

Scalable symmetric Tucker tensor decomposition

Ruhui Jin* Joe Kileel† Tamara G. Kolda‡ and Rachel Ward†

Abstract

We study the best low-rank Tucker decomposition of symmetric tensors. The motivating application is decomposing higher-order multivariate moments. Moment tensors have special structure and are important to various data science problems. We advocate for projected gradient descent (PGD) method and higher-order eigenvalue decomposition (HOEVD) approximation as computation schemes. Most importantly, we develop scalable adaptations of the basic PGD and HOEVD methods to decompose sample moment tensors. With the help of implicit and streaming techniques, we evade the overhead cost of building and storing the moment tensor. Such reductions make computing the Tucker decomposition realizable for large data instances in high dimensions. Numerical experiments demonstrate the efficiency of the algorithms and the applicability of moment tensor decompositions to real-world datasets. Finally we study the convergence on the Grassmannian manifold, and prove that the update sequence derived by the PGD solver achieves first- and second-order criticality.

1 Introduction

Tensor decompositions [36] provide powerful tools to represent and compress higher-order arrays. In this work, we focus on the *symmetric rank- r Tucker decomposition*. Applied to a d -th order symmetric tensor of size $n \times \dots \times n$, it expresses the input using a symmetric core tensor $\mathbf{c} \in \mathbb{R}^{r \times \dots \times r}$ ($r \ll n$) and a matrix $\mathbf{Q} \in \mathbb{R}^{n \times r}$ with orthonormal columns. See Figure 1 for an illustration.

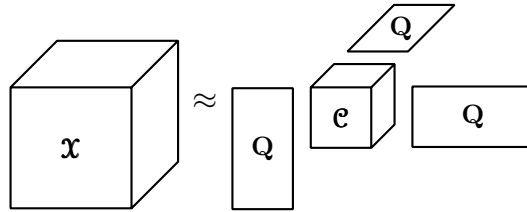


Figure 1: Symmetric Tucker decomposition.

Symmetric tensors abound in statistics as a means of representing higher-order moments and cumulants of a random vector. They are commonly used to analyze dependences in multivariate distributions [12, 30, 14], learn latent variable models via the method-of-moments [4, 24], solve important signal processing problems [9, 10], etc.

In most cases, the exact moment of the ground-truth distribution is not available. The idea of moment estimation is to use the d -th order *sample moment tensor*, denoted \mathcal{M}^d , instead. This is formed from observations $\{\mathbf{x}_1, \dots, \mathbf{x}_p\} \subseteq \mathbb{R}^n$ of a random variable \mathbf{x} as

$$\mathcal{M}^d = \frac{1}{p} \sum_{i=1}^p \mathbf{x}_i^{\otimes d} \approx \mathbb{E}(\mathbf{x}^{\otimes d}). \quad (1)$$

There is a need for economical representations of \mathcal{M}^d , which would otherwise take $\mathcal{O}(n^d)$ space to store. Using the symmetric Tucker decomposition, one approximates \mathcal{M}^d solely by determining the rank- r factor

*Department of Mathematics, University of Wisconsin-Madison, Madison, WI

†Department of Mathematics, University of Texas at Austin, Austin, TX

‡MathSci.ai, Dublin, CA

matrix \mathbf{Q} , which takes just rn memory space. (The core \mathbf{C} is a byproduct once \mathbf{Q} is identified; see [Remark 1](#)) However, decomposing the sample moment still has upfront costs of $\mathcal{O}(pn^d)$ and $\mathcal{O}(n^d)$ to initially construct and store \mathbf{M}^d . For large data volumes p and high dimensionalities n , we turn to *implicit* (or *tensor-free*) and *streaming* strategies to escape such prohibitive overhead.

The symmetric Tucker decomposition of moment tensors has other important motivations, besides compression. One major application is to extract the key features $\mathbf{Q} \in \mathbb{R}^{n \times r}$ from moment tensors. Without the need for the core tensor $\mathbf{C} \in \mathbb{S}^d(\mathbb{R}^r)$, this usage further reduces the output’s storage. In some scientific domains, data has high volatility, e.g., in combustion simulations [\[1\]](#), outlier detection [\[25, 55\]](#) and asset allocation [\[49, 8, 6\]](#). Higher-order moments and cumulants ($d \geq 3$) are sensitive to heavy tails and anomalies. Thus, finding the key subspace \mathbf{Q} of variability in higher-order statistics is important for non-Gaussian (also known as non-normal) data analysis. This is akin to the role of principal component analysis (PCA) and the covariance matrix for Gaussian data.

In other settings, data observations are assumed to lie in a low-dimensional subspace and follow the *low-rank factor model* [\[61\]](#). For example, this holds in the Fama-French model [\[22\]](#) of finance and also in psychometrics [\[48\]](#). The resulting moment tensors naturally admit a Tucker format. Thus the decomposition enforces the low Tucker rank structure and improves moment estimation accuracy.

1.1 Related works

Due to the NP-hard nature of tensor decompositions [\[29\]](#), an approximate symmetric Tucker decomposition is computed by higher-order eigenvector decomposition (HOEVD) [\[15\]](#). It is the eigendecomposition of the flattened tensor. It can be viewed as the symmetric version of higher-order singular value decomposition (HOSVD) [\[14\]](#) and efficient variants [\[64, 43, 62, 45, 11, 2\]](#). Despite their popularity, HOEVD offers no guarantee of returning the exact critical points for the problem.

We instead turn to iterative search methods to achieve the best possible solution, using HOEVD as an initialization. Newton [\[21, 33\]](#), quasi-Newton [\[57\]](#) and trust-region [\[32\]](#) methods have been considered for general Tucker decomposition. These second-order optimizations rely on Riemannian Hessian updates on a the manifold of orthonormal matrices. For symmetric tensors, higher-order orthogonal iterations (HOOI) [\[16\]](#), Jacobi rotations [\[31, 41\]](#) and power method [\[54\]](#) were proposed. Regalia briefly mentions a shifted variant of the power method in [\[54\]](#), which is akin to a gradient descent method, although strong convergence analysis is omitted in the work.

Our work is primarily motivated by moment tensor decompositions, which are fundamental objects in statistics and other disciplines. Neither the existing HOEVD nor iterative algorithms for Symmetric Tucker take advantage of the sample moment tensor structure [\(1\)](#). We show that streaming and implicit techniques can be applied to address the computational burden in the data-rich and high-dimensional regime. For covariance matrices which are centered second-order moments, this is relevant to the streaming PCA scheme, also known as Oja’s method [\[51\]](#), and its adaptations [\[46, 67, 28\]](#).

Recently, a similar strategy was applied to optimization with moment tensor CP decomposition for learning Gaussian mixture models, see [\[59, 52\]](#). Building on techniques from [\[59\]](#), the present work also passes over only a subset of samples at a time and drives the optimization without forming gigantic moment tensors. Despite using similar approaches for increasing efficiency, we emphasize that CP and Tucker decompositions are useful for their own independent purposes. The motivations we list for feature extraction and data factor analysis specifically apply to Tucker decompositions. We refer readers to [Section 4.5](#) for a concrete comparison.

1.2 Our contributions

In this paper, our major contribution is developing scalable algorithms for symmetric Tucker decomposition on moment tensors.

To start with, we propose the projected gradient descent (PGD) framework to the general symmetric Tucker tensor decomposition problem, see [Algorithm 1](#). The approach is led by a first-order gradient update governed by the Tucker approximation cost. It is free of non-trivial second-order updates on manifolds.

Driven by their importance in data science, our attention then focuses on decomposing sample moment tensors [\(1\)](#). By exploiting the special input structure, we develop scalable PGD (SPGD) in [Algorithm 2](#).

There are two main highlights. First, we adopt the implicit technique and operate on the input sample vectors directly. This is much cheaper in memory and computation than constructing the sample moment tensor. Proposed in [59, 52], the implicit strategy greatly reduces the effort for gradient computation. Second, SPGD is built on a streaming model. By passing over small batches of data observations, both storage and computational complexity are further reduced. Similar ideas have been used in CP moment tensor decomposition [59] for separate purposes.

We also design a scalable implementation of HOEVD (SHOEVD) for sample moment tensors (I), see Algorithm 3. This method is akin to the streaming PCA idea. We use the SHOEVD approximation (Algorithm 3) as the initialization to SPGD (Algorithm 2). Moreover, SHOEVD may be of independent interest due to its broad usages, e.g. [13, 3]. The savings from the streaming and implicit strategies allow SHOEVD and SPGD to handle large-scale computations. The gains are summarized in Table 1. Here p is the total sample number and b is the batch size at each iteration. Usually, the user sets $b \ll p$. These efficiency advantages are corroborated by extensive numerical experiments.

We demonstrate the key motivations of sample moment decompositions, including feature extraction and moment tensor low-rank estimation. Our proposed SPGD is respectively applied to real datasets in anomaly detection and portfolio allocation.

In terms of theoretical guarantee, the basic PGD (Algorithm 1) convergence result is provided. Employing the theories from manifold optimization and dynamical systems, we prove that for any symmetric tensor, PGD converges monotonically to a first-order critical point given arbitrary initialization. Furthermore, it converges to a second-order critical point almost surely.

Table 1: Computation and storage of SPGD (Algorithm 2) and SHOEVD (Algorithm 3). Here p is the number of samples, n is the vector size, d is the moment tensor order, and b is the batch size.

	overhead	storage per iter.
streaming & implicit	none	$nr + nb$
full & implicit	none	$nr + np$
full & explicit	$n^d p$	$nr + n^d$

2 Symmetric Tucker tensor decomposition

2.1 Notation and background

Scalars, vectors, matrices and tensors are represented as lowercase letters x , lowercase bold letters \mathbf{x} , uppercase bold letters \mathbf{X} , and calligraphic bold letters \mathfrak{X} , respectively. The (i_1, \dots, i_d) -th element of a d -way tensor is denoted by x_{i_1, \dots, i_d} . The j -th column of a matrix \mathbf{X} is \mathbf{x}_j . The spectral norm of a matrix is $\|\cdot\|_2$. The norm of a tensor is $\|\cdot\|$, see Definition 5. The identity matrix is $\mathbf{I}_n \in \mathbb{R}^{n \times n}$. The Moore-Penrose pseudoinverse of a matrix is denoted using \dagger . The matrix symmetrization operator is written $\text{sym} : \mathbb{R}^{n \times n} \rightarrow \mathbb{S}^2(\mathbb{R}^n)$ and given by $\text{sym}(\mathbf{X}) = \frac{1}{2}(\mathbf{X} + \mathbf{X}^\top)$ for $\mathbf{X} \in \mathbb{R}^{n \times n}$. We write orth for the function that selects the orthonormal factor $\mathbf{Q} \in \mathbb{R}^{n \times r}$ from the QR decomposition of a full-rank $n \times r$ matrix.¹ The notation $\mathbf{X}^{[d]}$ is the elementwise exponentiation, where each coordinate of \mathbf{X} is raised to the d -th power. The set $\{1, \dots, n\}$ is denoted by $[n]$ when n is a positive integer.

Definition 1 (symmetric tensors). *A tensor $\mathfrak{X} \in \mathbb{R}^{n \times \dots \times n}$ is symmetric if its entries are invariant under any permutation of indices, i.e.*

$$x_{i_1, \dots, i_d} = x_{i_{\sigma(1)}, \dots, i_{\sigma(d)}}, \quad \forall (i_1, \dots, i_d) \in [n]^d \text{ and } \sigma \in \Pi(d)$$

where $\Pi(d)$ is the permutation group on $[d]$. Here, d and n are called the order and dimension of \mathfrak{X} . The space of all real-valued symmetric tensors of order d and dimension n is denoted by $\mathbb{S}^d(\mathbb{R}^n)$.

¹For ease of analysis, we ensure the uniqueness of the QR factorization by using the one where all diagonal entries of the \mathbf{R} factor are positive. But none of our algorithms require this choice.

Definition 2 (symmetric outer product). The outer product of a vector $\mathbf{x} \in \mathbb{R}^n$ with itself d times is denoted as $\mathbf{x}^{\otimes d} \in \mathbb{S}^d(\mathbb{R}^n)$. The (i_1, \dots, i_d) -th entry of this product is

$$(\mathbf{x}^{\otimes d})_{i_1, \dots, i_d} = x_{i_1} \cdots x_{i_d}, \quad \forall (i_1, \dots, i_d) \in [n]^d.$$

Definition 3 (symmetric Tucker product). For a symmetric tensor $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ and a matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$, their symmetric Tucker product is defined as \mathbf{X} with \mathbf{Y} multiplied along each mode, written $\mathbf{X} \cdot (\mathbf{Y}, \dots, \mathbf{Y}) \in \mathbb{S}^d(\mathbb{R}^m)$. The (j_1, \dots, j_d) -th entry of the product is

$$(\mathbf{X} \cdot (\mathbf{Y}, \dots, \mathbf{Y}))_{j_1, \dots, j_d} = \sum_{i_1=1}^n \cdots \sum_{i_d=1}^n x_{i_1, \dots, i_d} y_{i_1, j_1} \cdots y_{i_d, j_d}, \quad \forall (j_1, \dots, j_d) \in [m]^d.$$

Definition 4 (symmetric tensor matricization). A symmetric tensor $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ can be flattened into a matrix, denoted $\text{mat}(\mathbf{X}) \in \mathbb{R}^{n \times n^{d-1}}$. The index $(i_1, \dots, i_d) \in [n]^d$ of \mathbf{X} is mapped to the index $(i_1, 1 + \sum_{\ell=2}^d (i_\ell - 1)n^{\ell-2}) \in [n] \times [n^{d-1}]$ of $\text{mat}(\mathbf{X})$.

Definition 5 (inner product). For (not necessarily symmetric) tensors $\mathbf{X}, \mathbf{y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ of the same size, their inner product is

$$\langle \mathbf{X}, \mathbf{y} \rangle = \sum_{i_d=1}^{n_d} \cdots \sum_{i_1=1}^{n_1} x_{i_1, \dots, i_d} y_{i_1, \dots, i_d} \in \mathbb{R}.$$

The norm of a tensor is $\|\mathbf{X}\| = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}$.

Definition 6. For tensors $\mathbf{X} \in \mathbb{R}^{m_1 \times n_2 \times \cdots \times n_d}$ and $\mathbf{y} \in \mathbb{R}^{m'_1 \times n_2 \times \cdots \times n_d}$, their inner product in all modes except the first one [21, 17] is defined as:

$$\mathbf{Z} = \langle \mathbf{X}, \mathbf{y} \rangle_{-1} \in \mathbb{R}^{m_1 \times m'_1}$$

$$\text{where } z_{j_1, j'_1} = \sum_{i_d=1}^{n_d} \cdots \sum_{i_2=1}^{n_2} x_{j_1, i_2, \dots, i_d} y_{j'_1, i_2, \dots, i_d} \quad \forall j_1 \in [m_1], j'_1 \in [m'_1].$$

2.2 Problem statement

Let $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ be a symmetric tensor. Fix a user-specified rank r (typically $r \ll n$). Our goal is to find the best approximation of \mathbf{X} represented by the symmetric Tucker product of a core tensor $\mathbf{C} \in \mathbb{S}^d(\mathbb{R}^r)$ and a rank- r orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n \times r}$. The problem is formulated as follows:

$$\min_{\substack{\mathbf{C} \in \mathbb{S}^d(\mathbb{R}^r) \\ \mathbf{Q} \in \mathbb{R}^{n \times r}}} \|\mathbf{X} - \mathbf{C} \cdot (\mathbf{Q}^\top, \dots, \mathbf{Q}^\top)\|^2, \quad \text{subject to } \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_r. \quad (2)$$

In fact, the minimization (2) is equivalent to solving for a rank- r basis \mathbf{Q} that maximizes the following cost function:

$$\max_{\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_r} F(\mathbf{Q}) \equiv \|\mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q})\|^2. \quad (*)$$

Remark 1. The problem formulation transforms from (2) to (*) by solving the unconstrained linear least squares in (2) for the core, i.e.,

$$\mathbf{C} = \mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q}). \quad (3)$$

Substituting (3) into (2) gives

$$\begin{aligned} \|\mathbf{X} - \mathbf{C} \cdot (\mathbf{Q}^\top, \dots, \mathbf{Q}^\top)\|^2 &= \|\mathbf{X} - \mathbf{X} \cdot (\mathbf{Q}\mathbf{Q}^\top, \dots, \mathbf{Q}\mathbf{Q}^\top)\|^2 \\ &= \|\mathbf{X}\|^2 - \|\mathbf{X} \cdot (\mathbf{Q}\mathbf{Q}^\top, \dots, \mathbf{Q}\mathbf{Q}^\top)\|^2 \\ &= \|\mathbf{X}\|^2 - \|\mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q})\|^2. \end{aligned}$$

As $\|\mathbf{X}\|^2$ is a constant in \mathbf{Q} , minimizing the above quantity is indeed the same as maximizing $\|\mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q})\|^2$.

