Robust Sparse Subspace Tracking from Corrupted Data Observations

Ta Giang Thuy Loan[*¶], Hoang-Lan Nguyen[*¶], Nguyen Thi Ngoc Lan[*], Do Hai Son[†],
Tran Thi Thuy Quynh[*], Nguyen Linh Trung[*], Karim Abed-Meraim[§], Thanh Trung Le[*¶]

[*] VNU University of Engineering and Technology, Hanoi, Vietnam
[†] VNU Information Technology Institute, Hanoi, Vietnam
[§] PRISME Laboratory, University of Orleans, Orleans, France

Abstract—**Subspace tracking is a fundamental problem in signal processing, where the goal is to estimate and track the underlying subspace that spans a sequence of data streams over time. In high-dimensional settings, data samples are often corrupted by non-Gaussian noises and may exhibit sparsity. This paper explores the alpha divergence for sparse subspace estimation and tracking, offering robustness to data corruption. The proposed method outperforms the state-of-the-art robust subspace tracking methods while achieving a low computational complexity and memory storage. Several experiments are conducted to demonstrate its effectiveness in robust subspace tracking and direction-of-arrival (DOA) estimation.**

Index Terms—**Sparse subspace estimation, Sparse subspace tracking, robust estimation, data corruption, non-Gaussian, outliers, α-divergence.**

I. INTRODUCTION

With the ubiquity of big data streams, modern online applications are continuously generating massive volumes of high-velocity data [1]. The dynamic and evolving nature of such data streams presents significant challenges for conventional data mining techniques, which often assume access to static or batch data. In many scenarios, data samples arrive sequentially, and it is desirable to update subspace estimates in (near) real time without revisiting old observations. Subspace tracking (ST) addresses this need by estimating and continuously updating a low-dimensional subspace that captures the underlying structure of streaming data [2].

Modern data are often incomplete, unreliable, or corrupted due to collection processes, inconsistencies, the presence of non-Gaussian noise, and sparse outliers [3]. High dimensionality further exacerbates these issues by increasing computational and memory demands, while also degrading algorithmic performance. A principled and effective way to mitigate these challenges is to exploit the empirical observation that high-dimensional data streams often lie near a low-dimensional and sparse subspace that can evolve over time [4], [5]. In the presence of data corruption, the classical subspace tracking problem naturally extends to the more general and challenging task of robust sparse subspace tracking.

Several ST algorithms have been proposed in recent years to address the challenges posed by data corruption [5]. Broadly, these methods fall into two categories: (ii) algorithms designed to handle outliers and missing data, and (ii) algorithms designed to cope with abrupt or impulsive noise. The first category includes several notable approaches such as Grassmannian-based methods (e.g., GROUSE [6], GRASTA [7], pROST [8]), recursive least squares-based methods (e.g., PETRELS [9], PETRELS-ADMM [10], PETRELS-CFAR [11]), recursive projected compressive sensing-based methods (e.g., ReProCS's variants [12]–[14]), and adaptive projected subgradient methods (e.g., [15]–[17]). Although these approaches are effective in handling outliers and missing entries, they typically rely on the assumption of slowly varying subspaces over time, making them less robust to abrupt changes or impulsive noise.

The second group includes robust variants of the Projection Approximation Subspace Tracking (PAST) algorithm (RPAST [18], MCC-PAST [19], TRPAST [20]); robust variants of adaptive or online power iteration methods (e.g., αFAPI [21], ROBUSTQR [22]); adaptive Kalman filtering (KFVM [23]), weighted recursive least-squares method (e.g., ROBUSTA [11]). These methods are designed to deal with impulsive noise but are generally sensitive to sparse corruptions and missing data. It is also worth noting that the aforementioned methods are not inherently designed for sparse subspace tracking. In the literature, only few sparse ST methods have been proposed. Notable examples include OIST [24], OVBSL [25], and OPIT [26]. However, these methods are not specifically designed to handle data corruption, particularly non-Gaussian noise. This limitation motivates the development of a new sparse subspace tracking algorithm that can both exploit sparsity in the subspace structure and maintain robustness against various types of data corruption. In this paper, we introduce αOPIT, an robust version of OPIT that leverages an α-divergence based weighting scheme for sparse subspace tracking with data corruption. Specifically, our method adaptively downweights anomalous/corrupted observations using

