig. 9: Effect of the target rank r on performance of OPITd: dimension $n = 100$, snapshots $T = 3000$, time-varying factor $\zeta = 10^{-3}$, noise level $\sigma_n = 10^{-3}$, sparsity level $\omega_{\text{sparse}} = 90\%$, forgetting factor $\beta = 0.97$, and two abrupt changes at $t = 1000$ and $t = 2000$.

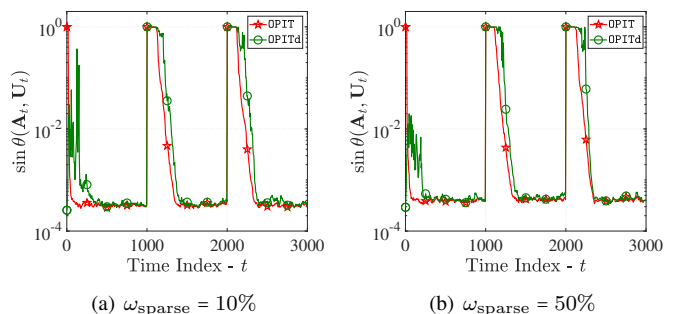


Fig. 10: Effect of the sparsity level ω_{sparse} on performance of OPITd: dimension $n = 100$, rank $r = 20$, snapshots $T = 3000$, time-varying factor $\zeta = 10^{-3}$, noise level $\sigma_n = 10^{-3}$, forgetting factor $\beta = 0.97$, and two abrupt changes at $t = 1000$ and $t = 2000$.

sparsity level ω_{sparse} was not too high, as shown in Fig. 10.

B. Experiments with Real Video Data

In this task, four different video sequences are used to illustrate the effectiveness and efficiency of OPIT for real data, including ‘‘Lobby’’, ‘‘Hall’’, ‘‘Highway’’, and ‘‘Park’’ whose details are reported in Tab. III, (see Fig. 11 for an illustration).⁹ The Lobby video comprises 1546 images, each with dimensions of 128×160 pixels. These images were captured within an office lobby, highlighting the background changes caused by the switching on and off of lights. The Hall video comprises 3584 images with dimensions of 176×144 pixels, taken in an airport hall. This video depicts a bustling airport hall with numerous people entering and exiting the premises. The Highway video consists of 1700 traffic images, where each frame has 240×320 pixels. It captures vehicles traveling on a two-lane highway, approaching the camera. Lastly, the Park video contains 600 frames of size 288×362 , capturing thermal images of moving objects within a park.

We here compared the video background tracking ability of OPIT with the state-of-the-art subspace tracking algorithms (i.e., ℓ_1 -PAST, SS-FAP1, and PETRELS-ADMM [39]) and

⁹Video sequences: <http://jacarini.dinf.usherbrooke.ca>.

Dataset		“Lobby”		“Hall”		“Highway”		“Park”	
Size	Tensor-based	128 × 160 × 1546		174 × 144 × 3584		320 × 240 × 1700		288 × 352 × 600	
	Matrix-based	20480 × 1546		25056 × 3584		76800 × 1700		101376 × 600	
Evaluation metrics		time(s)	error	time(s)	error	time(s)	error	time(s)	error
Tensor	SOAP	14.29	0.842	21.72	0.989	39.89	0.821	21.34	0.789
	OLCP	10.50	0.161	19.98	0.154	27.07	0.219	14.19	0.096
	OLSTEC	44.25	0.037	92.82	0.041	130.1	0.064	53.13	0.032
	ROLCP	4.32	0.114	10.74	0.120	11.45	0.154	4.47	0.086
Subspace	PETRELS-ADMM	118.4	0.015	305.5	0.018	452.6	0.009	203.6	0.032
	ℓ_1 -PAST	14.11	0.031	33.73	0.101	46.78	0.159	19.21	0.058
	SS-FAPI	12.99	0.023	32.72	0.100	46.37	0.160	17.56	0.056
	OPIT ($W = 1$)	16.32	0.013	50.78	0.056	56.78	0.102	26.94	0.042
	OPIT ($W = \lfloor \log(IJ) \rfloor$)	1.89	0.021	5.62	0.086	6.05	0.141	2.83	0.057

TABLE III: Runtime and averaged relative error of adaptive algorithms on tracking the four video sequences.