Remark 2. The full Tucker decomposition (2) takes $\mathcal{O}(nr + r^d)$ in memory to store the basis $\mathbf{Q} \in \mathbb{R}^{n \times r}$ and the core $\mathbf{C} \in \mathbb{S}^d(\mathbb{R}^r)$. Though still exponential in d , this cost savings can be substantial compared to the original $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ requiring $\mathcal{O}(n^d)$ space if the rank r is much less than the ambient dimension n . Further, in applications such as tensor PCA [47] and anomaly detection Section 5.1, one only desires the feature subspace \mathbf{Q} , and the core \mathbf{C} need not be computed or stored. This reduces the cost to $\mathcal{O}(nr)$.

The feasible domain in (8) for \mathbf{Q} is the Stiefel manifold:

$$\text{St}(n, r) := \{ \mathbf{Q} \in \mathbb{R}^{n \times r} \mid \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_r \}, \quad (4)$$

which is a compact Riemannian submanifold of $\mathbb{R}^{n \times r}$ with dimension $\frac{r(2n-r-1)}{2}$. Further, the cost function F is rotationally invariant. In fact, it only depends on the orthogonal projector $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top \in \mathbb{S}^2(\mathbb{R}^n)$, since $\|\mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q})\| = \|\mathbf{X} \cdot (\mathbf{P}, \dots, \mathbf{P})\|$. So, the optimization (8) naturally runs on the Grassmannian manifold, which consists of all orthogonal projectors \mathbf{P} :

$$\text{Gr}(n, r) := \{ \mathbf{P} \in \mathbb{S}^2(\mathbb{R}^n) \mid \mathbf{P}^2 = \mathbf{P}, \text{rank}(\mathbf{P}) = r \}. \quad (5)$$

Here, $\text{Gr}(n, r)$ is a compact Riemannian manifold of $\mathbb{S}^2(\mathbb{R}^n)$ with dimension $r(n-r)$.

Remark 3. No simple characterization of the critical points of (8) is known when $d \geq 3$. This is in contrast to case of $d = 2$, where the classical Eckart-Young theorem [20] holds. Various possible extensions of Eckart-Young to tensors have been shown to fail, e.g., see [35].

2.3 PGD algorithm

This paper develops a projected gradient descent (PGD) method to solve the constrained optimization problem (8). The PGD framework works as follows: given an initial point, we move the current iterate \mathbf{Q}_{t-1} along the negative gradient direction by a step size of $\gamma_t > 0$ where the gradient is:

$$\nabla F(\mathbf{Q}_{t-1}) = 2d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{Q}_{t-1}, \dots, \mathbf{Q}_{t-1}), \mathbf{X} \cdot (\mathbf{Q}_{t-1}, \dots, \mathbf{Q}_{t-1}) \rangle_{-1} \in \mathbb{R}^{n \times r}. \quad (6)$$

Then we project the update back to the Stiefel manifold (4) by selecting the \mathbf{Q} factor from the QR decomposition. The procedure is described in Algorithm 1. See Section 6 for a convergence analysis of PGD.

Algorithm 1 PGD for symmetric Tucker decomposition

Input: symmetric tensor $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$
number of iteration T , step sizes $\{\gamma_t\} \subseteq \mathbb{R}$
initial guess $\mathbf{Q}_0 \in \text{St}(n, r)$ ▷ initialization
Output: rank- r orthonormal basis $\mathbf{Q} \in \text{St}(n, r)$

- 1: **for** $t = 1, \dots, T$ **do**
- 2: $\mathbf{Q}_t \leftarrow \mathbf{Q}_{t-1} + 2d\gamma_t \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{Q}_{t-1}, \dots, \mathbf{Q}_{t-1}), \mathbf{X} \cdot (\mathbf{Q}_{t-1}, \dots, \mathbf{Q}_{t-1}) \rangle_{-1}$
- 3: ▷ gradient update
- 4: $\mathbf{Q}_t \leftarrow \text{qr}(\mathbf{Q}_t, 0)$ ▷ retraction
- 5: **end for**
- 6: **return** $\mathbf{Q} \leftarrow \mathbf{Q}_T$

2.4 HOEVD initialization

An approximate solution to the best symmetric Tucker decomposition (8) is the HOEVD [15] result:

$$\mathbf{Q}_{\text{hoevd}} \in \text{St}(n, r) \text{ given by the leading } r \text{ eigenvectors of } \text{mat}(\mathbf{X}) \text{mat}(\mathbf{X})^\top. \quad (7)$$

HOEVD is widely used for Tucker approximation, because it can be computed via a direct matrix eigendecomposition.

Generally speaking, the HOEVD solution is not a critical point of [\(8\)](#) [\[15\]](#). Its approximation error to a d -th order tensor \mathbf{X} is bounded [\[26\]](#), Theorems 6.9-10] as

$$\|\mathbf{X} - \hat{\mathbf{X}}_{\text{hoevd}}\| \leq \sqrt{d} \|\mathbf{X} - \hat{\mathbf{X}}_{\text{best}}\|.$$

Here, $\hat{\mathbf{X}}_{\text{hoevd}} = \mathbf{X} \cdot (\mathbf{Q}_{\text{hoevd}} \mathbf{Q}_{\text{hoevd}}^\top, \dots, \mathbf{Q}_{\text{hoevd}} \mathbf{Q}_{\text{hoevd}}^\top)$ as in [Remark 1](#) and $\hat{\mathbf{X}}_{\text{best}}$ is the optimum of [\(2\)](#). We observe the suboptimality of the HOEVD solution in [Section 4.4](#). However according to [\[14\]](#), section 3.4], the HOEVD solution usually belongs to an ‘‘attractive’’ region around the desired local optima in numerical experiments. So it is a good candidate for initialization.

3 Scalable sample moment tensor decomposition

The main interest in this work is the decomposition of the sample moment tensor $\mathcal{M}^d \in \mathbb{S}^d(\mathbb{R}^n)$ [\(1\)](#). We design scalable algorithms for the symmetric Tucker decomposition [\(5\)](#), and the HOEVD approximation [\(7\)](#), of \mathcal{M}^d . An average of symmetric outer products, the sample moment structure [\(1\)](#) is the key to the scalability of the decomposition computation. Tensor operations with outer products can be reduced to much cheaper computations involving only their factor vectors. Specifically we use the following identities: for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^m$ and a matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$, it holds [\[26\]](#)

$$\mathbf{x}^{\otimes d} \cdot (\mathbf{Y}, \dots, \mathbf{Y}) = (\mathbf{Y}^\top \mathbf{x})^{\otimes d} \in \mathbb{S}^d(\mathbb{R}^m), \quad (8)$$

$$\langle \mathbf{x}^{\otimes d}, \mathbf{y}^{\otimes d} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle^d \in \mathbb{R}, \quad \text{and} \quad (9)$$

$$\langle \mathbf{z} \otimes \mathbf{x}^{\otimes(d-1)}, \mathbf{y}^{\otimes d} \rangle_{-1} = \mathbf{z} \langle \mathbf{x}, \mathbf{y} \rangle^{d-1} \mathbf{y}^\top \in \mathbb{R}^{m \times n}. \quad (10)$$

3.1 Scalable PGD algorithm

In [Algorithm 2](#), we introduce the scalable PGD (SPGD) when the input tensor is a sample moment [\(1\)](#). The symmetric Tucker cost [\(8\)](#) now depends solely on the data observations $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$. Indeed due to [\(8\)](#),

$$F(\mathbf{Q}) := \left\| \left(\frac{1}{p} \sum_{i=1}^p \mathbf{x}_i^{\otimes d} \right) \cdot (\mathbf{Q}, \dots, \mathbf{Q}) \right\|^2 = \left\| \frac{1}{p} \sum_{i=1}^p (\mathbf{Q}^\top \mathbf{x}_i)^{\otimes d} \right\|^2. \quad (11)$$

We actually do not need to create or store the sample moment \mathcal{M}^d . Such an idea was called the implicit technique in [\[59, 52\]](#).

Commonly, there is a huge amount of data in exceedingly high dimensions. We hence further turn to the streaming strategy to speed up the computations. Here data is revealed as a sequence of sample batches $\{\mathbf{X}_1, \dots, \mathbf{X}_t, \dots\} \subseteq \mathbb{R}^{n \times b}$ with batch size b ($b \ll p$). We read each batch as it arrives but discard it afterwards. This streaming model is called the turnstile model [\[50, 42, 63\]](#).

Therefore, as the t -th ($t \geq 1$) batch $\mathbf{X}_t = [\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,b}]^\top \in \mathbb{R}^{n \times b}$ arrives, the cost function [\(11\)](#) is recast stochastically:

$$F_t(\mathbf{Q}) := \left\| \frac{1}{b} \sum_{i=1}^b (\mathbf{Q}^\top \mathbf{x}_{t,i})^{\otimes d} \right\|^2. \quad (12)$$

In light of [\(10\)](#), we can compute the stochastic gradient in a tensor-free manner:

$$\begin{aligned} \nabla F_t(\mathbf{Q}) &= 2d \left\langle \frac{1}{b} \sum_{i=1}^b \mathbf{x}_{t,i} \otimes (\mathbf{Q}^\top \mathbf{x}_{t,i})^{\otimes(d-1)}, \frac{1}{b} \sum_{j=1}^b (\mathbf{Q}^\top \mathbf{x}_{t,j})^{\otimes d} \right\rangle_{-1} \\ &= \frac{2d}{b^2} \sum_{i=1}^b \sum_{j=1}^b \mathbf{x}_{t,i} \langle \mathbf{Q}^\top \mathbf{x}_{t,i}, \mathbf{Q}^\top \mathbf{x}_{t,j} \rangle^{d-1} \mathbf{x}_{t,j}^\top \mathbf{Q} \\ &= \frac{2d}{b^2} \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{X}_t)^{[d-1]} \mathbf{X}_t^\top \mathbf{Q} \in \mathbb{R}^{n \times r}. \end{aligned} \quad (13)$$

Here, $(\cdot)^{[d-1]}$ means raising each element of the input matrix to the $(d-1)$ -th power. Computing the gradient implicitly via online samples reduces the cost to $\mathcal{O}(rnb + rb^2)$ flops per iteration. This improves upon $\mathcal{O}(rnp + rp^2)$ using the full amount of samples, and a cost that is exponential in d if we operate on the explicit tensor in $\mathbb{S}^d(\mathbb{R}^n)$.

Moreover, a scalable HOEVD solver, SHOEVVD in line 1 of [Algorithm 2](#), is utilized for good initialization; see [Section 3.2](#) for details. We also use AdaGrad [\[19\]](#) to automatically tune step sizes. We implement the column-wise extension of AdaGrad proposed in [\[28\]](#), since it is natural to independently update the orthonormal basis vectors in \mathbf{Q} . The step size $(\boldsymbol{\gamma})_t$ becomes a row vector in $\mathbb{R}^{1 \times r}$, rather than a scalar. See lines 4 and 5, where $\cdot/$ denotes columnwise division.

SPGD is a two-phased iterative algorithm. It consists of

- Phase I - SHOEVVD approximation (line 1),
- Phase II - symmetric Tucker decomposition (lines 2-8).

Algorithm 2 SPGD for sample moment Tucker decomposition

Input: numbers of iterations T_1, T_2 ; batch sizes b_1, b_2 ;
 datastreams $\{\mathbf{X}_1, \dots, \mathbf{X}_{T_1}\} \in \mathbb{R}^{n \times b_1}$ (Phase I), $\{\mathbf{X}_{T_1+1}, \dots, \mathbf{X}_{T_2}\} \in \mathbb{R}^{n \times b_2}$ (Phase II);
 hyperparameters: $(\boldsymbol{\gamma}_1)_0 = (\boldsymbol{\gamma}_2)_0 = 10^{-5} \text{ones}(1, r)$; $c_1, c_2 > 0$

Output: rank- r orthonormal basis $\mathbf{Q} \in \text{St}(n, r)$

- 1: $\mathbf{Q}_{T_1} \leftarrow \text{SHOEVVD}(T_1, b_1, \{\mathbf{X}_1, \dots, \mathbf{X}_{T_1}\}, (\boldsymbol{\gamma}_1)_0, c_1)$ ▷ initialization
 - 2: **for** $t = T_1 + 1, \dots, T_2$ **do**
 - 3: $\mathbf{G}_t \leftarrow \frac{2d}{b_2} \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{Q}_{t-1} \mathbf{Q}_{t-1}^\top \mathbf{X}_t)^{[d-1]} \mathbf{X}_t^\top \mathbf{Q}_{t-1}$ ▷ gradient computation
 - 4: $(\boldsymbol{\gamma}_2)_t \leftarrow \left((\boldsymbol{\gamma}_2)_{t-1}^{[2]} + \text{colnorm}(\mathbf{G}_t)^{[2]} \right)^{[1/2]}$ ▷ step size
 - 5: $\mathbf{Q}_t \leftarrow \mathbf{Q}_{t-1} + c_2 \mathbf{G}_t \cdot / (\boldsymbol{\gamma}_2)_t$
 - 6: $\mathbf{Q}_t \leftarrow \text{qr}(\mathbf{Q}_t, 0)$
 - 7: **end for**
 - 8: **return** $\mathbf{Q} \leftarrow \mathbf{Q}_{T_2}$
-

3.2 Scalable HOEVD

We present the scalable HOEVD (SHOEVVD) method in [Algorithm 3](#). This serves as the Phase I initialization in SPGD ([Algorithm 2](#)), but it may be of independent interest. As HOEVD is essentially a matrix eigendecomposition, we follow the streaming PCA idea. The solver uses the PGD framework, but is applied to a different cost function than [\(11\)](#). Specifically, we solve HOEVD iteratively finding $\mathbf{Q} \in \text{St}(n, r)$ that maximizes the following stochastic cost formed by a data batch $\mathbf{X}_t \in \mathbb{R}^{n \times b_1}$:

$$\left\| \text{mat} \left(\frac{1}{b_1} \sum_{i=1}^{b_1} \mathbf{x}_{t,i}^{\otimes d} \right)^\top \mathbf{Q} \right\|^2.$$

Using [\(10\)](#), the gradient can be calculated implicitly:

$$\begin{aligned} 2 \text{mat} \left(\frac{1}{b_1} \sum_{i=1}^{b_1} \mathbf{x}_{t,i}^{\otimes d} \right) \text{mat} \left(\frac{1}{b_1} \sum_{j=1}^{b_1} \mathbf{x}_{t,j}^{\otimes d} \right)^\top \mathbf{Q} &= \frac{2}{b_1^2} \sum_{i=1}^{b_1} \sum_{j=1}^{b_1} \mathbf{x}_{t,i} \langle \mathbf{x}_{t,i}, \mathbf{x}_{t,j} \rangle^{d-1} \mathbf{x}_{t,j}^\top \mathbf{Q} \\ &= \frac{2}{b_1^2} \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{X}_t)^{[d-1]} \mathbf{X}_t^\top \mathbf{Q} \in \mathbb{R}^{n \times r}. \end{aligned} \tag{14}$$

It costs $\mathcal{O}(rnb_1 + nb_1^2 + rb_1^2)$ flops to compute the gradient.

Remark 4. SPGD and SHOEVVD ([Algorithms 2](#) and [3](#)) obviate the upfront cost to form the tensor, which would require $\mathcal{O}(pn^d)$ flops. Each iteration temporarily stores an updated matrix $\mathbf{Q}_t \in \mathbb{R}^{n \times r}$ and an online sample batch $\mathbf{X}_t \in \mathbb{R}^{n \times b}$, taking only $nr + nb$ memory. This explains the complexity counts shown in [Table 1](#).

