Federated Over-Air Subspace Tracking From Incomplete and Corrupted Data

Praneeth Narayanamurthy, Namrata Vaswani, and Aditya Ramamoorthy

***Abstract*—In this work we study the problem of Subspace Tracking with missing data (ST-miss) and outliers (Robust ST-miss). We propose a novel algorithm, and provide a guarantee for both these problems. Unlike past work on this topic, the current work does not impose the piecewise constant subspace change assumption. Additionally, the proposed algorithm is much simpler (uses fewer parameters) than our previous work. Secondly, we extend our approach and its analysis to provably solving these problems when the data is federated and when the over-air data communication modality is used for information exchange between the K peer nodes and the center. We validate our theoretical claims with extensive numerical experiments.**

***Index Terms*—Principal component analysis, federated learning, matrix decomposition, adaptive algorithms.**

I. Introduction

SUBSPACE tracking (ST) with missing data, or outliers, or both has been extensively studied in the last few decades [2]–[6]. ST with outlier data is commonly referred to as Robust ST (RST); it is the dynamic or "tracking" version of Robust PCA [7], [8]. This work provides a new simple algorithm and guarantee for both ST with missing data (ST-miss) and RST-miss. Secondly, we extend our approach and its analysis to provably solving these problems when the data is federated and when the over-air data communication modality [9] is used for information exchange between the K peer nodes and the central server. (R)ST-miss has important applications in video analytics [10], social network activity learning [11] (anomaly detection) and recommendation system design [12] (learning time-varying low-dimensional user preferences from incomplete user ratings). The federated setting is most relevant for the latter two. At each time, each local node would have access to user ratings or messaging data from a subset of nearby users, but the subspace learning and matrix completion algorithm needs to use data from all the users.

Federated learning [13] refers to a distributed learning scenario in which individual nodes keep their data private and only share intermediate locally computed summary statistics with the central server at each algorithm iteration. The central server in turn, shares a global aggregate of these iterates with all the nodes. There has been extensive recent work on solving machine learning problems in a federated setting [14]–[18] but all these assume a perfect channel between the peer nodes and the central server. This is a valid assumption in the traditional digital transmission mode in which different peer nodes transmit in different time or frequency bands, and appropriate channel coding is done at lower network layers to enable error-free recovery with very high probability.

Advances in wireless communication technology now allow for (nearly) synchronous transmission by the various peer nodes and thus enable an alternate computation/communication paradigm for learning algorithms for which the aggregation step is a summation operation. In this alternate paradigm, the summation can be performed "over-air" using the superposition property of the wireless channel and the summed aggregate (or its processed version) can then be broadcasted to all the nodes [9], [19], [20]. Assuming K peer nodes, this over-air addition is up to K-times more time- or bandwidth-efficient than the traditional mode. In the absence of error control coding at the lower network layers, additive channel noise and channel fading effects corrupt the transmitted data. In general, there exist well-established physical layer communication techniques to estimate and compensate for channel fading [21]. Also, while perfect synchrony in transmission is impossible, small timing mismatches can be handled using standard techniques. We expand upon both these points in Section IV-A. From a signal processing perspective, therefore, the main issue to be tackled is that of additive channel noise which now corrupts each algorithm iterate.

1) Related Work: Provable ST with missing or corrupted data (ST-miss and RST-miss) in the centralized setting has been extensively studied in past work [3]–[5], [22]–[24]. Provable analyses can be one of two kinds – ones that come with a *complete guarantee or correctness result* and ones that come with only a *partial guarantee*. By *complete guarantee or correctness result*, we mean a result that makes assumptions only on the inputs to the algorithm (the observed data and the algorithm initialization if any) and guarantees that, the algorithm output will be close to the true value of the quantity of interest, either at all times, or after a finite delay. If a guarantee does not do this, we refer to it as a *partial guarantee*. Most existing works [3]–[5], [24], [25] are partial guarantees. Although, [22], [23] obtain complete guarantees, these works impose a piecewise constant

Manuscript received 17 June 2021; revised 17 February 2022 and 17 June 2022; accepted 20 June 2022. Date of publication 27 June 2022; date of current version 9 August 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ba Ngu Vo. This work was supported by the NSF under Grants CCF-1910840 and CCF-2115200. This work was presented at ICASSP 2022 [DOI:10.1109/ICASSP43922.2022.9747220]. *(Corresponding author: Praneeth Narayanamurthy.)*

The authors are with the Iowa State University, Iowa 50010 USA (e-mail: praneeth@usc.edu; namrata@iastate.edu; adityar@iastate.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2022.3186540, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3186540

subspace change assumption. This assumption is often not valid in practice, e.g, there is no reason for a "subspace change time" in case of slowly changing video backgrounds. The results of [22], [23] assume it in order to obtain simple guarantees for ϵ-accurate subspace recovery for any $\epsilon > 0$ (in the noise-free case) or for any ϵ larger than the noise-level (in the noisy case).

The only other existing works that also study unsupervised learning algorithms with noisy algorithm iterations are [26], [27]; both these works study the noisy iteration version of the power method (PM) for computing the top r singular vectors of a given data matrix. In these works, noise is deliberately added to each algorithm iterate in order to ensure privacy of the data matrix.

It should be noted that other solutions to batch low-rank matrix completion (LRMC) cannot be implemented to respect the federated constraints (the aggregation step needs to be a summation operation). We briefly discuss these in Section IV. Another somewhat related line of work involves distributed algorithms for PCA; these are reviewed in [12], and there is also one for distributed ST-miss [28], Most of these come without provable guarantees, and most also do not account for either missing data or iteration-noise or both. For example, the recent work [29] aims to optimize communication efficiency but the channel is assumed to be perfect, and so iteration noise is not considered. Moreover, the algorithm is computationally expensive (involves computing a full SVD of a large matrix); and the guarantee provided is a multiplicative one on the PCA reconstruction error. Finally, LRMC in a decentralized setting is studied in [30] with the goal of speeding up computation via parallel processing using multiple computing nodes. In this paper as well, the full data is communicated to the central server and hence this is not a federated setting. Also, no channel noise is considered. It is not clear if this algorithm or guarantee can be modified to deal with federated data or over-air communication. Finally, there also exist heuristics for various types of distributed LRMC such as [31]–[33].

Other works that also develop algorithms for the federated over-air aggregation setting include [9], [34]. However, all these develop stochastic gradient descent (SGD) based algorithms and the focus is on optimizing resource allocation to satisfy transmit power constraints. These do not provide performance guarantees for the resulting perturbed SGD algorithm. A different related line of work is in developing federated algorithms, albeit not in the over-air aggregation mode. Recent works such as [14], [35] attempt to empirically optimize the communication efficiency. Similarly, [36] studies federated PCA but it does not consider over-air communication paradigm, and does not deal with outliers or missing data.

2) Contributions: This work has two contributions. First, we obtain a new set of results that provide a complete guarantee for ST-miss and RST-miss without assuming piecewise constant subspace change. The tradeoff is our error bounds are a little more complicated. Another advantage of our new result with respect to previous ReProCS algorithms [23], [37] is that it analyzes a much simpler tracking algorithm (only one algorithm parameter needs to be set instead of three). Our guarantee is useful (improves upon the naive approach of standard PCA repeated every α frames) when the subspace changes are indeed slow enough. At the same time, we can still obtain a guarantee for our simpler algorithm that holds under piecewise constant subspace change but does not require an upper bound on the amount of change, i.e, we recover the result of [23].

The second contribution of this work is a provable solution to the above problem in the federated data setting when the data communication is done in the over-air mode. As explained above, the main new challenge here is to develop approaches that are provably robust to additive noise in the algorithm iterates. This setting of noisy iterations has received little attention in literature as noted above. To the best of our knowledge, this is the first provable algorithm that studies (R)ST-miss in a federated, over-air paradigm. The main challenges here are (i) a design of an algorithm for this setting (this requires use of a federated over-air power method (FedOA-PM) for solving the PCA step) and (ii) dealing with noise iterates due to the channel noise. For the latter, the main work is in obtaining a modified result for PCA in sparse data-dependent noise solved via the FedOA-PM; see Lemma 4.7.

3) Paper Organization: We give the centralized problem formulation next. After this, in Section III, we develop our solution for ST-miss in the centralized setting and explain how it successfully relaxes the piecewise constant subspace change assumption made by existing guarantees. Next, we directly consider RST-miss in the federated over-air setting in Section IV. Simulations are provided in Section V and we conclude in Section VI. We provide a few extensions of our proposed algorithms, and complete proof details in the Supplementary Material (https://arxiv.org/abs/2002.12873).

II. Notation and Problem Formulation

A. Notation

We use the interval notation $[a, b]$ to refer to all integers between a and b, inclusive, and we use $[a, b) := [a, b-1]$. We use $[K] := [1, K]$. $\|.\|$ denotes the l_2 norm for vectors and induced l_2 norm for matrices unless specified otherwise. We use $\boldsymbol{I}$ to denote the identity matrix of appropriate dimensions. We use $\boldsymbol{M}_{\mathcal{T}}$ to denote a sub-matrix of $\boldsymbol{M}$ formed by its columns indexed by entries in the set $\mathcal{T}$. A matrix $\boldsymbol{P}$ with mutually orthonormal columns is referred to as a *basis matrix*; it represents the subspace spanned by its columns. For basis matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$, $\mathrm{dist}(\boldsymbol{P}_1, \boldsymbol{P}_2) := \|(\boldsymbol{I} - \boldsymbol{P}_1\boldsymbol{P}_1^\top)\boldsymbol{P}_2\|$ quantifies the Subspace Error (distance) between their respective subspaces. This is equal to the sine of the largest principal angle between the subspaces. If $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are of the same dimension, $\mathrm{dist}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \mathrm{dist}(\boldsymbol{P}_2, \boldsymbol{P}_1)$. We reuse the letters C, c to denote different numerical constants in each use with the convention that $C \geq 1$ and $c < 1$.

We use *r-SVD* to refer to the matrix of top-r left singular vectors (vectors corresponding to the r largest singular values) of the given matrix. Finally, $\boldsymbol{M}^\dagger := (\boldsymbol{M}^\top\boldsymbol{M})^{-1}\boldsymbol{M}^\top$ is used to denote the pseudo inverse of $\boldsymbol{M}$.

B. ST With Missing Data (ST-Miss)

Assume that at each time t, we observe an n-dimensional data stream of the form

$$\boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\tilde{\boldsymbol{\ell}}_t), \quad t = 1, 2, \ldots, d \tag{1}$$

where $\mathcal{P}_{\Omega_t}(\cdot)$ is a binary mask that selects entries in the index set Ω_t (this is known), and $\tilde{\boldsymbol{\ell}}_t$ approximately lies in a low (at most r) dimensional subspace that is either constant or changes slowly over time. The goal is to track the subspace(s). This statement can be made precise in several ways. The first is as done in past work [23] and references therein. One assumes a "generative model": $\tilde{\boldsymbol{\ell}}_t = \boldsymbol{P}_t \boldsymbol{a}_t$ with $\boldsymbol{P}_t$ being a $n \times r$ basis matrix. The goal is to track the column span of $\boldsymbol{P}_t$, span$(\boldsymbol{P}_t)$. To make this problem well-posed (number of unknowns smaller than number of observed scalars), the piecewise constant subspace change model assumption becomes essential as explained in [23]. However, this is a restrictive assumption that is typically not valid for real data, e.g., there is no reason for the subspaces to change at certain select time instants in case of slowly changing videos.

A second approach to make our problem statement precise, and the one that we use in this work, is as follows. For an α large enough,[1] consider α-length sub-matrices formed by consecutive $\tilde{\boldsymbol{\ell}}_t$'s. Let $\tilde{\boldsymbol{L}}_1 := [\tilde{\boldsymbol{\ell}}_1, \tilde{\boldsymbol{\ell}}_2, \dots, \tilde{\boldsymbol{\ell}}_\alpha]$; $\tilde{\boldsymbol{L}}_2 := [\tilde{\boldsymbol{\ell}}_{\alpha+1}, \tilde{\boldsymbol{\ell}}_{\alpha+2}, \dots, \tilde{\boldsymbol{\ell}}_{2\alpha}]$ and so on. Let $\boldsymbol{P}_j$ be the r-SVD (matrix of top r singular vectors) of $\tilde{\boldsymbol{L}}_j$. Slow subspace change means that, for all j,

$$\Delta_j := \text{dist}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j) \leq \Delta_{tv}$$

for a $\Delta_{tv} \ll 1$. Note that the above problem formulation does not necessarily assume that the vectors $\{\tilde{\boldsymbol{\ell}}_t\}_{t=(j-1)\alpha+1}^{j\alpha}$ are drawn from a specific subspace, but rather the subspace is defined post-facto. This subtle, yet important, difference allows us to eliminate the piecewise constant subspace change assumption in this work. The goal is to track (sequentially estimate) the subspace spanned by the columns of $\boldsymbol{P}_j$ as well as the rank-r approximation, $\boldsymbol{L}_j := \boldsymbol{P}_j \boldsymbol{P}_j^\top \tilde{\boldsymbol{L}}_j$. As is well known from the Eckart-Young theorem, this minimizes $\|\tilde{\boldsymbol{L}}_j - \check{\boldsymbol{L}}\|_2$ over all rank r matrices $\check{\boldsymbol{L}}$. We will occasionally refer to $\boldsymbol{L}_j$ and its columns $\boldsymbol{\ell}_t$ as the *true data*.

Let $\boldsymbol{A}_j := \boldsymbol{P}_j^\top \tilde{\boldsymbol{L}}_j$ be the matrix of subspace coefficients along $\boldsymbol{P}_j$. Let $\boldsymbol{V}_j := \tilde{\boldsymbol{L}}_j - \boldsymbol{L}_j$ be the residual noise/error. Clearly, since

$$\tilde{\boldsymbol{L}}_j \overset{\text{SVD}}{=} [\boldsymbol{P}_j \underbrace{\boldsymbol{S}\boldsymbol{B}^\top}_{\boldsymbol{A}_j} + \underbrace{\boldsymbol{P}_{j,\perp}\boldsymbol{S}_\perp \boldsymbol{B}_\perp^\top}_{\boldsymbol{V}_j}] = \underbrace{\boldsymbol{P}_j \boldsymbol{A}_j}_{\boldsymbol{L}_j} + \boldsymbol{V}_j,$$

it is immediate that $\boldsymbol{L}_j \boldsymbol{V}_j^\top = 0$.

Let $\boldsymbol{a}_t$, $\boldsymbol{\ell}_t$ and $\boldsymbol{v}_t$ be the columns of $\boldsymbol{A}_j$, $\boldsymbol{L}_j$, and $\boldsymbol{V}_j$ respectively. Thus, for $t \in \mathcal{J}_j := [(j-1)\alpha + 1, (j-1)\alpha + 2, \dots j\alpha]$, $\boldsymbol{a}_t = \boldsymbol{P}_j^\top \tilde{\boldsymbol{\ell}}_t$, $\boldsymbol{\ell}_t = \boldsymbol{P}_j \boldsymbol{a}_t$, and $\boldsymbol{v}_t = \tilde{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t$.

