Dynamic Subspace Estimation from Undersampled Data using Grassmannian Geodesics

Anonymous authors
Paper under double-blind review

Abstract

This work considers recovering a sequence of low-rank matrices from undersampled measurements, where the underlying subspace varies across samples over time. Existing works involve concatenating all of the samples from each time point to recover the underlying matrix under the assumption that the data are well-approximated by a single, static subspace. However, this assumption is inappropriate for applications where the best low-rank approximations vary over time. To address this issue, we propose a Riemannian block majorize-minimization algorithm that constrains the time-varying subspaces as a geodesic along the Grassmann manifold. Our proposed method can faithfully estimate the best-fit subspaces at each time point, even when there are fewer samples at each time point than the subspace dimension. Theoretically, we show that our algorithm enjoys a monotonically non-increasing objective function while converging to an ϵ-stationary point within $\widetilde{\mathcal{O}}(\epsilon^{-2})$ iterations. We demonstrate the effectiveness of our algorithm on synthetic, dynamic fMRI, and video data, where the samples at each time are either compressed or partially missing.

1 Introduction

Subspace estimation and low-rank matrix approximation are fundamental problems that arise in many statistical signal processing and machine learning applications. By modeling the data using a linear subspace, practitioners can efficiently leverage the intrinsic, low-dimensional structure of the data for tasks such as classification (Basri et al., 2007; Hubert & Engelen, 2004), anomaly detection (Huang et al., 2006), and denoising (Zhang et al., 2010), among others. One of the most widely studied and applied methods for subspace estimation is principal component analysis (PCA) (Abdi & Williams, 2010).

In the literature, many existing algorithms, such as PCA, primarily focus on a setting where we wish to fit data with a single linear subspace (Cai et al., 2021; Balzano et al., 2010; 2018b; Mansour & Jiang, 2015; Jain et al., 2013; Chi et al., 2012; Tong et al., 2021; Zhang et al., 2021). In applications where data are collected over time, one may then concatenate data and identify a single static subspace or estimate different low-dimensional approximate subspaces at different time points. However, in applications such as array signal processing (Vaccaro, 2019; Lake & Keenan, 1998a), video processing (Vaswani et al., 2018), graph connectivity analysis Hume & Balzano (2024), and dynamic MRI (Otazo et al., 2015; Babu et al., 2023), the data generation process is dynamic in nature, so assuming the best-approximating subspace is static may be sub-optimal. Despite the abundance of such applications, the literature on dynamic subspace estimation often relies on assumptions such as slow changes or otherwise a static subspace (Narayanamurthy & Vaswani, 2018a; 2019). Furthermore, many works on dynamic subspace estimation assume that the data are fully observed (Saad-Falcon et al., 2024b), limiting applicability for data with missing entries.

In this work, we address these issues by proposing a Riemannian block majorize–minimize (RBMM) algorithm that constrains the time-varying subspaces to evolve along a geodesic on the Grassmann manifold, as illustrated in Figure 1. Instead of assuming a static subspace for low-rank approximation, this geodesic formulation estimates a sequence of subspaces that vary smoothly over time, enforcing smooth transitions between consecutive subspaces. By employing such a model, our algorithm can recover undersampled (missing or compressed) data more accurately under measurement noise compared to methods that assume a

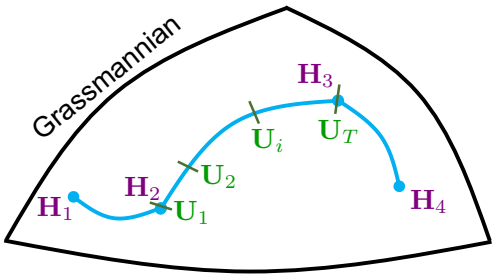

Figure 1: Cartoon illustration of the geodesic model. $\mathbf{H}_1, \ldots, \mathbf{H}_4$ are orthonormal bases for a point on the Grassmannian and $\mathbf{U}_i$ are connecting points along a geodesic between two points.

static subspace. We empirically demonstrate the effectiveness of our algorithm across a wide range of real video and dynamic fMRI reconstruction tasks. Theoretically, we show that our algorithm converges to an ϵ-stationary point in approximately $\widetilde{\mathcal{O}}(\epsilon^{-2})$ iterations, ignoring logarithmic factors, and that the objective function is monotonically non-increasing.

Notation. We denote scalars with unbolded letters (e.g., t, T), vectors with bold lower-case letters (e.g., $\mathbf{x}$) and matrices with bold upper-case letters (e.g., $\mathbf{X}$). For a matrix $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \cdots & \mathbf{x}_d\end{bmatrix}$ we define $\text{range}(\mathbf{X}) := \text{span}(\{\mathbf{x}_1, \ldots, \mathbf{x}_d\})$, which is the set of all possible linear combinations of the columns of $\mathbf{X}$. We use $\mathbf{I}_r$ to denote an identity matrix of size $r \in \mathbb{N}$ and $\mathcal{V}^{n\times r} = \{\mathbf{U}^{n\times r} \,|\, \mathbf{U}^\top\mathbf{U} = \mathbf{I}_r\}$ to denote the set of all $n \times r$ matrices with orthonormal columns. We use $\{\mathbf{U}_i\}$ as shorthand for the set ranging over the index i (i.e., $\{\mathbf{U}_i\}_{i=1}^T$), and similarly for sets over multiple indices (e.g., $\{\mathbf{A}_{i,j}\}$). We use $[L]$ to denote the set $\{1, 2, \ldots, L\}$. Lastly, $\widetilde{\mathcal{O}}(\cdot)$ denotes "big-O" notation that ignores logarithmic factors.

2 Problem Formulation

2.1 Data Model

Consider a setting where one observes measurements of a sequence of data matrices $\mathbf{X}_i \in \mathbb{R}^{n\times\ell}$ for times $i = 1, \ldots, T$. We may observe this sequence directly, or we may observe it indirectly as

$$\mathbf{Y}_i = [\mathbf{y}_{i,1} \ \ldots \ \mathbf{y}_{i,\ell}], \quad \text{where } \mathbf{y}_{i,j} = \mathbf{A}_{i,j}\mathbf{x}_{i,j} + \boldsymbol{\eta}_{i,j},$$

where $\mathbf{x}_{i,j} \in \mathbb{R}^n$ denotes the jth column of $\mathbf{X}_i$, each $\mathbf{A}_{i,j} \in \mathbb{R}^{m\times n}$ is a sensing matrix, and $\boldsymbol{\eta}_{i,j} \in \mathbb{R}^m$ is additive noise, typically with $m \ll n$. This setting arises in many signal processing applications; if $\mathbf{A}_{i,j}$ has iid Gaussian entries, we have a compressed sensing setting; if $\mathbf{A}_{i,j}$ are subsampled diagonal matrices, we have the missing data (or matrix completion) setting. Our goal in either of these cases is to estimate all latent signals $\{\mathbf{x}_{i,j}\}$ given the data $\{\mathbf{y}_{i,j}\}$ and the sensing matrices $\{\mathbf{A}_{i,j}\}$.

However, when $m < n$, the sensing matrix has a non-trivial nullspace, and the problem is under-determined. In the literature, a common approach is to assume a prior (or structure) on each signal $\mathbf{x}_{i,j}$ and use an appropriate regularizer. In this work, we start with a standard assumption that each $\mathbf{X}_i$ is nearly low-rank or can be well-approximated with its projection onto a low-rank subspace $\mathbf{U}_i$. In many applications where this is a good assumption, the latent signals are smoothly related over time in such a way that their low-rank approximations are also smoothly related.

This problem is also motivated by a generative model formulation. Suppose that at each time point $i \in [T]$, a total of ℓ data points $\mathbf{X}_i \in \mathbb{R}^{n\times\ell}$ are generated from a subspace $\mathbf{U}_i \in \mathcal{V}^{n\times r}$ to form a latent matrix

$$\mathbf{X} = [\mathbf{X}_1 \ \ldots \ \mathbf{X}_T] = [\mathbf{U}_1\mathbf{G}_1 \ \ldots \ \mathbf{U}_T\mathbf{G}_T] \in \mathbb{R}^{n\times\ell T}, \tag{1}$$

where each $\mathbf{G}_i \in \mathbb{R}^{r\times\ell}$ is a weighting matrix. Again, in many applications one might expect the sequence of subspaces $\{\mathbf{U}_i\}_{i=1}^T$ to be smoothly related over time. Our contribution is to model the sequence of best-fit subspaces $\mathbf{U}_i$ as being constrained to a geodesic on the Grassmannian. The proposed methodology applies

equally well to both problem perspectives: low-rank approximation and generative subspace estimation. Before presenting our dynamic approach, we first review commonly used static approaches to highlight the differences from our proposed method.

2.2 Existing Methods

A common approach to this challenging estimation problem is to drop dependence of the subspace on time, essentially assuming the subspace changes so slowly that it can be ignored. Mathematically, by setting $\mathbf{X} = \mathbf{U}\mathbf{G}$ for some weighting matrix $\mathbf{G} \in \mathbb{R}^{r \times \ell T}$, we could solve the following optimization problem:

$$\min_{\mathbf{U},\mathbf{G}} \quad \frac{1}{2}\|\mathbf{Y} - \mathcal{A}(\mathbf{U}\mathbf{G})\|_{\mathsf{F}}^2, \quad \text{where} \quad \mathbf{Y} \coloneqq [\mathbf{Y}_1 \ \ldots \ \mathbf{Y}_T] \in \mathbb{R}^{m \times \ell T}, \tag{2}$$

and $\mathcal{A}(\cdot) : \mathbb{R}^{n \times \ell T} \to \mathbb{R}^{m \times \ell T}$ is a linear sensing operator. We refer to this approach and related techniques as "static" methods, as they do not account for any temporal component of the underlying data. This problem has been widely studied in the literature and can be solved using various forms of alternating gradient-based methods (Jain et al., 2013). However, for applications with time-varying data, static methods may be suboptimal, as they ignore the temporal structure that could potentially be used to make more accurate data estimates.