Algorithm 3 SHOEVD for sample moment decomposition

Input: number of iterations T_1 ; batch size b_1 ;data stream $\{\mathbf{X}_1, \dots, \mathbf{X}_{T_1}\} \subseteq \mathbb{R}^{n \times b_1}$;hyperparameters: $(\gamma_1)_0 = 10^{-5} \text{ones}(1, r)$; $c_1 > 0$ **Output:** top- r eigenvectors $\mathbf{Q}_{\text{hoevd}} \in \text{St}(n, r)$

```

1: function S-HOEVD( $T_1, \{\mathbf{X}_1, \dots, \mathbf{X}_{T_1}\}, (\gamma_1)_0, c_1$ )
2:  $\mathbf{Q}_0 \leftarrow [\mathcal{N}(0, 1)]^{n \times r}$ 
3:  $\mathbf{Q}_0 \leftarrow \text{qr}(\mathbf{Q}_0, 0)$  ▷ initialization
4:   for  $t = 1, \dots, T_1$  do
5:      $\mathbf{G}_t \leftarrow \frac{2}{b_1^2} \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{X}_t)^{[d-1]} \mathbf{X}_t^\top \mathbf{Q}_{t-1}$ 
6:      $(\gamma_1)_t \leftarrow \left( (\gamma_1)_t^{[2]} + \text{colnorm}(\mathbf{G}_t)^{[2]} \right)^{[1/2]}$ 
7:      $\mathbf{Q}_t \leftarrow \mathbf{Q}_{t-1} + c_1 \mathbf{G}_t ./ (\gamma_1)_t$  ▷ gradient update
8:      $\mathbf{Q}_t \leftarrow \text{qr}(\mathbf{Q}_t, 0)$  ▷ retraction
9:   end for
10: return  $\mathbf{Q}_{\text{hoevd}} \leftarrow \mathbf{Q}_{T_1}$ 
11: end function

```

Remark 5. The core tensor $\mathbf{c} \in \mathbb{S}^d(\mathbb{R}^r)$ (3) can be evaluated as follows. After obtaining $\mathbf{Q} \in \text{St}(n, r)$ by SPGD or SHOEVD (Algorithms 2 and 3), we draw fresh data $\mathbf{X}_c = [(\mathbf{x}_c)_1, \dots, (\mathbf{x}_c)_{p_c}]^\top \in \mathbb{R}^{n \times p_c}$ and then set:

$$\mathbf{c} = \frac{1}{p_c} \sum_{i=1}^{p_c} (\mathbf{Q}^\top (\mathbf{x}_c)_i)^{\otimes d}.$$

The core computation costs $\mathcal{O}(r^d)$ in storage and $\mathcal{O}(p_c r n + p_c r^d)$ in flops.

4 Synthetic numerical tests

In this section, we utilize synthetic tests to study the proposed algorithms for sample moment tensor decomposition. And real-world datasets are considered in Section 5. In this section, we show:

- the scalability of implicit compared to explicit methods in Figure 2
- the speedup of streaming compared to full data usage in Figure 3
- the robustness of the HOEVD approximation and the symmetric Tucker decomposition to noise and target rank choice in Figures 4 and 5, and
- a comparison between symmetric Tucker and CP in Table 2

4.1 Set-up

In these tests, we generate moment tensors with approximate low-rank symmetric Tucker structure (3). One way is to assume data instances $\mathbf{x} \in \mathbb{R}^n$ follow a linear factor model:

$$\mathbf{x} = \mathbf{B}\mathbf{f} + \boldsymbol{\varepsilon}. \quad (15)$$

Here we fix the factor loading matrix $\mathbf{B} \in \mathbb{R}^{n \times r_{\text{true}}}$. The latent factor $\mathbf{f} \in \mathbb{R}^{r_{\text{true}}}$ is drawn from a coordinate-independent, zero-mean and unit variance normal-inverse Gaussian (NIG) distribution [5, 6]. This distribution is skewed and heavy-tailed, aligning with some motivations discussed in Section 1. The noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is Gaussian, following $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.

Note that in (15), the d -th moment of \mathbf{x} roughly admits a low-rank Tucker format:

$$\begin{aligned} \mathbf{M}^d &\approx \mathbb{E}(\mathbf{x}^{\otimes d}) \approx \mathbb{E}((\mathbf{B}\mathbf{f})^{\otimes d}) \\ &= \mathbb{E}((\mathbf{R}^* \mathbf{f})^{\otimes d}) \cdot (\mathbf{Q}^{*\top}, \dots, \mathbf{Q}^{*\top}). \end{aligned}$$

The matrices $\mathbf{Q}^* \in \mathbb{R}^{n \times r_{\text{true}}}$, $\mathbf{R}^* \in \mathbb{R}^{r_{\text{true}} \times r_{\text{true}}}$ are from the QR factorization of \mathbf{B} .

We define the inverse signal-to-noise ratio

$$\text{SNR}^{-1} := \frac{\|\mathbb{E}(\boldsymbol{\epsilon})\|}{\|\mathbb{E}(\mathbf{B}\mathbf{f})\|} = \frac{\sigma\sqrt{n}}{\|\mathbf{B}\|}. \quad (16)$$

The above quantity (rather than the deviation σ) is to measure the model’s noise level.

Remark 6. *Our goal is to simply perform the symmetric Tucker decomposition of the sample moment tensor $\mathcal{M}^d \in \mathbb{S}^d(\mathbb{R}^n)$. This is different from inference procedures on the linear factor model, e.g., independent component analysis (ICA) [14].*

Remark 7. *Besides the model (15) or other heavy-tailed datasets, we emphasize that our proposed algorithms are able to efficiently decompose \mathcal{M}^d when the sample vectors come from any probabilistic model.*

All numerical tests are implemented using MATLAB R2022a on a MacBook Pro with 16GB RAM memory. Each reported plot is averaged over 5 independent simulations with independent, random initializations. The matrix \mathbf{B} in (15) is randomly generated with i.i.d. standard normal entries and then fixed for all runs. We generate new \mathbf{f} and $\boldsymbol{\epsilon}$ to form the data samples in each stochastic simulation. The parameters of each experiment, namely n , r_{true} , r , p , d , SNR^{-1} , are specified below.

4.2 Implicit versus explicit algorithms

We illustrate the scalability of implicit and explicit implementations in Figure 2. The experiments are *non-streaming*. The implicit methods run on a fixed set of observations $\mathbf{X} \in \mathbb{R}^{n \times p}$. The explicit approaches require constructing the sample moment tensor $\mathcal{M}^d \in \mathbb{S}^d(\mathbb{R}^n)$ as overhead, and then operating on it. Due to resource limitations, the maximal tensor dimension n for explicit methods is smaller than maximal dimension for the implicit methods. We set the sample size to be $p = 5n$ and choose the rank as $r = r_{\text{true}} = 5$. We consider non-streaming methods, so $b_1 = b_2 = p$. The sample moment formation uses the `symktensor` package in the Tensor Toolbox for MATLAB [37]. The noise level is $\text{SNR}^{-1} = 0.5$, see (16).

We conduct the experiments in two phases for order-3 and 4 sample moments \mathcal{M}^d . For Phase I (initialization), we compare the efficiency of the implicit HOEVD result,

$$\mathbf{Q}_{\text{hoevd}} \in \text{St}(n, r) \text{ given by the leading } r \text{ eigenvectors of } \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{[d-1]} \mathbf{X}^\top / p^2, \quad (17)$$

and its explicit counterpart (7). The equivalence of the two methods is by (10).

Then in Phase II (iterations), we run the SPGD Algorithm 2 and various explicit algorithms, namely the explicit PGD Algorithm 1, HOOI [15], quasi-Newton [57] and trust-region [32] methods²

For the last three algorithms, we use the code `tucker_als.m` in Tensor Toolbox [37], `symalgQN1c.m` in package [56] and `mlra3_rtr.m` in TensorLab [65], respectively. HOOI and trust-region apply to general asymmetric tensors. We omit the trust-region method in $d = 4$ case as its code does not apply to the fourth order. The step size for Algorithm 2 is $c_2 = 1000$. Note that the implicit Algorithm 2 matches the explicit Algorithm 1 iteration-by-iteration, because there is no streaming. However, they calculate the updates differently because Algorithm 2 is implicit.

Figure 2 shows the running time of the two phases as the dimension n increases. The time on the vertical axes shows when the algorithm reaches a relative gradient to be less than 10^{-12} . This measure of convergence precision is used in [21, 57, 32] and defined as:

$$\text{relative gradient} := \frac{\|\text{grad } F(\mathbf{Q})\|}{F(\mathbf{Q})}.$$

Here $\text{grad } F(\mathbf{Q}) = (\mathbf{I}_n - \mathbf{Q}\mathbf{Q}^\top) \nabla F(\mathbf{Q}) \in \mathbb{R}^{n \times r}$ denotes the Riemannian gradient on the Stiefel manifold (4), where $\nabla F(\mathbf{Q}) \in \mathbb{R}^{n \times r}$ is the Euclidean gradient (6). We use the Riemannian gradient $\text{grad } F(\mathbf{Q})$ because

²Similar to our algorithm scheme, the comparable algorithms are mostly gradient and Hessian-driven methods. Since we could not identify an implementation for the Jacobi rotation method [31, 41], it is not included in the comparison.

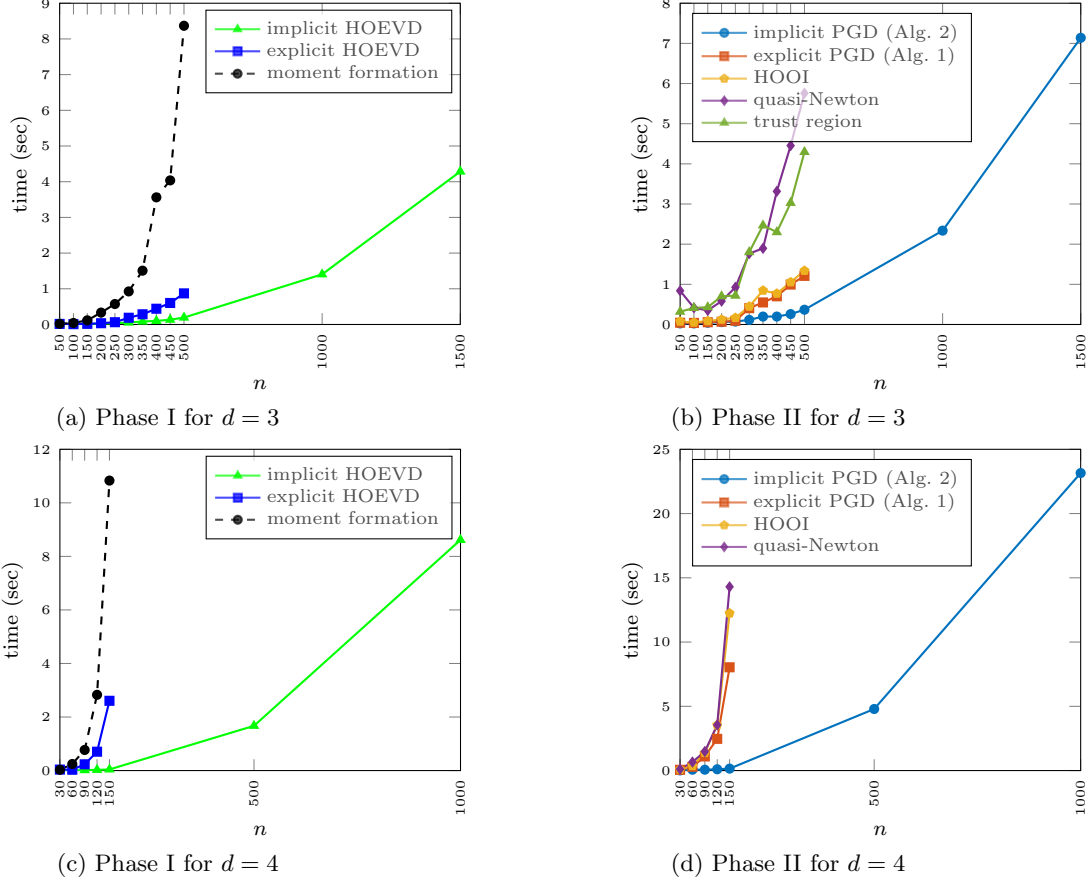


Figure 2: The scalability of implicit and explicit algorithms. Implicit method is much more scalable.

the possible update direction on the Stiefel manifold is constrained to the tangent space at \mathbf{Q} . The time for checking the relative gradient is included in the runtime.

It is evident from Figure 2 that the implicit HOEVD (17) and SPGD (Algorithm 2) are much faster than the explicit algorithms. The advantage of being tensor-free allows the implicit implementation to handle computation in higher dimensions. Meanwhile, the prohibitive upfront and storage costs in the explicit methods restrict their scalability.

4.3 Streaming versus full data usage

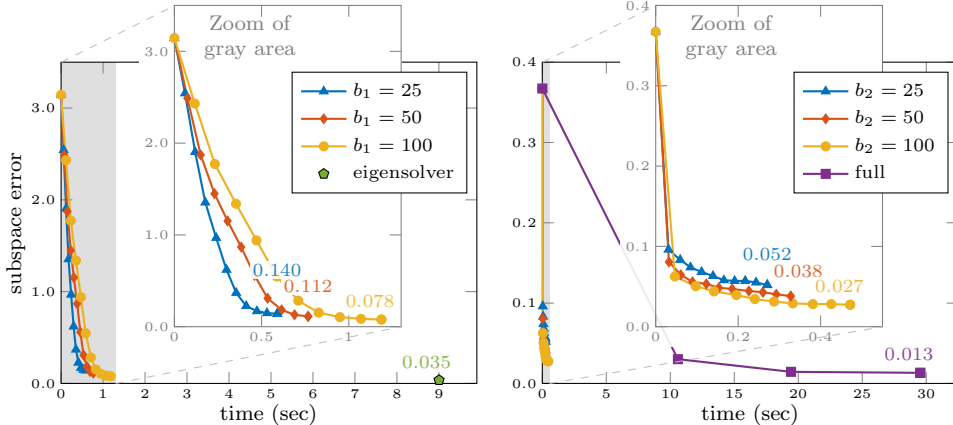
We stress the gains provided by the streaming technique. This is demonstrated in stochastic experiments for (a) SHOEVVD (Algorithm 3) and (b) SPGD (Phase II of Algorithm 2) in Figure 3. The vertical axes represent the approximation accuracy evaluated by the metric:

$$\text{subspace error} := \|\mathbf{Q}\mathbf{Q}^\top - \mathbf{Q}^*\mathbf{Q}^{*\top}\|.$$

The measure is rotationally invariant. We plot the subspace error against runtime.

To compare streaming and full data methods, we first generate a pool of observations $\mathbf{X} \in \mathbb{R}^{n \times p}$. Each streaming method selects a subset of samples $\mathbf{X}(:, b(t-1)+1 : bt)$ from the pool at step t . We ensure the total sample number p is just enough for the streaming case with the largest batch size b_{\max} . The non-streaming counterparts use all the samples \mathbf{X} at once. We set the number of iterations $T = 250$, and the total sample size $p = Tb_{\max} = 25,000$.

In Phase I (a), the full HOEVD refers to the implicit way (17), which is already more efficient than explicit HOEVD (7). The step size for all streaming cases of Algorithm 3 is $c_1 = 0.05$. In Phase II (b), all



(a) SHOEVD with $T_1 = 10$ and $c_1 = 0.05$. The parameter b_1 is the streaming block-size. (b) SPGD with $T_2 = 10$ and $c_2 = 1$. The parameter b_2 is the streaming blocksize. For full, $c_2 = 10$ and $b_2 = 25,000$.

Figure 3: Streaming versus full data usage on randomly-generated third-order moment tensor with $n = 500$, $p = 25,000$, and $\text{SNR}^{-1} = 0.5$, using $r = r_{\text{true}}$. The final error values for all methods are marked on the plot.

plots start at the same initialization by Algorithm 3 with $T_1 = 20$, $b_1 = 50$ and $c_1 = 1$. These parameters for Phase I are different from the (a) setting. They do not affect the efficiency of (b). The streaming and full cases for the actual Phase II use step size $c_2 = 1$ and $T_2 = 10$ respectively.