Also, let $\mathcal{M}_t = (\Omega_t)^c$ be the index set of missing entries at time t. With this, we can rewrite (1) as

$$\begin{aligned} \boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\tilde{\boldsymbol{\ell}}_t) &= \tilde{\boldsymbol{\ell}}_t - \boldsymbol{I}_{\mathcal{M}_t} \boldsymbol{I}_{\mathcal{M}_t}{}^\top \tilde{\boldsymbol{\ell}}_t \\ &= \boldsymbol{\ell}_t + \boldsymbol{v}_t - \boldsymbol{I}_{\mathcal{M}_t} \boldsymbol{I}_{\mathcal{M}_t}{}^\top (\boldsymbol{\ell}_t + \boldsymbol{v}_t) \end{aligned}$$

[1] as we show later $\alpha \geq Cr \log n$ suffices

C. Robust ST-Miss (RST-Miss)

Robust ST-miss assumes that there can also be additive sparse outliers in the observed data $\boldsymbol{y}_t$. Thus, for all $t = 1, 2, \dots, d$,

$$\boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\tilde{\boldsymbol{\ell}}_t) + \boldsymbol{s}_t \tag{2}$$

where $\boldsymbol{s}_t$'s are the sparse outliers with supports $\mathcal{M}_{\text{sparse},t}$. The assumptions on Ω_t, and the true data, $\tilde{\boldsymbol{\ell}}_t$ remain as in the previous section. Due to space constraints, we provide the complete algorithm and guarantee for this problem in the supplementary material.

D. Federated Over-Air Data Sharing Constraints and Iteration Noise

We also solve RST-miss in a federated over-air fashion. Concretely, this means the following for an iterative algorithm. At iteration l, the central server broadcasts the $(l-1)$-th estimate of the quantity of interest[2] denoted $\hat{\boldsymbol{U}}_{l-1}$ to each of the K nodes. Each node then uses this estimate and its (locally) available data to compute the new local estimate denoted $\tilde{\boldsymbol{U}}_{k,l}$. The nodes then synchronously transmit these to the central server but the transmission is corrupted by channel noise and thus the central server receives

$$\tilde{\boldsymbol{U}}_l := \sum_k \tilde{\boldsymbol{U}}_{k,l} + \boldsymbol{W}_l$$

where $\boldsymbol{W}_l$ is the channel noise. We assume that $\boldsymbol{W}_l$ is independent of data and that each entry of $\boldsymbol{W}_l$ is i.i.d. zero-mean Gaussian with variance σ_c^2. The central server then processes $\tilde{\boldsymbol{U}}_l$ to get the new estimate of the quantity of interest, $\hat{\boldsymbol{U}}_l$ which is then broadcast to all K nodes for the next iteration. The presence of $\boldsymbol{W}_l$ in each iteration introduces a new and different set of challenges in algorithm design and analysis compared to what has been largely explored in existing literature.

III. ST from Missing Data (ST-miss)

A. Proposed Algorithm

Recall that we split our data into mini-batches of size α; thus $\boldsymbol{Y}_1 := [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots \boldsymbol{y}_\alpha]$, $\boldsymbol{Y}_2 := [\boldsymbol{y}_{\alpha+1}, \boldsymbol{y}_{\alpha+2}, \dots \boldsymbol{y}_{2\alpha}]$ and so on. Thus $\boldsymbol{Y}_j := [\boldsymbol{y}_{(j-1)\alpha+1}, \boldsymbol{y}_{(j-1)\alpha+2}, \dots, \boldsymbol{y}_{j\alpha}]$. Without the slow subspace change assumption, the obvious way to solve ST-miss would be to use what can be called *simple PCA*: for each mini-batch j, compute $\hat{\boldsymbol{P}}_j$ as the r-SVD of $\boldsymbol{Y}_j$. However, when slow subspace change is assumed, a better approach is a simplification of our algorithm from [23]. We initialize via r-SVD: compute $\hat{\boldsymbol{P}}_1$ as the r-SVD of $\boldsymbol{Y}_1$. For the j-th mini-batch, we first obtain *an* estimate of the missing entries for each column using the previous subspace estimate and projected Least Squares (LS) as follows. For every $t \in ((j-1)\alpha, j\alpha]$, we compute

$$\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t} \boldsymbol{\Psi}_{\mathcal{M}_t}^\dagger \boldsymbol{\Psi} \boldsymbol{y}_t \tag{3}$$

[2] The quantity of interest could be a vector or a matrix depending on the application. For the problem we study (subspace learning/tracking), the quantity of interest is a $n \times r$ basis matrix.

Algorithm 1: STMiss-NoDet.

Input: $\boldsymbol{Y}, \mathcal{M}$

Parameters: α

Initialize: $\hat{\boldsymbol{P}}_1 \leftarrow r$-SVD$[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\alpha]$, $j \leftarrow 2$

for $j \geq 2$ **do**

  Projected LS:

  $\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}^\top$

  for all $t \in ((j-1)\alpha, j\alpha]$ **do**

    $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{\Psi}_{\mathcal{M}_t})^\dagger(\boldsymbol{\Psi}\boldsymbol{y}_t)$

  end for

  PCA on $\hat{\boldsymbol{L}}_j$*:*

  $\hat{\boldsymbol{P}}_j \leftarrow r$-SVD$(\hat{\boldsymbol{L}}_j)$ where $\hat{\boldsymbol{L}}_j := [\hat{\boldsymbol{\ell}}_{(j-1)\alpha+1}, \ldots, \hat{\boldsymbol{\ell}}_{j\alpha}]$

  for all $t \in ((j-1)\alpha, j\alpha]$ **do** ▷optional

    $\tilde{\boldsymbol{\Psi}} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_j\hat{\boldsymbol{P}}_j^\top$

    $\hat{\tilde{\boldsymbol{\ell}}}_t \leftarrow \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t}(\tilde{\boldsymbol{\Psi}}_{\mathcal{M}_t})^\dagger(\tilde{\boldsymbol{\Psi}}\boldsymbol{y}_t)$

  end for

end for

Output: $\hat{\boldsymbol{P}}_j, \hat{\boldsymbol{\ell}}_t, \hat{\tilde{\boldsymbol{\ell}}}_t$.

where $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}^\top$. This step works as long as (i) the span of $\hat{\boldsymbol{P}}_{j-1}$ is a good estimate of that of $\boldsymbol{P}_j$ and (ii) $\boldsymbol{\Psi}_{\mathcal{M}_t}$ is well conditioned (or has full-column rank). We argue the first point by assuming that the span of $\hat{\boldsymbol{P}}_{j-1}$ is a good estimate of that of $\boldsymbol{P}_{j-1}$ and furthermore, owing to slow subspace change, it is also a good estimate of the span of $\boldsymbol{P}_j$. We ensure the second point by bounding the number of missing entries in each column, $|\mathcal{M}_t|$ in our main result. This point is further explained in Remark 3.8.

Observe that (3) is a compact way to write the following: $(\hat{\boldsymbol{\ell}}_t)_{\mathcal{M}_t^c} = (\boldsymbol{y}_t)_{\mathcal{M}_t^c} = (\tilde{\boldsymbol{\ell}}_t)_{\mathcal{M}_t^c}$ (use the observed entries as is) and $(\hat{\boldsymbol{\ell}}_t)_{\mathcal{M}_t} = \boldsymbol{\Psi}_{\mathcal{M}_t}^\dagger(\boldsymbol{\Psi}\boldsymbol{y}_t)$. To understand this, notice that $\boldsymbol{\Psi}\boldsymbol{y}_t = -\boldsymbol{\Psi}_{\mathcal{M}_t}\boldsymbol{z}_t + (\boldsymbol{\Psi}\boldsymbol{\ell}_t + \boldsymbol{\Psi}\boldsymbol{v}_t)$ where $\boldsymbol{z}_t := (\boldsymbol{I}_{\mathcal{M}_t}{}^\top\tilde{\boldsymbol{\ell}}_t)$ is the vector of missing entries. The second two terms can be treated as small "noise"/disturbance[3] and so we can compute an estimate of $\boldsymbol{z}_t$ from $\boldsymbol{\Psi}\boldsymbol{y}_t$ by LS.

The second step is to compute $\hat{\boldsymbol{P}}_j$ as the r-SVD of $\hat{\boldsymbol{L}}_j := [\hat{\boldsymbol{\ell}}_{(j-1)\alpha+1}, \ldots, \hat{\boldsymbol{\ell}}_{j\alpha}]$.

Finally, we can use $\hat{\boldsymbol{P}}_j$ to obtain an optional improved estimate, $\hat{\tilde{\boldsymbol{\ell}}}_t = \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t}\tilde{\boldsymbol{\Psi}}_{\mathcal{M}_t}^\dagger(\tilde{\boldsymbol{\Psi}}\boldsymbol{y}_t)$ where $\tilde{\boldsymbol{\Psi}} = \boldsymbol{I} - \hat{\boldsymbol{P}}_j\hat{\boldsymbol{P}}_j^\top$. We summarize this approach in Algorithm 1. We show next that, under slow subspace change, Algorithm 1 yields significantly better subspace estimates than simple PCA (PCA on each $\boldsymbol{Y}_j$).

B. Assumptions and Main Result

It is well known from the LRMC literature [10] that for guaranteeing correct matrix recovery, we need to assume incoherence (w.r.t. the standard basis) of the left and right singular vectors of the matrix. We need a similar assumption on $\boldsymbol{P}_j$'s.

[3]The first is small because of slow subspace change and $\hat{\boldsymbol{P}}_{j-1}$ being a good estimate (if span$(\hat{\boldsymbol{P}}_{j-1}) =$ span$(\boldsymbol{P}_j)$ this term would be zero); the second is small because $\|\boldsymbol{v}_t\|$ is small due to the approximate low-rank assumption.

Assumption 3.1 (μ-Incoherence of $\boldsymbol{P}_j$s): Assume that

$$\max_{j\in[d/\alpha]}\max_{m\in[r]}\|\boldsymbol{P}_j^{(m)}\|_2^2 \leq \frac{\mu r}{n}$$

where $\boldsymbol{P}_j^{(m)}$ denotes the m-th row of $\boldsymbol{P}_j$ and $\mu \geq 1$ is a constant (incoherence parameter).

Since we study a tracking algorithm (we want to track subspace changes quickly), we replace the standard right singular vectors' incoherence assumption with the following simple statistical assumption on the subspace coefficients $\boldsymbol{a}_t$. This helps us obtain guarantees on our mini-batch algorithm that operates on α-size mini-batches of the data.

Assumption 3.2 (Statistical μ-Incoherence of $\boldsymbol{a}_i$s): Recall that $\boldsymbol{a}_t = \boldsymbol{P}_j^\top\tilde{\boldsymbol{\ell}}_t$ for all $t \in \mathcal{J}_j$. Assume that the $\boldsymbol{a}_t$'s are zero mean; mutually independent; have identical diagonal covariance matrix λ, i.e., that $\mathbb{E}[\boldsymbol{a}_t\boldsymbol{a}_t^\top] = \lambda$ with λ diagonal; and are bounded, i.e., $\max_t\|\boldsymbol{a}_t\|^2 \leq \mu r\lambda^+$, where $\lambda^+ := \lambda_{\max}(\lambda)$ and $\mu \geq 1$ is a small constant. Also, let $\lambda^- := \lambda_{\min}(\lambda)$ and $f := \lambda^+/\lambda^-$.

If a few complete rows (columns) of the entries are missing, in general it is not possible to recover the underlying matrix. This can be avoided by either assuming bounds on the number of missing entries in any row and in any column, or by assuming that each entry is observed uniformly at random with probability ρ independent of all others. In this work we assume the former which is a weaker assumption. We need the following definition.

Definition 3.3 (Bounded Missing Entry Fractions): Consider the $n \times \alpha$ observed matrix $\boldsymbol{Y}_j$ for the j-th mini-batch of data. We use max-miss-frac-col (max-miss-frac-row) to denote the maximum of the fraction of missing entries in any column (row) of this matrix.

Owing to the assumption that $\tilde{\boldsymbol{L}}_j$ is approximately low-rank, it follows that $\tilde{\boldsymbol{L}}_j - \boldsymbol{L}_j := \boldsymbol{V}_j$ is "small".

Assumption 3.4 (Small, Bounded, Independent Modeling Error): Let $\lambda_v^+ := \max_t\|\mathbb{E}[\boldsymbol{v}_t\boldsymbol{v}_t^\top]\|$. We assume that $\lambda_v^+ < \lambda^-$, $\max_t\|\boldsymbol{v}_t\|^2 \leq Cr\lambda_v^+$ and $\boldsymbol{v}_t$'s are mutually independent over time.

1) Main Result: We have the following result for the naive algorithm of PCA on every mini-batch of α observed samples $\boldsymbol{Y}_j$. We use the following definition of noise level

$$\text{no-lev} := \sqrt{\lambda_v^+/\lambda^-}$$

Theorem 3.5 (STmiss Algorithm 3): Set algorithm parameter $\alpha = Cf^2r\log n$.

Assume that no-lev < 0.2 and the following hold:

1) **Incoherence:** $\boldsymbol{P}_j$'s satisfy μ-incoherence, and $\boldsymbol{a}_t$'s satisfy statistical right μ-incoherence;
2) **Missing Entries:** max-miss-frac-col $\leq \rho_{\text{col}}/(\mu r)$, max-miss-frac-row $\leq \rho_{\text{row}}/f^2$ s.t.,
$$7\sqrt{\rho_{\text{row}}\rho_{\text{col}}} + \text{no-lev}^2 \leq \max(\text{no-lev}, 0.25\sqrt{\rho_{\text{col}}})$$
3) **Modeling Error:** Assumption 3.4 holds.
4) **Subspace Change:** $\max_j \text{dist}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j) := \Delta_{tv}$, s.t. $\Delta_{tv} \leq$ no-lev and
$$\max(\text{no-lev}, 0.25\sqrt{\rho_{\text{col}}} + 3/7\Delta_{tv}) \leq \Delta_{red} < 1$$

then, with probability at least $1 - 10dn^{-10}$, we have

$$\begin{aligned}&\operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j)\\ &\leq \max\\ &(\sqrt{\rho_{\text{col}}} \cdot 0.3^{j-1} + \Delta_{tv}(\Delta_{red} + \Delta_{red}^2 \ldots + \Delta_{red}^{j-1}), \text{no-lev})\\ &< \max\left(\sqrt{\rho_{\text{col}}} \cdot \Delta_{red}^{j-1} + \frac{\Delta_{red}\Delta_{tv}}{1-\Delta_{red}}, \text{no-lev}\right)\end{aligned}$$

Also, at all j, and for $t \in [(j-1)\alpha, j\alpha)$, $\|\hat{\tilde{\boldsymbol{\ell}}}_t - \tilde{\boldsymbol{\ell}}_t\| \leq 1.2 \cdot \operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j)\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\|$ while $\|\hat{\boldsymbol{\ell}}_t - \tilde{\boldsymbol{\ell}}_t\| \leq 1.2 \cdot \operatorname{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_j)\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\| \leq 1.2 \cdot (\Delta_{tv} + \operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j))\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\|$.

Proof: See Section III-E.

C. Discussion

First consider the noiseless setting, i.e., the data is exactly rank-r. The condition on the missing entries requires that $\rho_{\text{row}} \leq (0.25/7)^2 \leq 0.16$. While this might seem restrictive at first glance, these constants can be varied by modifying the PCA-SDDN result (Corollary 3.10). Next, from the final subspace error expression, note that $\Delta_{red} < 1$, governs the rate at which the error decays. As expected, increasing the number of missing entries in a column (proportional to ρ_{col}), or the maximum amount of subspace change, Δ_{tv} reduces the convergence rate to an ϵ-accurate solution. In the presence of noise, without further assumptions on $\boldsymbol{v}$, in general it is not possible to obtain a final error that is lower than no-lev and in this case, the tradeoffs are not as straightforward.

For ease of notation, we provide a special case of Theorem 3.5 with specific values of the various constants next.