For settings in which $\mathcal{A}(\cdot)$ is the identity map, there are many classical works on subspace tracking that approach the objective in Equation (2) with data streaming over time, but use constant step-sizes or linear-algebraic methods that essentially allow tracking of time-varying subspaces (Yang, 1995; Haghighatshoar & Caire, 2018; Narayanamurthy & Vaswani, 2018b; Vaswani et al., 2018; Comon & Golub, 1990). The theoretical results in these earlier works are generally limited to asymptotic convergence guarantees with static underlying subspaces. On the other hand, Narayanamurthy & Vaswani (2018b) relax the assumption of constant subspace to a very slowly varying subspace between the change points. For a review of these methods, see Vaswani et al. (2018). Dynamic subspace estimation has also been studied in the batch setting where one has access to all the data at once, such as (Saad-Falcon et al., 2024a) and the following papers that use the Grassmannian geodesic model discussed in this paper (Lake & Keenan, 1998b; Fuhrmann, 1997; Srivastava & Klassen, 2004; Hong et al., 2016).

When data are observed through compressive maps $\mathcal{A}(\cdot)$, such as missing data or general compressive operators, the existing works are those that build on the subspace tracking algorithms based on optimization techniques. For example, the PETRELS algorithm (Chi et al., 2012) is a recursive least-squares approach for the missing data setting, and provides convergence theory that assumes that the subspace changes at a particular instant and then stays constant for sufficient time so that the change can be tracked (also called the piecewise constant model). Oja's method and GROUSE (Allen-Zhu & Li, 2017; Balzano et al., 2010) can also handle missing data or compressive observations (Zhang & Balzano, 2016; Balzano, 2022). For a survey of missing data tracking methods, see Balzano et al. (2018a).

While we defer a comparison of our work with these existing methods in the case where the linear operator is the identity mapping to Appendix B (see Figure 10), we primarily compare our approach to the static method in Equation (2). This choice is motivated by two reasons: (i) it is often nontrivial to extend these methods to cases where the linear operator is not the identity, and (ii) many existing works assume that subspaces change one at a time and that the changes between subspaces are minimal—assumptions that are not made by our algorithm.

2.3 Proposed Method: A Dynamic Approach via Grassmannian Geodesics

In this work, we consider estimating all data points $\{\mathbf{x}_{i,j}\}$ generated by the sequence of subspaces $\{\mathbf{U}_i\}$ by modeling the subspaces as a geodesic on the Grassmann manifold. Similar to lines in Euclidean space, Grassmannian geodesics behave as interpolating curves, and can be expressed as (Absil et al., 2004, Section 3.8):

$$\mathbf{U}_i \coloneqq \mathbf{U}(t_i) = \mathbf{H}\cos(\mathbf{\Theta} t_i) + \mathbf{Z}\sin(\mathbf{\Theta} t_i), \tag{3}$$

where $\mathbf{H} \in \mathcal{V}^{n \times r}$ is an orthonormal basis for a point on the Grassmannian, $\mathbf{Z} \in \mathcal{V}^{n \times r}$ is a matrix with orthonormal columns whose span is in the tangent space of range($\mathbf{H}$) (i.e., $\{\mathbf{Z} \in \mathcal{V}^{n \times r} \,|\, \mathbf{H}^\top \mathbf{Z} = \mathbf{0}\}$), $\mathbf{\Theta} \in \mathbb{R}^{r \times r}$

is a diagonal matrix with entries θ_s, where θ_s denotes the principal angle between two endpoints of a geodesic, and $t_i \in [0,1]$ is a scalar that represents the time of each $\mathbf{U}_i$.

Geometrically, we can think of $\mathbf{H}$ as some starting point on the Grassmannian, $\mathbf{Z}$ as a normalized direction we want to travel, and $\mathbf{\Theta} t_i$ as the distance to reach $\mathbf{U}_i$. If we are given the time points t_i (which we treat as tunable hyperparameters), then $\mathbf{H}, \mathbf{Z}$, and $\mathbf{\Theta}$ are all learnable parameters of $\mathbf{U}_i$ that we can use to estimate $\{\mathbf{x}_{i,j}\}$. Then, instead of using a hard constraint $\mathbf{x}_{i,j} = \mathbf{U}_i \mathbf{g}_{i,j}$ as in Equation (2), we propose using a regularizer that penalizes data points that lie outside of their respective subspaces. This leads to the following optimization problem that forms the central component of this work:

$$\begin{aligned} \min_{\mathbf{H},\mathbf{Z},\mathbf{\Theta},\{\mathbf{x}_{i,j}\}} \quad & \sum_{i=1}^{T}\sum_{j=1}^{\ell} \underbrace{\frac{1}{2}\|\mathbf{y}_{i,j} - \mathbf{A}_{i,j}\mathbf{x}_{i,j}\|_2^2}_{\text{data consistency}} + \underbrace{\frac{\lambda}{2}\|(\mathbf{U}_i\mathbf{U}_i^\top - \mathbf{I})\mathbf{x}_{i,j}\|_2^2}_{\text{subspace regularizer}} \\ \text{s.t.} \quad & \underbrace{\mathbf{U}_i = \mathbf{H}\cos(\mathbf{\Theta} t_i) + \mathbf{Z}\sin(\mathbf{\Theta} t_i)}_{\text{constrain to Grassmannian geodesic}}. \end{aligned} \tag{4}$$

For simplicity, throughout this paper, we often write the objective function as

$$\begin{aligned} \min_{\mathbf{H},\mathbf{Y},\mathbf{\Theta},\{\mathbf{x}_{i,j}\}} \quad & \sum_{i=1}^{T}\sum_{j=1}^{\ell} \frac{1}{2}\left\|\widetilde{\mathbf{y}}_{i,j} - \widetilde{\mathbf{A}}_{i,j}\mathbf{x}_{i,j}\right\|_2^2 \\ \text{s.t.} \quad & \mathbf{U}_i = \mathbf{H}\cos(\mathbf{\Theta} t_i) + \mathbf{Z}\sin(\mathbf{\Theta} t_i), \end{aligned} \tag{5}$$

where we define

$$\widetilde{\mathbf{y}}_{i,j} \coloneqq \begin{bmatrix} \mathbf{y}_{i,j} \\ \mathbf{0} \end{bmatrix}, \quad \widetilde{\mathbf{A}}_{i,j} \coloneqq \begin{bmatrix} \mathbf{A}_{i,j} \\ \sqrt{\lambda}\,(\mathbf{U}_i\mathbf{U}_i^\top - \mathbf{I}_n) \end{bmatrix}.$$

Remarks. Here, we provide some intuition as to why the subspace regularizer may be useful. Define $\mathbf{B}_i \coloneqq \mathbf{U}_i\mathbf{U}_i^\top - \mathbf{I}_n$ and

$$\mathbb{A}_{i,j} \coloneqq \operatorname{range}(\mathbf{A}_{i,j}^\top), \qquad \mathbb{B}_i \coloneqq \operatorname{range}(\mathbf{B}_i^\top).$$

If each of the measurement matrices in $\{\mathbf{A}_{i,j}\}$ has rank $m \geq r$ and $\dim(\mathbb{A}_{i,j} \cap \mathbb{B}_i) \leq m - r$, then we can lower bound the rank of $\widetilde{\mathbf{A}}_{i,j} \in \mathbb{R}^{(m+n)\times n}$ by

$$\begin{aligned} \operatorname{rank}\left(\widetilde{\mathbf{A}}_{i,j}\right) = \operatorname{rank}\left(\widetilde{\mathbf{A}}_{i,j}^\top\right) &= \dim(\mathbb{A}_{i,j}) + \dim(\mathbb{B}_i) - \dim(\mathbb{A}_{i,j} \cap \mathbb{B}_i) \\ &\geq m + (n - r) - (m - r) = n. \end{aligned}$$

Hence, $\widetilde{\mathbf{A}}_{i,j}$ would have full column rank and $\{\mathbf{x}_{i,j}\}$ would be identifiable, assuming we knew the subspaces $\{\mathbf{U}_i\}$. The assumption on the intersection between two subspaces is natural in a random, high-dimensional setting – if $\{\mathbf{A}_{i,j}\}$ are random with a continuous distribution over $\mathbb{R}^{m\times n}$ that were independent of $\{\mathbf{U}_i\}$, then $\dim(\mathbb{A}_{i,j} \cap \mathbb{B}_i) = m - r$.

Of course, since we do not know the subspaces $\{\mathbf{U}_i\}$ a priori, we need to estimate them from the data. Without the geodesic constraint, the problem would amount to estimating a total of T unrelated subspaces. For applications with time-varying data, it is often favorable to impose a relationship between consecutive subspaces $\mathbf{U}_i$. The Grassmannian geodesic constraint serves as a natural remedy for these problems, as we only need to estimate a $2r$-dimensional basis represented by $\mathbf{H} \in \mathcal{V}^{n\times r}$ and $\mathbf{Z} \in \mathcal{V}^{n\times r}$ across *all* time points, while enforcing smoothness between consecutive subspaces. Consequently, another advantage of this constraint is that we can allow $\ell < r$ (i.e., observe fewer data points than the intrinsic rank of the subspace) and still achieve accurate recovery.