TABLE I: Conventional Notations

$x, \boldsymbol{x}, \boldsymbol{X}, \mathbb{X}$	scalar, vector, matrix, and set/support
x_i or $\boldsymbol{x}(i)$	i-th entry of $\boldsymbol{x}$
x_{ij} or $\boldsymbol{X}(i,j)$	(i,j)-th entry of $\boldsymbol{X}$
$\mathbf{X}(i,:), \mathbf{X}(:,j)$	i-th row and j-th column of $\boldsymbol{X}$
$\boldsymbol{X}^\top, \boldsymbol{X}^{-1}$	transpose, inverse of $\boldsymbol{X}$
$\mathrm{QR}(\mathbf{X})$	QR decomposition of $\mathbf{X}$
$\|\cdot\|_F$	Frobenius norm
$\mathrm{tr}\{\cdot\}$	trace operator
$\mathbb{E}\{\cdot\}$	expectation operator
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{D}_\alpha\{g\|f\}$	α-divergence between two distributions g and f

α-divergence, thereby enhancing its robustness against outliers and non-Gaussian noise. Additionally, αOPIT incorporates a recursive covariance and hence subspace update with a forgetting factor, allowing for efficient online adaptation to evolving subspaces in streaming data. Experimental results indicate that αOPIT consistently outperforms existing subspace tracking algorithms, particularly under challenging conditions involving non-Gaussian and impulsive noise, offering robustness and improved tracking accuracy.

Paper Organization: The rest of this paper is organized as follows. Section II provides background on subspace tracking, the OPIT algorithm, and α-divergence. Section III introduces our proposed method. Section IV presents the experimental results, and Section V concludes the paper. For easy reference, Table I summarizes frequently used notations in this paper.

II. BACKGROUND

In this section, we begin by formulating the problem of robust subspace tracking, followed by a brief overview of the classical OPIT method. We then introduce the α-divergence, which is employed to develop a weighting scheme that mitigates the influence of corrupted data.

A. Subspace Tracking

Assume that at each time t, we collect a data sample $\mathbf{x}_t \in \mathbb{R}^n$ which is generated under the following model

$$\mathbf{x}_t = \mathbf{A}\mathbf{w}_t + \boldsymbol{\nu}_t, \quad t = 1, 2, \ldots, T. \tag{1}$$

Here, $\mathbf{A} \in \mathbb{R}^{n \times r}$ is the underlying subspace matrix with $r < n$, $\mathbf{w}_t \in \mathbb{R}^r$ is a weight vector, and $\boldsymbol{\ell}_t = \mathbf{A}\mathbf{w}_t$ represents the low-rank component of $\mathbf{x}_t$. The vector $\boldsymbol{\nu}_t \in \mathbb{R}^n$ denotes data corruption present in the observation. The problem of subspace tracking can be stated as follows:

Subspace Tracking Problem: On the arrival of a new data sample $\mathbf{x}_t$ at each time t, our goal is to estimate the underlying subspace $\mathbf{A}$ that spans the low-rank components $\{\boldsymbol{\ell}_i\}_{i=1}^t$.

Algorithm 1: THRESHOLDING - $\hat{\mathbf{S}} = \tau(\mathbf{S}, k)$

Input: Matrix $\mathbf{S}$ and a thresholding factor k
Main Procedure:
  $[n, r] = \mathrm{size}(\mathbf{S})$
  for $i = 1, 2, \ldots, r$ **do**
    $\mathbf{s}_i = \mathbf{S}(:, i)$
    Select the set $\mathcal{T}_t$ that contains indices of k strongest entries (w.r.t. absolute value) of $\mathbf{s}_i$
    Form $\hat{\mathbf{S}}(j, i) = \begin{cases} \mathbf{s}_i(j) & \text{if } j \in \mathcal{T}_t \\ 0 & \text{if } j \notin \mathcal{T}_t \end{cases}$
  end
End

When the subspace matrix $\mathbf{A}$ is sparse, robust ST becomes robust sparse subspace tracking (SST) problem.[1]