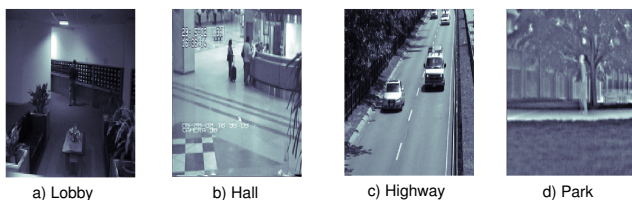


Fig. 11: Four video sequences used in this paper.

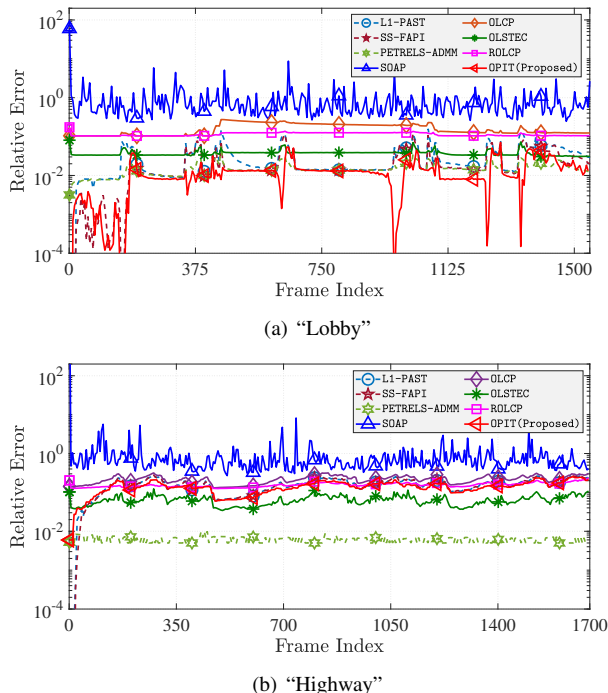


Fig. 12: Tracking ability of algorithms on the video datasets.

tensor tracking algorithms (i.e., SOAP [66], OLCP [67], OLSTEC [68], and ROLCP [69]). In order to apply these subspace tracking algorithms to the video sequences, each video frame of size $I \times J$ was reshaped into a $IJ \times 1$ vector. Following the

studies on video tracking in [39] and [69], the tensor rank and subspace rank were set to 10 for all simulations.

In this task, our main goal is to estimate the background of a given video sequence by leveraging the assumption that video frames can be expressed as a combination of a low-rank background and a sparse foreground component. To accomplish this, we first apply subspace/tensor trackers to estimate the low-rank subspace/tensor from the video frames over time. Once the subspace is estimated/tracked at each time step, we projected the video frame onto this subspace, thereby obtaining the corresponding subspace coefficient and subsequently the low-rank component/vector. This vector is then reshaped into a matrix of the same size as the video frame, which we refer to as the video background. In the case of the tensor-based approach, the video background is directly recovered by utilizing the tensor product of the tracked loading tensor factors.

In order to evaluate the estimation performance, we use the following relative error metric

$$\text{Relative_Error}(t) = \frac{\|\mathbf{X}_{\text{true}}(t) - \mathbf{X}_{\text{est}}(t)\|_F^2}{\|\mathbf{X}_{\text{true}}(t)\|_F^2}, \quad (33)$$

where $\mathbf{X}_{\text{true}}(t)$ is the ground truth and $\mathbf{X}_{\text{est}}(t)$ is the estimation of the video background at each time t .

Simulation results are shown in Tab. III and Fig. 12. As can be seen that OPIT provided a competitive estimation accuracy as compared to PETRELS-ADMM while its runtime was much faster than that of the ADMM-based tracking algorithm. Indeed, OPIT had a better performance than PETRELS-ADMM on the “Lobby” data, see Fig. 12(a). Also, OPIT outperformed most tracking algorithms, apart from PETRELS-ADMM. With respect to runtime, ROLCP was the fastest “one-pass” tracking algorithm, several times faster than the second-best. Interestingly, our algorithm is also designed for handling a block of multiple incoming samples at each time (i.e., the length of window $W > 1$). When $W = \lfloor \log(IJ) \rfloor$, OPIT was even faster than ROLCP while still retaining a reasonable video tracking accuracy.