3 Riemannian Block Majorize-Minimize Algorithm

This section outlines a Riemannian block majorize-minimization (RBMM) algorithm for solving Equation (4). Section 3.1 presents the steps for alternately updating the block matrix $\mathbf{Q} \coloneqq [\mathbf{H}, \mathbf{Z}] \in \mathcal{V}^{n\times 2r}$, each diagonal

coordinate of $\boldsymbol{\Theta} \in \mathbb{R}^{r\times r}$, and each data sample $\mathbf{x}_{i,j} \in \mathbb{R}^n$. Section 3.2 introduces a spectral scheme for initializing each of the factors. Algorithm 1 summarizes the steps. For our theoretical analysis, we introduce proximal parameters $\lambda_{\mathbf{Q}}$, $\lambda_{\boldsymbol{\Theta}}$, $\lambda_{\mathbf{x}}$ corresponding to each block to ensure convergence to a stationary point. However, our experiments show that these parameters do not play a significant role in performance and do not need to be tuned.

3.1 Riemannian Block Update Steps

This section summarizes the update steps for our RBMM algorithm. Appendix A derives the complete update steps. Throughout this section, for simplicity, we denote

$$\mathbf{U}_i = \mathbf{H}\cos(\boldsymbol{\Theta} t_i) + \mathbf{Z}\sin(\boldsymbol{\Theta} t_i) = \mathbf{Q}\mathbf{R}_i, \quad \text{where} \quad \mathbf{Q} \coloneqq [\mathbf{H}, \mathbf{Z}], \quad \mathbf{R}_i \coloneqq \begin{bmatrix}\cos(\boldsymbol{\Theta} t_i)\\ \sin(\boldsymbol{\Theta} t_i)\end{bmatrix}.$$

Updates for X. Suppose $\mathbf{U}_i = \mathbf{Q}\mathbf{R}_i$ are fixed and define $f(\mathbf{x}_{i,j}) = \frac{1}{2}\|\widetilde{\mathbf{y}}_{i,j} - \widetilde{\mathbf{A}}_{i,j}\mathbf{x}_{i,j}\|_2^2$. We construct a majorizer g^k at iteration k for f from the second-order Taylor expansion and then update $\mathbf{X}$ by minimizing that majorizer:

$$\mathbf{x}_{i,j}^{k+1} = \underset{\mathbf{x}_{i,j}}{\arg\min}\, g^k(\mathbf{x}_{i,j}; \mathbf{x}_{i,j}^k) = \underset{\mathbf{x}_{i,j}}{\arg\min} \left\| \mathbf{x}_{i,j} - \left(\mathbf{x}_{i,j}^k - \frac{1}{L+\lambda_{\mathbf{x}}} \nabla f(\mathbf{x}_{i,j}^k) \right) \right\|_2^2 = \mathbf{x}_{i,j}^k - \frac{1}{L+\lambda_{\mathbf{x}}} \nabla_{\mathbf{x}} f(\mathbf{x}_{i,j}^k), \quad (6)$$

where L is the Lipschitz constant of f, $\nabla_{\mathbf{x}} f(\mathbf{x}_{i,j}^k) = \widetilde{\mathbf{A}}_{i,j}^\top(\widetilde{\mathbf{A}}_{i,j}\mathbf{x}_{i,j}^k - \widetilde{\mathbf{y}}_{i,j})$ and $\lambda_{\mathbf{x}} > 0$ is the parameter introduced by adding a proximal term to ensure a quadratic majorization gap (see Appendix A). This update step is a gradient descent step with a fixed step size that ensures a monotonically non-increasing objective function value.

Updates for Q. To update $\mathbf{Q}$, suppose that each $\mathbf{R}_i$ and $\mathbf{X}_i$ are fixed. Then, notice that the data consistency term in Equation (4) is not a function of $\mathbf{Q}$, so we only need to consider the subspace regularizer:

$$\widehat{\mathbf{Q}} = \underset{\mathbf{Q}\in\mathcal{V}^{n\times 2r}}{\arg\min}\, f(\mathbf{Q}) = \underset{\mathbf{Q}\in\mathcal{V}^{n\times 2r}}{\arg\min} \frac{\lambda}{2} \sum_{i=1}^{T}\sum_{j=1}^{\ell} \left\| \left(\mathbf{Q}\mathbf{R}_i\mathbf{R}_i^\top\mathbf{Q}^\top - \mathbf{I}_n\right)\mathbf{x}_{i,j} \right\|_2^2 \quad (7)$$

$$= \underset{\mathbf{Q}\in\mathcal{V}^{n\times 2r}}{\arg\min} \frac{\lambda}{2} \sum_{i=1}^{T} \left\| \left(\mathbf{Q}\mathbf{R}_i\mathbf{R}_i^\top\mathbf{Q}^\top - \mathbf{I}_n\right)\mathbf{X}_i \right\|_{\mathsf{F}}^2. \quad (8)$$

By constructing a linear majorizer g^k and projecting onto the Stiefel manifold (Absil et al., 2007; Higham, 1989), we obtain

$$\mathbf{Q}^{k+1} = \underset{\mathbf{Q}\in\mathcal{V}^{n\times 2r}}{\arg\min}\, g^k(\mathbf{Q}; \mathbf{Q}^k) = \left\| \mathbf{Q} - \left(\lambda_{\mathbf{Q}}\mathbf{Q}^k - \nabla f_{\mathbf{Q}}(\mathbf{Q}^k)\right) \right\|_{\mathsf{F}}^2 \quad (9)$$

$$= \mathbf{W}\mathbf{V}^\top, \quad \text{where } \left(\lambda_{\mathbf{Q}}\mathbf{Q}^k - \nabla f_{\mathbf{Q}}(\mathbf{Q}^k)\right) = \mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^\top, \quad (10)$$

and $\nabla f_{\mathbf{Q}}(\mathbf{Q}) = -\lambda\sum_{i=1}^{T} \mathbf{X}_i\mathbf{X}_i^\top\mathbf{Q}\mathbf{R}_i\mathbf{R}_i^\top$ and $\lambda_{\mathbf{Q}} > 0$ is the proximal parameter for $\mathbf{Q}$.

Updates for Θ. To preserve the diagonal structure of $\boldsymbol{\Theta}$, we update each coordinate of $\boldsymbol{\Theta}$ present in each $\mathbf{R}_i$, which we denote as $\theta_s \in \mathbb{R}$, $\forall s \in [r]$. Similarly, consider the objective function

$$\widehat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\arg\min}\, f(\boldsymbol{\Theta}) = \underset{\boldsymbol{\Theta}}{\arg\min} \frac{\lambda}{2} \sum_{i=1}^{T} \left\| \left(\mathbf{Q}\mathbf{R}_i\mathbf{R}_i^\top\mathbf{Q}^\top - \mathbf{I}_n\right)\mathbf{X}_i \right\|_{\mathsf{F}}^2.$$

Simplifying this loss function in terms of θ_s leads to

$$\widehat{\theta}_s = \underset{\theta_s}{\arg\min}\, -\frac{\lambda}{2}\sum_{i=1}^{T}\sum_{s=1}^{r} f_{i,s}(\theta_s) = \underset{\theta_s}{\arg\min}\, -\frac{\lambda}{2}\sum_{i=1}^{T}\sum_{s=1}^{r} r_{i,s}\cdot\cos(2\theta_s t_i - \phi_{i,s}), \quad (11)$$

Algorithm 1 RBMM Algorithm for Dynamic Subspace Estimation

Require: Observations $\{\mathbf{y}_{i,j}\} \in \mathbb{R}^m$; Sensing matrices $\{\mathbf{A}_{i,j}\} \in \mathbb{R}^{m\times n}$; Parameters $\lambda_{\mathbf{x}}, \lambda_{\mathbf{Q}}, \lambda_{\mathbf{\Theta}}, \lambda \geq 0$; Iterations K; Geodesic parameters $t_i \in [0,1], \forall i \in [T]$

for $i = 1, \ldots, T$ **do**
  for $j = 1, \ldots, \ell$ **do**
    $\mathbf{C}_{i,j} = \mathbf{B}_{i,j}\mathbf{A}_{i,j}$ ▷ Construct surrogate matrix in Equation (15)
    $\mathbf{x}_{i,j}^0 \leftarrow \mathrm{SVD}(\mathbf{C}_{i,j})$ ▷ Initialize via top left singular vector of $\mathbf{C}_{i,j}$
  end for
end for
$\mathbf{X}_i^0 = [\mathbf{x}_{i,1}^0 \ \ldots \ \mathbf{x}_{i,\ell}^0] \in \mathbb{R}^{n\times \ell}$ ▷ Concatenate initialized samples at time point i
$\mathbf{X}^0 = [\mathbf{X}_1^0 \ \ldots \ \mathbf{X}_T^0] \in \mathbb{R}^{n\times \ell T}$ ▷ Concatenate all initial samples across time
$\mathbf{H}^0, \mathbf{Z}^0, \mathbf{\Theta}^0 \leftarrow \mathrm{SVD}(\mathbf{X}^0)$ ▷ Initialize via Equations (16, 17)
for $k = 1, \ldots, K$ **do** ▷ Start RBMM algorithm
  for $i = 1, \ldots, T$ **do**
    for $j = 1, \ldots, \ell$ **do**
      $\mathbf{x}_{i,j}^{k+1} = \mathbf{x}_{i,j}^k - \frac{1}{L+\lambda_{\mathbf{x}}}\nabla_{\mathbf{x}} f(\mathbf{x}_{i,j}^k)$ ▷ Update signals $\mathbf{x}_{i,j}$ from Equation (6)
    end for
  end for
  $\left(\lambda_{\mathbf{Q}}\mathbf{Q}^k - \nabla f_{\mathbf{Q}}(\mathbf{Q}^k)\right) = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^\top$ ▷ Compute SVD
  $\mathbf{Q}^{k+1} = \mathbf{W}\mathbf{V}^\top$ ▷ Update $\mathbf{Q}$ from Equation (9)
  for $s = 1, \ldots, r$ **do**
    $\theta_s^{k+1} = \theta_s^k - \sum_{i=1}^T \frac{\dot{f}_{i,s}(\theta_s^k)}{w_{f_{i,s}}(\theta_s^k)}$ ▷ Update each θ_s from Equations (13, 14)
  end for
end for
Return: $\mathbf{X}_i^K, \ \forall i \in [T]$ ▷ Return estimated data points