B. OPIT Method

Online Power Iteration by Thresholding (OPIT) offers a fast method for tracking the underlying sparse subspace of data streams over time [26]. It builds upon the standard Power Iteration (PI) method for computing the dominant eigenvectors of the covariance matrix $\mathbf{C}_t = \mathbb{E}\{\mathbf{x}_t\mathbf{x}_t^\top\}$. Particularly at the k-th iteration, PI updates

$$\mathbf{S}_\ell \leftarrow \mathbf{C}_t\mathbf{U}_{\ell-1}, \quad \mathbf{U}_\ell \overset{\text{Q-factor}}{\leftarrow} \mathrm{QR}(\mathbf{S}_\ell), \tag{2}$$

where $\mathrm{QR}(\cdot)$ denotes the QR factorization [2]. PI starts from an initial matrix $\mathbf{U}_0 \in \mathbb{R}^{n \times r}$ and returns an orthonormal matrix $\mathbf{U}_\ell$, where L is the number of iterations. OPIT modifies PI by recursively updating the "scaled" version of $\mathbf{C}_t$ and introducing a forgetting factor $0 < \beta \leq 1$ to exponentially discount the effect of old observations

$$\mathbf{R}_t = \beta\mathbf{R}_{t-1} + \mathbf{x}_t\mathbf{x}_t^\top, \tag{3}$$

where $\mathbf{C}_t = t^{-1}\mathbf{R}_t$. According, OPIT rewrites the first step of (2) as follows

$$\mathbf{S}_t = \mathbf{R}_t\mathbf{U}_{t-1} = \beta\mathbf{R}_{t-1}\mathbf{U}_{t-1} + \mathbf{x}_t\mathbf{z}_t^\top, \tag{4}$$

where $\mathbf{z}_t = \mathbf{U}_{t-1}^\top\mathbf{x}_t$. As small perturbations do not significantly affect the performance of power methods [27, Proposition 2], OPIT derives the following rule

$$\mathbf{S}_t \simeq \beta\mathbf{S}_{t-1}\mathbf{E}_{t-1} + \mathbf{x}_t\mathbf{z}_t^\top, \tag{5}$$

where $\mathbf{E}_{t-1} = \mathbf{U}_{t-1}^\top\mathbf{U}_{t-2}$. After that, OPIT employs the thresholding operation on (5) before employing the QR factorization of $\mathbf{S}_t$, see Algorithm 1. We refer the readers to our work [26] for further details. Since OPIT is not inherently designed to handle data corruption, we incorporate the α-divergence (introduced in the following subsection) to enhance its robustness.

---

[1]In nonstationary environments, the subspace matrix $\mathbf{A}$ can be slowly varying with time, i.e., $\mathbf{A} = \mathbf{A}_t$. Our method not only estimates it accurately but also effectively tracks its variation over time. See Fig. 2 for an illustration.

Algorithm 2: αOPIT - ONLINE POWER ITERATION BY THRESHOLDING WITH α DIVERGENCE

Input: $\{\mathbf{x}_t\}_{t=1}^T, \mathbf{x}_t \in \mathbb{R}^n$, rank r, a forgetting factor $0 \le \lambda \le 1$, alpha divergence with parameter $0 < \alpha < 1$, $0 < p \le 2$, 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 subspace.

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

Main Procedure:
  for $t = 1, 2, \ldots, T$ **do**

$\mathbf{w}_t = \mathbf{U}_{t-1}^\top \mathbf{x}_t$	$\mathcal{O}(nr)$
$\mathbf{e}_t = \mathbf{x}_t - \mathbf{U}_{t-1}\mathbf{w}_t$	$\mathcal{O}(nr)$
$\omega_t = \exp\left(-\dfrac{1-\alpha}{2}\|\mathbf{e}_t\|_F^p\right)$	$\mathcal{O}(n)$
$\mathbf{S}_t = (1-\lambda)\mathbf{S}_{t-1}\mathbf{E}_{t-1} + \lambda\omega_t\mathbf{x}_t\mathbf{w}_t^\top$	$\mathcal{O}(nr^2 + nr)$
$\hat{\mathbf{S}}_t = \tau(\mathbf{S}_t, k)$	$\mathcal{O}(nr + rk\log k)$
$\mathbf{U}_t = \text{QR}(\hat{\mathbf{S}}_t)$	$\mathcal{O}(nr^2)$
$\mathbf{E}_t = \mathbf{U}_{t-1}^\top \mathbf{U}_t$	$\mathcal{O}(nr^2)$