VII. CONCLUSIONS

In this paper, we have proposed a new provable OPIT algorithm which is fully capable of tracking the sparse principal subspace over time in both classical regime and high-dimension, low-sample-size regime. OPIT provides a competitive performance in terms of both subspace estimation accuracy and convergence rate in the classical regime, especially when the SNR level is high. In high dimensions, OPIT outperforms other sparse subspace tracking algorithms, its estimation accuracy is much better than that of the second-best. Besides, a fast variant of OPIT has been obtained using deflation called OPITd. Its computational complexity and memory storage are linear to the input size and they are lower than that of OPIT. Simulations carried out on real video sequences indicated that the proposed method has potential for real applications.

REFERENCES

- [1] J. P. Delmas, "Subspace tracking for signal processing," *Adaptive Signal Process.: Next Generation Solutions*, pp. 211–270, 2010.
- [2] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, 2018.
- [3] N. Vaswani and P. Narayanamurthy, "Static and dynamic robust PCA and matrix completion: A review," *Proc. IEEE*, vol. 106, no. 8, pp. 1359–1379, 2018.
- [4] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking algorithms in signal processing: A brief survey," *REV J. Electr. Commun.*, vol. 11, no. 1–2, pp. 16–25, 2021.
- [5] N. El Karoui, "Spectrum estimation for large dimensional covariance matrices using random matrix theory," *Ann. Stat.*, vol. 36, no. 6, pp. 2757–2790, 2008.
- [6] X. Mestre, "On the asymptotic behavior of the sample estimates of eigenvalues and eigenvectors of covariance matrices," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5353–5368, 2008.
- [7] R. Vershynin, "How close is the sample covariance matrix to the actual covariance matrix?" *J. Theor. Probab.*, vol. 25, no. 3, pp. 655–686, 2012.
- [8] N. El Karoui, "Operator norm consistent estimation of large-dimensional sparse covariance matrices," *Ann. Stat.*, vol. 36, no. 6, pp. 2717–2756, 2008.
- [9] P. J. Bickel and E. Levina, "Covariance regularization by thresholding," *Ann. Stat.*, vol. 36, no. 6, pp. 2577–2604, 2008.
- [10] I. M. Johnstone and A. Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," *J. Am. Stat. Assoc.*, vol. 104, no. 486, pp. 682–693, 2009.
- [11] D. Shen, H. Shen, and J. S. Marron, "Consistency of sparse PCA in high dimension, low sample size contexts," *J. Mult. Anal.*, vol. 115, pp. 317–333, 2013.
- [12] A. A. Amini and M. J. Wainwright, "High-dimensional analysis of semidefinite relaxations for sparse principal components," *Ann. Stat.*, vol. 37, no. 5B, p. 2877–2921, 2009.
- [13] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, no. 2, 2010.
- [14] Z. Ma, "Sparse principal component analysis and iterative thresholding," *Ann. Stat.*, vol. 41, no. 2, pp. 772–801, 2013.
- [15] T. T. Cai, Z. Ma, and Y. Wu, "Sparse PCA: Optimal rates and adaptive estimation," *Ann. Stat.*, vol. 41, no. 6, pp. 3074–3110, 2013.
- [16] T. T. Cai, Z. Ren, and H. H. Zhou, "Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation," *Electr. J. Stat.*, vol. 10, no. 1, pp. 1–59, 2016.
- [17] H. Zou and L. Xue, "A selective overview of sparse principal component analysis," *Proc. IEEE*, vol. 106, no. 8, pp. 1311–1320, 2018.
- [18] L. T. Thanh, K. Abed-Meraim, A. Hafiane, and N. L. Trung, "Sparse subspace tracking in high dimensions," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 5892–5896.
- [19] N. Lassami, K. Abed-Meraim, and A. Aïssa-El-Bey, "Low cost subspace tracking algorithms for sparse systems," in *European Signal Process. Conf.*, 2017, pp. 1400–1404.
- [20] N. Lassami, A. Aïssa-El-Bey, and K. Abed-Meraim, "Fast sparse subspace tracking algorithm based on shear and givens rotations," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2019, pp. 1667–1671.