with the following definitions:

$$\begin{aligned} \alpha_{i,s} &\coloneqq \left[\mathbf{H}^\top \mathbf{X}_i \mathbf{X}_i^\top \mathbf{H}\right]_{s,s} & r_{i,s} &\coloneqq \sqrt{\left(\frac{\alpha_{i,s}-\gamma_{i,s}}{2}\right)^2 + \beta_{i,s}^2} \\ \beta_{i,s} &\coloneqq \mathrm{real}\left\{\left[\mathbf{Z}^\top \mathbf{X}_i \mathbf{X}_i^\top \mathbf{H}\right]_{s,s}\right\} & \phi_{i,s} &\coloneqq \arctan2\left(\beta_{i,s}, \frac{\alpha_{i,s}-\gamma_{i,s}}{2}\right) \\ \gamma_{i,s} &\coloneqq \left[\mathbf{Z}^\top \mathbf{X}_i \mathbf{X}_i^\top \mathbf{Z}\right]_{s,s}. \end{aligned} \tag{12}$$

Upon constructing a quadratic majorizer (Funai et al., 2008) (see Appendix A.3), we have the update step

$$\theta_s^{k+1} = \theta_s^k - \sum_{i=1}^T \frac{\dot{f}_{i,s}(\theta_s^k)}{w_{f_{i,s}}(\theta_s^k)}, \tag{13}$$

where $\dot{f}_{i,s}(\theta_s^k) = \lambda r_{i,s} t_i \sin(2\theta_s t_i - \phi_{i,s})$ and

$$w_{f_{i,s}}(\theta_s^k) = \begin{cases} \frac{\dot{f}_{i,s}(\theta_s^k)}{\mathrm{mod}\left((\theta_s^k - \frac{\phi_{i,s}}{2t_i}) + \frac{\pi}{2t_i}, \frac{2\pi}{2t_i}\right) - \frac{\pi}{2t_i}} + \lambda_{\mathbf{\Theta}} & \theta_s^k \neq \frac{\phi_{i,s}+2\pi h}{2t_i},\ h \in \mathbb{Z} \\ 2\lambda r_{i,s} t_i^2 + \lambda_{\mathbf{\Theta}} & \theta_s^k = \frac{\phi_{i,s}+2\pi h}{2t_i},\ h \in \mathbb{Z}, \end{cases} \tag{14}$$

where $\lambda_{\mathbf{\Theta}} > 0$ is the proximal parameter for $\mathbf{\Theta}$.

3.2 Spectral Initialization

Due to the non-convex nature of our problem, a smart initialization facilitates convergence to good solutions. This section presents a spectral initialization scheme for each of the blocks that empirically improves the performance of our algorithm.

Initialization for X. Suppose that every entry of the sensing matrix $\mathbf{A}_{i,j}$ had an independent, random distribution over $\mathbb{R}^{m\times n}$ with zero mean and unit variance. To initialize each $\mathbf{x}_{i,j}$, consider the surrogate matrix

$$\mathbf{C}_{i,j} = \mathbf{B}_{i,j}\mathbf{A}_{i,j}, \quad \text{where } \mathbf{B}_{i,j} = \text{Diag}(\mathbf{y}_{i,j}). \tag{15}$$

After some elementary calculations, we have

$$\mathbb{E}\left[\mathbf{C}_{i,j}^\top\right] = \begin{bmatrix}\mathbf{x}_{i,j} & \mathbf{x}_{i,j} & \dots & \mathbf{x}_{i,j}\end{bmatrix},$$

which is a rank-1 matrix, and so we initialize each $\mathbf{x}_{i,j}$ by taking the first left singular vector of $\mathbf{C}_{i,j}$.

Initialization for H, Z, Θ. We first introduce a *random* scheme for initializing the geodesic parameters. Let $\mathbf{H}_1, \mathbf{H}_2 \in \mathcal{V}^{n\times r}$ be two orthonormal bases for two points on the Grassmannian. Given $t_i \in [0,1]$, a point in between these two endpoints is given

$$\mathbf{U}_i = \text{span}\left(\mathbf{H}_1\mathbf{V}\cos(\mathbf{\Theta}t_i) + \mathbf{Z}\sin(\mathbf{\Theta}t_i)\right), \tag{16}$$

where $\left(\mathbf{I} - \mathbf{H}_1\mathbf{H}_1^\top\right)\mathbf{H}_2\left(\mathbf{H}_1^\top\mathbf{H}_2\right)^{-1} = \mathbf{Z}\mathbf{S}\mathbf{V}^\top$ is a compact SVD and $\mathbf{\Theta} = \tan^{-1}(\mathbf{S})$. Thus, a random initialization for $\mathbf{H} \in \mathcal{V}^{n\times r}, \mathbf{Z} \in \mathcal{V}^{n\times r}$ and $\mathbf{\Theta} \in \mathbb{R}^{r\times r}$ takes any two orthonormal bases and applies Equation (16).

However, we can obtain more accurate initial points by using information from our data. Suppose $n \geq 2r$ and that we observed the data $\mathbf{X}$ directly (i.e., $\mathbf{A}_{i,j} = \mathbf{I}_n$). If we assumed that $\mathbf{X}$ was truly generated from a geodesic, based on Figure 1, we can expect the two endpoints of the geodesic, $\mathbf{U}_1$ and $\mathbf{U}_T$, to serve as bases for two endpoints of the geodesic. Consider taking two subsets of the concatenated data as follows:

$$\mathbf{X} = [\underbrace{\mathbf{X}_1}_{=\mathbf{D}_1\in\mathbb{R}^{n\times r}} \overbrace{\mathbf{x}_{2,1} \dots \mathbf{x}_{2,\ell}}^{\mathbf{X}_2} \dots \overbrace{\mathbf{x}_{T-1,1} \dots \mathbf{x}_{T-1,\ell}}^{\mathbf{X}_{T-1}} \underbrace{\mathbf{X}_T}_{=\mathbf{D}_2\in\mathbb{R}^{n\times r}}]. \tag{17}$$

Then, we use the left singular vectors of $\mathbf{D}_1$ and $\mathbf{D}_2$ as $\mathbf{H}_1$ and $\mathbf{H}_2$, respectively, and apply Equation (16). We call this procedure *spectral* initialization. Since we do not observe $\mathbf{X}$ directly, we use the initialization method for $\mathbf{X}$ above and apply spectral initialization for the geodesic parameters.

4 THEORETICAL RESULTS

This section presents the theoretical guarantees of our RBMM algorithm.

Theorem 1 (Convergence to a Stationary Point)**.** *Let $\mathbf{\Psi}^k = \{\mathbf{X}_i^k\}_{i=1}^T \cup \{\mathbf{Q}^k, \mathbf{\Theta}^k\}$ denote the iterates generated by Algorithm 1 at iteration k starting from any initialization with proximal parameters $\lambda_\mathbf{x}, \lambda_\mathbf{Q}, \lambda_\mathbf{\Theta} > 0$. For any $\epsilon > 0$, if $k = \widetilde{\mathcal{O}}(\epsilon^{-2})$, then we have*

$$\sum_{p=1}^{T+2} \|\nabla f(\mathbf{\Psi}_p^k)\| \leq \epsilon, \tag{18}$$

where p indexes the $T+2$ sequentially updated blocks of variables in $\mathbf{\Psi}^k$.

Appendix C provides all proofs. The proximal parameters in Theorem 1 can be viewed as values that ensure each update step is bounded in some sense – values closer to zero result in larger update steps, while larger values restrict the update steps, consequently leading to slower convergence. Due to the non-convex nature of our problem, it is very challenging to obtain guarantees of convergence to the global minimum. However, from our synthetic experiments in a planted geodesic setting, we observe empirically that our method with spectral initialization often converges to the ground truth (see Figure 5).

Theorem 2 (Monotonic Objective Function)**.** *Let $\mathbf{\Psi}^k = \{\mathbf{X}_i^k\}_{i=1}^T \cup \{\mathbf{Q}^k, \mathbf{\Theta}^k\}$ denote the iterates generated by Algorithm 1 at iteration k. Algorithm 1 produces blocks that are monotonically non-increasing in the loss: $f(\mathbf{\Psi}^{k+1}) \leq f(\mathbf{\Psi}^k)$.*

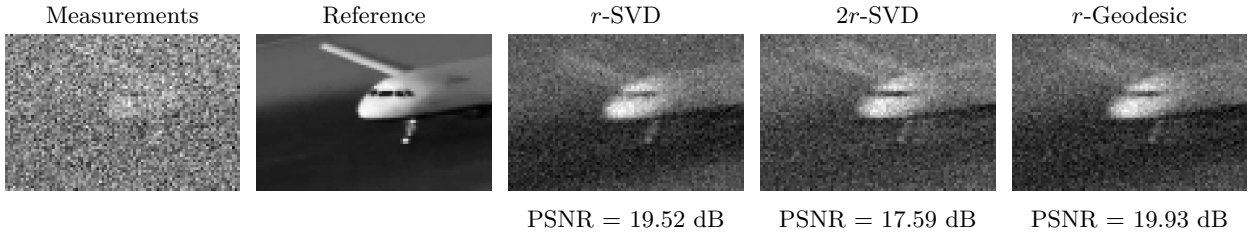

Figure 2: Visual results with corresponding PSNRs for denoising the plane video dataset with Gaussian noise with standard deviation $\sigma_{\mathbf{y}} = 0.4$ and rank $r = 1$.