Theorem 3.6 (STmiss – Special Case): Set algorithm parameter $\alpha = Cf^2 r \log n$.

Assume that no-lev < 0.2 and the following hold:

1) **Incoherence:** $\boldsymbol{P}_j$'s satisfy μ-incoherence, and $\boldsymbol{a}_t$'s satisfy statistical right μ-incoherence;
2) **Missing Entries:** max-miss-frac-col $\leq 0.01/(\mu r)$, max-miss-frac-row $\leq 0.0001/f^2$;
3) **Modeling Error:** Assumption 3.4 holds
4) **Subspace Change:** $\max_j \operatorname{dist}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j) \leq \Delta_{tv} = 0.1$,

then, with probability at least $1 - 10dn^{-10}$, we have

$$\begin{aligned}&\operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j)\\ &\leq \max(0.1 \cdot 0.3^{j-1} + \Delta_{tv}(0.3 + 0.3^2 \ldots + 0.3^{j-1}), \text{no-lev})\\ &< \max(0.1 \cdot 0.3^{j-1} + 0.5\Delta_{tv}, \text{no-lev})\end{aligned}$$

Also, at all j, and for $t \in [(j-1)\alpha, j\alpha)$, $\|\hat{\tilde{\boldsymbol{\ell}}}_t - \tilde{\boldsymbol{\ell}}_t\| \leq 1.2 \cdot \operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j)\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\|$ while $\|\hat{\boldsymbol{\ell}}_t - \tilde{\boldsymbol{\ell}}_t\| \leq 1.2 \cdot \operatorname{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_j)\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\| \leq 1.2 \cdot (\Delta_{tv} + \operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j))\|\tilde{\boldsymbol{\ell}}_t\| + \|\boldsymbol{v}_t\|$.

Proof: See Section III-E.

In the sequel, we only build upon the special case, Theorem 3.6. As a baseline, consider the following naive approach to solve the ST-miss problem and its associated result:

Theorem 3.7 (Simple PCA): Let $\hat{\boldsymbol{P}}_j$ be the r-SVD of $\boldsymbol{Y}_j$ with $\alpha = Cf^2 r \log n$. Assume μ-incoherence of $\boldsymbol{P}_j$s, statistical μ-incoherence of $\boldsymbol{a}_i$s, modeling error assumption given in Assumption 3.4, max-miss-frac-col $\leq 0.01/(\mu r)$, max-miss-frac-row $\leq 0.01/f^2$. Then, with probability at least $1 - 10dn^{-10}$,

$$\operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) \leq \max(0.1 \cdot 0.25, \text{no-lev})$$

Proof: The proof is the same as that for the initialization step of Algorithm 1; see Section III-E.

To compare our main result, Theorem 3.6, consider the practically relevant setting of approximately rank r $\tilde{\boldsymbol{L}}_j$'s so that the noise level $\sqrt{\lambda_v^+/\lambda^-}$ is small. In particular, assume it is smaller than $0.1 \cdot 0.25$. Then, if Δ_{tv} is small enough, the bound of Theorem 3.6 is significantly smaller. If the noise level is larger, then in both cases, the noise level term dominates and both results give the same bound. Thus, in all cases, as long as Δ_{tv} is small (slow subspace change holds), Theorem 3.6 gives an as good or better bound as the naive approach. We demonstrate this in Fig. 1.

Note: Our result assumes a mix of deterministic and stochastic assumptions due to the following. As mentioned earlier, Algorithm 1 is modification of an algorithm for RST-miss from [37] in which we treat *missing entry recovery* as a special case of sparse recovery using ideas from Compressive Sensing (CS), and the subspace update step as a Dynamic PCA problem. For the CS problem, we require a (deterministic) bound on the number of non-zero entries (missing entries) but not on *how* the support set is generated, i.e., we can tolerate deterministic patterns on set of missing entries. Furthermore, for the sparse recovery step to work, the CS result [38] requires that the $2|\mathcal{M}_t|$-level incoherence of the measurement matrix, $\boldsymbol{\Psi}_{\mathcal{M}_t}$ is bounded by $\sqrt{2} - 1$. This translates to our incoherence bounds on the subspaces. For the dynamic PCA problem, it is customary to impose stochastic assumptions on either the subspaces or the subspace coefficients and in this paper, we choose the latter. For a detailed comparison with the best known results in subspace tracking, we refer the reader to [23, Sec. III].

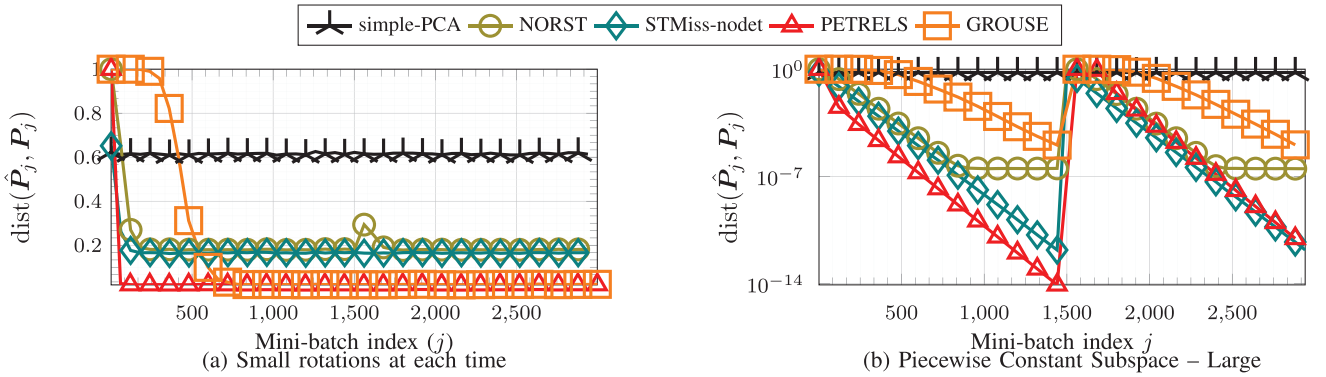
Remark 3.8 (Demonstrating full Column-Rank of $\boldsymbol{\Psi}_{\mathcal{M}_t}$): Notice that $\boldsymbol{\Psi}_{\mathcal{M}_i} \in \mathbb{R}^{n \times |\mathcal{M}_t|}$ and under the conditions of Theorem 3.6, we assume that (for some $c < 1$) $|\mathcal{M}_i| \leq cn/(\mu r)$ and thus for $\boldsymbol{\Psi}_{\mathcal{M}_i}$ to have full column rank, one needs,

$$\frac{cn}{\mu r} \leq n - r \Rightarrow \mu \geq \frac{cn}{r(n-r)}$$

notice that $r(n-r) \in [0, n^2/4]$ and thus one only needs that $\mu \geq 4c/n$. Now, as long as $c \in [0, 1/4]$, this bound is satisfied for all n since $\mu \geq 1$ by definition. If $c \in (1/4, 1]$, as long as $n > 4/c$, the condition is again satisfied. In other words, the matrix $\boldsymbol{\Psi}_{\mathcal{M}_i}$ has full-column rank for all $n > 4$.

D. Guarantee for Piecewise Constant Subspace Change

Previous work on provable ST-miss [23] assumed piecewise constant subspace change (required the subspace to be constant for long enough), but did not require an upper bound on the amount of change. As we show next STmiss-NoDet is able



(a) Small rotations at each time (b) Piecewise Constant Subspace – Large

Fig. 1. Comparison of ST-Miss Algorithms in the centralized setting.

to track such changes as well and provide similar tracking guarantees even under a (mild) generalization of the previous model.

Theorem 3.9: Set algorithm parameter $\alpha = Cf^2 r \log n$. Assume that no-lev < 0.02 and the first three assumptions of Theorem 3.6 hold. Under an approximately piecewise constant subspace change model ($\Delta_j \leq$ no-lev for all j except for $j = j_\gamma$, for $\gamma = 1, 2, \ldots,$) with the subspace change times satisfying $j_\gamma - j_{\gamma-1} > K := C \log(1/\text{no-lev})$, then, w.p. at least $1 - dn^{-10}$,

$$\text{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) \leq$$

$$\begin{cases} (0.2 + 2\text{no-lev}) \cdot 0.25 + \text{no-lev}), & \text{if} \quad j = j_\gamma \\ (0.2 + 2\text{no-lev}) \cdot 0.3^{(j-j_\gamma)-1} + \text{no-lev}, & \text{if} \quad j_\gamma < j < j_{\gamma+1} \end{cases}$$

Notice that for $j_{\gamma+1} > j > j_\gamma + K$, the bound is at most 2no-lev.

The subspace change model in this result does not require an upper bound on the amount of subspace change as long as the change occurs infrequently. However, it still allows for small rotations to the subspace at each time. The exponential decay in the subspace recovery error bound is the same as that guaranteed by the results is [23]. STmiss-NoDet does not detect subspace changes. However, a detection step similar to that used in previous work can be included and then a similar detection guarantee can also be proved. We provide these in the Supplementary Material (https://arxiv.org/abs/2002.12873).

E. Proof of Theorem 3.6 and 3.7

The proof follows by a careful application of a result from [37] that analyzes PCA in sparse data-dependent noise (SDDN) along with simple linear algebra tricks, some of which are also borrowed from there. The novel contribution here is the application of the same ideas for providing a result that holds under a much simpler and practically valid assumption of slow changing subspaces (without any artificial piecewise constant assumption). Also, the proof provided here is much shorter.

1) Subspace Error Bounds: Consider the projected LS step. Recall that $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}^\top$. Since $\boldsymbol{y}_t$ can be expressed as $\boldsymbol{y}_t = \tilde{\boldsymbol{\ell}}_t - \boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{I}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t)$, using the idea explained while developing the algorithm,

$$\begin{aligned} \hat{\boldsymbol{\ell}}_t &= \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{\Psi}_{\mathcal{M}_t}^\dagger \boldsymbol{\Psi}(-\boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{I}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t + \tilde{\boldsymbol{\ell}}_t) \\ &= \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}(-\boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{I}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t + \tilde{\boldsymbol{\ell}}_t) \\ &= \boldsymbol{y}_t + \boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{I}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t - \boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t \\ &= \tilde{\boldsymbol{\ell}}_t - \boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \tilde{\boldsymbol{\ell}}_t \\ &= \boldsymbol{\ell}_t + \boldsymbol{v}_t - \boldsymbol{I}_{\mathcal{M}_t}\left(\boldsymbol{\Psi}_{\mathcal{M}_t}\right)^\dagger \boldsymbol{\Psi}_{\mathcal{M}_t}^\top (\boldsymbol{\ell}_t + \boldsymbol{v}_t) \end{aligned}$$

This final expression can be reorganized as follows.

$$\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t + \underbrace{\boldsymbol{v}_t - \boldsymbol{I}_{\mathcal{M}_t}\left(\boldsymbol{\Psi}_{\mathcal{M}_t}\right)^\dagger \boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{v}_t}_{\text{small, unstructured noise}} - \underbrace{\boldsymbol{I}_{\mathcal{M}_t}\left(\boldsymbol{\Psi}_{\mathcal{M}_t}\right)^\dagger \boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\ell}_t}_{\text{sparse, data dependent noise}}$$

$$:= \boldsymbol{\ell}_t + \boldsymbol{e}_t \tag{4}$$

Thus, recovering $\boldsymbol{P}_j$ from estimates $\hat{\boldsymbol{L}}_j$ is a problem of PCA in sparse data-dependent noise (SDDN): the "noise" $\boldsymbol{e}_t$ consists of two terms, the first is just small unstructured noise (depends on $\boldsymbol{v}_t$) while the second is sparse with support $\mathcal{M}_t$ and depends linearly on the true data $\boldsymbol{\ell}_t$. We studied PCA-SDDN in detail in [37] where we showed the following.

Lemma 3.10 (PCA-SDDN): For $i = 1, \ldots, \alpha$, assume that $\boldsymbol{z}_i = \boldsymbol{\ell}_i + \boldsymbol{w}_i + \boldsymbol{v}_i$ with $\boldsymbol{w}_i = \boldsymbol{I}_{\mathcal{M}_i}\boldsymbol{B}_i\boldsymbol{\ell}_i$ being sparse, data-dependent noise with support $\mathcal{M}_i$; $\boldsymbol{\ell}_i = \boldsymbol{P}\boldsymbol{a}_i$ with $\boldsymbol{P}$ being an $n \times r$ basis matrix that satisfies μ-incoherence, and $\boldsymbol{a}_i$'s satisfy statistical μ-incoherence; and $\boldsymbol{v}_i$ is small bounded noise with $\lambda_v^+ := \|\mathbb{E}[\boldsymbol{v}_i\boldsymbol{v}_i^\top]\| < \lambda^-$ and $\max_i \|\boldsymbol{v}_i\|^2 \leq Cr_v\lambda_v^+$. Let $q := max_i\|\boldsymbol{B}_i\boldsymbol{P}\|$ and let b be the maximum fraction of nonzeros in any row of the matrix $[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_\alpha]$. Let $\hat{\boldsymbol{P}}$ be the matrix of top r eigenvectors of $\frac{1}{\alpha}\sum_i \boldsymbol{z}_i\boldsymbol{z}_i^\top$. Assume that $q \leq 3$. Pick an $\epsilon > 0$. If

$$7\sqrt{b}qf + \frac{\lambda_v^+}{\lambda^-} < 0.4\epsilon, \text{ and} \tag{5}$$

$$\alpha \geq \alpha^* := C \max\left(\frac{q^2 f^2}{\epsilon^2} r \log n, \frac{\frac{\lambda_v^+}{\lambda^-} f}{\epsilon^2} r \log n\right), \tag{6}$$

then, w.p. at least $1 - 10n^{-10}$, $\text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) \leq \epsilon$.

This result says that, under the incoherence assumptions, and assuming that the unstructured noise satisfies the stated

assumptions, if the support of the SDDN, $\boldsymbol{w}_i$, changes enough over time so that b, which is the maximum fraction of nonzeros in any row of the matrix $[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_\alpha]$, is sufficiently small, if the unstructured noise power is small enough compared to the r-th eigenvalue of the true data covariance matrix and it is bounded with small effective dimension, $\|\boldsymbol{v}_i\|^2/\lambda_v^+ \leq Cr$, and if α is large enough, then span($\hat{\boldsymbol{P}}$), is a good approximation of span($\boldsymbol{P}$). Notice here that for SDDN, the true data and noise correlation, $\mathbb{E}[\boldsymbol{\ell}_i \boldsymbol{w}_i{}^\top]$, is not zero, and the noise power, $\mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i{}^\top]$, itself is also not small. However, the key idea used to obtain this result is the following: enough support changes over time (small b) helps ensure that the upper bounds on sample averaged values of both these quantities, $\|(1/\alpha)\sum_i \mathbb{E}[\boldsymbol{\ell}_i \boldsymbol{w}_i{}^\top]\|$ and $\|(1/\alpha)\sum_i \mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i{}^\top]\|$ are $\sqrt{b}$ times smaller than those on their maximum instantaneous values, $\|\mathbb{E}[\boldsymbol{\ell}_i \boldsymbol{w}_i{}^\top]\|$ and $\|\mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i{}^\top]\|$.

Our proof uses Lemma 3.10 applied on the j-th mini-batch of estimates, $\hat{\boldsymbol{L}}_j$ along with the following simple facts.