- [21] N. Lassami, A. Aïssa-El-Bey, and K. Abed-Meraim, "Low cost sparse subspace tracking algorithms," *Signal Process.*, vol. 173, p. 107522, 2020.
- [22] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, 1995.
- [23] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2931–2941, 2005.
- [24] X. Yang, Y. Sun, T. Zeng, T. Long, and T. K. Sarkar, "Fast STAP method based on PAST with sparse constraint for airborne phased array radar," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4550–4561, 2016.
- [25] P. Xiao and L. Balzano, "Online sparse and orthogonal subspace estimation from partial information," in *Proc. Allerton Conf. Commun. Control Comput.*, 2016, pp. 284–291.
- [26] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas, "Online sparse and low-rank subspace learning from incomplete data: A Bayesian view," *Signal Process.*, vol. 137, pp. 199–212, 2017.
- [27] Chuang Wang and Y. M. Lu, "Online learning for sparse PCA in high dimensions: Exact dynamics and phase transitions," in *Proc. IEEE Inf. Theory Works.*, 2016, pp. 186–190.
- [28] W. Yang and H. Xu, "Streaming sparse principal component analysis," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 494–503.
- [29] K. Abed-Meraim, S. Attallah, A. Chkeif, and Y. Hua, "Orthogonal Oja algorithm," *IEEE Signal Process. Lett.*, vol. 7, no. 5, pp. 116–119, 2000.
- [30] Z. Allen-Zhu and Y. Li, "First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate," in *IEEE Ann. Symp. Found. Comput. Sci.*, 2017, pp. 487–492.
- [31] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "A new look at the power method for fast subspace tracking," *Digital Signal Process.*, vol. 9, no. 4, pp. 297–314, 1999.
- [32] K. Abed-Meraim, A. Chkeif, Y. Hua, and S. Attallah, "On a class of orthonormal algorithms for principal and minor subspace tracking," *J. VLSI Signal Process. Syst.*, vol. 31, no. 1, pp. 57–70, 2002.
- [33] X. G. Doukopoulos and G. V. Moustakides, "Fast and stable subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1452–1465, 2008.
- [34] R. Wang, M. Yao, D. Zhang, and H. Zou, "A novel orthonormalization matrix based fast and stable DPM algorithm for principal and minor subspace tracking," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 466–472, 2011.
- [35] Q. Wu, J. Zheng, Z. Dong, E. Panayirci, Z. Wu, and R. Qingnuobu, "An improved adaptive subspace tracking algorithm based on approximated power iteration," *IEEE Access*, vol. 6, pp. 43 136–43 145, 2018.
- [36] J. He, L. Balzano, and A. Sztam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2012, pp. 1568–1575.
- [37] Y. Chi, Y. C. Eldar, and R. Calderbank, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5947–5959, Dec 2013.
- [38] N. L. Trung, V. D. Nguyen, M. Thameri, T. M. Chinh, and K. Abed-Meraim, "Low-complexity adaptive algorithms for robust subspace tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1197–1212, 2018.
- [39] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee," *IEEE Trans. Signal Process.*, vol. 69, pp. 2070–2085, 2021.
- [40] P. Narayanamurthy, V. Daneshpajoo, and N. Vaswani, "Provable subspace tracking from missing data and matrix completion," *IEEE Trans. Signal Process.*, pp. 4245–4260, 2019.
- [41] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, 2019.
- [42] S.-C. Chan, Y. Wen, and K.-L. Ho, "A robust PAST algorithm for subspace tracking in impulsive noise," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 105–116, 2006.
- [43] J. Zhang and T. S. Qiu, "A robust correntropy based subspace tracking algorithm in impulsive noise environments," *Digital Signal Process.*, vol. 62, pp. 168–175, 2017.
- [44] S. Chan, Z. Zhang, and Y. Zhou, "A new adaptive Kalman filter-based subspace tracking algorithm and its application to DOA estimation," in *Proc. IEEE Symp. Circuits Syst.*, 2006, pp. 129–132.