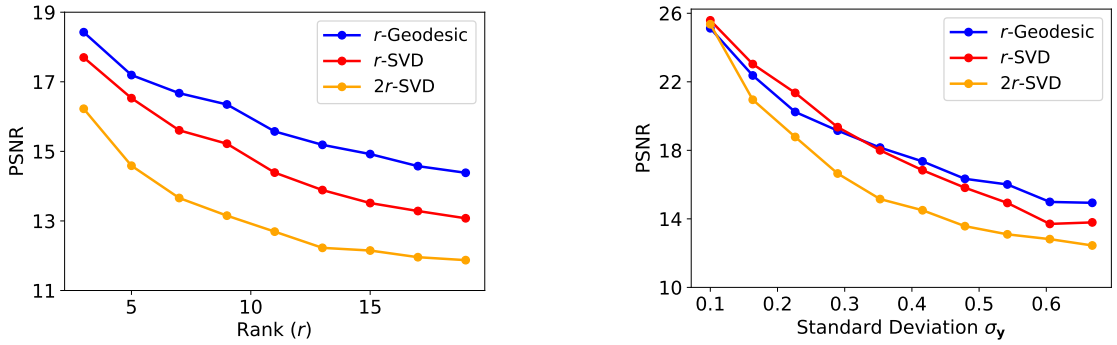

Figure 3: Experimental results on the plane video dataset for denoising with fully observed data. Left: Plot of the PSNR for varying values of the rank r for fixed $\sigma_{\mathbf{y}} = 0.40$. Right: Plot of the PSNR for varying values of the standard deviation of the added noise $\sigma_{\mathbf{y}}$ for a fixed rank $r = 5$. The PSNR is averaged across all video frames.

Generally, existing algorithms that use optimization methods such as gradient descent do not guarantee a monotonically non-increasing objective function unless the step size is appropriately tuned. Our algorithm possesses this property without requiring additional step size tuning, which is particularly advantageous in domains with intrinsically high-dimensional data, where cross-validation can be costly. Theorem 2 holds independently of the proximal parameters – they can be set to any non-negative value and the statement will remain valid.

5 EXPERIMENTS

This section presents our experimental results that are divided into two parts: experiments with fully sampled data (Section 5.1) and with undersampled data (Section 5.2). Both parts include experiments on synthetic and real video data, aimed at analyzing the behavior and performance of our proposed algorithm. For our algorithm, we set $\lambda_{\mathbf{x}} = \lambda_{\mathbf{Q}} = \lambda_{\mathbf{\Theta}} = 0$ and $\lambda = 1$, unless otherwise specified. We considered the three following performance metrics:

$$\text{Subspace Error} = \frac{1}{2rT} \sum_{i=1}^{T} \|\mathbf{U}_i \mathbf{U}_i^\top - \widehat{\mathbf{U}}_i \widehat{\mathbf{U}}_i^\top\|_{\mathsf{F}}^2,$$

for any two subspaces $\mathbf{U}_i$ and $\widehat{\mathbf{U}}_i$, the normalized root mean-squared error (NRMSE), defined by $\|\widehat{\mathbf{X}} - \mathbf{X}\|_{\mathsf{F}} / \|\mathbf{X}\|_{\mathsf{F}}$, and the peak-signal-to-noise ratio (PSNR).

5.1 Experiments with Fully Sampled Data

Before presenting experimental results for the undersampled data case, we first consider the fully sampled setting, i.e., $\mathbf{A}_{i,j} = \mathbf{I}_n$. In this case, the data consistency term in Equation (4) simplifies to a denoising task,

where the data points are regularized to lie on a geodesic. The goal of this setting is to (i) understand the behavior of the RBMM algorithm and how it differs from the undersampled case, and (ii) investigate how enforcing a geodesic structure aids in denoising compared to static approaches, i.e., the truncated SVD.

Synthetic Experiments. Here, we performed experiments in a planted geodesic setting to investigate the dependence of the algorithm's performance on its parameters. To this end, we generated measurements according to

$$\mathbf{y}_{i,j} = \mathbf{U}_i \mathbf{g}_{i,j} + \boldsymbol{\eta}_{i,j} \in \mathbb{R}^n, \qquad \forall j \in [\ell],\ \forall i \in [T], \tag{19}$$

where $\mathbf{g}_{i,j} \sim \mathcal{N}(0, \mathbf{I}_r)$ and $\boldsymbol{\eta}_{i,j} \sim \mathcal{N}(0, \sigma_{\mathbf{y}} \cdot \mathbf{I}_m)$, with $n = 40$. Figure 4 plots the subspace error for our algorithm and the r-SVD for different values of ℓ, where $T = 51$, $r = 4$, and $\sigma_{\mathbf{y}} = 10^{-2}$. The SVD method involves computing the top-r SVD components of the data at each time point and, hence, can only be computed when $\ell \geq r$. On the other hand, our algorithm works even for $\ell < r$ while achieving a lower error.

Figure 5a shows the fraction of times our algorithm converged to the ground-truth solution as measured by the subspace error, for $\ell = 1$, across varying values of T and r. There is a clear phase transition at $T = 2r$, as the geodesic assumption amounts to estimating $2nr$ parameters from a total of $Tn\ell$ data points.

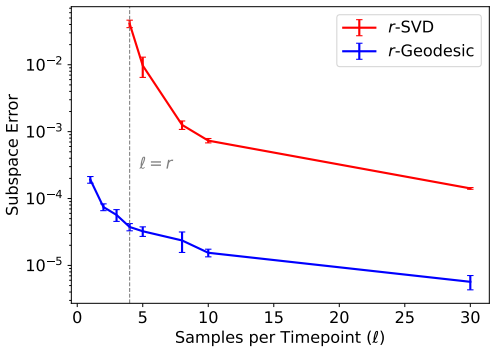

Figure 4: Subspace error for varying ℓ in the fully sampled case.

Video Data Experiments. Next, we present results on a video denoising task, where the video consists of frames showing a plane landing on a runway. The video has a total of 105 frames, each with dimensions (60×90). This video was used in other inverse problem contexts (Nayer et al., 2020; Kwon et al., 2022), albeit for a different application. For preprocessing, we normalized pixel values to $[0, 1]$ and mean-centered the data after vectorizing each frame. For our algorithm, we chose $r = 5$ and set $\ell = r - 1$ unless otherwise specified. This case emphasizes the fact that our algorithm can handle $\ell < r$. We evenly sampled the time steps $t_i \in [0.15, 0.85]$, as this range generally yielded the best performance. We compared our algorithm against the r-SVD and $2r$-SVD baselines, which performed an SVD of the corresponding rank on the data and projected the frames onto the resulting subspaces. We consider $2r$ as a baseline, as $\mathbf{Q}$ spans a $2r$-dimensional subspace.

Figure 2 presents visual results on a single frame, demonstrating that our algorithm denoised the video more effectively than the SVD-based baselines in terms of PSNR when $\sigma_{\mathbf{y}} = 0.40$. This suggests that incorporating additional structure can be beneficial for such tasks, especially in highly noisy settings. Figure 3 shows the PSNR for varying values of the rank for a fixed $\ell = 4$ (left) and the standard deviation of the added noise (right). In less noisy settings, the SVD-based solutions appear to outperform our method, but the gap quickly disappears as the noise level increases.

5.2 Experiments with Undersampled Data

This section presents experimental results in the under-sampled setting. Similar to Section 5.1, we experimented with both synthetic and video data, and we also include experiments on real data, which involve reconstructing fMRI images from sub-sampled k-space data. In the under-sampled setting, since we do not have access to the full dataset, we selected the rank for the video data through trial and error, and we present results using two different rank choices. For the fMRI data, we chose the rank based on a data-sharing method, which involves searching for the nearest row of the non-missing data sample and using that to reconstruct the data. In Figure 17a, we show that the singular value plots for the data-shared reconstructed frames and the original frames exhibit similar patterns, justifying the choice of rank.

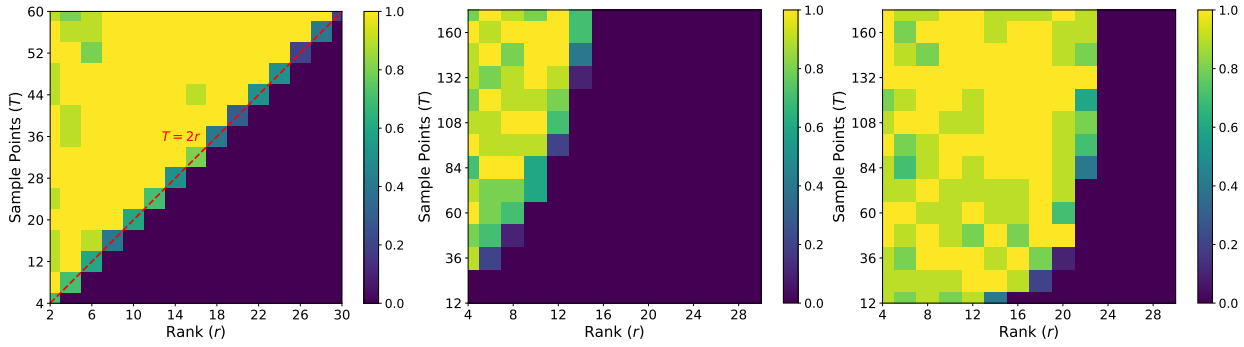

Figure 5: Fraction of convergence to ground truth for varying values of r and T over 10 random runs for a fixed m. The fraction is the number of times the subspace error went below a threshold of 10^{-4} over the total number of trials. (a): Fully-sampled case with $m = n = 40$ and $\ell = 1$. (b): Under-sampled case with $m = \frac{1}{2}n = 20$ and $\ell = 1$. (c): Under-sampled case with $m = \frac{1}{2}n = 20$ and $\ell = r - 1$. For the under-sampled settings, the number of data points ℓ for each time point makes a notable difference in the recovery of the underlying subspaces.