Fact 3.11:

1) From [5, Remark 3.6] we have: let $\boldsymbol{P}$ be an μ-incoherent, $n \times r$ basis matrix. Then, for any set $\mathcal{M} \subseteq [n]$, we have
$$\|\boldsymbol{I}_{\mathcal{M}}^\top \boldsymbol{P}\|^2 \leq |\mathcal{M}| \cdot \frac{\mu r}{n}$$
2) For $n \times r$ basis matrices $\boldsymbol{P}$, $\hat{\boldsymbol{P}}$ (useful when the column span of $\hat{\boldsymbol{P}}$ is a good approximation of that of $\boldsymbol{P}$), and any set $\mathcal{M} \subseteq [n]$, we have
$$\|\boldsymbol{I}_{\mathcal{M}}^\top \hat{\boldsymbol{P}}\| \leq \text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) + \|\boldsymbol{I}_{\mathcal{M}}^\top \boldsymbol{P}\|$$
3) For a μ-incoherent $n \times r$ basis matrix, $\boldsymbol{P}$, and any set $\mathcal{M} \subseteq [n]$,
$$\lambda_{\min}(\boldsymbol{I}_{\mathcal{M}}{}^\top (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\top)\boldsymbol{I}_{\mathcal{M}}) = 1 - \|\boldsymbol{I}_T{}^\top \boldsymbol{P}\|^2$$

Thus, combining the above three facts,
$$\|(\boldsymbol{I}_{\mathcal{M}}{}^\top (\boldsymbol{I} - \hat{\boldsymbol{P}}\hat{\boldsymbol{P}}^\top)\boldsymbol{I}_{\mathcal{M}})^{-1}\| \leq \frac{1}{1 - (\text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) + \sqrt{|\mathcal{M}|\mu r/n})^2}$$

The proof for $j = 1$ is a little different from $j > 1$. For $j = 1$, $\boldsymbol{\Psi} = \boldsymbol{I}$ and $\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t$. Also, $i = t$. For $j > 1$, $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}{}^\top$ and $i = t - (j-1)\alpha$. Consider $j = 1$ (initialization). In this case, $\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t$ satisfies (4) with $\boldsymbol{\Psi} = \boldsymbol{I}$. We apply Lemma 3.10 with $i = t$, $\boldsymbol{z}_i \equiv \hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t$, $\boldsymbol{\ell}_i \equiv \boldsymbol{\ell}_t$, $\boldsymbol{P} \equiv \boldsymbol{P}_1$, $\boldsymbol{w}_i \equiv -\boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{\ell}_t$, $\boldsymbol{v}_i \equiv \boldsymbol{v}_t - \boldsymbol{I}_{\mathcal{M}_t}\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{v}_t$, $\boldsymbol{B}_i \equiv \boldsymbol{I}_{\mathcal{M}_t}^\top$. Notice that the fraction of non-zeros in the matrix $[\boldsymbol{w}_1, \cdots \boldsymbol{w}_\alpha]$ is bounded by max-miss-frac-row and thus $b \equiv$ max-miss-frac-row. To obtain q, we need to bound $\max_{t \in \mathcal{J}_1} \|\boldsymbol{B}_t \boldsymbol{P}_1\| = \max_{t \in \mathcal{J}_1} \|\boldsymbol{I}_{\mathcal{M}_t}{}^\top \boldsymbol{P}_1\|$. By item 1 of Fact 3.11, $\|\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{P}_1\|^2 \leq |\mathcal{M}_t|\mu r/n \leq$ max-miss-frac-col $\cdot\, n\mu r/n$. Under the assumptions of Theorem 3.6, |max-miss-frac-col| $\leq \rho_{\text{col}}/\mu r$ and thus $\max_t \|\boldsymbol{B}_t \boldsymbol{P}\| \leq \sqrt{\rho_{\text{col}}} = q_1 \equiv q$. We pick $\epsilon = \max(\text{no-lev}, 0.25q_1)$. From the Theorem assumptions (missing entry fractions), $b =$ max-miss-frac-row $\leq \rho_{\text{row}}/f^2$ and no-lev ≤ 0.2 and so (5) is satisfied. Furthermore, since $\epsilon = \max(\text{no-lev}, 0.25q_1)$, the value of α used in the Theorem satisfies the requirements of Lemma 3.10. Thus, we can apply this lemma to conclude that $\text{dist}(\hat{\boldsymbol{P}}_1, \boldsymbol{P}_1) \leq \epsilon = \max(\text{no-lev}, 0.25q_1)$ with $q_1 = 0.1 = \sqrt{\rho_{\text{col}}}$. This completes the proof of Theorem 3.7 since simple-PCA just repeats this step at each j.

Now consider any $j > 1$. We claim that for $j > 1$,
$$\text{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) \leq \epsilon_j$$
with ϵ_j satisfying the following recursion: $\epsilon_1 = \max(\text{no-lev}, 0.25q_1)$ with $q_1 = 0.1$, and
$$\epsilon_j = \max(\text{ no-lev},\ 0.25 \cdot 1.2 \cdot (\epsilon_{j-1} + \Delta_{tv})) \tag{7}$$

This can be simplified to show that
$$\begin{aligned}\epsilon_j &\leq \max\left(\text{no-lev}, (\text{no-lev} + \Delta_{tv})\sum_{j'=1}^{j-1}(0.3)^{j'}, \right. \\ &\qquad \left. 0.3^j \cdot 0.25q_1 + \Delta_{tv}\sum_{j'=1}^{j-1}(0.3)^{j'}\right) \\ &\leq \max\left(\text{no-lev}, 0.3^j(0.25q_1) + \Delta_{tv}\sum_{j'=1}^{j-1}(0.3)^{j'}\right) \end{aligned} \tag{8}$$
where the second inequality follows by using $\Delta_{tv} \leq$ no-lev and $\sum_{j'=1}^{j-1}(0.3)^{j'} \leq \sum_{j'=1}^{\infty}(0.3)^{j'} = 3/7$.

To prove the above claim, we use induction. Base case: $j = 1$ done above. Induction assumption: assume $\text{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_{j-1}) \leq \epsilon_{j-1}$. The application of the PCA-SDDN lemma is similar to that for $j = 1$ with the difference being that $i = t - (j-1)\alpha$ and $\boldsymbol{B}_i$ is different now. We now have $\boldsymbol{B}_i \equiv (\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top$ and so $\max_{t \in \mathcal{J}_j} \|\boldsymbol{B}_t \boldsymbol{P}\| = \max_t \|(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{P}_j\|$. This can be bounded using Fact 3.11 as follows
$$\begin{aligned}&\max_t \|(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{P}_j\| \\ &\leq \max_t \|(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\|\|\boldsymbol{I}_{\mathcal{M}_t}^\top\|\|\boldsymbol{\Psi}\boldsymbol{P}_j\| \\ &\leq \frac{1}{1 - (\epsilon_{j-1} + \sqrt{0.01})^2} \cdot 1 \cdot \text{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_j) \\ &\leq \frac{1}{1 - (\epsilon_{j-1} + \sqrt{0.01})^2}(\epsilon_{j-1} + \Delta_{tv}) := q_j\end{aligned}$$

Using (8), $\epsilon_{j-1} \leq \max(\text{no-lev}, 0.25q_1 + \Delta_{tv}(3/7))$ and recalling that $\max(\text{no-lev}, 0.35\sqrt{\rho_{\text{col}}} + \Delta_{tv}(3/7)) < 0.3$. Using this upper bound on ϵ_{j-1} in the denominator expression of above,
$$q_j \leq 1.2(\epsilon_{j-1} + \Delta_{tv}) \tag{9}$$

Apply the PCA-SDDN lemma with $q \equiv q_j$ and $\epsilon = \max(\text{no-lev}, 0.25q_j)$. With this choice of ϵ, it is easy to see that $7\sqrt{b}q_j f + \frac{\lambda_v^+}{\lambda^-} \leq 0.4\epsilon$. Also, α given in the Theorem again satisfies the requirements of the lemma. Applying the PCA-SDDN lemma, and using (9) to bound $q \equiv q_j$,
$$\begin{aligned}\text{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) &\leq \max(\text{no-lev}, 0.25q_j) \\ &\leq \max(\text{no-lev}, 0.25 \cdot 1.2(\epsilon_{j-1} + \Delta_{tv})) = \epsilon_j\end{aligned}$$
This proves our claim.

2) Bounds on Error in Estimating $\tilde{\boldsymbol{\ell}}_t$: From (4), $\hat{\boldsymbol{\ell}}_t - \tilde{\boldsymbol{\ell}}_t = -\boldsymbol{I}_{\mathcal{M}_t}(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}\tilde{\boldsymbol{\ell}}_t$ with $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}{}^\top$ for

$t \in \mathcal{J}_j$. Using this, $\tilde{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t + \boldsymbol{v}_t = \boldsymbol{P}_j \boldsymbol{a}_t + \boldsymbol{v}_t$, and Fact 3.11, we can get

$$\begin{aligned}\|\hat{\boldsymbol{\ell}}_t - \tilde{\boldsymbol{\ell}}_t\| &\le \operatorname{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_j)\|\boldsymbol{\ell}_t\| + \|\boldsymbol{v}_t\| \\ &\le (\epsilon_{j-1} + \Delta_{tv})\|\boldsymbol{\ell}_t\| + \|\boldsymbol{v}_t\|\end{aligned}$$

Using the same approach that we used to derive (4), we get that $\hat{\tilde{\boldsymbol{\ell}}}_t - \tilde{\boldsymbol{\ell}}_t$ has the same expression as $\hat{\boldsymbol{\ell}}_t - \tilde{\boldsymbol{\ell}}_t$ but with $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_j \hat{\boldsymbol{P}}_j{}^\top$ for $t \in \mathcal{J}_j$. Thus,

$$\|\hat{\tilde{\boldsymbol{\ell}}}_t - \tilde{\boldsymbol{\ell}}_t\| \le \operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j)\|\boldsymbol{\ell}_t\|_2 + \|\boldsymbol{v}_t\| \le \epsilon_{j-1}\|\boldsymbol{\ell}_t\| + \|\boldsymbol{v}_t\|$$

F. Proof of Theorem 3.9

The proof again follows by using the PCA-SDDN lemma given above along with use of Fact 3.11. The main difference is the use of the following idea.

Consider the interval just before the subspace change, i.e., the j-th interval with $j = j_\gamma - 1$. At this time, by our delay assumption, $\operatorname{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) \le$ 2no-lev and thus, using Fact 3.11, $\|\boldsymbol{I}_{\mathcal{M}_t}{}^\top \hat{\boldsymbol{P}}_j\| \le$ 2no-lev $+ 0.1$. Also, using Fact 3.11,

$$\begin{aligned}&\max_t \|(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1} \boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{P}_j\| \\ &\quad\le \max_t \|(\boldsymbol{\Psi}_{\mathcal{M}_t}^\top \boldsymbol{\Psi}_{\mathcal{M}_t})^{-1}\| \|\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{\Psi} \boldsymbol{P}_j\| \\ &\quad\le \frac{1}{1 - (\text{2no-lev} + 0.1)^2} \cdot (\|\boldsymbol{I}_{\mathcal{M}_t}^\top \hat{\boldsymbol{P}}_{j-1}\| + \|\boldsymbol{I}_{\mathcal{M}_t}^\top \boldsymbol{P}_j\|) \\ &\quad\le \frac{1}{1 - (\text{2no-lev} + 0.1)^2} \cdot ((0.1 + \text{2no-lev}) + 0.1)\end{aligned}$$

Combining with the bound from the previous section, the final bound for this term is

$$\frac{\min(\operatorname{dist}(\hat{\boldsymbol{P}}_{j-1}, \boldsymbol{P}_j), ((0.1 + \text{2no-lev}) + 0.1))}{1 - (\text{2no-lev} + 0.1)^2}$$

IV. Fedrated Over-Air Robust ST-Miss

In this section, we study robust ST-miss in the federated, over-air learning paradigm. There are two important distinctions with respect to the centralized ST-miss problem from Section III namely (a) data is now available across different nodes and the proposed algorithm must obey the federated data sharing constraints and (b) the proposed algorithm must be able to deal with gross and sparse outliers.

An example where such a problem formulation is valid is as follows. Consider the recommendation system design problem. Assume that there are n products and a total of d users/buyers distributed across a geographical area. The "products" could be movies, news sites, Facebook pages, blogs or even survey questions. A subset of d_k users sends their "ratings" of these products to worker node k. There are a total of K worker nodes. The master node would like to compute a low-dimensional subspace approximation of the $n \times d$ ratings' matrix, denoted by $\boldsymbol{Y}$, in order to use this information to recommend relevant movies to them. Note that the dataset is also potentially dynamic; every day new users enter the system and provide more ratings of the movies or the news sites or blogs. Thus, at time t, across all users, we have an $n \times \alpha$ data matrix $\boldsymbol{Y}_{(t)}$. This typically has many missing entries (set to zero), and gross outliers (that arise either from unintentional rating mistakes, or presence of malicious users). Collating all such matrices together we have a very big $n \times d$ matrix with $d = t\alpha$ at time t. The goal is to track the underlying true data subspace at each time t; this could be fixed or slow time varying. The assumption here is that user preferences are actually governed by a small number of factors r; this number is much smaller than the number of products n or the total number of users d.

A key observation that allows us to build upon Section III is that only Line 10 of Algorithm 1 needs to be federated (all other operations are performed locally on each vector). To this end, we first explain why tackling iteration noise is sufficient to satisfy the Fed-OA constraints in Section IV-A, we then present our result for PCA in the Fed-OA setting in Section IV-B (federated version of Line 10 of Algorithm 1), and finally show how this is used to develop an algorithm that solves Robust ST-Miss in the Fed-OA setting in Section IV-C.

A. Dealing With Mild Asynchrony and Channel Fading

As discussed previously, the three key challenges while working with over-air aggregation are (a) small timing mismatches, (b) channel fading, and (c) iteration noise. There exist a plethora of techniques within physical layer communications for dealing with channel fading and mild asynchrony. The main idea is to use carefully designed pilot sequences. Pilot sequences are symbols that the transmitter-receiver pairs agree on in advance and are transmitted in the beginning of a data frame. For instance, suppose that there are only $K = 2$ transmitters and the relative offsets between the transmitters is at most j symbols. In this case, both transmitters can use pilot sequences of length $2j + 1$, $[a_1, a_1, \ldots, a_1]$ and $[a_2, a_2, \ldots, a_2]$ respectively. Since the offset is at most j, the central node receives at least one symbol with values $a_1 + a_2$. It can determine the relative offset by determining the start location of the value $a_1 + a_2$. Once the estimated offset is communicated back to the nodes, the center can then receive the correct sum by having the nodes appropriately zero pad their transmissions. Extensions of these ideas can be utilized to handle the case of $K > 2$ nodes. Similarly, some amount of channel fading can compensated for by estimating the fading coefficients which can be done since the values of the pilot symbols are assumed to be known. These techniques are by now quite well-known in the single and multiple antenna scenarios [21]. As correctly noted by anonymous reviewer, it may be impossible to compensate for a very weak channel gain since that would require a transmit power that's above the limit. Thus, the main problem to be addressed is iteration noise which is the focus of this paper.

B. Federated Over-Air PCA Via the Power Method (PM)

Here we provide a result for subspace learning while obeying the federated data sharing constraints.