It is seen in Figure 3 that the streaming Algorithms 2 and 3 have limited loss of accuracy compared to the methods that use the full dataset. On the other hand, the runtimes for all streaming cases are nearly negligible compared to timings for the full methods.

4.4 Robustness

We test how the noise level in the observations and the choice of target rank affect HOEVD and symmetric Tucker decomposition in Figures 4 and 5. In the two experiments below, we plot the evolution of both phases of Algorithm 2, see Figures 4 and 5. As the HOEVD phase (in dashed lines) plateaus, its last iterate gives the initialization for Phase II (in solid lines). The number of iterations are shown on the horizontal axes. The parameters in Algorithm 2 are set to be $r_{\text{true}} = 5$, $b_1 = b_2 = 50$, $c_1 = c_2 = 1$.

We first consider the algorithms' sensitivity to noise in the data samples. To test this, we generate a sequence of clean samples and then plant Gaussian noise with different levels of SNR^{-1} . The algorithms' results are shown in Figure 4. In particular, note the improvement in Phase II over the HOEVD phase. The discrepancy is enlarged in the high noise regime ($\text{SNR}^{-1} \geq 1$), when the HOEVD solver provides a poor solution. In contrast, relatively good approximation is obtained using the symmetric Tucker decomposition (Phase II) when SNR^{-1} is no bigger than 1.

Next, we test various rank choices in Figure 5. All cases pass over the same set of data observations. The algorithms' recovery qualities are reported in terms of the objective value $F(\mathbf{Q})$, and compared to the "ground-truth" value $F(\mathbf{Q}^*)$. As there is no deterministic moment \mathcal{M}^d in the streaming case, we estimate F using an additional set of testing samples $\mathbf{X}_{\text{test}} \in \mathbb{R}^{n \times p_{\text{test}}}$ drawn from (15). We calculate F implicitly using (9):

$$\begin{aligned} \text{objective value: } F(\mathbf{Q}) &= \frac{1}{p_{\text{test}}^2} \left\| \sum_{i=1}^{p_{\text{test}}} (\mathbf{Q}^\top \mathbf{x}_{\text{test}})_i^{\otimes d} \right\|^2 \\ &= \frac{1}{p_{\text{test}}^2} \left\| (\mathbf{X}_{\text{test}}^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{X}_{\text{test}})^{[d]} \right\|. \end{aligned}$$

When $r \geq r_{\text{true}}$, the solutions obtained by Algorithm 2 fully approximate the moment tensor in the sense that their objective values match the ground truth. When $r < r_{\text{true}}$, we can only partially recover the tensor.

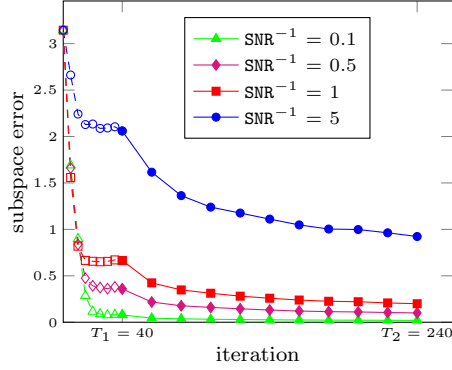


Figure 4: Two phases of SPGD with different noise levels. Tensor has $n = 500$, $p = 12000$, $d = 3$, and $r = r_{\text{true}} = 5$. In Algorithm 2, $b_1 = b_2 = 50$ and $c_1 = c_2 = 1$.

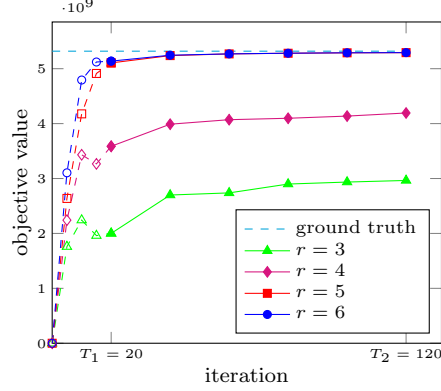


Figure 5: Two phases of SPGD with different target ranks. Tensor has $n = 500$, $p = 6000$, $d = 3$, and $r_{\text{true}} = 5$, and $\text{SNR}^{-1} = 0.5$. In Algorithm 2, $b_1 = b_2 = 50$ and $c_1 = c_2 = 1$.

The less the rank is, the lower the objective value. Also note that as r decreases, HOEVD’s initialization moves further away from the ground truth and the gap between the phases is more pronounced.

Figures 4 and 5 show that the robustness of the symmetric Tucker decomposition is stronger than that of the HOEVD approximation. For “hard” problem instances where noise is high or the rank is under-specified, the HOEVD phase (in dashed lines) is clearly improved upon during Phase II (in solid lines).

4.5 Tucker versus CP decomposition

This subsection shows the situations where symmetric Tucker decomposition is more appropriate to use than the symmetric CP counterpart. We provide numerical evidence in the moment tensor approximation quality for low-rank factor model, while maintaining a low storage budget.

We employ the relative error as the accuracy metric:

$$\text{relative error} := \frac{\|\mathcal{M}^d - \mathcal{M}_{\text{test}}^d\|^2}{\|\mathcal{M}_{\text{test}}^d\|^2} \times 100\%.$$

The testing sample moment $\mathcal{M}_{\text{test}}^d \in \mathbb{S}^d(\mathbb{R}^n)$ serves as the reference solution and is built from the dataset $\mathbf{X}_{\text{test}} \in \mathbb{R}^{n \times p_{\text{test}}}$ via (1). The approximation tensor $\mathcal{M}^d \in \mathbb{S}^d(\mathbb{R}^n)$ is obtained via either the symmetric Tucker or the symmetric CP [59] decompositions. These solutions admit the formats:

$$\mathcal{M}_{\text{Tucker}}^d = \mathcal{M}_{\text{test}}^d \cdot (\mathbf{Q}\mathbf{Q}^\top, \dots, \mathbf{Q}\mathbf{Q}^\top) \in \mathbb{S}^d(\mathbb{R}^n), \quad \mathcal{M}_{\text{CP}}^d = \sum_{i=1}^{r'} \lambda_i \mathbf{v}_i^{\otimes d} \in \mathbb{S}^d(\mathbb{R}^n). \quad (18)$$

In (18), we denote the Tucker and CP rank by r and r' respectively. The Tucker approximation is determined by the basis $\mathbf{Q} \in \mathbb{R}^{n \times r}$ and its core $\mathbf{c} \in \mathbb{S}^d(\mathbb{R}^r)$ is via (3). The CP format consists of the coefficients $\{\lambda_i\}_{i=1}^{r'} \subset \mathbb{R}$ and the vector components $\{\mathbf{v}_i\}_{i=1}^{r'} \subset \mathbb{R}^n$. For the Tucker and CP solutions, we respectively apply implicit stochastic methods: our Algorithm 2 and Algorithm cp_ism [59] with Adam optimization from Tensor Toolbox [37].

The data observations are drawn from the linear factor model (15). Here, the latent factor $\mathbf{f} \in \mathbb{R}^{r_{\text{true}}}$ follows the standard Gaussian distribution with zero mean and identity covariance. The noise level is $\text{SNR}^{-1} = 0.05$. In one simulation, we first draw the full data set $\mathbf{X}_{\text{test}} \in \mathbb{R}^{n \times p_{\text{test}}}$ with dimension $n = 500$ and sample size $p_{\text{test}} = 10000$. We form order $d = 4$ moment tensors. At each iteration of both algorithms, we compute the gradient based on a sample batch $\mathbf{X}_t \in \mathbb{R}^{n \times b}$ with batch size $b = 50$. Their objectives

are evaluated based on the full testing tensor $\mathcal{M}_{\text{test}}^d$. The algorithms take a complete pass over \mathbf{X}_{test} in one simulation (for Tucker) and one epoch (for CP). All epochs of one CP simulation operate on the same data \mathbf{X}_{test} .

In [Table 2](#), we compare the accuracies of the symmetric Tucker versus CP decompositions, in reference to $\mathcal{M}_{\text{test}}^d$. Regarding the target rank, the Tucker method sets it equal to the true rank: r_{true} . There are two choices for the CP method: r_{true} and r_{true}^{d-1} . For each table entry, we report the lowest error from 5 independent simulations.

Tucker rank: $r = r_{\text{true}}$	3	4	5	6	7
relative error (%)	0.0027	0.0035	0.0039	0.0048	0.0059
CP rank: $r' = r_{\text{true}}$	3	4	5	6	7
relative error (%)	32.74	42.36	45.94	52.77	57.59
CP rank: $r' = r_{\text{true}}^{d-1}$	27	64	125	216	343
relative error (%)	0.0025	0.0060	0.0084	0.0089	0.0106

Table 2: The error comparisons of the symmetric Tucker and CP decompositions when $n = 500, d = 4$. The row indicates the target rank choice. The moment tensor estimation errors are displayed along the columns. In the symmetric Tucker method ([Algorithm 2](#)), Phase I and II have $T_1 = 20, T_2 = 180$ iterations. AdaGrad step sizes are $c_1 = c_2 = 1$. In the CP algorithm, the iteration number per epoch is 200. The maximal epoch number is 100. Adam learning rate is 0.05 and the decay rate is 0.1.

We see that the Tucker estimation achieves the greatest accuracy on a low rank choice. On the other hand, CP decomposition suffers substantially larger error (10000 times worse than Tucker) when its target rank is small. CP manages to fit moment tensors with high precision similar to Tucker’s level for $r' = r_{\text{true}}^{d-1}$. However, comparing the storage costs, Tucker is much more economical in representing the moment tensor. From [\(18\)](#), Tucker costs $rn + \binom{r+d-1}{d} \approx r_{\text{true}}n$ to store \mathbf{Q} and \mathcal{C} , whereas the CP decomposition costs $r'(n+1) \approx r_{\text{true}}^{d-1}n$ in storage to achieve a similar approximation.

We conclude that there are important situations where the Tucker structure provides a more natural and more accurate format for moment tensors than CP does. This holds especially when the data samples lie close to a low-rank subspace.

5 Applications

This section illustrates the applicability of moment Tucker decompositions to real datasets from anomaly detection and portfolio allocation.

The computations are not wholly streaming, as we need to store and preprocess the data first. However, only data batches are used while running [Algorithms 2](#) and [3](#). The task results are based on one simulation.

5.1 Anomaly detection in hyperspectral imaging

Outliers can be discovered using higher-order statistics, when the mean and covariance information are not sufficient [\[18, 1\]](#). In hyperspectral imaging (HSI) [\[53\]](#), the detection goal is to find a low-rank feature subspace, denoted by $\mathbf{Q} \in \text{St}(n, r)$. After transforming HSI pixels by \mathbf{Q} , unusual targets in the image are revealed [\[25\]](#). Intuitively if the anomalies are captured by $\text{colspan}(\mathbf{Q})$, they should stand out as bright spots in the transformed image.

This section tests three methods to extract the feature subspace from HSI data: Tucker factorization of the skewness (order-3 standardized moment) [\[44\]](#), HOEVD applied to the skewness, and eigendecomposition of the covariance. Here, skewness and covariance are formed from HSI pixels. The results are qualitatively and quantitatively evaluated according to their performance in capturing anomalies, as in [\[25\]](#).

[Figure 6](#) shows the data structure of a typical HSI image. It is a 3-way hypercube $\mathcal{X} \in \mathbb{R}^{n \times p_1 \times p_2}$. The data consists of n band images in $\mathbb{R}^{p_1 \times p_2}$ and $p = p_1 p_2$ HSI pixels in \mathbb{R}^n . We use a real hyperspectral image from the Airport-Beach-Urban dataset [\[3\]](#). The scene was taken at Gulfport, Florida by the Airborne

³The dataset is available at <http://xudongkang.weebly.com/data-sets.html>.

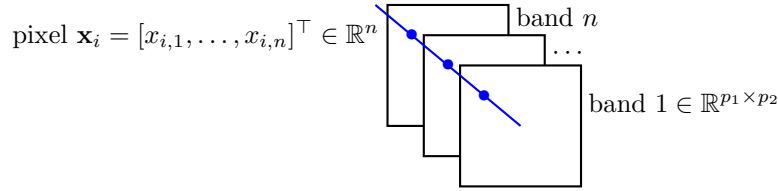


Figure 6: HSI hypercube.

Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. The top two images of [Figure 7](#) are the false-color composite image and the detection reference image (with white spots depicting the anomalous aircraft). The number of bands is $n = 191$ and the size of each band is $p_1 \times p_2 = 100 \times 100$. We use scalable algorithms for the moment decomposition task.

First, the HSI cube $\mathbf{X} \in \mathbb{R}^{n \times p_1 \times p_2}$ is reshaped into an $n \times p$ matrix. This is whitened [\[23\]](#) to give $\mathbf{X} \in \mathbb{R}^{n \times p}$. We choose rank $r = 4$, and seek the leading rank- r subspaces $\mathbf{Q} \in \mathbb{R}^{n \times r}$ from the skewness tensor $\mathbf{M}^3 \in \mathbb{S}^3(\mathbb{R}^n)$ and the covariance matrix $\mathbf{M}^2 \in \mathbb{S}^2(\mathbb{R}^n)$, both formed using the columns in \mathbf{X} . The subspaces are found using SPGD ([Algorithm 2](#)), SHOEVD ([Algorithm 3](#)), and PCA. The parameters for [Algorithms 2](#) and [3](#) are $b_1 = b_2 = 100$, $c_1 = 0.35$, $c_2 = 0.5$ and $T_1 = 500$, $T_2 = 1500$. For each subspace \mathbf{Q} obtained, \mathbf{X} is transformed into $\mathbf{Q}^\top \mathbf{X} \in \mathbb{R}^{r \times p}$. Each row of $\mathbf{Q}^\top \mathbf{X}$ is reshaped to a $p_1 \times p_2$ matrix. The resulting bands 1 - 4 are shown in [Figure 7](#).

Since the reference map in [Figure 7](#) is sparse, we further pursue a sparsest detection image by optimizing over weights on the bands. This is achieved by a constrained ℓ_1 -minimization:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^r} \|\mathbf{X}^\top \mathbf{Q} \mathbf{w}\|_1, \quad \text{subject to } \|\mathbf{w}\|_2 = 1.$$

This is solved by the MATLAB Optimization Toolbox function `fmincon`. Then we reshape $\mathbf{X}^\top \mathbf{Q} \mathbf{w}^*$ into a band image. The resulting sparse images are in the first column of [Figure 7](#).

Inspecting [Figure 7](#), we see that SPGD and SHOEVD based on skewness distinguish between abnormal and normal pixels well. The subspace provided by SPGD enables especially clear detection, since the targets are the most pronounced. It outlines both aircraft individually in the sparse image. The covariance information, however, fails to identify anomalies.

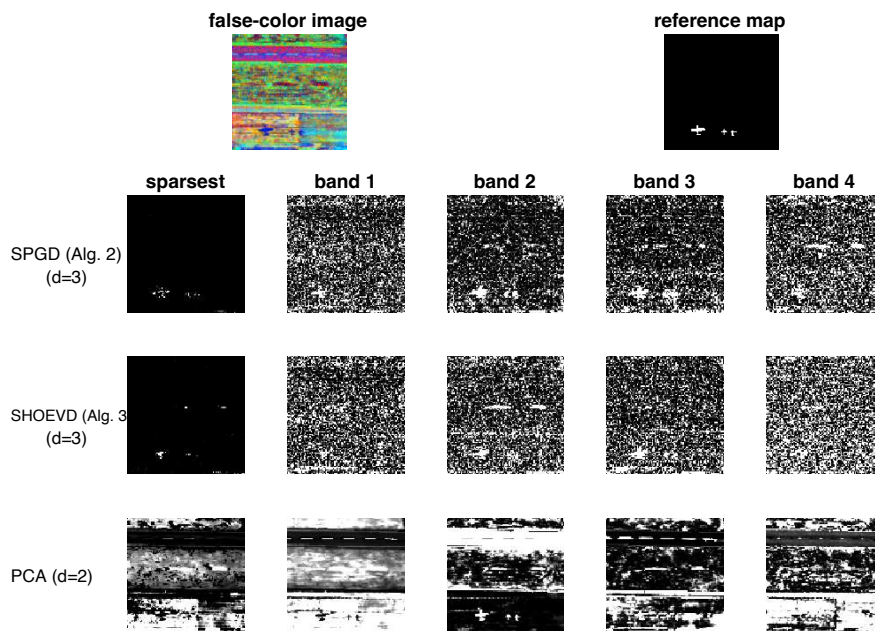


Figure 7: Qualitative detection comparison.