- [45] B. Liao, Z. Zhang, and S.-C. Chan, "A new robust Kalman filter-based subspace tracking algorithm in an impulsive noise environment," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 57, no. 9, pp. 740–744, 2010.
- [46] V.-D. Nguyen, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking algorithms using fast adaptive Mahalanobis distance," *Signal Process.*, vol. 195, p. 108402, 2022.
- [47] A. M. Rekaivandi, A.-K. Seghouane, and K. Abed-Meraim, "TRPAST: A tunable and robust projection approximation subspace tracking method," *IEEE Trans. Signal Process.*, vol. 71, pp. 2407–2419, 2023.
- [48] L. T. Thanh, A. M. Rekaivandi, S. Abd-Krim, and K. Abed-Meraim, "Robust subspace tracking with contamination via α -divergence," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
- [49] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1493–1516, 1997.
- [50] N. V. Dung, K. Abed-Meraim, N. L. Trung, and R. Weber, "Generalized minimum noise subspace for array processing," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3789–3802, 2017.
- [51] J.-M. Chaufray, W. Hachem, and P. Loubaton, "Asymptotic analysis of optimum and suboptimum CDMA downlink MMSE receivers," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2620–2638, 2004.
- [52] X. Mestre and M. Á. Lagunas, "Modified subspace algorithms for DoA estimation with large arrays," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 598–614, 2008.
- [53] T. T. Cai, C.-H. Zhang, and H. H. Zhou, "Optimal rates of convergence for covariance matrix estimation," *Ann. Stat.*, vol. 38, no. 4, pp. 2118–2144, 2010.
- [54] A. J. Rothman, E. Levina, and J. Zhu, "Generalized thresholding of large covariance matrices," *J. Am. Stat. Assoc.*, vol. 104, no. 485, pp. 177–186, 2009.
- [55] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
- [56] L. Mackey, "Deflation methods for sparse PCA," *Adv. Neural Inf. Process. Syst.*, vol. 21, 2008.
- [57] J. Camacho, A. Smilde, E. Saccenti, J. Westerhuis, and R. Bro, "All sparse PCA models are wrong, but some are useful. Part II: Limitations and problems of deflation," *Chemometr. Intell. Lab. Syst.*, vol. 208, p. 104212, 2021.
- [58] X.-T. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *J. Mach. Learn. Res.*, vol. 14, no. 4, 2013.
- [59] Y. Deshpande and A. Montanari, "Sparse pca via covariance thresholding," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 4913–4953, 2016.
- [60] T. Wang, Q. Berthet, and R. J. Samworth, "Statistical and computational trade-offs in estimation of sparse principal components," *Ann. Stat.*, vol. 44, no. 5, pp. 1896–1930, 2016.
- [61] R. Krauthgamer, B. Nadler, and D. Vilenchik, "Do semidefinite relaxations solve sparse PCA up to the information limit?" *The Annals of Statistics*, vol. 43, no. 3, pp. 1300–1322, 2015.
- [62] V. Q. Vu and J. Lei, "Minimax sparse principal subspace estimation in high dimensions," *Ann. Stat.*, vol. 41, no. 6, pp. 2905–2947, 2013.
- [63] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Process. Lett.*, vol. 7, no. 3, pp. 60–62, 2000.
- [64] P. Strobach, "Low-rank adaptive filters," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 2932–2947, 1996.
- [65] M. Arjomandi-Lari and M. Karimi, "Generalized YAST algorithm for signal subspace tracking," *Signal Process.*, vol. 117, pp. 82–95, 2015.
- [66] N. V. Dung, K. Abed-Meraim, and N. L. Trung, "Second-order optimization based adaptive PARAFAC decomposition of three-way tensors," *Digit. Signal Process.*, vol. 63, pp. 100–111, 2017.
- [67] S. Zhou, N. X. Vinh, J. Bailey, Y. Jia, and I. Davidson, "Accelerating online CP decompositions for higher order tensors," in *Proc. ACM Int. Conf. Knowl. Discover. Data Min.*, 2016, pp. 1375–1384.
- [68] H. Kasai, "Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations," *Neurocomput.*, vol. 347, pp. 177–190, 2019.
- [69] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "A fast randomized adaptive CP decomposition for streaming tensors," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 2910–2914.