1) Problem Setting: The goal of PCA (subspace learning) is to compute an r-dimensional subspace approximation in which a given data matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ approximately lies. The k-th node

observes a columns' sub-matrix $\boldsymbol{Z}_k \in \mathbb{R}^{n\times d_k}$. We have $\boldsymbol{Z} := [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_k, \ldots, \boldsymbol{Z}_K] \in \mathbb{R}^{n\times d}$ with $d = \sum_{k=1}^{K} d_k$ and the goal of PCA is to find an $n \times r$ basis matrix $\boldsymbol{U}$ that minimizes $\|\boldsymbol{Z} - \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{Z}\|_F^2$. As is well known, the solution, $\boldsymbol{U}$, is given by the top r eigenvectors of $\boldsymbol{Z}\boldsymbol{Z}^\top$. Thus the goal is to estimate the span of $\boldsymbol{U}$ in a federated over-air (FedOA) fashion.

2) Federated Over-Air Power Method (FedOA-PM): The simplest algorithm for computing the top eigenvectors is the Power Method (PM) [39]. The distributed PM is well known, but most previous works assume the iteration-noise-free setting, e.g., see the review in [12]. On the other hand, there is recent work that studies the iteration-noise-corrupted PM [26], [27] but in the centralized setting. In this line of work, the authors consider two models for iteration-noise. The noise could either be deterministic, or statistical noise could be added to ensure differential privacy. Our setting is easier than the deterministic noise model, since we assume a statistical channel noise model, but is harder than the privacy setting since we do not have control over the amount of noise observed at the central server (here use the term channel noise and iteration-noise interchangeably).

The vanilla PM estimates $\boldsymbol{U}$ by iteratively updating $\tilde{\boldsymbol{U}}_l = \boldsymbol{Z}\boldsymbol{Z}^\top \hat{\boldsymbol{U}}_{l-1}$ followed by QR decomposition to get $\hat{\boldsymbol{U}}_l$. FedOA-PM approximates this computation as follows. At iteration l, each node k computes $\tilde{\boldsymbol{U}}_{k,l} := \boldsymbol{Z}_k \boldsymbol{Z}_k^\top \hat{\boldsymbol{U}}_{l-1}$ and synchronously transmits it to the central server which receives the sum corrupted by channel noise, i.e., it receives

$$\tilde{\boldsymbol{U}}_l := \sum_{k=1}^{K} \tilde{\boldsymbol{U}}_{k,l} + \boldsymbol{W}_l = \boldsymbol{Z}\boldsymbol{Z}^\top \hat{\boldsymbol{U}}_{l-1} + \boldsymbol{W}_l.$$

since $\sum_k \boldsymbol{Z}_k \boldsymbol{Z}_k^\top = \boldsymbol{Z}\boldsymbol{Z}^\top$. Here $\boldsymbol{W}_l$ is the channel noise. It then computes a QR decomposition of $\tilde{\boldsymbol{U}}_l$ to get a basis matrix $\hat{\boldsymbol{U}}_l$ which is broadcast to all the K nodes for use in the next iteration. We summarize this complete FedOA-PM algorithm in Algorithm 2. If no initialization is available, it starts with a random initialization. When we use FedOA-PM for subspace tracking in the next section, the input will be the subspace estimate from the previous time instant.

We use σ_i to denote the i-th largest eigenvalue of $\boldsymbol{Z}\boldsymbol{Z}^\top$, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$. We have the following guarantee for Algorithm 2.

Lemma 4.1 (FedOA-PM): Consider Algorithm 2. Pick the desired final accuracy $\epsilon \in (0, 1/3)$. Assume that, at each iteration, the channel noise $\boldsymbol{W}_l \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_c^2)$ with (i) $\sigma_c < \epsilon\sigma_r/(5\sqrt{n})$ and (ii) $R := \sigma_{r+1}/\sigma_r < 0.99$.

When using random initialization, if the number of iterations, $L = \Omega(\frac{1}{\log(1/R)} \log(\frac{nr}{\epsilon}))$. then, with probability at least $0.9 - L\exp(-cr)$, $\mathrm{dist}(\boldsymbol{U}, \hat{\boldsymbol{U}}_L) \leq \epsilon$.

When using an available initialization with $\mathrm{dist}(\hat{\boldsymbol{U}}_0, \boldsymbol{U}) < \epsilon_0$, if $L = \Omega(\frac{1}{\log(1/R)} \log(\frac{1}{\epsilon\sqrt{1-\epsilon_0^2}}))$, then, with probability at least $1 - L\exp(-cr)$, $\mathrm{dist}(\boldsymbol{U}, \hat{\boldsymbol{U}}_L) \leq \epsilon$.

Lemma 4.1 is similar to the one proved in [26], [27] for private PM but with a few key differences which we discuss in the Supplementary Material (Appendix D) due to space constraints. We also provide a guarantee for the convergence of the maximum eigenvalue (Lines $10-13$ of Algorithm 2) below.

Algorithm 2: FedOA-PM: Federated Over-Air PM.

Input: $\boldsymbol{Z}$ (data matrix), r (rank), L (# iterations), $\hat{\boldsymbol{U}}_0$ (optional initial subspace estimate)

1: K nodes, $\boldsymbol{Z}_k \in \mathbb{R}^{n\times d_k}$ local data at k-th node.
2: If no initial estimate provided, at central node, do $\tilde{\boldsymbol{U}}_0 \overset{i.i.d.}{\sim} \mathcal{N}(0, I)_{n\times r}$; $\hat{\boldsymbol{U}}_0 \leftarrow \tilde{\boldsymbol{U}}_0$, transmit to all K nodes.
3: **for** $l = 1, \ldots, L$ **do**
4: At k-th node, for all $k \in [K]$, compute $\tilde{\boldsymbol{U}}_{k,l} = \boldsymbol{Z}_k \boldsymbol{Z}_k^\top \hat{\boldsymbol{U}}_{l-1}$
5: All K nodes transmit $\tilde{\boldsymbol{U}}_{k,l}$ synchronously to central node.
6: Central node receives $\tilde{\boldsymbol{U}}_l := \sum_k \tilde{\boldsymbol{U}}_{k,l} + \boldsymbol{W}_l$.
7: Central node computes $\hat{\boldsymbol{U}}_l \boldsymbol{R}_l \overset{QR}{\leftarrow} \boldsymbol{U}_l$
8: Central node broadcasts $\hat{\boldsymbol{U}}_l$ to all nodes
9: **end for**
10: At k-th node, compute $\tilde{\boldsymbol{U}}_{k,L+1} = \boldsymbol{Z}_k \boldsymbol{Z}_k^\top \hat{\boldsymbol{U}}_L$
11: All K nodes transmit $\tilde{\boldsymbol{U}}_{k,L+1}$ synchronously to the central node.
12: Central node receives $\tilde{\boldsymbol{U}}_{L+1} := \sum_k \tilde{\boldsymbol{U}}_{k,L+1} + \boldsymbol{W}_{L+1}$
13: Central node computes $\hat{\lambda} = \hat{\boldsymbol{U}}_L^\top \tilde{\boldsymbol{U}}_{L+1}$ and its top eigenvalue, $\hat{\sigma}_1 = \lambda_{\max}(\hat{\lambda})$.

Output: $\hat{\boldsymbol{U}}_L$, $\hat{\sigma}_1$.

Lemma 4.2 (FedOA-PM: Maximum Eigenvalue): Let σ_i be the i-th largest eigenvalue of $\boldsymbol{Z}\boldsymbol{Z}^\top$. Under the assumptions of Lemma 4.1, $\hat{\sigma}_1$ computed in line 13 of Algorithm 2 satisfies

$$(1 - 4\epsilon^2)\sigma_1 - \epsilon^2\sigma_{r+1} - \epsilon\sigma_r \leq \hat{\sigma}_1 \leq (1+\epsilon)\sigma_1$$

To our best knowledge, the Lemma 4.2 has not been proved in earlier work. This result is useful because thresholding the top eigenvalue of an appropriately defined matrix is typically used for subspace change detection, see for example [23]. The proof of Lemma 4.2 given in Supplementary Material requires use of Weyl's inequality and the careful bounding of two error terms.

Note: The reason we obtain a constant probability 0.9 in the Lemma 4.1 is as follows: for any given r-dimensional subspace, $\boldsymbol{U}$ and a random Gaussian matrix $\hat{\boldsymbol{U}}$, the matrix $\hat{\boldsymbol{U}}^\top \boldsymbol{U}$ is an $r \times r$ random Gaussian matrix with independent entries. The singular values of $\hat{\boldsymbol{U}}^\top \boldsymbol{U}$ equal the cosine of the r principal angles between $\hat{\boldsymbol{U}}_0$ and $\boldsymbol{U}$. For successfully estimation (through *any* iterative method) it is necessary that none of the principal angles are $\pi/2$. To ensure this, we need to lower bound the smallest singular value of $\hat{\boldsymbol{U}}^\top \boldsymbol{U}$. This is difficult because the smallest singular value of square or "almost" square random matrices can be arbitrarily close to zero [40], [41]. The same issue is also seen in [26], [27].[4] In fact, this is an issue for any randomized algorithm for estimating only the top r singular vectors (without a full SVD), e.g., see [42]–[44].

[4]These papers also provide a more general result that allows one to compute an r'-dimensional subspace approximation for an $r' > r$. If r' is picked sufficiently large, e.g., if $r' = 2r$, then the guarantee holds with probability at least $1 - 0.1^r$.

We next define the federated over-air robust subspace tracking with missing entries (Fed-OA-RSTMiss) problem, and show how Algorithm 2 and Lemma 4.1 is used to solve Fed-OA-RSTMiss.

C. Fed-OA-RSTMiss: Problem Setting

In this section, we use α_k to denote the number of data points at node k at time t and $\alpha := \sum_k \alpha_k$ to denote the total number at time t. We do this to differentiate from d (in Section IV-B) which is used to indicate the total number of data vectors. Thus, at time t, $d = t\alpha$ and $d_k = t\alpha_k$. At time t and node k, we observe a possibly incomplete and noisy data matrix $\boldsymbol{Y}_{k,t}$ of dimension $n \times \alpha_k$ with the missing entries being replaced by a zero. This means the following: let $\tilde{\boldsymbol{L}}_{k,t}$ denote the unknown, complete, approximately low-rank matrix at node k at time t. Then

$$\boldsymbol{Y}_{k,t} = \mathcal{P}_{\Omega_{k,t}}(\tilde{\boldsymbol{L}}_{k,t} + \boldsymbol{G}_{k,t}) = \mathcal{P}_{\Omega_{k,t}}(\tilde{\boldsymbol{L}}_{k,t}) + \boldsymbol{S}_{k,t}$$

where $\boldsymbol{G}_{k,t}$'s are sparse outliers and $\boldsymbol{S}_{k,t} := \mathcal{P}_{\Omega_{k,t}}(\boldsymbol{G}_{k,t})$, and $\mathcal{P}_{\Omega_{k,t}}$ sets entries outside the set $\Omega_{k,t}$ to zero. The full matrix available from all nodes at time t is denoted $\boldsymbol{Y}_t := [\boldsymbol{Y}_{1,t}, \boldsymbol{Y}_{2,t}, \dots, \boldsymbol{Y}_{K,t}]$. This is of size $n \times \alpha$. The true (approximately) rank-r matrix $\tilde{\boldsymbol{L}}_t$ is similarly defined. Define the index sets $\mathcal{I}_{1,t} := [1, 2, \dots, \alpha_1]$, $\mathcal{I}_{2,t} := [\alpha_1 + 1, \alpha_1 + 2, \dots, \alpha_1 + \alpha_2]$ and so on. Denote the i-th column of $\boldsymbol{Y}_t$ by $\boldsymbol{y}_i$, $i = 1, 2, \dots, \alpha$. And with slight abuse of notation, we define (the matrix binary masks) $\Omega_{1,t} := [(\mathcal{M}_{1,t})^c, (\mathcal{M}_{2,t})^c, \dots, (\mathcal{M}_{\alpha_1,t})^c]$, $\Omega_{2,t} := [(\mathcal{M}_{\alpha_1+1,t})^c, (\mathcal{M}_{\alpha_1+2,t})^c, \dots, (\mathcal{M}_{\alpha_1+\alpha_2,t})^c]$ and so on where $\mathcal{M}_{i,t}$ is the set of missing entries in column i of the data matrix at time t, $(\mathcal{M}_{i,t})^c$ is its complement w.r.t $[n]$. Thus, the observations satisfy

$$\boldsymbol{y}_i = \mathcal{P}_{\mathcal{M}_{i,t}^c}(\tilde{\boldsymbol{\ell}}_i) + \boldsymbol{s}_i, \quad i \in \mathcal{I}_{k,t}, \quad k \in [K] \tag{10}$$

where $\boldsymbol{s}_i$ are sparse vectors with support $\mathcal{M}_{\text{sparse},i}$. Notice that it is impossible to recover $\boldsymbol{g}_i$ on the set $\mathcal{M}_{i,t}$ and so by definition, $\mathcal{M}_{\text{sparse},i}, \mathcal{M}_{i,t}$ are disjoint. Let $\boldsymbol{P}_t$ denote the ($n \times r$ dimensional) matrix of top r left singular vectors of $\tilde{\boldsymbol{L}}_t$. In general, our assumptions imply that $\tilde{\boldsymbol{L}}_t$ is only approximately rank r. As done in our result for ST-miss (in a centralized setting), we define the matrix of the principal subspace coefficients at time t as $\boldsymbol{A}_t := \boldsymbol{P}_t^\top \tilde{\boldsymbol{L}}_t$, the rank-$r$ approximation, $\boldsymbol{L}_t := \boldsymbol{P}_t \boldsymbol{P}_t^\top \tilde{\boldsymbol{L}}_t$ and the "noise" orthogonal to the span($\boldsymbol{P}_t$) as $\boldsymbol{V}_t := \tilde{\boldsymbol{L}}_t - \boldsymbol{L}_t$. With these definitions, for all $i \in \mathcal{I}_{k,t}$ and $k \in [K]$, we can equivalently express the measurements as follows

$$\begin{aligned}
\boldsymbol{y}_i &= \mathcal{P}_{\mathcal{M}_{i,t}^c}(\tilde{\boldsymbol{\ell}}_i) + \boldsymbol{s}_i \\
&= \tilde{\boldsymbol{\ell}}_i - \boldsymbol{I}_{\mathcal{M}_{i,t}} \boldsymbol{I}_{\mathcal{M}_{i,t}}^\top \tilde{\boldsymbol{\ell}}_i + \boldsymbol{s}_i \\
&:= \tilde{\boldsymbol{\ell}}_i + \boldsymbol{z}_i + \boldsymbol{s}_i \\
&= \boldsymbol{\ell}_i + \boldsymbol{z}_i + \boldsymbol{s}_i + \boldsymbol{v}_i
\end{aligned}$$

The goal is to track the subspaces $\boldsymbol{P}_t$ quickly and reliably, and hence also reliably estimate the columns of the rank r matrix $\boldsymbol{L}_t$, under the FedOA constraints given earlier. Our problem can also be understood as a dynamic (changing subspace) version of robust matrix completion [45].

D. Algorithm

The overall idea of the solution is similar to that for ST-miss. The algorithm still consists of two parts: (a) obtain *an* estimate of the columns $\tilde{\boldsymbol{L}}_t$ using the previous subspace estimate $\hat{\boldsymbol{P}}_{t-1}$; and (b) use this estimated matrix $\hat{\boldsymbol{L}}_t$ to update the subspace estimate, i.e., obtain $\hat{\boldsymbol{P}}_t$ by r-SVD. The algorithm can be initialized via r-SVD (as done in ST-miss) if we assume that $\boldsymbol{Y}_1$ (the set of data available at $t = 1$) contains no outliers and if not, one would need to use a batch RPCA approach such as AltProj [8] to obtain the initial subspace estimate $\hat{\boldsymbol{P}}_1$.