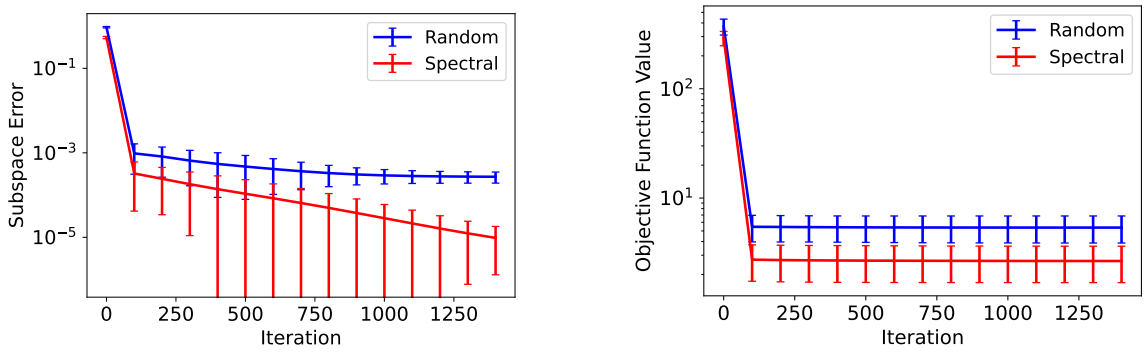

Figure 6: Results showing the benefits of spectral initialization with $\sigma_{\mathbf{y}} = 0.01$ averaged over 10 runs for matrix completion. The spectral initialization scheme leads to faster convergence in terms of iterations.

Synthetic Experiments. Similar to Section 5.1, we considered experiments in a planted geodesic setting, where the objective is to: (i) demonstrate the efficiency of our spectral initialization method compared to random initialization; (ii) observe the dependency of the parameters and their effects on the performance in contrast to the fully-sampled setting; (iii) compare the performance of our algorithm on noisy matrix sensing to the baseline AltMin (Jain et al., 2013), which is an alternating minimization scheme for updating two r-dimensional factors. Since AltMin is a matrix recovery algorithm, we vectorized and stacked all of the signals to construct a single matrix. Appendix B.2 provides more experimental details on the baselines. For the last experiment, our goal was to investigate if the "smoothness" between consecutive subspaces, enforced by the geodesic constraint, assists in dealing with measurement noise.

The results in this section used the parameters $n = 40$, $m = 20$, $r = 3$, $\ell = 1$, and $T = 80$, unless otherwise specified. For each 3-dimensional subspace, we observed only a single vector. To generate data from a geodesic, we first took two randomly generated orthonormal bases and applied Equation (16) to obtain $\mathbf{U}_i \in \mathcal{V}^{n\times r}$ for all $i \in [T]$. Then, in contrast to Equation (19), we generate measurements according to

$$\mathbf{y}_{i,j} = \mathbf{A}_{i,j}\mathbf{U}_i\mathbf{g}_{i,j} + \boldsymbol{\eta}_{i,j} \in \mathbb{R}^m, \qquad \forall j \in [\ell],\ \forall i \in [T], \tag{20}$$

where $\mathbf{g}_{i,j} \sim \mathcal{N}(0, \mathbf{I}_r)$, $\boldsymbol{\eta}_{i,j} \sim \mathcal{N}(0, \sigma_{\mathbf{y}} \cdot \mathbf{I}_m)$, and each column of $\mathbf{A}_{i,j} \in \mathbb{R}^{m\times n}$ is drawn iid from a Gaussian distribution (i.e., $\mathbf{a}_{i,j,k} \sim \mathcal{N}(0, \mathbf{I}_n)$) in the matrix sensing setting and sub-sampled diagonal matrices for the matrix completion setting. Here, $\sigma_{\mathbf{y}}$ is assumed to be 0 unless otherwise stated.

Dataset	Method	$r = 3$		$r = 5$	
		PSNR ↑	RMSE ↓	PSNR ↑	RMSE ↓
Mouse	r-AltMin (Jain et al., 2013)	20.55 ± 0.030	0.108 ± 0.001	22.05 ± 0.061	0.089 ± 0.001
	$2r$-AltMin (Jain et al., 2013)	$\underline{22.23} \pm 0.080$	$\underline{0.086} \pm 0.001$	$\underline{22.92} \pm 0.098$	$\mathbf{0.080} \pm 0.001$
	r-GNMR (Zilber & Nadler, 2022)	19.61 ± 0.095	0.128 ± 0.003	20.73 ± 0.348	0.114 ± 0.011
	$2r$-GNMR (Zilber & Nadler, 2022)	20.88 ± 0.095	0.115 ± 0.014	19.22 ± 0.314	0.129 ± 0.010
	r-Geodesic (Ours)	$\mathbf{22.70} \pm 0.158$	$\mathbf{0.083} \pm 0.001$	$\mathbf{23.28} \pm 0.164$	$\underline{0.082} \pm 0.001$
Plane	r-AltMin (Jain et al., 2013)	25.27 ± 0.058	0.058 ± 0.001	25.25 ± 0.047	0.058 ± 0.001
	$2r$-AltMin (Jain et al., 2013)	$\underline{28.47} \pm 0.274$	$\mathbf{0.040} \pm 0.001$	$\underline{28.61} \pm 0.114$	$\underline{0.040} \pm 0.001$
	r-GNMR (Zilber & Nadler, 2022)	24.37 ± 0.854	0.072 ± 0.022	24.39 ± 0.333	0.067 ± 0.002
	$2r$-GNMR (Zilber & Nadler, 2022)	27.00 ± 0.827	0.061 ± 0.014	27.14 ± 0.832	0.055 ± 0.006
	r-Geodesic (Ours)	$\mathbf{29.01} \pm 0.207$	$\underline{0.040} \pm 0.002$	$\mathbf{29.20} \pm 0.379$	$\mathbf{0.039} \pm 0.003$

Table 1: Quantitative results for varying ranks for different metrics for both datasets across 10 trials. Both datasets are corrupted with Gaussian noise of standard deviation $\sigma_{\mathbf{y}} = 0.01$, where for the mouse data we observe 50% of each frame, and for the plane data we observe 40% of each frame. Best results are bolded and second best results are underlined.

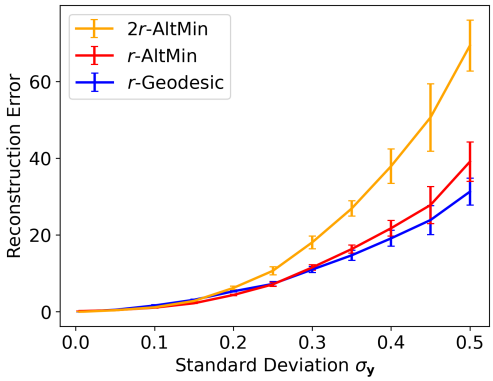

Figure 7: Reconstruction error (NRMSE) over $\sigma_{\mathbf{y}}$. For a larger noise variance, $2r$-AltMin seems to overfit the noise rapidly, whereas our method is more robust.

Figure 6 demonstrates that our algorithm can recover the ground truth data and subspace more quickly (in terms of iterations) using our spectral initialization method compared to random initialization. While random initialization often converges to poor local minima, spectral initialization often converges to a solution with near-zero subspace error. Nevertheless, this illustrates that even within noisy settings, our algorithm can faithfully estimate the data. Appendix B.1 illustrates the benefits of the spectral method and its failure cases.

Figures 5b and 5c show phase transitions similar to Figure 5a to obtain convergence to the ground truth solution. In Figure 5b, we vary T and r while fixing $m = 0.5n = 20$ and $\ell = 1$. Figure 5c is a similar setting, but with $\ell = r - 1$. Unlike the fully-sampled case, the phase transition is less clear in both cases and seem to require more samples once the rank r gets closer to d. For the under-sampled cases, it seems that the number of data points at each time point ℓ plays an important role in ensuring accurate recovery. Nevertheless, for our applications of interest, since we generally assume that $r \ll d$, we can still faithfully use our method with a smaller number of time points T. We leave a rigorous analysis of the relationship amongst the parameters for convergence for future work.

Lastly, we compare the performance of our algorithm compared to AltMin with a rank choice of r and $2r$ with varying $\sigma_{\mathbf{y}}$ for matrix sensing. By enforcing the smoothness constraint via the geodesic, we expect our model to handle measurement noise more efficiently. Figure 7 verifies this conjecture: our algorithm obtains a much lower reconstruction error than $2r$-AltMin as a function of noise.

fMRI Data Experiments. Here, we present experiments on reconstructing dynamic fMRI images. We considered an oscillating steady state imaging (OSSI) dataset (Guo et al., 2020) that was acquired on a 3T GE MR750 scanner with a 32-channel head coil. The data were comprised of 167 slow-time, 10 fast-time, and (128×128) spatial samples. We considered a single point in fast time and 160 slow-time points, which gave us a dataset of 160 frames, where each frame was of dimension (128×128). We considered the slow-time points because we hypothesize that the slowly changing movements were suitable for a geodesic model.