In [Figure 8](#), we show a quantitative evaluation in anomaly detection, via the receiver operating characteristic-area under curve (ROC-AUC). This quantifies how accurate a method is in classifying normal and abnormal objects, where a higher AUC value indicates better detection. To draw the ROC curve, we whiten the matrix $\mathbf{X}^\top \mathbf{Q}$ and assign the anomaly score for each pixel as its ℓ_2 -norm. The dashed curve indicates a random classifier. Note that the ROC of skewness-based detection schemes lie strictly above that of PCA. The AUC values corresponding to SPGD, SHOEVD and PCA are 0.9947, 0.9900 and 0.8893.

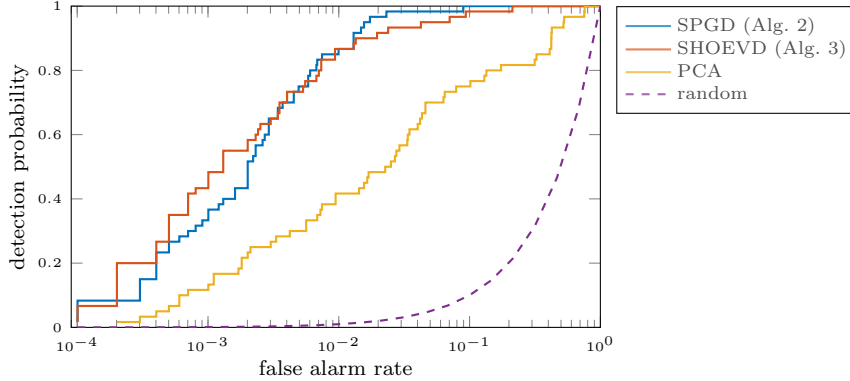


Figure 8: Quantitative detection comparison.

5.2 Moment estimation for portfolio allocation

Moment estimation is sometimes improved by enforcing a low-rank structure [\[6\]](#). In portfolio allocation, we seek a distribution over investments to maximize future returns. Judicious predictions can be obtained by including higher-order moment estimations in allocation optimization, as financial data is heavy-tailed [\[34\]](#).

Higher moments of financial data should roughly admit a low-rank symmetric Tucker format, because of the Fama-French model [\[22\]](#). In this section, we test two ways to calculate moments: the sample moment \mathcal{M}^d [\(1\)](#) and the low-rank symmetric Tucker approximation to \mathcal{M}^d . We evaluate the moments by deriving portfolios based on them and comparing the portfolio performances, as in [\[6\]](#).

We use the real HFR hedge fund dataset⁴. It consists of daily return rates (in percentage) of $n = 30$ HFRX indices from 01/02/2009 to 12/31/2019. The time series is split into in-sample data (2009 - 2016) $\mathbf{X}_{\text{in}} \in \mathbb{R}^{n \times p_{\text{in}}}$ ($p_{\text{in}} = 2016$) and out-of-sample data (2017 - 2019) $\mathbf{X}_{\text{out}} \in \mathbb{R}^{n \times p_{\text{out}}}$ ($p_{\text{out}} = 754$). The in-sample data is centered. We calculate the averaged sample moments [\(1\)](#): $\mathcal{M}^2 \in \mathbb{S}^2(\mathbb{R}^n)$, $\mathcal{M}^3 \in \mathbb{S}^3(\mathbb{R}^n)$ and $\mathcal{M}^4 \in \mathbb{S}^4(\mathbb{R}^n)$. Then their low-rank Tucker approximations are computed by SPGD [\(Algorithm 2\)](#). The rank is set to $r = 15$, and [\(3\)](#) is used to solve for the cores once the subspaces \mathbf{Q} are determined. We run Phase II of [Algorithm 2](#) twice. Starting from a random initialization, we obtain \mathbf{Q} for \mathcal{M}^2 . This is used as an initialization for decomposing \mathcal{M}^3 . Finally we tackle \mathcal{M}^4 , starting from the \mathbf{Q} for \mathcal{M}^3 . The parameters are set to $c_2 = 1, 10^{-2}, 10^{-5}$; $T_2 = 1000, 1000, 200$; and $b_2 = 500, 500, 500$ for the three runs respectively. This produces $\mathcal{M}_{\text{Tucker}}^2 \in \mathbb{S}^2(\mathbb{R}^n)$, $\mathcal{M}_{\text{Tucker}}^3 \in \mathbb{S}^3(\mathbb{R}^n)$ and $\mathcal{M}_{\text{Tucker}}^4 \in \mathbb{S}^4(\mathbb{R}^n)$.

Given moment estimates, we select the allocation weights that maximize the expected utility (EU) [\[6\]](#), [Supplementary Material](#), Eqn. (17)]:

$$\begin{aligned} \mathbf{w}^* &:= \arg \max_{\substack{\sum_i w_i = 1 \\ w_i \geq 0}} \text{EU}(\mathbf{w}) \\ &\equiv -\frac{\mu}{2} \langle \mathbb{E}((\mathbf{x} - \bar{\mathbf{x}})^{\otimes 2}), \mathbf{w}^{\otimes 2} \rangle + \frac{\mu(\mu+1)}{6} \langle \mathbb{E}((\mathbf{x} - \bar{\mathbf{x}})^{\otimes 3}), \mathbf{w}^{\otimes 3} \rangle \\ &\quad - \frac{\mu(\mu+1)(\mu+2)}{24} \langle \mathbb{E}((\mathbf{x} - \bar{\mathbf{x}})^{\otimes 4}), \mathbf{w}^{\otimes 4} \rangle, \end{aligned} \quad (19)$$

where the risk aversion parameter μ is set to be 1. We estimate $\mathbb{E}((\mathbf{x} - \bar{\mathbf{x}})^{\otimes d}) \approx \mathcal{M}^d$ in [\(19\)](#), and then apply the MATLAB Optimization Toolbox function `fmincon` to output the optimal weights. Meanwhile, estimating $\mathbb{E}((\mathbf{x} - \bar{\mathbf{x}})^{\otimes d}) \approx \mathcal{M}_{\text{Tucker}}^d$ results in another weight vector.

⁴The HFRX indices dataset can be found at: <https://www.hfr.com/family-indices/hfrx>

The two portfolios are evaluated on the out-sample data. The acquired daily returns $\{y_t\} \subseteq \mathbb{R}$ are collected in a vector $\mathbf{y} = \mathbf{X}_{\text{out}}^\top \mathbf{w}^* \in \mathbb{R}^{p_{\text{out}}}$. In [Figure 9](#), we plot the cumulative daily return rates $(\prod_{t'=1}^t (1 + \frac{y_{t'}}{100}) - 1) \times 100$ at days $t = 1, \dots, p_{\text{out}}$. We observe that the low-rank moment estimates produce a better return rate than the plain sample moments do. Despite mild fluctuations, the two strategies always produce positive return rates. A naive strategy, the equally-weighted portfolio, is the most unstable. It sometimes yields negative returns.

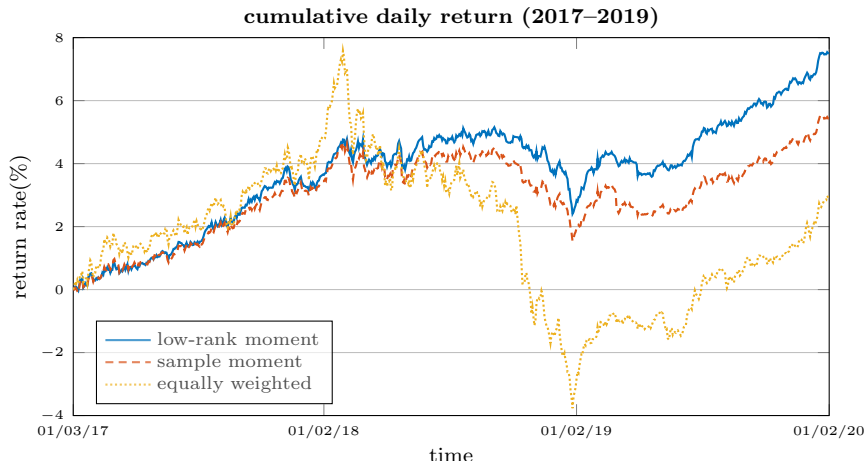


Figure 9: Portfolio returns under different weights.

Other statistical measures are shown in [Table 3](#). The annual geometric mean of returns over the years 2017 – 2019 is $\left(\left(\prod_{t=1}^{p_{\text{out}}} \left(1 + \frac{y_t}{100} \right) \right)^{1/3} - 1 \right) \times 100$. The centered d -th moment ($d = 2, 3, 4$) of daily returns $\{y_t\}$ is $m^d = \sum_{t=1}^{p_{\text{out}}} \frac{(y_t - \bar{y})^d}{p_{\text{out}}}$. Large geometric mean and low absolute values of moments are desirable allocation outcomes. In each row, we highlight the preferred score in bold. They are all achieved by the portfolio based on low-rank Tucker approximations to the moments.

Table 3: Statistical performance of portfolios.

	Equally weighted	Sample	Low rank ($r = 15$)
annual geometric mean	9.4143×10^{-1}	2.2434	2.3268
m^2	2.6914×10^{-2}	8.9233×10^{-3}	8.5933×10^{-3}
m^3	-2.9724×10^{-3}	-7.7193×10^{-4}	-6.0692×10^{-4}
m^4	3.4507×10^{-3}	5.4339×10^{-4}	4.7906×10^{-4}

6 Convergence analysis

This section presents a convergence theory for the basic PGD method ([Algorithm 1](#)) applied to symmetric Tucker decomposition ([3](#)). We note several points in the analysis: 1) The tensor \mathcal{X} needs not arise as a moment tensor; even when it does, we do not analyze the effects of streaming. 2) Proving convergence of the non-streaming algorithm to critical points is needed first, because ([3](#)) is non-convex, with a complex landscape when $d \geq 3$ (recall [Remark 3](#)). 3) Our results do not immediately follow from known general properties of plain GD or projected GD. We use the specific properties of the Tucker objective function and QR retraction.

6.1 Main statements

Recall the set-up in [Section 2.2](#). Because the Tucker factorization cost [\(8\)](#) is non-unique as

$$\|\mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q})\|^2 = \|\mathbf{X} \cdot (\mathbf{Q}\mathbf{O}, \dots, \mathbf{Q}\mathbf{O})\|^2 \quad \text{for all } r \times r \text{ orthogonal matrices } \mathbf{O},$$

the convergence of $\{\mathbf{Q}_t\}$ is not relevant. Instead we focus on the convergence of the projectors, $\mathbf{P}_t := \mathbf{Q}_t \mathbf{Q}_t^\top \in \text{Gr}(n, r)$. Thus, while [Algorithm 1](#) runs on the Stiefel manifold and generates the sequence of *bases* $\{\mathbf{Q}_t\} \subseteq \text{St}(n, r)$, we analyze the sequence of *subspaces* $\{\mathbf{P}_t\} \subseteq \text{Gr}(n, r)$ on the Grassmannian manifold.

Given scalar step sizes $\{\gamma_t\} \subseteq \mathbb{R}_{\geq 0}$, the update scheme starts from the PGD iteration on \mathbf{Q} :

$$\mathbf{Q}_{t+1} = \Phi(\mathbf{Q}_t) := \text{orth}(\mathbf{Q}_t + \gamma_t \nabla F(\mathbf{Q}_t)) \in \text{St}(n, r). \quad (20)$$

The corresponding update on \mathbf{P} is defined as

$$\mathbf{P}_{t+1} = \phi(\mathbf{P}_t) := \mathbf{Q}_{t+1} \mathbf{Q}_{t+1}^\top = \Phi(\mathbf{Q}_t) \Phi(\mathbf{Q}_t)^\top \in \text{Gr}(n, r). \quad (21)$$

$$\begin{array}{ccc} \mathbf{Q}_{t+1} = \Phi(\mathbf{Q}_t) & \longrightarrow & \mathbf{P}_{t+1} = \mathbf{Q}_{t+1} \mathbf{Q}_{t+1}^\top = \phi(\mathbf{P}_t) \\ \text{St}(n, r) & & \text{Gr}(n, r) \\ \uparrow \Phi & & \uparrow \phi \\ \text{St}(n, r) & \longrightarrow & \text{Gr}(n, r) \\ \mathbf{Q}_t & & \mathbf{P}_t = \mathbf{Q}_t \mathbf{Q}_t^\top \end{array}$$

Figure 10: Update diagram.

The sequence $\{\mathbf{P}_t\}$ is assessed in terms of the cost function

$$f(\mathbf{P}) := \|\mathbf{X} \cdot (\mathbf{P}, \dots, \mathbf{P})\|^2. \quad (22)$$

As noted in [Section 2.2](#), it is equivalent to the cost in [\(8\)](#), as $f(\mathbf{P}) = F(\mathbf{Q})$ when $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top$. We prove that the iterates $\{\mathbf{P}_t\}$ converge to a critical point of f on the Grassmannian. This is stronger than just showing convergence of the objective values $\{f(\mathbf{P}_t)\}$ in \mathbb{R} .

The first and second-order criticality conditions are stated as follows in the manifold setting.

Definition 7. A point $\mathbf{P} \in \text{Gr}(n, r)$ is first-order critical for the cost function [\(22\)](#) if

$$\text{grad } f(\mathbf{P}) = \mathbf{0}. \quad (23)$$

Here, $\text{grad } f(\mathbf{P}) \in T_{\mathbf{P}} \text{Gr}(n, r) \subseteq \mathbb{S}^2(\mathbb{R}^n)$ denotes the Riemannian gradient, which is defined as the Euclidean gradient $\nabla f(\mathbf{P}) \in \mathbb{S}^2(\mathbb{R}^n)$ projected to $T_{\mathbf{P}} \text{Gr}(n, r)$. The tangent space at \mathbf{P} is given by

$$T_{\mathbf{P}} \text{Gr}(n, r) := \{\Delta \mathbf{P} \in \mathbb{S}^2(\mathbb{R}^n) \mid \mathbf{P} \Delta \mathbf{P} + (\Delta \mathbf{P}) \mathbf{P} = \Delta \mathbf{P}, \text{trace}(\Delta \mathbf{P}) = 0\}. \quad (24)$$

Definition 8. A point $\mathbf{P} \in \text{Gr}(n, r)$ is second-order critical for the cost function [\(22\)](#) if

$$\text{grad } f(\mathbf{P}) = \mathbf{0} \quad \text{and} \quad \langle \Delta \mathbf{P}, \text{Hess } f(\mathbf{P})[\Delta \mathbf{P}] \rangle \leq 0 \quad \text{for all } \Delta \mathbf{P} \in T_{\mathbf{P}} \text{Gr}(n, r). \quad (25)$$

Here, $\text{Hess } f(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow T_{\mathbf{P}} \text{Gr}(n, r)$ is the Riemannian Hessian. It is calculated in [\[27, Eqn. \(2.109\)\]](#).

The first theorem is a uniform guarantee: [Algorithm 1](#) always converges to a first-order critical point regardless where the iteration begins.

Theorem 6.1. Let $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ be any symmetric tensor, and consider the objective function f defined by [\(22\)](#). There exists a constant $\Gamma^*(\mathbf{X}, r) > 0$ such that if the step sizes $\{\gamma_t\} \subseteq \mathbb{R}_{\geq 0}$ satisfy $\sup_t \gamma_t \leq \Gamma^*$ and $\inf_t \gamma_t > 0$, the following holds. For all initializations $\mathbf{Q}_0 \in \text{St}(n, r)$, the sequence $\{\mathbf{P}_t\} \subseteq \text{Gr}(n, r)$ generated via [\(20\)](#)-[\(21\)](#) by [Algorithm 1](#) converges monotonically to a first-order critical point [\(23\)](#) of f . The convergence is at no less than an algebraic rate.