In the federated setting (a) is done locally at each node, while (b) requires a Fed-OA algorithm for SVD which is done using Algorithm 2. If one were to consider a federated but noise-free setting, there would be no need for new analysis (standard guarantees for PM would apply).

For step (a) (obtaining an estimate of $\tilde{\boldsymbol{L}}_t$ column-wise), we use the projected Compressive Sensing (CS) idea [5]. This relies on the slow-subspace change assumption. Let $\hat{\boldsymbol{P}}_{t-1}$ denote the subspace basis estimate from the previous time and let $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1} \hat{\boldsymbol{P}}_{t-1}^\top$. Projecting $\boldsymbol{y}_i$ orthogonal to $\hat{\boldsymbol{P}}_{t-1}$ helps mostly nullify $\boldsymbol{\ell}_i$ but gives projected measurements of the missing entries, $\boldsymbol{I}_{\mathcal{M}_i} \boldsymbol{I}_{\mathcal{M}_i}^\top \boldsymbol{\ell}_i$ and the sparse outliers, $\boldsymbol{s}_i$ as follows

$$\boldsymbol{\Psi} \boldsymbol{y}_i = \underbrace{\boldsymbol{\Psi}(\boldsymbol{s}_i - \boldsymbol{I}_{\mathcal{M}_i} \boldsymbol{I}_{\mathcal{M}_i}^\top \boldsymbol{\ell}_i)}_{\text{projected sparse vector}} + \underbrace{\boldsymbol{\Psi}(\boldsymbol{\ell}_i + \boldsymbol{v}_i)}_{\text{error}}$$

If the previous subspace estimate is good enough, and the noise is small, the error term above will be small. Now recovering the vector $\boldsymbol{s}_i - \boldsymbol{I}_{\mathcal{M}_i} \boldsymbol{I}_{\mathcal{M}_i}^\top \boldsymbol{\ell}_i$ is from $\boldsymbol{\Psi} \boldsymbol{y}_i$ is a problem of noisy compressive sensing with partial support knowledge (since we know $\mathcal{M}_i$). We first recover the support of $\boldsymbol{s}_i$ using the approach of [46], and then perform a least-squares based debiasing to estimate the magnitude of the entries. Following this, *an* estimate of the true data, $\hat{\boldsymbol{\ell}}_i$ is computed by subtraction from the observed data $\boldsymbol{y}_i$. We show in Lemma 4.6 that $\hat{\boldsymbol{\ell}}_i$ satisfies

$$\hat{\boldsymbol{\ell}}_i = \boldsymbol{\ell}_i - \boldsymbol{I}_{\hat{\mathcal{M}}_i} \left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top \boldsymbol{\Psi}_{\hat{\mathcal{M}}_i} \right)^{-1} \boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top \boldsymbol{\Psi}(\boldsymbol{\ell}_i + \boldsymbol{v}_i) + \boldsymbol{v}_i \tag{11}$$

Now we have $\hat{\boldsymbol{L}}_t := [\hat{\boldsymbol{L}}_{1,t}, \hat{\boldsymbol{L}}_{2,t}, \dots, \hat{\boldsymbol{L}}_{K,t}]$ with $\hat{\boldsymbol{L}}_{k,t}$ available only at node k. To goal is to compute an estimate ($\hat{\boldsymbol{P}}_t$) of its top r left singular vectors while obeying the federated data sharing constraints. We implement this through FedOA-PM (Algorithm 2) with $\boldsymbol{Z}_k \equiv \hat{\boldsymbol{L}}_{k,t}$ being the data matrix at node k. We invoke FedOA-PM with an initial estimate $\hat{\boldsymbol{P}}_{t-1}$. This simple change allows the probability of success of the overall algorithm to be close to 1 rather than 0.9 which is what the result of Lemma 4.1 predicts. This result is obtained by carefully combining the result for PCA-SDDN in a centralized setting (Lemma 3.10) and the result for FedOA-PM (Lemma 4.1). The result is summarized in Lemma 4.7. Applying these results in exactly the same manner as we did in Section III-E (with a few minor differences we point out in the next section), we get the main result.

E. Guarantee for Fed-OA RST-Miss

Before we state the main result, we need a few definitions.

Algorithm 3: Fed-OA-RSTMiss-NoDet.

Input: $\boldsymbol{Y}, \mathcal{M}$
Parameters: $L \leftarrow C\log(1/\text{no-lev}), \omega_{supp}, \xi, \alpha$
Init: $\tau \leftarrow 1, j \leftarrow 1, \hat{\boldsymbol{P}}_1$
for $t > 1$ **do**
  $\hat{\boldsymbol{L}}_t \leftarrow$ FED-MODCS($\boldsymbol{y}_i, \mathcal{I}_{k,t}, \mathcal{M}_i, \hat{\boldsymbol{P}}_{t-1}$)
  $\hat{\boldsymbol{P}}_t \leftarrow$ FEDOA-PM($\hat{\boldsymbol{L}}_t, r, L, \hat{\boldsymbol{P}}_{t-1}$)
  $\hat{\hat{\boldsymbol{L}}}_t \leftarrow$ FED-MODCS($\boldsymbol{y}_i, \mathcal{I}_{k,t}, \mathcal{M}_i, \hat{\boldsymbol{P}}_t$) ▷optional
end for
Output: $\hat{\boldsymbol{P}}$

Algorithm 4: Federated Modified Compressed Sensing.

procedure Fed-ModCS($\boldsymbol{y}_i, \mathcal{I}_{k,t}, \mathcal{M}_i, \hat{\boldsymbol{P}}_{t-1}$)
  for all node k, $i \in \mathcal{I}_{k,t}$ **do**
    $\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}^{\top}$
    $\tilde{\boldsymbol{y}}_i \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_i$
    $\hat{\boldsymbol{s}}_{i,cs} \leftarrow \arg\min_{\boldsymbol{s}} \|(\boldsymbol{s})_{(\mathcal{M}_i)^c}\|_1$ s.t. $\|\tilde{\boldsymbol{y}}_i - \boldsymbol{\Psi}\boldsymbol{s}\| \leq \xi$.
    $\hat{\mathcal{M}}_i \leftarrow \mathcal{M}_i \cup \{j : |(\hat{\boldsymbol{s}}_{i,cs})_j| > \omega_{supp}\}$
    $\hat{\boldsymbol{\ell}}_i \leftarrow \boldsymbol{y}_i - \boldsymbol{I}_{\hat{\mathcal{M}}_i}(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i})^{\dagger}\tilde{\boldsymbol{y}}_i$.
  end for
  Output: $\hat{\boldsymbol{L}}_t$
end procedure

Definition 4.3 (Sparse outlier fractions): Consider the $n \times \alpha$ sparse outlier matrix $\boldsymbol{S}_t := [\boldsymbol{S}_{1,t}, \dots, \boldsymbol{S}_{K,t}]$ at time t. We use max-out-frac-col (max-out-frac-row) to denote the maximum of the fraction of non-zero elements in any column (row) of this matrix. Also define $s_{\min} = \min_{i\in\mathcal{I}_{k,t}} \min_{j\in\mathcal{M}_{\text{sparse},i}} |(\boldsymbol{s}_i)_j|$.

Let $\lambda_v^+ := \max_{i\in\mathcal{I}_{k,t}} \|\mathbb{E}[\boldsymbol{v}_i\boldsymbol{v}_i^{\top}]\|$ and $\max_{i\in\mathcal{I}_{k,t}} \|\boldsymbol{v}_i\|^2 \leq Cr\lambda_v^+$ for all $k \in [K]$.

Theorem 4.4 (Federated Robust Subspace Tracking NoDet): Consider Algorithm 3. Assume that $\sqrt{\lambda_v^+/\lambda^-} :=$ no-lev ≤ 0.2. Set $L = C\log(1/\text{no-lev})$ and $\omega_{supp} = s_{\min}/2$, $\xi = s_{\min}/15$. Assume that the following hold:

1) At $t = 1$ we are given a $\hat{\boldsymbol{P}}_1$ s.t. $\text{dist}(\boldsymbol{P}_1, \hat{\boldsymbol{P}}_1) \leq \epsilon_{\text{init}}$.
2) **Incoherence:** $\boldsymbol{P}_t$'s satisfy μ-incoherence, and $\boldsymbol{a}_i$'s satisfy statistical right μ-incoherence;
3) **Missing Entries:** max-miss-frac-col $\in O(1/\mu r)$, max-miss-frac-row $\in O(1)$;
4) **Sparse Outliers:** max-out-frac-col $\in O(1/\mu r)$, max-out-frac-row $\in O(1)$;
5) **Channel Noise:** the channel noise seen by each FedOA-PM iteration is mutually independent at all times, isotropic, and zero mean Gaussian with standard deviation $\sigma_c \leq$ no-lev$\lambda^-/10\sqrt{n}$.
6) **Subspace Model:** The total data available at each time t, $\alpha \in \Omega(r\log n)$ and $\Delta_{tv} := \max_t \text{dist}(\boldsymbol{P}_{t-1}, \boldsymbol{P}_t)$ s.t.

$$0.3\epsilon_{\text{init}} + 0.5\Delta_{tv} \leq 0.28 \quad \text{and}$$

$$C\sqrt{r\lambda^+}(0.3^{t-1}\epsilon_{\text{init}} + 0.5\Delta_{tv}) + \sqrt{r_v\lambda_v^+} \leq s_{\min}$$

then, with probability at least $1 - 10dn^{-10}$, for $t > 1$, we have

$$\begin{aligned}&\text{dist}(\hat{\boldsymbol{P}}_t, \boldsymbol{P}_t)\\ &\leq \max(0.3^{t-1}\epsilon_{\text{init}} + \Delta_{tv}(0.3 + 0.3^2 \dots + 0.3^{t-1}), \text{no-lev})\\ &< \max(0.3^{t-1}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev})\end{aligned}$$

Also, at all times t, $\|\hat{\hat{\boldsymbol{\ell}}}_i - \boldsymbol{\ell}_i\| \leq 1.2 \cdot \text{dist}(\hat{\boldsymbol{P}}_t, \boldsymbol{P}_t)\|\boldsymbol{\ell}_i\| + \|\boldsymbol{v}_i\|$ for all $i \in \mathcal{I}_{k,t}, k \in [K]$.

1) Discussion: Items 2-4 of Theorem 4.4 are necessary to ensure that the RST-miss and robust matrix completion problems are well posed [23], [45]. The initialization assumption of Theorem 4.4 is different from the requirement of Theorem 3.6 due to the presence of outliers. Just performing a r-SVD on $\boldsymbol{Y}_1$ as done in Algorithm 1 does not work since even a few outliers can make the output arbitrarily far from the "true subspace". Additionally, without a "good initialization" Algorithm 3 cannot obtain good estimates of the sparse outliers since the noise in the sparse recovery step would be too large. One possibility to extend our result is to assume that there are no outliers at $t = 1$, i.e., $\boldsymbol{S}_1 = \boldsymbol{0}$ in which case, we use the initialization idea of Algorithm 1 (see Remark 4.5). Item 5 is standard in the federated learning/differential privacy literature [26], [27] as without bounds on iteration noise, it is not possible to obtain a final estimate that is close to the ground truth. Finally, consider item 6: the first part is required to ensure that the projection matrices, $\boldsymbol{\Psi}$'s satisfy the restricted isometry property [38], [46] which is necessary for provable sparse recovery (with partial support knowledge). This is a more stringent assumption than $\Delta_{tv} \leq 0.1$ assumed in Theorem 3.6 due to the presence of outliers. The second part of item 6 is an artifact of our analysis and arises due to the fact that it is hard to obtiain element-wise error bounds for Compressive Sensing.

In Theorem 4.4 we assumed that we are given a good enough initialization. If however, $\boldsymbol{S}_1$ were 0, we have the following result.

Remark 4.5: Under the conditions of Theorem 4.4, if $\boldsymbol{S}_1 = 0$, then all conclusions of Theorem 4.4 hold with the following changes

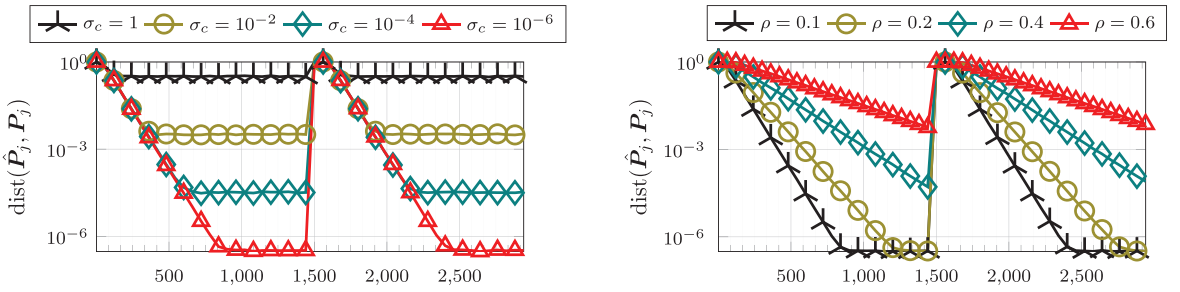
1) The number of iterations is set as $L = C\log(n/\text{no-lev})$
2) The subspace model (item 6 satisfies all conditions with ϵ_{init} replaced by $0.01 \cdot 0.3$
3) The probability of success is now $0.9 - 10dn^{-10}$.

F. Proof Outline

Here we prove our main result for robust ST-Miss under the federated data sharing constraints. The proof relies on two main results given below – (i) the result of (centralized) RST-Miss proved in the Supplementary Material (Appendix. C) and (ii) our result for federated over-air power method from Section IV-B.

Lemma 4.6 (Projected-CS with partial support knowledge): Consider Lines $5-7$ of Algorithm 4. Under the conditions of Theorem 4.4, we have for all t and all $i \in \mathcal{I}_{k,t}$, the error seen by the compressed sensing step satisfies

$$\|\boldsymbol{\Psi}(\boldsymbol{\ell}_i + \boldsymbol{v}_i)\| \leq (0.3^{t-1}\epsilon_{\text{init}} + 2.5\Delta_{tv})\sqrt{\mu r\lambda^+} + \sqrt{r_v\lambda_v^+}$$



(a) Comparison with respect to channel noise (b) Comparison with respect to fraction of missing entries

Fig. 2. Performance of Algorithm 3 under varying model parameters.