  end

End

Output: $\mathbf{U}_T$

C. Alpha Divergence

The α-divergence is a family of measures used to quantify the difference between two probability distributions [28]. Particularly given two distributions $g(\theta)$ and $f(\theta)$, its α-divergence $D_\alpha\{g \| f\}$ is defined as

$$D_\alpha\{g \| f\} = \frac{1}{\alpha(1-\alpha)}\left[\int g(\theta)^\alpha f(\theta)^{1-\alpha}\,d\theta - 1\right], \quad (6)$$

where $0 < \alpha < 1$ is a tunable parameter to control the asymmetry of the divergence.[2] Here, (6) can be considered as a generalization of the well-known Kullback–Leibler (KL) divergence. In particular, $\lim_{\alpha \to 1} D_\alpha(g \| f) = \text{KL}(g \| f)$ and $\lim_{\alpha \to 0} D_\alpha(g \| f) = \text{KL}(f \| g)$. The α divergence can offer robustness to several types of non-Gaussian noises which is exploited in the next section.

III. PROPOSED METHOD

In this section, we introduce a novel robust variant of OPIT, called αOPIT, which is designed to enhance OPIT's robustness and effectiveness against data corruption. The proposed method incorporates advanced weighting and subspace estimation techniques derived from α-divergence and OPIT, allowing it to handle non-Gaussian noise and high-dimensional settings. In what follows, we detail how α-divergence is incorporated into OPIT to improve its robustness.

Following the robust statistical approach for sample covariance estimation proposed in [21], we first modify the "scaled" covariance matrix (3) as follows

$$\mathbf{R}_t = (1-\lambda)\mathbf{R}_{t-1} + \lambda\omega_t\mathbf{x}_t\mathbf{x}_t^\top, \quad (7)$$

[2]In the literature, there exist some other forms of α-divergence, see [28] for more details.

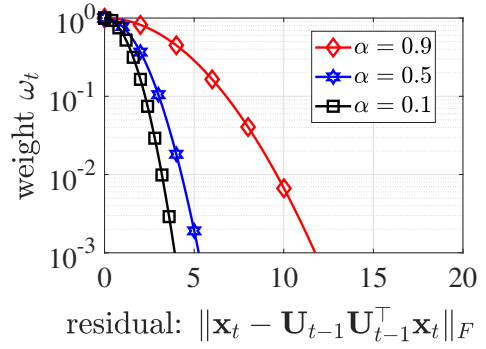

Fig. 1: The weight ω_t is computed based on α divergence with $p = 2$. When the residual is large, ω_t approaches zero; conversely, when the residual is small, ω_t approaches one.

where $1 - \lambda$ plays the same role as to the forgetting factor β and the weight ω_t is chosen as

$$\omega_t = \exp\left(-\frac{1-\alpha}{2}\left\|\mathbf{x}_t - \mathbf{U}_{t-1}\mathbf{U}_{t-1}^\top\mathbf{x}_t\right\|_F^p\right), \quad (8)$$

with $0 < \alpha < 1$ and $0 < p \le 2$. Here, $\mathbf{U}_{t-1}$ denotes the previous estimate of the subspace matrix $\mathbf{A}$ at time $t - 1$. The residual between the data observation and its projection is defined as $\|\mathbf{e}_t\|_F = \|\mathbf{x}_t - \mathbf{U}_{t-1}\mathbf{U}_{t-1}^\top\mathbf{x}_t\|_F$. See Fig. 1 for an illustration. A large residual suggests that $\mathbf{x}_t$ may not lie within the current subspace and could be a corrupted data sample. In such cases, the corresponding weight ω_t becomes small (close to zero), reducing the influence of the corrupted data. Conversely, as the residual approaches zero, ω_t approaches 1, indicating that the data sample is clean and should fully contribute to the subspace update.