To this end, we used an approximately 50% missing k-space sampling mask on each frame, where the center of the mask was fully observed but rows above and below the center were missing (see Figure 16 for examples). For our algorithm, we chose parameters $\ell = 4$ to match the slow time frames, which leads to $T = 40$ and $\lambda = 0.1$. We evenly sampled t_i from the range $[0, 0.5]$ and chose the rank to be $r = 5$. For the baselines,

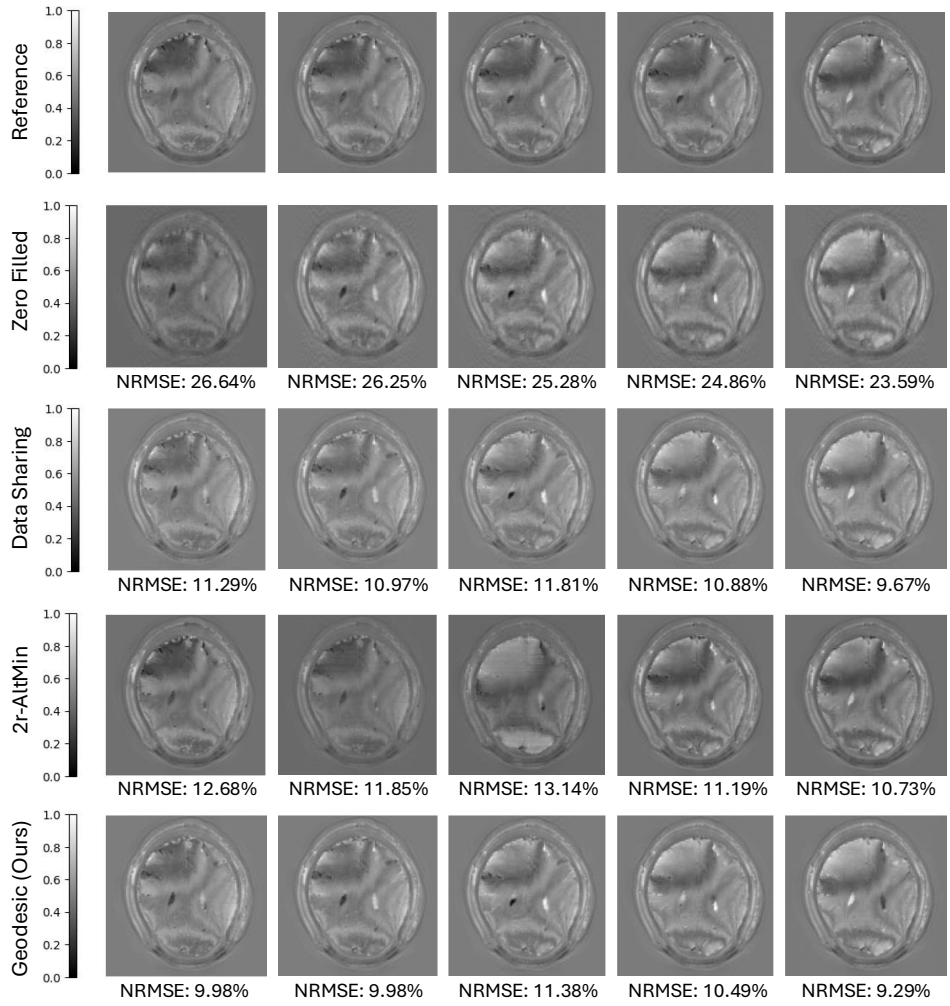

Figure 8: fMRI reconstruction results, where each frame was masked by a 50% sampled k-space mask. Both our algorithm and AltMin used a subspace dimension of $r = 5$, with the data-sharing method used for initialization. Figure 18 presents difference images between the methods and the reference images to more clearly illustrate each algorithm's performance.

we considered zero-filled reconstructions, data sharing, and $2r$-AltMin. For data sharing, we identified the nearest data sample that contained the observed row and used it to impute the missing row for each frame. We do not include results for r-AltMin, as it was generally superseded by $2r$-AltMin. For both AltMin and our algorithm, we used the zero-filled method as the initial starting point for our algorithm. This is to investigate if we can improve upon the nearest-neighbors method by enforcing the low-rank structure. Both algorithms were run until the change in the NRMSE between two successive iterations were less than 10^{-6}, unless otherwise stated.

Figure 8 shows that the geodesic model significantly improved upon the nearest-neighbors method and achieved the lowest NRMSE scores. On the other hand, while $2r$-AltMin improved upon the initial point, there exist some time frames in which it performed worse. Appendix B.2 provides additional fMRI experiments, particularly on dynamic Shepp-Logan phantoms. Here, we expect AltMin to perform best since the Shepp-Logan phantoms have an *exact* (planted) low-rank structure. Nonetheless, our method performed closely to AltMin, highlighting the ability of our approach to reconstruct fMRI data.

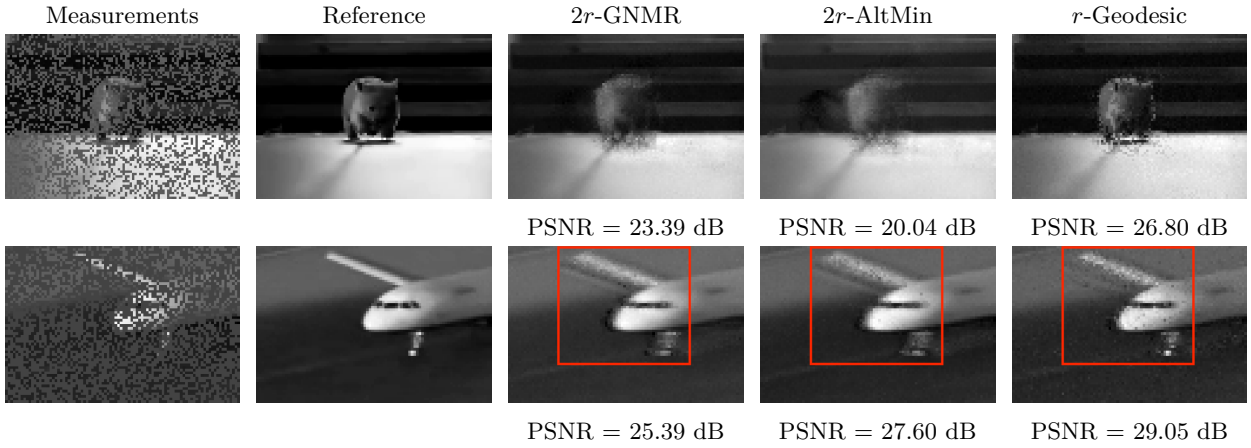

Figure 9: Visual results with corresponding PSNRs for a single frame of the mouse and plane video dataset with Gaussian noise with standard deviation $\sigma_{\mathbf{y}} = 0.01$. Top: Mouse frame with 50% missing pixels at random for $r = 5$. Bottom: Plane frame with 60% missing pixels at random for $r = 3$.

Video Data Experiments. We considered two video datasets: a video of a mouse moving towards a camera and the plane video from Section 5.1. The mouse video has a total of 90 frames, each with dimensions (70×100). This video was also used in earlier works (Nayer et al., 2020; Kwon et al., 2022). Our goal was to see if we can improve upon the globally low-rank methods by incorporating a temporal component. Since these videos are neither exactly low-rank nor lie on a geodesic, we do not expect perfect recovery.

For the mouse video dataset, we consider the noisy matrix completion setting, in which approximately 50% of the pixels in each frame are missing and corrupted with Gaussian noise of $\sigma_{\mathbf{y}} = 0.01$. For the plane video dataset, we observe only 40% of the data, also with $\sigma_{\mathbf{y}} = 0.01$. Alongside AltMin, we consider the baseline GNMR (Zilber & Nadler, 2022), a matrix recovery algorithm based on Gauss-Newton linearization. For GNMR, we also vectorize and stack all the frames to construct a single matrix. We picked the ranks to be $r = 3, 5$, where for our algorithm we set $\ell = r - 1$. This was to emphasize the fact that our algorithm can handle the case in which $\ell < r$. We evenly sample the timesteps $t_i \in [0.3, 0.7]$ and set $\lambda = 0.01$.

Figure 9 displays frames from both video data experiments. Intriguingly, we observe that our algorithm is less prone to the "blurring" effects commonly seen in models that use low-rank approximations. For example, in the bottom row of Figure 9, the plane appears less blurry with our algorithm, especially in the areas highlighted by the red box. We see similar results in the mouse frame, where the mouse is clearly more prominent using our algorithm. In Table 1, we present quantitative results across all frames for our algorithm and the baselines. Overall, our algorithm outperforms the others in terms of PSNR and RMSE across different ranks. However, in some trials, $2r$-AltMin slightly surpasses our algorithm in RMSE. We believe this is due to slightly more noise or "speckles" in the frames reconstructed by our algorithm. This highlights a tradeoff: when the dataset does not lie on a geodesic, our algorithm more accurately captures the object of interest but may introduce additional noise, since it estimates more parameters than a static model.