$\|\hat{\boldsymbol{s}}_{i,cs} - \boldsymbol{s}_i\| \leq 7x_{\min}/15 < x_{\min}/2$, $\hat{\mathcal{M}}_{\text{sparse},i} = \mathcal{M}_{\text{sparse},i}$, the error $\boldsymbol{e} := \hat{\boldsymbol{\ell}}_i - \boldsymbol{\ell}_i$ satisfies

$$\begin{aligned}\boldsymbol{e} &= -\boldsymbol{I}_{\hat{\mathcal{M}}_i}\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top \boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1} \boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top \boldsymbol{\Psi}(\boldsymbol{\ell}_i + \boldsymbol{v}_i) + \boldsymbol{v}_i, \\ &= (\boldsymbol{e}_i)_{\boldsymbol{\ell}} + (\boldsymbol{e}_i)_{\boldsymbol{v}} + \boldsymbol{v}_i \qquad (12)\end{aligned}$$

and $\|\boldsymbol{e}_i\| \leq 1.2(0.3^{t-1}\epsilon_{\text{init}} + 2.5\Delta_{tv})\sqrt{\mu r \lambda^+} + 2.2\sqrt{r_v \lambda_v^+}$. Here, $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}^\top$

Lemma 4.7 (FedOA PCA-SDDN (available init)): Consider the output $\hat{\boldsymbol{P}}$ of FedOA-PM (Algorithm 2) applied on data vectors $\boldsymbol{z}_i$ distributed across K nodes, when $\boldsymbol{z}_i = \boldsymbol{\ell}_i + \boldsymbol{e}_i + \boldsymbol{v}_i$, $i = 1, 2, \ldots, \alpha$ with $\boldsymbol{\ell}_i = \boldsymbol{P}\boldsymbol{a}_i$, $\boldsymbol{e}_i = \boldsymbol{I}_{\mathcal{M}_i}\boldsymbol{B}_i\boldsymbol{\ell}_i$ being sparse, data-dependent noise with support $\mathcal{M}_i$; the modeling error $\boldsymbol{v}_i$ is bounded with $\max_i \|\boldsymbol{v}_i\|^2 \leq C r_v \lambda_v^+$ where $\lambda_v^+ := \|\mathbb{E}[\boldsymbol{v}_i \boldsymbol{v}_i^\top]\|$. The matrix of top-$r$ left singular vectors, $\boldsymbol{P}$ satisfies μ-incoherence, and $\boldsymbol{a}_i$'s satisfy μ-statistical right-incoherence. The channel noise is zero mean i.i.d. Gaussian with standard deviation $\sigma_c \leq \epsilon_{PM}\lambda^-/10\sqrt{n}$ and is independent of the $\boldsymbol{\ell}_i$'s. Let $q := \max_i \|\boldsymbol{B}_i\boldsymbol{P}\|$ and let b denote the fraction of non-zeros in any row of the SDDN matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\alpha]$. Pick an $\epsilon_{PM} > 0$. If

$$7\sqrt{b}qf + \lambda_v^+/\lambda^- < 0.4\epsilon_{PM},$$

$\alpha \geq Cr\log n \max(\frac{q^2}{\epsilon_{PM}^2}f^2, \frac{\lambda_v^+}{\epsilon_{PM}^2\lambda^-}f)$, and if FedOA-PM is initialized with a matrix $\boldsymbol{P}_{\text{init}}$ such that $\mathrm{dist}(\boldsymbol{P}_{\text{init}}, \boldsymbol{P}) \leq \epsilon_{\text{init},PM}$, then after $L = C\log(1/(\epsilon_{PM}\sqrt{1-\epsilon_{\text{init},PM}^2}))$ iterations, with probability at least $1 - L\exp(-cr) - n^{-10}$, $\hat{\boldsymbol{P}}$ satisfies $\mathrm{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) \leq \epsilon_{PM}$.

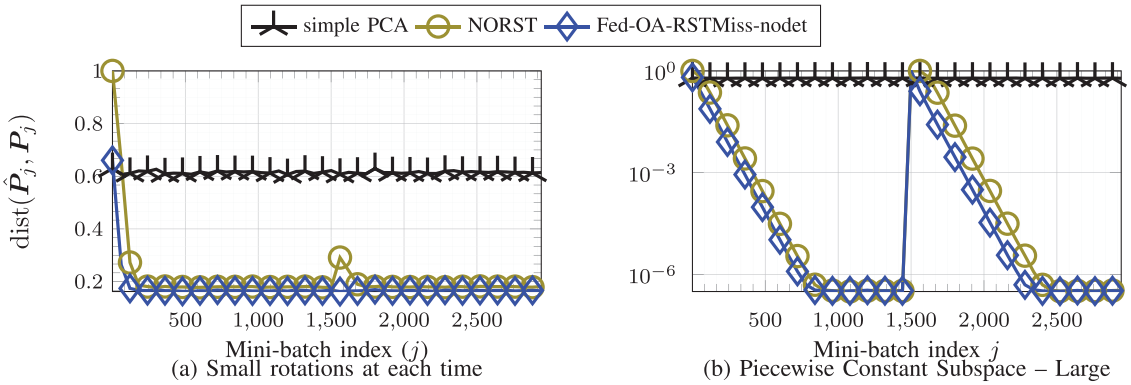
With these two Lemmas, the proof of Theorem 4.4 is similar to the proof of Theorem 3.6. Firstly, consider the projected CS with partial support knowledge step. Lemma 4.6 applied to each vector locally gives us $\hat{\boldsymbol{\ell}}_i = \boldsymbol{\ell}_i - \boldsymbol{e}_i$ with $\boldsymbol{e}_i$ satisfying (12). Next, at each time t, we update the subspace as the top r left singular vectors of $\hat{\boldsymbol{L}}_t$, where the k-th node only has access to the submatrix $\hat{\boldsymbol{L}}_{k,t}$. For a $t > 1$, we assume that the previous subspace estimate, $\hat{\boldsymbol{P}}_{t-1}$ satisfies $\mathrm{dist}(\hat{\boldsymbol{P}}_{t-1}, \boldsymbol{P}_{t-1}) \leq \max(0.3^{t-2}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev})$. We invoke Lemma 4.7 with $\hat{\boldsymbol{P}}_{init} \equiv \hat{\boldsymbol{P}}_{t-1}$ and thus, $\epsilon_{\text{init},PM} \equiv \max(0.3^{t-2}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev})$; $\boldsymbol{z}_i \equiv \hat{\boldsymbol{\ell}}_i, i \in \mathcal{I}_{k,t}$; $\boldsymbol{P} \equiv \boldsymbol{P}_t, \boldsymbol{e}_i \equiv (\boldsymbol{e}_i)_{\boldsymbol{\ell}}, \boldsymbol{v}_i \equiv (\boldsymbol{e}_i)_{\boldsymbol{v}} + \boldsymbol{v}_i$; and $\epsilon_{PM} \equiv \max(0.3^{t-2}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev})$. Under the conditions of theorem 4.4, we conclude that w.h.p., $\mathrm{dist}(\hat{\boldsymbol{P}}_t, \boldsymbol{P}_t) \leq \max(0.3^{t-1}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev})$. Thus, applying this argument inductively proves the result. For the second optional FedOA-PM step, the same ideas from the proof of Theorem 3.6 apply.

V. Numerical Experiments

The codes are available at https://github.com/praneethmurthy/distributed-pca.

A. Centralized STMiss

1) Small Rotations at Each Time: We first consider the centralized setting for Subspace Tracking with missing data (Section III). We demonstrate results under two sets of subspace change models. First we consider the "rotation model" that has been commonly used in the literature [3], [4]. At each time t, we generate a $n \times r$ dimensional subspace $\boldsymbol{P}_{(t)} = e^{-\delta_t \boldsymbol{B}_t}\boldsymbol{P}_{(t-1)}$ with $\boldsymbol{P}_{(0)}$ generated by orthonormalizing the columns of a i.i.d. standard Gaussian matrix and $\boldsymbol{B}_t$ is some skew symmetric matrix to simulate rotations and δ_t controls the amount of rotation (for this experiment we set $\delta_t = 10^{-4}$ which ensures that $\Delta_{tv} \approx 10^{-2}$). We generate matrix $\tilde{\boldsymbol{A}}$ as a i.i.d. uniform random matrix of size $r \times d$ and set the t-th column of the true data matrix $\tilde{\boldsymbol{\ell}}_t = \boldsymbol{P}_{(t)}\tilde{\boldsymbol{a}}_t$. Thus, in the notation of our result, $\boldsymbol{P}_j$ is the matrix of the top r left singular vectors of $\tilde{\boldsymbol{L}}_j = [\tilde{\boldsymbol{\ell}}_{(j-1)\alpha+1}, \cdots \tilde{\boldsymbol{\ell}}_{j\alpha}]$ and $\boldsymbol{A}_j = \boldsymbol{P}_j^\top \tilde{\boldsymbol{L}}_j$. In all experiments, we choose $n = 1000$ and $d = 3000$. We simulate the set of observed entries using a Bernoulli model where each element of the matrix is observed with probability 0.9. For all experiments, we set $r = 30$ and the fraction of missing entries to be 0.1. We implement STMiss-nodet (Algorithm 1) and set $r = 30$. We compare with NORST [23] (the state-of-the-art theoretically), GROUSE [4], and PETRELS [3] (the state-of-the-art experimentally). For all algorithms, we used default parameters mentioned in the codes. We also implement the simple PCA method wherein we estimate $\hat{\boldsymbol{P}}_j$ as the top-r left singular vectors of $\boldsymbol{Y}_j$ for each mini-batch. For all algorithms, the mini-batch size was chosen as $\alpha = 60$. The results are shown in Fig. 1(a). We see that as specified by Theorem 3.7, the simple PCA algorithm does not improve the recovery errors since it is not exploiting slow subspace change. However, all other algorithms exploit slow-subspace change and thus are able to provide better estimates with time. We also notice that PETRELS is the fastest to converge, followed by NORST



(a) Small rotations at each time (b) Piecewise Constant Subspace – Large

Fig. 3. Corroborating the claims of Theorem 4.4.

and STMiss-nodet, and finally GROUSE. This is consistent with the previous set of results in [23].

2) Piecewise Constant: Next, we consider a piecewise constant subspace change model that has been considered in the provable subspace tracking literature [23]. In this, we simulate a large subspace change at $t_1 = 1500$. The subspace is fixed until then, i.e., $\boldsymbol{P}_j = \boldsymbol{P}_1$ for all $j \in [1, \lceil t_1/\alpha \rceil)$ and $\boldsymbol{P}_j = \boldsymbol{P}_2$ for all $j \in [\lceil t_1/\alpha \rceil, \lceil d/\alpha \rceil]$. The results are shown in Fig. 1(b). Notice that NORST and STMiss-nodet significantly outperform simple PCA as both exploit slow subspace change. Additionally, even though the change is large (in the notation of Definition B.1 given in the supplementary material, $\Delta_{\text{large}} \approx 1$ and $\Delta_{tv} = 0$), STMiss-nodet is also able to adapt without requiring a detection step. Finally, since the updates are always improving, after a certain time, NORST stops improving the subspace estimates, but STMiss-nodet improves it and gets a better result.

B. Federated ST-Miss

We also implement Algorithm 3 to corroborate our theoretical claims. We use the exact data generation parameters as we did in the centralized setting. To simulate over-air communication, we replace the inbuilt `SVD` routine of MATLAB by a power method code snippet, and by adding iteration noise.[5] In each iteration, we add i.i.d. Gaussian noise with variance 10^{-6}. The results are presented in Fig. 3. Notice that in both cases, Algorithm 3 works as well as NORST even though NORST cannot deal with iteration noise. Additionally, as opposed to the centralized setting (Fig. 1(b)), the error of Fed-OA-RSTMiss-nodet in Fig. 3 does not improve beyond the iteration noise level of 10^{-6}.

We next validate the performance of Algorithm 3 with respect to different values of the channel noise. We generate the data as done in the previous experiment, but vary the iteration noise level. In particular, we choose $\sigma_c = \{1, 10^{-2}, 10^{-4}, 10^{-6}\}$ and provide the results in Fig. 2(a). Notice that in all the cases, the subspace error saturates at roughly σ_c as predicted.

Finally we analyze the performance of Algorithm 3 with respect to different values of missing entries. The data is generated as in the previous experiment with $\sigma_c = 10^{-6}$ and we vary the fraction of missing entries, $\rho = \{0.1, 0.2, 0.4, 0.6\}$. The results are given in Fig. 2(b) and we notice that in all cases, the algorithm works, but as the fraction of missing entries increases, more samples are required for convergence.

[5]Note that in this approach, it is not required to explicitly set a value of K

VI. Conclusions and Future Work

In this paper we studied the problem of Subspace Tracking from missing data and outliers. In particular, we consider a generalized problem formulation that does not make the *piecewise constant* subspace change assumption that is common in the provable subspace tracking literature. We proposed a simple algorithm to solve this problem provably and efficiently. We also developed a an algorithm to solve (robust) subspace tracking with missing entries when the data is federated, and over-air data communication modality is used. As part of future work, there are several open questions such as (i) is it possible to modify the proposed analysis to provide a guarantee for differentially private subspace tracking? (ii) is it possible to consider row-wise federated data model with appropriate analytical modifications?

Appendix A
Proof of Key Lemmas for Theorem 4.4

Proof of Lemma 4.6: Recall from Algorithm 4 that we need solve

$$\hat{\boldsymbol{s}}_{i,cs} = \arg\min_{\boldsymbol{s}} \|(\boldsymbol{s})_{\mathcal{M}_i^c}\|_1 \text{ s.t.} \|\boldsymbol{\Psi}\boldsymbol{y}_t - \boldsymbol{\Psi}\boldsymbol{s}\| \le \xi$$

This is a problem of sparse recovery from partial subspace knowledge. To prove the correctness of the result, we first need to bound the s-level RIC of $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}^\top$ where $s := (2\text{max-out-frac-col} + \text{max-miss-frac-col}) \cdot n$. Under the assumptions of Theorem 4.4 (we only assumed that max-out-frac-col $\in P(1/\mu r)$ and max-miss-frac-col $\in O(1/\mu r)$ but the actual requirement is $(2\text{max-miss-frac-col} + \text{max-out-frac-col}) \cdot n \le 0.01/\mu r$), and Fact 3.11, we have that

$$\begin{aligned}\delta_s(\boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}^\top) &= \max_{|\mathcal{M}|\le s} \|\boldsymbol{I}_{\mathcal{M}}^\top \hat{\boldsymbol{P}}_{t-1}\|^2 \\ &\le \max_{|\mathcal{M}|\le s} (\text{dist}(\hat{\boldsymbol{P}}_{t-1}, \boldsymbol{P}_{t-1}) + \|\boldsymbol{I}_{\mathcal{M}}^\top \boldsymbol{P}_{t-1}\|)^2\end{aligned}$$

Recall that for $t > 1$, $\text{dist}(\hat{\boldsymbol{P}}_j, \boldsymbol{P}_j) \le \max(0.1 \cdot 0.3^{j-1}\epsilon_{\text{init}} + 0.5\Delta_{tv}, \text{no-lev}) \le 0.2$ and from the incoherence assumption on $\boldsymbol{P}_t$'s, the second term above is upper bounded by 0.01. Thus, $\delta_s(\boldsymbol{\Psi}) \le 0.3^2 < 0.15$. Next, consider the error seen by

the modified-CS step,

$$\begin{aligned}\|\boldsymbol{b}_i\| &= \|\boldsymbol{\Psi}(\boldsymbol{\ell}_i+\boldsymbol{v}_i)\| \le \left\|(\boldsymbol{I}-\hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1})\boldsymbol{P}_t\boldsymbol{a}_i\right\| + \|\boldsymbol{v}_i\| \\ &\le \operatorname{dist}(\hat{\boldsymbol{P}}_{t-1},\boldsymbol{P}_t)\,\|\boldsymbol{a}_i\| + \|\boldsymbol{v}_i\| \\ &\le (\operatorname{dist}(\hat{\boldsymbol{P}}_{t-1},\boldsymbol{P}_{t-1}) + \operatorname{dist}(\boldsymbol{P}_{t-1},\boldsymbol{P}_t)) \\ &\quad \times \sqrt{\mu r\lambda^+} + C\sqrt{r\lambda_v^+} \\ &\le (0.3^{t-1}\epsilon_{\text{init}} + 1.5\Delta_{tv})\sqrt{\mu r\lambda^+} + C\sqrt{r_v\lambda_v^+}\end{aligned}$$

under the assumptions of Theorem 4.4, the RHS of the above is bounded by $s_{\min}/15$. This is why we have set $\xi = s_{\min}/15$ in Algorithm 3. Using these facts, and $\delta_s(\boldsymbol{\Psi}) < 0.15$, we have that

$$\|\hat{\boldsymbol{s}}_{i,cs} - \boldsymbol{s}_i\| \le 7\xi = 7s_{\min}/15 < s_{\min}/2$$