The use of the weight ω_t in (8) is motivated from the following observation. The signal model (1) implies the empirical distribution $g(\mathbf{x}, \mathbf{U})$ of data samples is a mixture of a true one $f(\mathbf{x}, \mathbf{U})$ (corresponding to the low-rank signal) and a contaminated component, i.e.,

$$g(\mathbf{x}, \mathbf{U}) = (1-\delta)f(\mathbf{x}, \mathbf{U}) + \delta h(\mathbf{x}), \quad (9)$$

where $h(\mathbf{x})$ represents impulsive or non-Gaussian noises and $0 < \delta < 1$ is denotes a trade-off parameter between two distributions. Accordingly, the α divergence $\mathcal{D}_\alpha(g(\mathbf{x}, \mathbf{U}) \| f(\mathbf{x}, \mathbf{U}))$ provides a robust estimation criterion for estimating the underlying subspace as follows

$$\mathbf{A} = \underset{\mathbf{U}}{\arg\min} \; \mathcal{D}_\alpha\{g(\mathbf{x}, \mathbf{U}) \| f(\mathbf{x}, \mathbf{U})\}. \quad (10)$$

Since $g(\mathbf{x}, \mathbf{U})$ is generally unknown in practice, the work [20] indicated that, in such cases, (10) reduces to

$$\mathbf{A} = \underset{\mathbf{U}}{\arg\max} \; \frac{1}{1-\alpha}\sum_{k=1}^t f(\mathbf{x}_k, \mathbf{U})^{1-\alpha}. \quad (11)$$

As indicated in our companion work [21], (11) is approximately equivalent to

$$\mathbf{A} = \underset{\mathbf{U}}{\operatorname{argmin}} \sum_{k=1}^{t} \widetilde{\omega}_k \left\| \mathbf{x}_k - \mathbf{U}\mathbf{U}^\top \mathbf{x}_k \right\|_F^2, \quad \text{with} \qquad (12)$$

$$\widetilde{\omega}_k = \exp\left(-\frac{1-\alpha}{2} \left\| \mathbf{x}_k - \mathbf{U}_{k-1}\mathbf{U}_{k-1}^\top \mathbf{x}_k \right\|_F^2 \right). \qquad (13)$$

In parallel, minimizing a weighted least-square objective function with a weight $\widetilde{\omega}_k$ results in the principal subspace of the weighted covariance matrix $\mathbf{C}_k = \mathbf{C}_{k-1} + \widetilde{\omega}_k \mathbf{x}_k \mathbf{x}_k^\top$. As a result, we can adopt the form (8) to set the weight ω_t in (7). The inclusion of $p \leq 2$. Accordingly, we reformulate the main step (4) of the classical OPIT as follows

$$\mathbf{S}_t = (1 - \lambda)\mathbf{S}_{t-1}\mathbf{E}_{t-1} + \lambda\omega_t\mathbf{x}_t\mathbf{w}_t^\top. \qquad (14)$$

Other steps of αOPIT can be given in Algorithm 2.

Complexity Analysis: αOPIT shares the same computational complexity and memory storage as the classical OPIT method. Specifically, its overall computational cost is $\mathcal{O}(nr^2)$ flops. The cost for each step of αOPIT is provided in details in Algorithm 2. In terms of memory storage, αOPIT requires a total of $2nr + r^2$ words of memory to save $\mathbf{U}_t, \mathbf{S}_t$, and $\mathbf{E}_t$ at each iteration.

IV. Experiments

This section presents several experiments conducted to investigate the performance of αOPIT. Its effectiveness is evaluated in comparison with several state-of-the-art ST algorithms, including αFAPI, TRPAST, ROBUSTA, and OPIT.

A. Robust Sparse Subspace Tracking

The signal samples $\{\mathbf{x}_t\}_{t\geq 1}$ are generated based on the data model (1) where $\mathbf{A} = \mathbf{A}_t$ is a slowly time-varying sparse subspace matrix, generated recursively as follows

$$\mathbf{A}_t = \mathbf{\Omega} \circledast (\mathbf{A}_{t-1} + \varepsilon_t \mathbf{V}_t), \qquad (15)$$

where $\mathbf{\Omega} \in \mathbb{R}^{n \times r}$ is a binary matrix indicating the sparsity level of the subspace matrix, and $\mathbf{V}_t \in \mathbb{R}^{n \times r}$ is Gaussian noise with zero mean and unit variance. The parameter $\varepsilon_t \geq 0$ controls the level of subspace variation at each time step t. The vector $\mathbf{w}_t \in \mathbb{R}^r$ denotes the coefficient vector. The vector $\boldsymbol{\nu}_t \in \mathbb{R}^n$ accounts for non-Gaussian noises. In our experiments, we consider the following three cases for the noise $\boldsymbol{\nu}_t$:

$$\boldsymbol{\nu}_t(i) \sim (1-\delta)\mathcal{N}(0, \sigma_n^2) + \delta\,\mathrm{Laplace}(\mu, \gamma), \qquad (16)$$
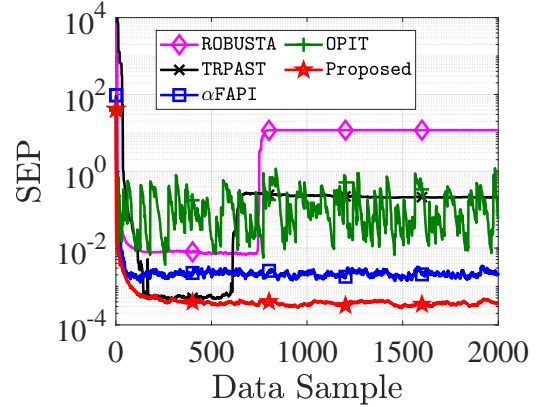
$$\boldsymbol{\nu}_t(i) \sim (1-\delta)\mathcal{N}(0, \sigma_n^2) + \delta\,\mathrm{Cauchy}(\mu, \gamma), \qquad (17)$$
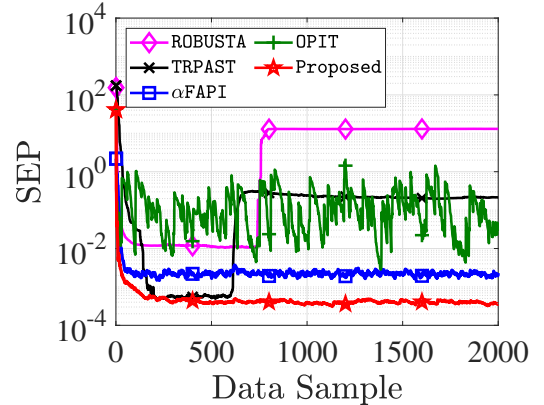
and

$$\begin{aligned} \boldsymbol{\nu}_t(i) \sim\ & (1-\delta)\mathcal{N}(0, \sigma_n^2) \\ & + \frac{\delta}{2}\mathrm{Laplace}(\mu, \gamma) + \frac{\delta}{2}\mathrm{Cauchy}(\mu, \gamma), \qquad (18) \end{aligned}$$
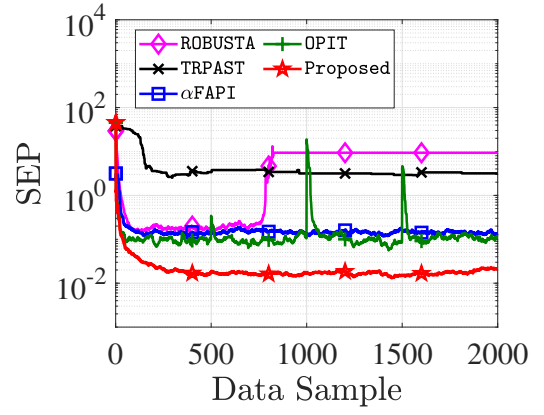
(a) Laplace noise

(b) Cauchy noise

(c) Laplace + Cauchy

Fig. 2: Experiment setup: data dimension $n = 200$, true rank $r = 5$, number of data samples $T = 2000$, time-varying factor $\varepsilon_t = 10^{-2}$, subspace sparsity 80%, the alpha divergence with $\alpha = 0.9$.

where δ represents the proportion of corrupted data, and their probability density functions are given by

$$\mathcal{N}(x; \mu, \sigma_n^2) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma_n^2} \right), \qquad (19)$$

$$\mathrm{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x-\mu|}{\gamma} \right), \qquad (20)$$

and

$$\text{Cauchy}(x; \mu, \gamma) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x-\mu}{\gamma}\right)^2\right]}. \qquad (21)$$

Furthermore, two abrupt changes are made at $t = 1000$ and $t = 1500$ to evaluate the robustness of all ST algorithms.