Next, in fact, [Algorithm 1](#) converges to a second-order critical point, for almost all initializations.

Theorem 6.2. *Let $\mathbf{X} \in \mathbb{S}^d(\mathbb{R}^n)$ be any symmetric tensor, and consider the cost f defined by [\(22\)](#). Use a constant step size γ for all iterates. There exist a constant $\Gamma^{**}(\mathbf{X}, r) > 0$ and a measure-zero subset $\mathcal{I} \subseteq \text{St}(n, r)$ such that if $0 < \gamma \leq \Gamma^{**}$, then for all initializations $\mathbf{Q}_0 \in \text{St}(n, r) \setminus \mathcal{I}$, the sequence $\{\mathbf{P}_t\} \subseteq \text{Gr}(n, r)$ generated via [\(20\)](#)-[\(21\)](#) converges to a second-order critical point [\(25\)](#) of f .*

We build the proofs of [Theorems 6.1](#) and [6.2](#) respectively in [Sections 6.2](#) and [6.3](#). The main technical tools are a convergence theorem for real-analytic cost functions (based on the Lojasiewicz inequality) and the center-stable manifold theorem. These are stated in [Appendix A](#). Detailed proofs for supporting propositions and lemmas are in the Supplementary Materials. They involve non-trivial calculations for the QR retraction and the symmetric Tucker cost function [\(22\)](#).

Remark 8. *The convergence of PGD in $\text{Gr}(n, r)$ is not for free. Previously, Regalia found example tensors \mathbf{X} where our [Algorithm 2](#) with a step size approaching infinity shows an oscillatory cycle, rather than convergence. See [\[54\]](#), [Figures 1 and 4](#).*

6.2 Monotonic convergence to first-order critical points

To establish [Theorem 6.1](#), we apply the guarantee stated in [Theorem A.1](#), originally from [\[58\]](#), [Theorem 2.3](#). It requires us to prove $\{\mathbf{P}_t\}$ meets **C1** - **C4** conditions in [Theorem A.1](#).

We first compute the Riemannian gradient of f [\(22\)](#) on $\text{Gr}(n, r)$. Although [Theorem 6.1](#) is in terms of $\mathbf{P} \in \text{Gr}(n, r)$, our proof uses $\mathbf{Q} \in \text{St}(n, r)$ due to the explicit update rule in [\(20\)](#). Thus, we state the relation between $\text{grad } f(\mathbf{P})$ and $\mathbf{Q}, \nabla F(\mathbf{Q})$. As short-hand notation, define $w : \text{Gr}(n, r) \rightarrow \mathbb{S}^2(\mathbb{R}^n)$ by

$$w(\mathbf{P}) = d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \in \mathbb{S}^2(\mathbb{R}^n). \quad (26)$$

Note that $w(\mathbf{P})$ is a positive semidefinite (PSD) matrix.

Proposition 6.3. *For $\mathbf{Q} \in \text{St}(n, r)$ and $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top \in \text{Gr}(n, r)$, the following holds*

$$\text{grad } f(\mathbf{P}) = \text{sym} \left(2(\mathbf{I}_n - \mathbf{P}) w(\mathbf{P}) \mathbf{P} \right) = \text{sym} \left((\mathbf{I}_n - \mathbf{Q}\mathbf{Q}^\top) \nabla F(\mathbf{Q}) \mathbf{Q}^\top \right). \quad (27)$$

[Proposition 6.3](#) is proven in [Appendix B](#).

Next, we analyze the update rule on $\mathbf{P} \in \text{Gr}(n, r)$. For $t \geq 0$ and step size $\gamma_t > 0$, define $\alpha_t : \text{Gr}(n, r) \rightarrow \mathbb{S}^2(\mathbb{R}^n)$ by

$$\alpha_t(\mathbf{P}) = \mathbf{P} + 2\gamma_t \mathbf{P} w(\mathbf{P}) \mathbf{P} \in \mathbb{S}^2(\mathbb{R}^n). \quad (28)$$

The expression $\alpha_t(\mathbf{P})$ and the gradient $\text{grad } f(\mathbf{P})$ help to represent the update on \mathbf{P} .

Lemma 6.4. *Let a step $t \geq 0$. The difference between the current iterate \mathbf{P}_t and the next \mathbf{P}_{t+1} [\(20\)](#)-[\(21\)](#) is approximated as follows:*

$$\begin{aligned} \mathbf{P}_{t+1} - \mathbf{P}_t &= 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \right) \\ &\quad + \mathcal{O} \left(\gamma_t^2 \|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger\|^2 \right) + \mathcal{O} \left(\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \right). \end{aligned} \quad (29)$$

The proof of [Lemma 6.4](#) is in [Appendix C](#). It relies on the Taylor expansion of [\(20\)](#).

The following lemma gives properties of $\alpha_t(\mathbf{P})$ and shows that when the step size is sufficiently small, $\|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger\|$ and $\|\text{grad } f(\mathbf{P}_t)\|$ are on the same order.

Lemma 6.5. *Let $t \geq 0$ and $\mathbf{P} \in \text{Gr}(n, r)$. The pseudoinverse $\alpha_t(\mathbf{P})^\dagger \in \mathbb{S}^2(\mathbb{R}^n)$ satisfies $\alpha_t(\mathbf{P})^\dagger \alpha_t(\mathbf{P}) = \mathbf{P}$. Moreover, there exists a constant $\Gamma_1(\mathbf{X}, r) > 0$ such that if γ_t satisfies $0 < \gamma_t \leq \Gamma_1$, then $\|\mathbf{P} - \alpha_t(\mathbf{P})^\dagger\| \leq \frac{1}{2}$ and*

$$\frac{\sqrt{2}}{3} \|\text{grad } f(\mathbf{P}_t)\| \leq \|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger\| \leq \frac{\sqrt{2}r}{2} \|\text{grad } f(\mathbf{P}_t)\|. \quad (30)$$

Please refer to [Appendix D](#) for the proof of [Lemma 6.5](#). It utilizes the fact that α_t is a bounded continuous function of \mathbf{P} .

Proof of [Theorem 6.7](#). With the above useful lemmas and proposition in place, in this proof, we show that our update scheme for $\mathbf{P} \in \text{Gr}(n, r)$ [\(20\)](#)-[\(21\)](#) satisfies the conditions **C1** - **C4** in [Theorem A.1](#). Fix a step $t \geq 0$. Assume $0 < \gamma_t \leq \Gamma_1$ where Γ_1 is as in [Lemma 6.5](#).

(C1) We apply [\(30\)](#) that $\mathcal{O}(\|\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger\|)$ has the same scale as $\mathcal{O}(\|\text{grad } f(\mathbf{P}_t)\|)$. By [Lemma 6.4](#), it follows

$$\mathbf{P}_{t+1} - \mathbf{P}_t = 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger \right) + \mathcal{O} \left(\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \right). \quad (31)$$

It implies that there exists a constant $C_1(\mathfrak{X}, r) > 0$ such that

$$\left\| \mathbf{P}_{t+1} - \mathbf{P}_t - 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger \right) \right\| \leq C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2.$$

By [\(59\)](#) in [Appendix C](#),

$$\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger = (\mathbf{I}_n - \mathbf{P}_t)w(\mathbf{P}_t)\mathbf{P}_t\alpha_t(\mathbf{P}_t)^\dagger = (\mathbf{I}_n - \mathbf{P}_t)w(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger,$$

and so

$$\begin{aligned} & \langle \text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger, \alpha_t(\mathbf{P}_t)^\dagger \text{grad } f(\mathbf{P}_t) \rangle \\ &= \text{trace} \left((\mathbf{I}_n - \mathbf{P}_t)w(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger (\mathbf{I}_n - \mathbf{P}_t)w(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger \right) = 0. \end{aligned}$$

It implies

$$\left\| \text{sym} \left(\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger \right) \right\| = \frac{\sqrt{2}}{2} \|\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger\|.$$

By triangle inequality,

$$\begin{aligned} \|\mathbf{P}_{t+1} - \mathbf{P}_t\| &\geq 4 \times \frac{\sqrt{2}}{2} \gamma_t \|\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger\| - C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \\ &\geq 2\sqrt{2} \times \frac{\sqrt{2}}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\| - C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \\ &= \frac{4}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\| - C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2. \end{aligned} \quad (32)$$

Here, the second inequality is due to the first half of [\(30\)](#). We further constrain the step size as follows

$$0 < \gamma_t \leq \Gamma_2 := \min \left(\Gamma_1, \frac{\frac{2}{3}}{C_1 \sup_{\mathbf{P} \in \text{Gr}(n, r)} \|\text{grad } f(\mathbf{P})\|} \right), \quad (33)$$

so that

$$C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \leq \frac{2}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\|. \quad (34)$$

Note that $\sup_{\mathbf{P} \in \text{Gr}(n, r)} \|\text{grad } f(\mathbf{P})\|$ is finite because $\text{grad } f$ is continuous on $\text{Gr}(n, r)$ and the Grassmann is compact. (We also exclude the trivial case when $\text{grad } f(\mathbf{P}) \equiv 0$.) Continuing from [\(32\)](#) and substituting [\(34\)](#), we have

$$\|\mathbf{P}_{t+1} - \mathbf{P}_t\| \geq \left(\frac{4}{3} - \frac{2}{3} \right) \gamma_t \|\text{grad } f(\mathbf{P}_t)\| = \frac{2}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\|.$$

Letting $\kappa = \inf_{t \rightarrow \infty} \frac{2}{3} \gamma_t > 0$ ensures that **C1** [\(46\)](#) holds.

In preparation for **C2**, we derive an upper bound for $\|\mathbf{P}_{t+1} - \mathbf{P}_t\|$. The idea is similar to [\(32\)](#):

$$\begin{aligned} \|\mathbf{P}_{t+1} - \mathbf{P}_t\| &\leq 2\sqrt{2}\gamma_t \|\text{grad } f(\mathbf{P}_t)\alpha_t(\mathbf{P}_t)^\dagger\| + C_1\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \\ &\leq 2\sqrt{2} \times \frac{\sqrt{2r}}{2} \gamma_t \|\text{grad } f(\mathbf{P}_t)\| + \frac{2}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\| \\ &\leq \frac{8\sqrt{r}}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\|. \end{aligned} \quad (35)$$

The second inequality is obtained from the latter half of [\(30\)](#) and [\(34\)](#).

(C2) Keep the assumption (33) on γ_t . Consider the first-order Taylor's expansion for f on $\text{Gr}(n, r)$ [7, section 10.7]. Inserting (31), the difference of the current and next f is:

$$\begin{aligned} f(\mathbf{P}_{t+1}) - f(\mathbf{P}_t) &= \langle \text{grad } f(\mathbf{P}_t), \mathbf{P}_{t+1} - \mathbf{P}_t \rangle + \mathcal{O}(\|\mathbf{P}_{t+1} - \mathbf{P}_t\|^2) \\ &= \left\langle \text{grad } f(\mathbf{P}_t), 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \right) \right\rangle \\ &\quad + \mathcal{O} \left(\gamma_t^2 \max \left(\|\text{grad } f(\mathbf{P}_t)\|^3, \|\text{grad } f(\mathbf{P}_t)\|^2 \right) \right). \end{aligned} \quad (36)$$

Since $\text{grad } f(\mathbf{P}_t)$ is symmetric, the first term in the result of (36) is

$$\begin{aligned} &\left\langle \text{grad } f(\mathbf{P}_t), 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \right) \right\rangle \\ &= 4\gamma_t \langle \text{grad } f(\mathbf{P}_t), \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \rangle \\ &= 4\gamma_t \langle \text{grad } f(\mathbf{P}_t) \mathbf{P}_t, \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t \rangle \\ &= 4\gamma_t \langle \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \alpha_t(\mathbf{P}_t), \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t \rangle. \end{aligned}$$

The second equation is due to $\alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t = \alpha_t(\mathbf{P}_t)^\dagger$. The definition (28) gives $\alpha_t(\mathbf{P}_t) = \mathbf{P}_t(\mathbf{I}_n + 2\gamma_t w(\mathbf{P}_t))\mathbf{P}_t$. Continuing from above,

$$\begin{aligned} &\left\langle \text{grad } f(\mathbf{P}_t), 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \right) \right\rangle \\ &= 4\gamma_t \langle \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t (\mathbf{I}_n + 2w(\mathbf{P}_t)), \text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t \rangle \\ &\geq 4\gamma_t \|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger \mathbf{P}_t\|^2 = 4\gamma_t \|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger\|^2 \\ &\geq \frac{8}{9}\gamma_t \|\text{grad } f(\mathbf{P}_t)\|^2. \end{aligned} \quad (37)$$

Here we used that $\mathbf{I}_n + 2\gamma_t w(\mathbf{P}_t) \succeq \mathbf{I}_n$ for $\gamma_t \geq 0$, since $w(\mathbf{P}_t) \in \mathbb{S}^2(\mathbb{R}^n)$ is PSD. The second inequality is derived from (30).

Considering the remainder term in (36), there is a constant $C_2(\mathbf{X}, r) > 0$ such that

$$\begin{aligned} f(\mathbf{P}_{t+1}) - f(\mathbf{P}_t) &\geq \frac{8}{9}\gamma_t \|\text{grad } f(\mathbf{P}_t)\|^2 \\ &\quad - C_2\gamma_t^2 \max \left(\|\text{grad } f(\mathbf{P}_t)\|^3, \|\text{grad } f(\mathbf{P}_t)\|^2 \right). \end{aligned} \quad (38)$$

If the step size γ_t is small enough to satisfy

$$0 < \gamma_t \leq \Gamma^* := \min \left(\Gamma_2, \frac{4}{9C_2} \right), \quad (39)$$

where $\Gamma_2 > 0$ is defined in (33), then

$$C_2\gamma_t^2 \max \left(\|\text{grad } f(\mathbf{P}_t)\|^3, \|\text{grad } f(\mathbf{P}_t)\|^2 \right) \leq \frac{4}{9}\gamma_t \|\text{grad } f(\mathbf{P}_t)\|^2.$$

Inserting into (38),

$$\begin{aligned} f(\mathbf{P}_{t+1}) - f(\mathbf{P}_t) &\geq \left(\frac{8}{9} - \frac{4}{9} \right) \gamma_t \|\text{grad } f(\mathbf{P}_t)\|^2 \\ &= \frac{1}{6\sqrt{r}} \left(\frac{8\sqrt{r}}{3} \gamma_t \|\text{grad } f(\mathbf{P}_t)\| \right) \|\text{grad } f(\mathbf{P}_t)\| \\ &\geq \frac{1}{6\sqrt{r}} \|\mathbf{P}_{t+1} - \mathbf{P}_t\| \|\text{grad } f(\mathbf{P}_t)\|. \end{aligned}$$

The final inequality is by (35). Thus C2 holds if we set $\sigma = \frac{1}{6\sqrt{r}}$ in (47).

(C3) “ \Rightarrow ”: From $\text{grad } f(\mathbf{P}) = \mathbf{0}$ and (27), we know

$$\text{grad } f(\mathbf{P})\mathbf{Q} = \mathbf{0} = \text{grad } F(\mathbf{Q}) = (\mathbf{I}_n - \mathbf{Q}\mathbf{Q}^\top) \nabla F(\mathbf{Q}),$$

where $\mathbf{Q} \in \text{St}(n, r)$ is a basis of $\mathbf{P} \in \text{Gr}(n, r)$. It implies that $\text{colspan}(\nabla F(\mathbf{Q}))$ is in $\text{colspan}(\mathbf{Q})$. Note that $\mathbf{Q} + \gamma \nabla F(\mathbf{Q}) = (\mathbf{I}_n + 2\gamma w(\mathbf{P})) \mathbf{Q}$ has full column rank, like \mathbf{Q} , because $\mathbf{I}_n + 2\gamma w(\mathbf{P}) \in \mathbb{S}^2(\mathbb{R}^n)$ is positive definite. Hence $\text{colspan}(\mathbf{Q} + \gamma \nabla F(\mathbf{Q})) = \text{colspan}(\mathbf{Q})$. Then the next iterate $\Phi(\mathbf{Q})$ obtained by QR decomposition still outputs a basis for $\text{colspan}(\mathbf{Q})$. So the orthogonal projectors remain the same, i.e. $\phi(\mathbf{P}) = \mathbf{P}$.