Consider support recovery. From above,

$$|(\hat{\boldsymbol{s}}_{i,cs} - \boldsymbol{s}_i)_m| \le \|\hat{\boldsymbol{s}}_{i,cs} - \boldsymbol{s}_i\| \le 7s_{\min}/15 < s_{\min}/2$$

The Algorithm sets $\omega_{supp} = s_{\min}/2$. Consider an index $m \in \mathcal{M}_{\text{sparse},i}$. Since $|(\boldsymbol{s}_i)_m| \ge s_{\min}$,

$$\begin{aligned}s_{\min} - |(\hat{\boldsymbol{s}}_{i,cs})_m| &\le |(\boldsymbol{s}_i)_m| - |(\hat{\boldsymbol{s}}_{i,cs})_m| \\ &\le |(\boldsymbol{s}_i - \hat{\boldsymbol{s}}_{i,cs})_m| < \frac{s_{\min}}{2}\end{aligned}$$

Thus, $|(\hat{\boldsymbol{s}}_{i,cs})_m| > \frac{s_{\min}}{2} = \omega_{supp}$ which means $m \in \hat{\mathcal{M}}_{\text{sparse},i}$. Hence $\mathcal{M}_{\text{sparse},i} \subseteq \hat{\mathcal{M}}_{\text{sparse},i}$. Next, consider any $m \notin \mathcal{M}_{\text{sparse},i}$. Then, $(\boldsymbol{s}_i)_m = 0$ and so

$$\begin{aligned}|(\hat{\boldsymbol{s}}_{i,cs})_m| &= |(\hat{\boldsymbol{s}}_{i,cs})_m)| - |(\boldsymbol{s}_i)_m| \\ &\le |(\hat{\boldsymbol{s}}_{i,cs})_m - (\boldsymbol{s}_i)_m| < \frac{s_{\min}}{2}\end{aligned}$$

which implies $m \notin \hat{\mathcal{M}}_{\text{sparse},i}$ and $\hat{\mathcal{M}}_{\text{sparse},i} \subseteq \mathcal{M}_{\text{sparse},i}$ implying that $\hat{\mathcal{M}}_{\text{sparse},i} = \mathcal{M}_{\text{sparse},i}$ and consequently that $\hat{\mathcal{M}}_i := \mathcal{M}_i \cup \hat{\mathcal{M}}_{\text{sparse},i} = \mathcal{M}_i \cup \mathcal{M}_{\text{sparse},i}$.

With $\hat{\mathcal{M}}_{\text{sparse},i} = \mathcal{M}_{\text{sparse},i}$ and since $\mathcal{M}_{\text{sparse},i}$ is the support of $\boldsymbol{s}_i$, $\boldsymbol{s}_i = \boldsymbol{I}_{\mathcal{M}_{\text{sparse},i}}\boldsymbol{I}_{\mathcal{M}_{\text{sparse},i}}^\top \boldsymbol{s}_i$, and so

$$\begin{aligned}\hat{\boldsymbol{s}}_i &= \boldsymbol{I}_{\hat{\mathcal{M}}_i}\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1}\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top(\boldsymbol{\Psi}\boldsymbol{\ell}_i + \boldsymbol{\Psi}\boldsymbol{z}_i + \boldsymbol{\Psi}\boldsymbol{s}_i + \boldsymbol{\Psi}\boldsymbol{v}_i) \\ &= \boldsymbol{I}_{\hat{\mathcal{M}}_i}\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1}\boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}(\boldsymbol{\ell}_i+\boldsymbol{v}_i) + \boldsymbol{s}_i + \boldsymbol{z}_i\end{aligned}$$

Thus, the estimate of the true-data $\hat{\boldsymbol{\ell}}_i = \boldsymbol{y}_i - \hat{\boldsymbol{s}}_i$ satisfies

$$\hat{\boldsymbol{\ell}}_i = \boldsymbol{\ell}_i + \boldsymbol{v}_i - \boldsymbol{I}_{\hat{\mathcal{M}}_i}\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1}\boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}(\boldsymbol{\ell}_i+\boldsymbol{v}_i)$$

and thus $\boldsymbol{e}_i = \hat{\boldsymbol{\ell}}_i - \boldsymbol{\ell}_i$ satisfies

$$\begin{aligned}\boldsymbol{e}_i &= -\boldsymbol{I}_{\hat{\mathcal{M}}_i}\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1}\boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}(\boldsymbol{\ell}_i+\boldsymbol{v}_i) + \boldsymbol{v}_i \\ \|\boldsymbol{e}_i\| &\le \left\|\left(\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}_{\hat{\mathcal{M}}_i}\right)^{-1}\right\|\|\boldsymbol{I}_{\hat{\mathcal{M}}_i}^\top\boldsymbol{\Psi}(\boldsymbol{\ell}_i+\boldsymbol{v}_i)\| + \|\boldsymbol{v}_i\| \\ &\le 1.2\|\boldsymbol{b}_i\| + \|\boldsymbol{v}_i\|\end{aligned}$$

■

We next prove Lemma 4.7. But before we prove this, under the conditions of Lemma 3.10, the result from [37] also shows the following:

$$\begin{aligned}\|\text{perturb}\| &:= \left\|\frac{1}{\alpha}\sum_i(\boldsymbol{z}_i\boldsymbol{z}_i^\top - \boldsymbol{\ell}_i\boldsymbol{\ell}_i^\top)\right\| \\ &\le \left\|\frac{1}{\alpha}\sum_i \boldsymbol{e}_i\boldsymbol{e}_i^\top\right\| + 2\left\|\frac{1}{\alpha}\sum_i\boldsymbol{\ell}_i\boldsymbol{e}_i^\top\right\| + 2\left\|\frac{1}{\alpha}\sum_i\boldsymbol{\ell}_i\boldsymbol{v}_i^\top\right\| \\ &\quad + 2\left\|\frac{1}{\alpha}\sum_i\boldsymbol{v}_i\boldsymbol{e}_i^\top\right\| + \left\|\frac{1}{\alpha}\sum_i\boldsymbol{v}_i\boldsymbol{v}_i^\top\right\|, \\ &\le \left(6.6\sqrt{b}qf + 4.4\frac{\lambda_v^+}{\lambda^-}\right)\lambda^- \qquad (13)\end{aligned}$$

and

$$\lambda_r\left(\frac{1}{\alpha}\sum_i\boldsymbol{\ell}_i\boldsymbol{\ell}_i^\top\right) \ge 0.99\lambda^-.$$

Proof of Lemma 4.7: Before we prove There are the following two parts in the proof:

1) First, we show that $\hat{\boldsymbol{P}}$ is *close* to $\tilde{\boldsymbol{P}}$ where $\tilde{\boldsymbol{P}}$ is the top r left singular vectors of $\boldsymbol{Z}$. In particular, we show that $\operatorname{dist}(\hat{\boldsymbol{P}},\tilde{\boldsymbol{P}}) \le \epsilon_{PM}/2$. This relies on application of Lemma 4.1 to the matrix $\boldsymbol{Z}\boldsymbol{Z}^\top/\alpha$ with the appropriate parameters.
2) Next, we use centralized Principal Components Analysis in Sparse, Data-Dependent Noise (PCA SDDN) with $\boldsymbol{z}_i \equiv \boldsymbol{y}_i$ to show that the $\tilde{\boldsymbol{P}}$ is *close* to the true subspace, $\boldsymbol{P}$. Here too we show that $\operatorname{dist}(\tilde{\boldsymbol{P}},\boldsymbol{P}) \le \epsilon_{PM}/2$. Combining the above two results, and the triangle inequality gives $\operatorname{dist}(\hat{\boldsymbol{P}},\boldsymbol{P}) \le \operatorname{dist}(\hat{\boldsymbol{P}},\tilde{\boldsymbol{P}}) + \operatorname{dist}(\tilde{\boldsymbol{P}},\boldsymbol{P}) \le \epsilon_{PM}$.

Notice from (13), with high probability, the matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ has a good eigen-gap, i.e.,

$$\begin{aligned}\lambda_r(\boldsymbol{Z}\boldsymbol{Z}^\top) &= \lambda_r(\boldsymbol{L}\boldsymbol{L}^\top + \text{perturb}) \ge \lambda_r(\boldsymbol{L}\boldsymbol{L}^\top) - \|\text{perturb}\| \\ &\ge 0.99\lambda^- - \left(7.7\sqrt{b}qf + 4.4\frac{\lambda_v^+}{\lambda^-}\right)\lambda^- \\ \lambda_{r+1}(\boldsymbol{Z}\boldsymbol{Z}^\top) &\le \lambda_{r+1}(\boldsymbol{L}\boldsymbol{L}^\top) + \|\text{pertub}\| \\ &\le \left(7.7\sqrt{b}qf + 4.4\frac{\lambda_v^+}{\lambda^-}\right)\lambda^-\end{aligned}$$

Under the assumptions of Lemma 4.7, $7.7\sqrt{b}qf + 4.4\lambda_v^+/\lambda^- \le 2.5\epsilon_{SE}$. Thus, for this matrix, $R < 0.99$ with high probability. The standard deviation of the channel noise in each iteration satisfies, $\sigma_c \le \epsilon_{PM}\lambda^-/10\sqrt{n}$. Furthermore, since we initialize Fed-PM with $\boldsymbol{P}_{\text{init}}$ that satisfies $\operatorname{dist}(\boldsymbol{P}_{\text{init}},\boldsymbol{P}) \le \epsilon_{\text{init},PM}$ it follows from second part of Lemma 4.1 that after $L = C\log(1/(\epsilon_{PM}\sqrt{1-\epsilon_{\text{init},PM}^2}))$ iterations, with probability at least $1 - L\exp(-cr)$, the output $\hat{\boldsymbol{P}}$ satisfies $\operatorname{dist}(\hat{\boldsymbol{P}},\tilde{\boldsymbol{P}}) \le \epsilon_{PM}/2$.

Next, observe that the conditions required to apply Lemma 3.10 is satisfied under the assumptions of Lemma 4.7. Thus, we apply Lemma 3.10 with $\epsilon_{\text{SE}} \equiv \epsilon_{PM}/2$. This ensures that with probability at least $1 - 10n^{-10}$, the eigenvectors of the empirical covariance are close to that of the the population covariance, i.e., $\operatorname{dist}(\tilde{\boldsymbol{P}},\boldsymbol{P}) \le \epsilon_{PM}/2$.

Combining the above two results we have with probability at least $1 - L\exp(-cr) - 10n^{-10}$, $\text{dist}(\hat{\boldsymbol{P}}, \boldsymbol{P}) \leq \text{dist}(\hat{\boldsymbol{P}}, \tilde{\boldsymbol{P}}) + \text{dist}(\tilde{\boldsymbol{P}}, \boldsymbol{P}) \leq \epsilon_{PM}$. ■

The proof of the subspace detection step (Lemma B.4) is similar to that of [37] applied with Lemma 4.2.

Proof of Lemma 4.1: The proof of Lemma 4.1 is a special case of Lemma D.1 that is proved in the Supplementary Material. The proof of Lemma 4.2 is also provided in the Supplementary Material. ■

REFERENCES

[1] P. Narayanamurthy, N. Vaswani, and A. Ramamoorthy, "Federated over-air robust subspace tracking from missing data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5598–5602.

[2] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[3] 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.

[4] D. Zhang and L. Balzano, "Global convergence of a grassmannian gradient descent algorithm for subspace estimation," in *Proc. Artif. Intell. Statist.*, 2016, pp. 1460–1468.

[5] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007–5039, Aug. 2014.

[6] C. Wang, Y. C. Eldar, and Y. M. Lu, "Subspace estimation from incomplete observations: A high-dimensional analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1240–1252, Dec. 2018.

[7] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[8] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust PCA," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 1107–1115.

[9] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 1432–1436.

[10] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, pp. 717–772, 2008.

[11] A. Zare, A. Ozdemir, M. A. Iwen, and S. Aviyente, "Extension of PCA to higher order data structures: An introduction to tensors, tensor decompositions, and tensor PCA," *Proc. IEEE*, vol. 106, no. 8, pp. 1341–1358, Aug. 2018.

[12] S. X. Wu, H-T Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," *Proc. IEEE*, vol. 106, no. 8, pp. 1321–1340, Aug. 2018.

[13] P. Kairouz et al., "Advances and open problems in federated learning," *Foundations Trends Mach. Learn.*, vol. 14, no. 1/2, pp. 1–210, 2021.

[14] J. Konecny, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[15] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[16] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[17] K. Bonawitz et al., "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[18] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[19] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[20] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[21] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[22] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, Mar. 2019.

[23] P. Narayanamurthy, V. Daneshpajooh, and N. Vaswani, "Provable subspace tracking from missing data and matrix completion," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4245–4260, Aug. 2019.

[24] A. Gonen, D. Rosenbaum, Y. C. Eldar, and S. Shalev-Shwartz, "Subspace learning with partial information," *J. Mach. Learn. Res.*, vol. 17, no. 52, pp. 1–21, 2016.

[25] 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.

[26] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2861–2869.

[27] M-F. Balcan, S. S. Du, Y. Wang, and A. W. Yu, "An improved gap-dependency analysis of the noisy power method," in *Proc. Conf. Learn. Theory*, 2016, pp. 284–309.

[28] Y. Kopsinis, S. Chouvardas, and S. Theodoridis, "Distributed robust subspace tracking," in *Proc. Eur. Signal Process. Conf.*, 2015, pp. 2531–2535.

[29] Y. Liang, M-F Balcan, V. Kanchanapally, and D. Woodruff, "Improved distributed principal component analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3113–3121.

[30] L. Mackey, A. Talwalkar, and M. I. Jordan, "Distributed matrix completion and robust factorization," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 913–960, 2015.

[31] C. Teflioudi, F. Makari, and R. Gemulla, "Distributed matrix completion," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 655–664.

[32] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4613–4623.

[33] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 261–270.

[34] D. Gunduz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184–2199, Oct. 2019.

[35] J. Konecny, H. N. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[36] A. Grammenos, R. Mendoza-Smith, J. Crowcroft, and C. Mascolo, "Federated principal component analysis," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6453–6464, 2020.

[37] 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, Nov. 2020.

[38] E. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9/10, pp. 589–592, 2008.

[39] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1989.

[40] M. Rudelson and R. Vershynin, "The Littlewood–Offord problem and invertibility of random matrices," *Adv. Math.*, vol. 218, no. 2, pp. 600–633, 2008.

[41] M. Rudelson and R. Vershynin, "Smallest singular value of a random rectangular matrix," *Commun. Pure Appl. Math.*, vol. 62, no. 12, pp. 1707–1739, 2009.

[42] C. Musco and C. Musco, "Randomized block Krylov methods for stronger and faster approximate singular value decomposition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1396–1404.

[43] I. Mitliagkas, C. Caramanis, and P. Jain, "Memory limited, streaming PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2886–2894.

[44] P. Jain, C. Jin, S. M. Kakade, P. Netrapalli, and A. Sidford, "Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm," in *Proc. Conf. Learn. Theory*, 2016, pp. 1147–1164.

[45] Y. Cherapanamjeri, K. Gupta, and P. Jain, "Nearly-optimal robust matrix completion," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 797–805.

[46] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, Sep. 2010.