To measure the accuracy of subspace tracking algorithms, we use the subspace estimation performance (SEP) metric, defined as follows.

$$\text{SEP}\left(\mathbf{U}_{\text{true}}, \mathbf{U}_{\text{est}}\right) = \frac{\text{tr}\left\{\mathbf{U}_{\text{est}}^{\top}\left(\mathbf{I} - \mathbf{U}_{\text{true}}\mathbf{U}_{\text{true}}^{\top}\right)\mathbf{U}_{\text{est}}\right\}}{\text{tr}\left\{\mathbf{U}_{\text{est}}^{\top}\left(\mathbf{U}_{\text{true}}\mathbf{U}_{\text{true}}^{\top}\right)\mathbf{U}_{\text{est}}\right\}}. \qquad (22)$$

The lower SEP indicates the better algorithm performance. Fig. 2 illustrates the performance of all subspace tracking algorithms across three case studies involving non-Gaussian noise. As demonstrated, our method consistently outperforms other state-of-the-art robust subspace tracking methods in all settings. In particular, ROBUSTA and TRPAST are sensitive to both Laplace and Cauchy noise. Although OPIT performs well under Laplace noise, it exhibits performance degradation during abrupt changes at $t = 1000$ and $t = 1500$ and struggles with Cauchy noise. αFAPI can handle both types of corruption. However, its estimation accuracy is lower than that of our proposed method.

B. Direction-of-arrival (DOA) Tracking

We further evaluate the performance of the proposed αOPIT algorithm in the context of direction-of-arrival (DOA) estimation for wireless communication systems. Assume that the received signal at each time t follows the data model

$$\mathbf{x}_t = \mathbf{A}_t\mathbf{s}_t + \boldsymbol{\nu}_t. \qquad (23)$$

Here, $\mathbf{A}_t = \left[\mathbf{a}(\omega_{1,t}), \mathbf{a}(\omega_{2,t}), \ldots, \mathbf{a}(\omega_{K,t})\right]$ is the time-varying steering matrix whose column is defined as

$$\mathbf{a}(\omega_{k,t}) = \left[1, \exp(j\omega_{k,t}), \ldots, \exp(j(n-1)\omega_{k,t})\right]^{\top}, \qquad (24)$$

where $\omega_{k,t} = \pi\sin\theta_{k,t}$ denotes the angular frequency associated with the DOA of the k-th user. The vector $\mathbf{s}_t$ represents the user signal, modeled as a complex Gaussian random vector with covariance matrix $\mathbf{C}_s = \mathbf{I}_K$. The noise vector $\boldsymbol{\nu}_t$ follows the mixed distribution (17) described in the previous task of robust subspace tracking.

The main objective is to estimate the DOAs from the corrupted observations $\mathbf{x}_t$. To this end, αOPIT is applied to recursively and robustly estimate the underlying signal subspace. Once the subspace is obtained, the ESPRIT algorithm [29] is applied to extract angular frequencies $\omega_k(t)$, and thus recover the DOAs $\theta_k(t)$.

The experimental results in Fig. 3 demonstrate that αOPIT achieves high accuracy in DOA estimation under non-Gaussian noise conditions. Compared to existing algorithms such as ROBUSTA, TRPAST, and αFAPI, our method consistently offers superior robustness for the DOA tracking task. Notably,
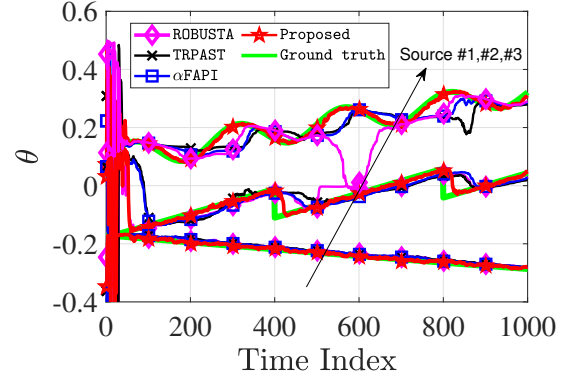

Fig. 3: DOA tracking setup: $n = 20$ sensors, $r = 3$ signal sources, and $T = 1000$ data samples. Alpha divergence with $\alpha = 0.9$ is used. The three underlying DOAs consist of a linear source (#1), a sawtooth source (#2) and a sinusoidal source (#3), illustrated by (—).

in scenarios where the angular frequency θ changes fast (e.g., source #1 or source #2 at $t = 400$ and $t = 800$), αOPIT adapts more quickly and tracks the variations more accurately than the competing methods.