“ \Leftarrow ”: This directly follows from **C1** if γ_t is sufficiently small to satisfy (33).

- (C4) The Grassmannian is a real-analytic submanifold of $\text{Gr}(n, r)$, because $\text{Gr}(n, r)$ satisfies property (3) in [38, Proposition 2.7.3]. Also, the cost function f (22) is a polynomial in $\mathbf{P} \in \text{Gr}(n, r)$, so real-analytic as well. By [58, Proposition 2.2], all points $\mathbf{P} \in \text{Gr}(n, r)$ satisfy a Łojasiewicz inequality for f .

We have shown that the sequence $\{\mathbf{P}_t\}$ meets all **C1** - **C4** provided $\{\gamma_t\}$ satisfies $\inf \gamma_t > 0$ and $\sup \gamma_t \leq \Gamma^*$, with Γ^* given in (39). By Theorem A.1, it follows Algorithm 1 achieves convergence to a first-order critical point of f . The proof of Theorem 6.1 is complete. \square

6.3 Convergence to second-order critical points

We establish Theorem 6.2 regarding second-order criticality when the step size is constant. The proof relies on the center-stable manifold theorem [60], which is recalled in Theorem A.2. It characterizes the local structure of a smooth local diffeomorphism around a fixed point.

The calculation of the Riemannian Hessian, $\text{Hess } f(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow T_{\mathbf{P}} \text{Gr}(n, r)$, and the differential of the update map, $D\phi(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow T_{\phi(\mathbf{P})} \text{Gr}(n, r)$, are essential to showing the second-order guarantee. As short-hand, we define a linear operator $v(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow \mathbb{R}^{n \times n}$ by

$$v(\mathbf{P})[\Delta \mathbf{P}] = d(d-1) \langle \mathbf{X} \cdot (\mathbf{I}_n, \Delta \mathbf{P}, \mathbf{P}, \dots, \mathbf{P}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \mathbf{P} \in \mathbb{R}^{n \times n}. \quad (40)$$

Proposition 6.6. For $\mathbf{P} \in \text{Gr}(n, r)$ and $\Delta \mathbf{P} \in T_{\mathbf{P}} \text{Gr}(n, r)$, the Riemannian Hessian of the cost f (22) on $\text{Gr}(n, r)$ is given by

$$\text{Hess } f(\mathbf{P})[\Delta \mathbf{P}] = 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P}) v(\mathbf{P})[\Delta \mathbf{P}] + w(\mathbf{P}) (\Delta \mathbf{P}) \mathbf{P} - \Delta \mathbf{P} w(\mathbf{P}) \mathbf{P} \right). \quad (41)$$

The differential of ϕ (21) on $\text{Gr}(n, r)$ is given by

$$\begin{aligned} D\phi(\mathbf{P})[\Delta \mathbf{P}] \\ = 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P})) (\Delta \mathbf{P} + 2\gamma w(\mathbf{P}) \Delta \mathbf{P} + 2\gamma v(\mathbf{P})[\Delta \mathbf{P}]) (\mathbf{P} + 2\gamma w(\mathbf{P}) \mathbf{P})^\dagger \right). \end{aligned} \quad (42)$$

The calculations in Proposition 6.6 are in Appendix E. These results are derived from standard formula for Riemannian Hessians on $\text{Gr}(n, r)$ and the first-order Taylor expansion of $\phi(\mathbf{P})$, respectively.

We next relate the eigenvalues of $\text{Hess } f(\mathbf{P})$ and $D\phi(\mathbf{P})$ when \mathbf{P} is a first-order critical point of f . In this case, \mathbf{P} is a fixed point of ϕ by **C3** in the proof of Theorem 6.1.

Lemma 6.7. Let \mathbf{P} be a first-order critical point (23). If $\text{Hess } f(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow T_{\mathbf{P}} \text{Gr}(n, r)$ has a positive eigenvalue, then $D\phi(\mathbf{P}) : T_{\mathbf{P}} \text{Gr}(n, r) \rightarrow T_{\mathbf{P}} \text{Gr}(n, r)$ has an eigenvalue that is greater than 1.

Lemma 6.7 implies that any first-order, but not second-order, critical point of f is an unstable fixed point of ϕ . The proof is in Appendix F. It is needed to express the Jacobian $D\phi(\mathbf{P})$ in terms of an eigenbasis for the Riemannian Hessian $\text{Hess } f(\mathbf{P})$.

The next lemma guarantees that the update map ϕ has a nice local property given a sufficiently small step size.

Lemma 6.8. There exists a constant $0 < \Gamma^{**}(\mathbf{X}, r) \leq \Gamma^*$ (cf. Theorem 6.1) such that if the step size satisfies $0 < \gamma \leq \Gamma^{**}$, then the update map $\phi : \text{Gr}(n, r) \rightarrow \text{Gr}(n, r)$ (21) is a local diffeomorphism.

The proof of Lemma 6.8 is in Appendix G. We show that ϕ is C^∞ and $D\phi(\mathbf{P})$ is invertible for all $\mathbf{P} \in \text{Gr}(n, r)$, when γ is small enough.

Proof of Theorem 6.2. Fix the step size $0 < \gamma \leq \Gamma^{**}$. From Theorem 6.1, the sequence $\{\mathbf{P}_t\}$ always converges to a first-order critical point of f (23), thus a fixed point of ϕ (21) by C3 in Section 6.2. In this proof, we consider the case when the limit $\mathbf{P} \in \text{Gr}(n, r)$ is a “bad” critical point, meaning it is a first-order critical point but fails the second-order criticality (25). The idea is to apply Theorem A.2 to show that the set of \mathbf{Q}_0 that converge to bad critical points has zero measure in $\text{St}(n, r)$.

Because ϕ is a local diffeomorphism by Lemma 6.8, we are eligible to apply Theorem A.2. It gives an open neighborhood $B_{\mathbf{P}} \subseteq \text{Gr}(n, r)$ around each bad critical point \mathbf{P} satisfying the conditions in Theorem A.2. As $\text{Gr}(n, r)$ is second-countable, it has the Lindelöf property, i.e. every open cover has a countable subcover. So, there exists a countable set $\mathcal{C} \subseteq \text{Gr}(n, r)$ consisting of some of the bad critical points such that

$$\bigcup_{\mathbf{P} \in \text{Gr}(n, r) \text{ is a bad critical point}} B_{\mathbf{P}} = \bigcup_{\mathbf{P} \in \mathcal{C}} B_{\mathbf{P}}. \quad (43)$$

Define the subset

$$\mathcal{J} := \bigcup_{\mathbf{P} \in \mathcal{C}} \bigcup_{t=0}^{\infty} \phi^{-t}(W_{\mathbf{P}}) \subseteq \text{Gr}(n, r). \quad (44)$$

We claim that

$$\{\mathbf{P}_0 \in \text{Gr}(n, r) : \phi^t(\mathbf{P}_0) \text{ converges to a bad critical point as } t \rightarrow \infty\} \subseteq \mathcal{J}. \quad (45)$$

To see this, assume $\lim_{t \rightarrow \infty} \phi^t(\mathbf{P}_0) = \mathbf{P}$ is bad. By (43), $\mathbf{P} \in B_{\mathbf{P}'}$ for some $\mathbf{P}' \in \mathcal{C}$. Since $B_{\mathbf{P}'}$ is open, there is $t_0 \geq 0$ such that $\phi^t(\mathbf{P}_0) \in B_{\mathbf{P}'}$ for all $t \geq t_0$. As \mathbf{P}' is a fixed point of ϕ , by (50) in Theorem A.2, we obtain $\phi^{t_0}(\mathbf{P}_0) \in W_{\mathbf{P}'}$. In other words, $\mathbf{P}_0 \in \phi^{-t_0}(W_{\mathbf{P}'})$. So, $\mathbf{P}_0 \in \mathcal{J}$ as claimed.

We claim that \mathcal{J} is measure-zero in $\text{Gr}(n, r)$. For each $\mathbf{P} \in \mathcal{C}$, at least one eigenvalue of $\text{Hess } f(\mathbf{P})$ is greater than 0. By Lemma 6.7, we know at least one eigenvalue of $D\phi(\mathbf{P})$ is greater than 1. By the first result of Theorem A.2, the dimension of $W_{\mathbf{P}}$ is strictly less than the dimension of $\text{Gr}(n, r)$. So $W_{\mathbf{P}}$ is measure-zero in $\text{Gr}(n, r)$ by [40, Corollary 6.12]. Given that the step size satisfies $0 < \gamma \leq \Gamma^{**}$, the map ϕ is a local diffeomorphism by Lemma 6.8, and so is ϕ^t for all $t \geq 0$. Thus their pre-images of measure-zero sets have zero measure. As $(\phi^t)^{-1} \equiv \phi^{-t}$, the set $\phi^{-t}(W_{\mathbf{P}})$ has zero measure for each $t \geq 0$. Then \mathcal{J} is a countable union of measure-zero sets, so has zero measure.

To conclude, define $\mathcal{I} \subseteq \text{St}(n, r)$ as the pre-image of $\mathcal{J} \subseteq \text{Gr}(n, r)$ under the map $\mathbf{Q} \mapsto \mathbf{Q}\mathbf{Q}^{\top}$. Since the map is a submersion from the Stiefel manifold onto the Grassmannian, \mathcal{I} has zero measure in $\text{St}(n, r)$ as \mathcal{J} does in $\text{Gr}(n, r)$. For all initializations \mathbf{Q}_0 in $\text{St}(n, r)$ but outside the null set \mathcal{I} , the sequence $\{\phi^t(\mathbf{P}_0)\} = \{\mathbf{P}_t\}$ converges to a second-order critical point of f if $0 < \gamma \leq \Gamma^{**}$. The proof of Theorem 6.2 is complete. \square

7 Conclusion and future work

In this paper, we developed the PGD method (Algorithm 1) for symmetric Tucker tensor decomposition. It has a simple update scheme compared to other iterative solvers running on Riemannian manifolds. We further designed the SPGD method (Algorithm 2) and SHOEVD solver (Algorithm 3) to decompose the sample moment tensor (1). The algorithms are free of constructing and storing moment tensors, and their iterations are conducted by streaming only a small subset of samples. Tremendous computational savings were seen in numerical experiments with minimal loss of accuracy. We also showed the applicability of the Tucker decomposition of moment tensors in comparison to the CP format, as well as to detect anomalies and allocate assets on real datasets. Finally, utilizing Riemannian manifold optimization theory, we derived theoretical guarantees for the PGD sequence on the Grassmannian. The sequence always converges to a first-order critical point, and almost surely to a second-order critical point.

There are a number of directions that would be interesting to study in the future.

1. **Implementation for cumulants.** Cumulant tensors are central to non-Gaussian data analysis. They are nonlinear combinations of moments. It would be useful to derive scalable implementations for sample cumulants.
2. **Extending the analysis.** It is certainly worth investigating convergence guarantees in the streaming setting with adaptive step sizes of Algorithms 2 and 3. We also want to explore under what conditions is the limiting point guaranteed to be a global optimizer.

3. **Estimating the core without fresh samples.** Currently, the core tensor \mathbf{C} is computed using independent data samples after the subspace \mathbf{Q} has been calculated. We would like to figure out a good way of updating \mathbf{C} while updating \mathbf{Q} , using the same data stream.

Acknowledgements

The authors are especially grateful to Dries Cornilly and Xiurui Geng for their help with the real data. We also thank Hemanth Kolla, Amit Singer and Mihai Sirbu for useful discussions. R.J. was supported by NSF DMS 1952735. J.K. was supported by start-up grants from the College of Natural Sciences and Oden Institute for Computational Engineering and Sciences at UT Austin. R.W. was supported by AFOSR MURI FA9550-19-1-0005, NSF DMS 1952735, NSF HDR-1934932 and NSF 2019844.

References

- [1] K. ADITYA, H. KOLLA, W. P. KEGELMEYER, T. M. SHEAD, J. LING, AND W. L. DAVIS, *Anomaly detection in scientific data using joint statistical moments*, Journal of Computational Physics, 387 (2019), pp. 522–538, <https://doi.org/10.1016/j.jcp.2019.03.003>.
- [2] S. AHMADI-ASL, S. ABUKHOVICH, M. G. ASANTE-MENSAH, A. CICHOCKI, A. H. PHAN, T. TANAKA, AND I. OSELEDETS, *Randomized algorithms for computation of Tucker decomposition and higher order SVD (HOSVD)*, IEEE Access, 9 (2021), pp. 28684–28706, <https://doi.org/10.1109/ACCESS.2021.3058103>.
- [3] O. ALTER AND G. H. GOLUB, *Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations*, Proceedings of the National Academy of Sciences, 102 (2005), pp. 17559–17564, <https://doi.org/10.1073/pnas.0509033102>.
- [4] A. ANANDKUMAR, R. GE, D. HSU, S. M. KAKADE, AND M. TELGARSKY, *Tensor decompositions for learning latent variable models*, Journal of Machine Learning Research, 15 (2014), pp. 2773–2832, <http://jmlr.org/papers/v15/anandkumar14b.html>.
- [5] O. E. BARNDORFF-NIELSEN, *Normal inverse Gaussian distributions and stochastic volatility modelling*, Scandinavian Journal of Statistics, 24 (1997), pp. 1–13, <https://doi.org/10.1111/1467-9469.00045>.
- [6] K. BOUDT, D. CORNILLY, AND T. VERDONCK, *Nearest comoment estimation with unobserved factors*, Journal of Econometrics, 217 (2020), pp. 381–397, <https://doi.org/10.1016/j.jeconom.2019.12.009>.
- [7] N. BOUMAL, *An introduction to optimization on smooth manifolds*, 2022, <http://www.nicolasboumal.net/book>. To be published by Cambridge University Press.
- [8] W. BRIEC, K. KERSTENS, AND O. JOKUNG, *Mean-variance-skewness portfolio performance gauging: A general shortage function and dual approach*, Management Science, 53 (2007), pp. 135–149, <https://doi.org/10.1287/mnsc.1060.0596>.
- [9] J.-F. CARDOSO, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, in Proceedings ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, 1991, pp. 3109–3112, vol.5, <https://doi.org/10.1109/ICASSP.1991.150113>.
- [10] J.-F. CARDOSO, *Blind signal separation: Statistical principles*, Proceedings of the IEEE, 86 (1998), pp. 2009–2025, <https://doi.org/10.1109/5.720250>.
- [11] M. CHE, Y. WEI, AND H. YAN, *The computation of low multilinear rank approximations of tensors via power scheme and random projection*, SIAM Journal on Matrix Analysis and Applications, 41 (2020), pp. 605–636, <https://doi.org/10.1137/19M1237016>.