6 CONCLUSION AND FUTURE WORK

This work proposed an RBMM algorithm for estimating low-dimensional, time-varying data from undersampled measurements. Our algorithm constrains the time-varying subspaces to follow a geodesic on the Grassmannian. We demonstrated that our algorithm can effectively handle measurement noise and missing data compared to existing baselines in both synthetic and real settings. We also showed that our algorithm enjoys two convergence properties: a monotonically decreasing objective function and convergence to a stationary point in $\widetilde{\mathcal{O}}(\epsilon^{-2})$, where $\epsilon > 0$ is an error term representing the norm of the gradients for each block.

There are several exciting avenues for future work. While our experiments demonstrated that our model and algorithm are already useful and applicable to real data, it would be of interest to develop a continuous piecewise-geodesic model rather than the single geodesic considered here. This extension may improve applicability across a wider range of data types, and our work provides a key building block toward such a piecewise model. Furthermore, for fMRI data, the measurements are intrinsically tensor-valued, whereas our current approach vectorizes each image for reconstruction. Extending our algorithm to the tensor case could better exploit spatial structure for faster and more accurate reconstruction.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, USA, 2007. ISBN 0691132984.

Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k-pca: a global, gap-free, and near-optimal rate. In *IEEE 58th Annu. Symp. Foundations Comput. Sci. (FOCS)*, pp. 487–492. IEEE, 2017.

S. Babu, S. G. Lingala, and N. Vaswani. Fast low rank column-wise compressive sensing for accelerated dynamic MRI. *IEEE Trans. Computational Imaging*, 9:409–24, 2023. doi: 10.1109/TCI.2023.3263810.

Laura Balzano. On the equivalence of oja's algorithm and grouse. In *International Conference on Artificial Intelligence and Statistics*, pp. 7014–7030. PMLR, 2022.

Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 704–711, 2010. doi: 10.1109/ALLERTON.2010.5706976.

Laura Balzano, Yuejie Chi, and Yue Lu. Streaming PCA and subspace tracking: The missing data case. *Proc. of IEEE Special Issue on Rethinking PCA for Modern Datasets: Theory, Algorithms, and Applications*, 2018a.

Laura Balzano, Yuejie Chi, and Yue M. Lu. Streaming PCA and subspace tracking: The missing data case. *Proceedings of the IEEE*, 106(8):1293–1310, 2018b. doi: 10.1109/JPROC.2018.2847041.

Ronen Basri, Tal Hassner, and Lihi Zelnik-Manor. Approximate nearest subspace search with applications to pattern recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007. doi: 10.1109/CVPR.2007.383201.

Changxiao Cai, Gen Li, Yuejie Chi, H. Vincent Poor, and Yuxin Chen. Subspace estimation from unbalanced and incomplete data matrices: $\ell_{2,\infty}$ statistical guarantees. *The Annals of Statistics*, 49(2):944 – 967, 2021. doi: 10.1214/20-AOS1986. URL https://doi.org/10.1214/20-AOS1986.

Yuejie Chi, Yonina C. Eldar, and Robert Calderbank. Petrels: Subspace estimation and tracking from partial observations. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3301–3304, 2012. doi: 10.1109/ICASSP.2012.6288621.

Pierre Comon and Gene H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

Daniel R Fuhrmann. A geometric approach to subspace tracking. In *Asilomar Conf. Signals, Syst. and Comput.*, volume 1, pp. 783–787. IEEE, 1997.

A. K. Funai, J. A. Fessler, D. T. B. Yeo, V. T. Olafsson, and D. C. Noll. Regularized field map estimation in MRI. *IEEE Trans. Med. Imag.*, 27(10):1484–94, October 2008. doi: 10.1109/TMI.2008.923956.

S. Guo, J. A. Fessler, and D. C. Noll. Oscillating steady state imaging (OSSI) for fMRI using 3D sparse acquisition and model-based image reconstruction. In *BRAIN PI Meeting*, 2020.

Saeid Haghighatshoar and Giuseppe Caire. Low-complexity massive MIMO subspace estimation and tracking from low-dimensional projections. *IEEE Trans. Signal Process.*, 66(7):1832–1844, 2018.

N J Higham. Matrix nearness problems and applications. *APPLICATIONS OF MATRIX THEORY*, 22, 1989.

Yi Hong, Roland Kwitt, Nikhil Singh, Nuno Vasconcelos, and Marc Niethammer. Parametric regression on the grassmannian. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(11):2284–2297, 2016.

Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael Jordan, Anthony Joseph, and Nina Taft. In-network PCA and anomaly detection. In *Advances in Neural Information Processing Systems*, volume 19, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/2227d753dc18505031869d44673728e2-Paper.pdf.

Mia Hubert and Sanne Engelen. Robust pca and classification in biosciences. *Bioinformatics*, 20(11):1728–1736, 2004.

Jacob Hume and Laura Balzano. A spectral framework for tracking communities in evolving networks. In *The Third Learning on Graphs Conference*, 2024.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pp. 665–674, 2013. ISBN 9781450320290. doi: 10.1145/2488608.2488693. URL https://doi.org/10.1145/2488608.2488693.

Soo Min Kwon, Xin Li, and Anand D. Sarwate. Low-rank phase retrieval with structured tensor models. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3643–3647, 2022. doi: 10.1109/ICASSP43922.2022.9746452.

Kimmo Kärkkäinen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1547–1557, 2021. doi: 10.1109/WACV48630.2021.00159.

D.E. Lake and D.M. Keenan. Maximum likelihood estimation of geodesic subspace trajectories using approximate methods and stochastic optimization. In *Ninth IEEE Signal Processing Workshop on Statistical Signal and Array Processing (Cat. No.98TH8381)*, pp. 148–151, 1998a. doi: 10.1109/SSAP.1998.739356.

Douglas E Lake and DM Keenan. Maximum likelihood estimation of geodesic subspace trajectories using approximate methods and stochastic optimization. In *IEEE Signal Process. Workshop Statis. Signal and Array Process.*, pp. 148–151. IEEE, 1998b.

Yuchen Li, Laura Balzano, Deanna Needell, and Hanbaek Lyu. Convergence and complexity of block majorization-minimization for constrained block-riemannian optimization. *arXiv preprint arXiv:2312.10330*, 2023.

Hassan Mansour and Xin Jiang. A robust online subspace estimation and tracking algorithm. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4065–4069, 2015. doi: 10.1109/ICASSP.2015.7178735.

Marc Moonen, Paul Van Dooren, and Joos Vandewalle. A singular value decomposition updating algorithm for subspace tracking. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1015–1038, 1992. doi: 10.1137/0613061. URL https://doi.org/10.1137/0613061.

Praneeth Narayanamurthy and Namrata Vaswani. A fast and memory-efficient algorithm for robust pca (merop). In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4684–4688, 2018a. doi: 10.1109/ICASSP.2018.8461540.

Praneeth Narayanamurthy and Namrata Vaswani. Provable dynamic robust PCA or robust subspace tracking. *IEEE Trans. Inform. Theory*, 65(3):1547–1577, 2018b.

Praneeth Narayanamurthy and Namrata Vaswani. Provable dynamic robust pca or robust subspace tracking. *IEEE Transactions on Information Theory*, 65(3):1547–1577, 2019. doi: 10.1109/TIT.2018.2872023.

Seyedehsara Nayer, Praneeth Narayanamurthy, and Namrata Vaswani. Provable low rank phase retrieval. *IEEE Transactions on Information Theory*, 66(9):5875–5903, 2020. doi: 10.1109/TIT.2020.2984478.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, November 1982. ISSN 0303-6812. doi: 10.1007/BF00275687. URL http://dx.doi.org/10.1007/BF00275687.

Ricardo Otazo, Emmanuel Candès, and Daniel K Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 73(3):1125–1136, 2015.

Alex Saad-Falcon, Brighton Ancelin, and Justin Romberg. Subspace tracking with dynamical models on the grassmannian. In *2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 1–5. IEEE, 2024a.

Alex Saad-Falcon, Brighton Ancelin, and Justin Romberg. Subspace tracking with dynamical models on the grassmannian. *arXiv preprint arXiv:2402.10352*, 2024b.

L. A. Shepp and B. F. Logan. The fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, 1974. doi: 10.1109/TNS.1974.6499235.

Anuj Srivastava and Eric Klassen. Bayesian and geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, 2004.

Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2017. doi: 10.1109/TSP.2016.2601299.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021. URL http://jmlr.org/papers/v22/20-1067.html.

Richard J. Vaccaro. The role of subspace estimation in array signal processing. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1566–1572, 2019. doi: 10.1109/IEEECONF44664.2019.9048994.

Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE signal processing magazine*, 35(4):32–55, 2018.

Bin Yang. Projection approximation subspace tracking. *IEEE Trans. Signal process.*, 43(1):95–107, 1995.

Dejiao Zhang and Laura Balzano. Global convergence of a grassmannian gradient descent algorithm for subspace estimation. In *Artificial Intelligence and Statistics*, pp. 1460–1468. PMLR, 2016.

Jialun Zhang, Salar Fattahi, and Richard Y Zhang. Preconditioned gradient descent for over-parameterized nonconvex matrix factorization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5985–5996. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2f2cd5c753d3cee48e47dbb5bbaed331-Paper.pdf.

Lei Zhang, Weisheng Dong, David Dian Zhang, and Guangming Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43:1531–1549, 2010. URL https://api.semanticscholar.org/CorpusID:6212299.

Pini Zilber and Boaz Nadler. Gnmr: A provable one-line algorithm for low rank matrix recovery. *SIAM Journal on Mathematics of Data Science*, 4(2):909–934, 2022. doi: 10.1137/21M1433812. URL https://doi.org/10.1137/21M1433812.