V. CONCLUSIONS

In this paper, we addressed the problem of robust subspace tracking with data corruption. We proposed a novel robust and adaptive algorithm, called αOPIT, for tracking the principal sparse subspace of streaming data over time. Experimental results demonstrated that αOPIT effectively handles various types of data contamination and consistently outperforms existing subspace tracking algorithms. Future works will explore its performance in high-dimensional settings and fast time-varying cases. In addition, when handling higher-order streaming data, subspace tracking naturally extends to tensor tracking [30]. Therefore, exploring robust tensor tracking through α-divergence is a promising direction for future research.

REFERENCES

[1] T. Kolajo, O. Daramola, and A. Adebiyi, "Big data stream analysis: A systematic literature review," *J. Big Data*, vol. 6, no. 1, pp. 1–30, 2019.

[2] J.-P. Delmas, "Subspace tracking for signal processing," *Adaptive Signal Processing: Next Generation Solutions*, pp. 211–270, 2010.

[3] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *National Science Review*, vol. 1, no. 2, pp. 293–314, 2014.

[4] 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.

[5] 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.

[6] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Annual Allerton Conf. Commun. Control Comput.*, pp. 704–711, 2010.

[7] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pp. 1568–1575, 2012.

[8] F. Seidel, C. Hage, and M. Kleinsteuber, "pROST: a smoothed ℓ_p-norm robust online subspace tracking method for background subtraction in video," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1227–1240, 2014.

[9] 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, pp. 5947–5959, Dec 2013.

[10] 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.

[11] 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.

[12] J. Zhan, B. Lois, H. Guo, and N. Vaswani, "Online (and offline) robust PCA: Novel algorithms and performance guarantees," in *Proc. Artif. Intell. Stat.*, pp. 1488–1496, 2016.

[13] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust pca or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, 2018.

[14] P. Narayanamurthy and N. Vaswani, "Fast robust subspace tracking via PCA in sparse data-dependent noise," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 3, pp. 723–744, 2020.

[15] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, "An adaptive projected subgradient based algorithm for robust subspace tracking," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 5497–5501, 2014.

[16] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, "Robust subspace tracking with missing entries: The set-theoretic approach," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5060–5070, 2015.

[17] K. Slavakis and I. Yamada, "The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning," *SIAM J. Opt.*, vol. 23, no. 1, pp. 126–152, 2013.

[18] 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.

[19] J. Zhang, T. Qiu, and S. LI, "A robust past algorithm based on maximum correntropy criterion for impulsive noise environments," *Acta Electronica Sinica*, vol. 43, no. 3, pp. 483–488, 2015.

[20] A. M. Rekavandi, 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.

[21] L. T. Thanh, A. M. Rekavandi, S. Abd-Krim, and K. Abed-Meraim, "Robust subspace tracking with contamination via α-divergence," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 1–5, 2023.

[22] 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.

[23] 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.

[24] C. Wang and Y. M. Lu, "Online learning for sparse PCA in high dimensions: Exact dynamics and phase transitions," in *Proc. IEEE Inf. Theory Works.*, pp. 186–190, 2016.

[25] 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.

[26] T. T. Le, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "OPIT: a simple but effective method for sparse subspace tracking in high-dimension and low-sample-size context," *IEEE Trans. Signal Process.*, vol. 72, pp. 521–534, 2024.

[27] T. T. Le, K. Abed-Meraim, A. Hafiane, and N. L. Trung, "Sparse subspace tracking in high dimensions," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, pp. 5892–5896, 2022.

[28] A. Cichocki and S.-i. Amari, "Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities," *Entropy*, vol. 12, no. 6, pp. 1532–1568, 2010.

[29] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 7, pp. 984–995, 1989.

[30] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "A contemporary and comprehensive survey on streaming tensor decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 10897–10921, 2023.