- [12] P. COMON, *Independent component analysis, a new concept?*, Signal Processing, 36 (1994), pp. 287–314, [https://doi.org/10.1016/0165-1684\(94\)90029-9](https://doi.org/10.1016/0165-1684(94)90029-9).
- [13] L. DE LATHAUWER, *Signal Processing based on Multilinear Algebra*, PhD thesis, Katholieke Universiteit Leuven, 1997.
- [14] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *An introduction to independent component analysis*, Journal of Chemometrics, 14 (2000), pp. 123–149, [https://doi.org/10.1002/1099-128x\(200005/06\)14:3<123::aid-cem589>3.0.co;2-1](https://doi.org/10.1002/1099-128x(200005/06)14:3<123::aid-cem589>3.0.co;2-1).
- [15] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1253–1278, <https://doi.org/10.1137/S0895479896305696>.
- [16] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *On the best rank-1 and rank- (R_1, R_2, \dots, R_n) approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1324–1342, <https://doi.org/10.1137/S0895479898346995>.
- [17] V. DE SILVA AND L.-H. LIM, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal of Matrix Analysis and Applications, 30 (2008), pp. 1084–1127, <https://doi.org/10.1137/06066518X>.
- [18] K. DOMINO, P. GAWRON, AND L. PAWELA, *Efficient computation of higher-order cumulant tensors*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1590–A1610, <https://doi.org/10.1137/17M1149365>.
- [19] J. DUCHI, E. HAZAN, AND Y. SINGER, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159, <http://jmlr.org/papers/v12/duchi11a.html>.
- [20] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218, <https://doi.org/10.1007/BF02288367>.
- [21] L. ELDÉN AND B. SAVAS, *A Newton-Grassmann method for computing the best multilinear rank- (r_1, r_2, r_3) approximation of a tensor*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 248–271, <https://doi.org/10.1137/070688316>.
- [22] E. F. FAMA AND K. R. FRENCH, *Common risk factors in the returns on stocks and bonds*, Journal of Financial Economics, 33 (1993), pp. 3–56, [https://doi.org/10.1016/0304-405X\(93\)90023-5](https://doi.org/10.1016/0304-405X(93)90023-5).
- [23] J. H. FRIEDMAN, *Exploratory projection pursuit*, Journal of the American Statistical Association, 82 (1987), pp. 249–266, <https://doi.org/10.1080/01621459.1987.10478427>.
- [24] R. GE, Q. HUANG, AND S. M. KAKADE, *Learning mixtures of Gaussians in high dimensions*, in Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, 2015, pp. 761–770, <https://doi.org/10.1145/2746539.2746616>.
- [25] X. GENG, K. SUN, L. JI, AND Y. ZHAO, *A high-order statistical tensor based algorithm for anomaly detection in hyperspectral imagery*, Scientific Reports, 4 (2014), p. 6869, <https://doi.org/10.1038/srep06869>.
- [26] W. HACKBUSCH, *Numerical tensor calculus*, Acta Numerica, 23 (2014), pp. 651–742, <https://doi.org/10.1017/S0962492914000087>.
- [27] U. HELMKE, K. HÜPER, AND J. TRUMPF, *Newton’s method on Grassmann manifolds*, arXiv preprint arXiv:0709.2205, (2007), <https://arxiv.org/abs/0709.2205>.
- [28] A. HENRIKSEN AND R. WARD, *AdaOja: Adaptive learning rates for streaming PCA*, arXiv preprint arXiv:1905.12115, (2019), <https://arxiv.org/abs/1905.12115>.

- [29] C. J. HILLAR AND L.-H. LIM, *Most tensor problems are NP-hard*, Journal of the ACM (JACM), 60 (2013), pp. 1–39, <https://doi.org/10.1145/2512329>.
- [30] A. HYVÄRINEN AND E. OJA, *Independent component analysis: Algorithms and applications*, Neural Networks, 13 (2000), pp. 411–430, [https://doi.org/10.1016/S0893-6080\(00\)00026-5](https://doi.org/10.1016/S0893-6080(00)00026-5).
- [31] M. ISHTEVA, P.-A. ABSIL, AND P. VAN DOOREN, *Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors*, SIAM Journal on Matrix Analysis and Applications, 34 (2013), pp. 651–672, <https://doi.org/10.1137/11085743X>.
- [32] M. ISHTEVA, P.-A. ABSIL, S. VAN HUFFEL, AND L. DE LATHAUWER, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 115–135, <https://doi.org/10.1137/090764827>.
- [33] M. ISHTEVA, L. DE LATHAUWER, P.-A. ABSIL, AND S. HUFFEL, *Differential-geometric Newton method for the best rank- (r_1, r_2, r_3) approximation of tensors*, Numerical Algorithms, 51 (2009), pp. 179–194, <https://doi.org/10.1007/s11075-008-9251-2>.
- [34] E. JONDEAU AND M. ROCKINGER, *Optimal portfolio allocation under higher moments*, European Financial Management, 12 (2006), pp. 29–55, <https://doi.org/10.1111/j.1354-7798.2006.00309.x>.
- [35] T. G. KOLDA, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM Journal on Matrix Analysis and Applications, 24 (2003), pp. 762–767, <https://doi.org/10.1137/S0895479801394465>.
- [36] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500, <https://doi.org/10.1137/07070111X>.
- [37] T. G. KOLDA, B. W. BADER, AND ET AL., *Tensor toolbox for matlab v. 3.2.1*, 2021, <https://www.tensortoolbox.org>.
- [38] S. KRANTZ AND H. PARKS, *A Primer of Real Analytic Functions*, Birkhauser, 2002, <https://doi.org/10.1007/978-0-8176-8134-0>.
- [39] J. D. LEE, I. PANAGEAS, G. PILIOURAS, M. SIMCHOWITZ, M. I. JORDAN, AND B. RECHT, *First-order methods almost always avoid strict saddle points*, Mathematical Programming, 176 (2019), pp. 311–337, <https://doi.org/10.1007/s10107-019-01374-3>.
- [40] J. M. LEE, *Introduction to Smooth Manifolds*, Springer, 2012, <https://doi.org/10.1007/978-1-4419-9982-5>.
- [41] J. LI, K. USEVICH, AND P. COMON, *Globally convergent Jacobi-type algorithms for simultaneous orthogonal symmetric tensor diagonalization*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 1–22, <https://doi.org/10.1137/17M1116295>.
- [42] Y. LI, H. L. NGUYEN, AND D. P. WOODRUFF, *Turnstile streaming algorithms might as well be linear sketches*, in Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, 2014, pp. 174–183, <https://doi.org/10.1145/2591796.2591812>.
- [43] O. A. MALIK AND S. BECKER, *Low-rank Tucker decomposition of large tensors using TensorSketch*, in Advances in Neural Information Processing Systems, vol. 31, 2018, <https://proceedings.neurips.cc/paper/2018/file/45a766fa266ea2eb6680fa139d2a3d-Paper.pdf>.
- [44] K. V. MARDIA, *Measures of multivariate skewness and kurtosis with applications*, Biometrika, 57 (1970), pp. 519–530, <https://doi.org/10.1093/biomet/57.3.519>.
- [45] R. MINSTER, A. K. SAIBABA, AND M. E. KILMER, *Randomized algorithms for low-rank tensor decompositions in the Tucker format*, SIAM Journal on Mathematics of Data Science, 2 (2020), pp. 189–215, <https://doi.org/10.1137/19M1261043>.

- [46] I. MITLIAGKAS, C. CARAMANIS, AND P. JAIN, *Memory limited, streaming PCA*, in *Advances in Neural Information Processing Systems*, vol. 26, 2013, <https://proceedings.neurips.cc/paper/2013/file/76cf99d3614e23eabab16fb27e944bf9-Paper.pdf>.
- [47] A. MONTANARI AND E. RICHARD, *A statistical model for tensor pca*, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, Cambridge, MA, USA, 2014, MIT Press, p. 2897–2905.
- [48] A. MOOIJART, *Factor analysis for non-normal variables*, *Psychometrika*, 50 (1985), pp. 323–342, <https://doi.org/10.1007/BF02294108>.
- [49] J. MORTON AND L.-H. LIM, *Principal cumulant component analysis*. Preprint, 2009, <http://galton.uchicago.edu/~lekheng/work/pcca.pdf>.
- [50] S. MUTHUKRISHNAN, *Data streams: Algorithms and applications*, *Foundations and Trends in Theoretical Computer Science*, 1 (2005), pp. 117–236, <https://doi.org/10.1561/04000000002>.
- [51] E. OJA, *Simplified neuron model as a principal component analyzer*, *Journal of Mathematical Biology*, 15 (1982), pp. 267–273, <https://doi.org/10.1007/BF00275687>.
- [52] J. M. PEREIRA, J. KILEEL, AND T. G. KOLDA, *Tensor moments of Gaussian mixture models: Theory and applications*, arXiv preprint arXiv:2202.06930, (2022), <https://arxiv.org/abs/2202.06930>.
- [53] A. PLAZA, J. A. BENEDIKTSSON, J. W. BOARDMAN, J. BRAZILE, L. BRUZZONE, G. CAMPS-VALLS, J. CHANUSSOT, M. FAUVEL, P. GAMBA, A. GUALTIERI, M. MARCONCINI, J. C. TILTON, AND G. TRIANNI, *Recent advances in techniques for hyperspectral image processing*, *Remote Sensing of Environment*, 113 (2009), pp. S110–S122, <https://doi.org/10.1016/j.rse.2007.07.028>. *Imaging Spectroscopy Special Issue*.
- [54] P. A. REGALIA, *Monotonically convergent algorithms for symmetric tensor approximation*, *Linear Algebra and its Applications*, 438 (2013), pp. 875–890, <https://doi.org/10.1016/j.laa.2011.10.033>.
- [55] V. RENGANATHAN, N. HASHEMI, J. RUTHS, AND T. H. SUMMERS, *Higher-order moment-based anomaly detection*, *IEEE Control Systems Letters*, 6 (2022), pp. 211–216, <https://doi.org/10.1109/LCSYS.2021.3058269>.
- [56] B. SAVAS, *Best low rank tensor approximation*, 2009, <https://liu.se/en/employee/bersa48>.
- [57] B. SAVAS AND L.-H. LIM, *Quasi-Newton methods on Grassmannians and multilinear approximations of tensors*, *SIAM Journal on Scientific Computing*, 32 (2010), pp. 3352–3393, <https://doi.org/10.1137/090763172>.
- [58] R. SCHNEIDER AND A. USCHMAJEW, *Convergence results for projected line-search methods on varieties of low-rank matrices via Lojasiewicz inequality*, *SIAM Journal on Optimization*, 25 (2015), pp. 622–646, <https://doi.org/10.1137/140957822>.
- [59] S. SHERMAN AND T. G. KOLDA, *Estimating higher-order moments using symmetric tensor decomposition*, *SIAM Journal on Matrix Analysis and Applications*, 41 (2020), pp. 1369–1387, <https://doi.org/10.1137/19m1299633>.
- [60] M. SHUB, *Global Stability of Dynamical Systems*, Springer, 1987, <https://doi.org/10.1007/978-1-4757-1947-5>.
- [61] C. SPEARMAN, *The proof and measurement of association between two things*, *The American Journal of Psychology*, 15 (1904), pp. 72–101, <https://doi.org/10.2307/1422689>. Republished in 1987.
- [62] Y. SUN, Y. GUO, C. LUO, J. TROPP, AND M. UDELL, *Low-rank Tucker approximation of a tensor from streaming data*, *SIAM Journal on Mathematics of Data Science*, 2 (2020), pp. 1123–1150, <https://doi.org/10.1137/19M1257718>.

- [63] J. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Streaming low-rank matrix approximation with an application to scientific simulation*, SIAM Journal on Scientific Computing, 41 (2019), pp. A2430–A2463, <https://doi.org/10.1137/18M1201068>.
- [64] N. VANNIEUWENHOVEN, R. VANDEBRIL, AND K. MEERBERGEN, *A new truncation strategy for the higher-order singular value decomposition*, SIAM Journal on Scientific Computing, 34 (2012), pp. 1027–1052, <https://doi.org/10.1137/110836067>.
- [65] N. VERVLIIET, O. DEBALS, L. SORBER, M. VAN BAREL, AND L. DE LATHAUWER, *Tensorlab 3.0*, 3 2016, <https://www.tensorlab.net>.
- [66] S. F. WALTER, L. LEHMANN, AND R. LAMOUR, *On evaluating higher-order derivatives of the QR decomposition of tall matrices with full column rank in forward and reverse mode algorithmic differentiation*, Optimization Methods and Software, 27 (2012), pp. 391–403, <https://doi.org/10.1080/10556788.2011.610454>.
- [67] P. YANG, C.-J. HSIEH, AND J.-L. WANG, *History PCA: A new algorithm for streaming PCA*, arXiv preprint arXiv:1802.05447, (2018), <https://arxiv.org/abs/1802.05447>.
- [68] R. ZIMMERMANN, *Hermite interpolation and data processing errors on Riemannian matrix manifolds*, SIAM Journal on Scientific Computing, 42 (2020), pp. A2593–A2619, <https://doi.org/10.1137/19M1282878>.

A Supporting theorems

A.1 Convergence guarantee

Let $f : \text{Gr}(n, r) \rightarrow \mathbb{R}$ be a real-analytic function. Consider the optimization $\max_{\mathbf{P} \in \text{Gr}(n, r)} f(\mathbf{P})$. Let $\{\mathbf{P}_t\} \subseteq \text{Gr}(n, r)$ be a sequence. We make the following assumptions:

- (C1) There exists $\kappa > 0$ such that

$$\|\mathbf{P}_{t+1} - \mathbf{P}_t\| \geq \kappa \|\text{grad} f(\mathbf{P}_t)\|; \quad (46)$$

- (C2) There exists $\sigma > 0$ such that

$$f(\mathbf{P}_{t+1}) - f(\mathbf{P}_t) \geq \sigma \|\mathbf{P}_{t+1} - \mathbf{P}_t\| \|\text{grad} f(\mathbf{P}_t)\|; \quad (47)$$

- (C3) First-order optimality is equivalent to the fixed point condition, i.e.

$$\text{grad} f(\mathbf{P}_t) = \mathbf{0} \Leftrightarrow \mathbf{P}_{t+1} = \mathbf{P}_t; \quad (48)$$

- (C4) [Lojasiewicz inequality] There exist $\delta, \rho > 0$ and $\theta \in (0, \frac{1}{2}]$ such that for all first-order critical points $\mathbf{P} \in \text{Gr}(n, r)$ of f (23) and all points $\mathbf{P}' \in \text{Gr}(n, r)$ with $\|\mathbf{P}' - \mathbf{P}\| \leq \delta$,

$$|f(\mathbf{P}') - f(\mathbf{P})|^{1-\theta} \leq \rho \|\text{grad} f(\mathbf{P}')\|. \quad (49)$$

Theorem A.1 (Theorem 2.3 in [58]). *Under the above set-up, if $\{\mathbf{P}_t\}$ satisfies conditions C1 - C4, it converges monotonically to a first-order critical point of f at no less than an algebraic rate.*

A.2 Center-stable manifold theorem

Theorem A.2 (Theorem III.7(2) in [60]). *Let $\phi : \text{Gr}(n, r) \rightarrow \text{Gr}(n, r)$ be a local diffeomorphism. Let $\mathbf{P} \in \text{Gr}(n, r)$ be a fixed point of ϕ . There exist an open neighborhood $B_{\mathbf{P}} \subseteq \text{Gr}(n, r)$ of \mathbf{P} and a smoothly embedded disk $W_{\mathbf{P}} \subseteq \text{Gr}(n, r)$ containing \mathbf{P} such that*

1. *the dimension of $W_{\mathbf{P}}$ is the number of linearly independent eigenmatrices of $D\phi(\mathbf{P})$ with eigenvalues of magnitude no more than 1;*
2. *and*

$$\{\mathbf{X} \in \text{Gr}(n, r) : \phi^t(\mathbf{X}) \in B_{\mathbf{P}}, \forall t \geq 0\} \subseteq W_{\mathbf{P}}. \quad (50)$$

Remark 9. *Although the center-stable manifold statement is discussed for Euclidean space in [60], as noted in [39, Theorem 1], it can be extended to any manifold. We specify the manifold to be the Grassmannian in this paper.*

In this supplemental document, we present the detailed proofs for all of the supporting propositions and lemmas in [Section 6](#).

B Proof of [Proposition 6.3](#)

Proof. We first compute the Riemannian gradient, $\text{grad} f(\mathbf{P}) \in T_{\mathbf{P}} \text{Gr}(n, r)$. It is the orthogonal projection of the Euclidean gradient $\nabla f(\mathbf{P}) \in \mathbb{R}^{n \times r}$ onto the tangent space at \mathbf{P} . The projection $\pi_{\mathbf{P}} : \mathbb{S}^2(\mathbb{R}^n) \rightarrow T_{\mathbf{P}} \text{Gr}(n, r)$ is computed as follows [27, Proposition 2.1]:

$$\pi_{\mathbf{P}}(\mathbf{X}) = \text{ad}_{\mathbf{P}}^2(\mathbf{X}) \in T_{\mathbf{P}} \text{Gr}(n, r) \text{ with } \text{ad}_{\mathbf{P}}^2 = \text{ad}_{\mathbf{P}} \circ \text{ad}_{\mathbf{P}} \text{ and } \text{ad}_{\mathbf{P}}(\mathbf{X}) = \mathbf{P}\mathbf{X} - \mathbf{X}\mathbf{P}$$

for $\mathbf{X} \in \mathbb{S}^2(\mathbb{R}^n)$. Here, ad stands for the adjoint operator.

The Euclidean gradient of f (22) is

$$\begin{aligned}\nabla f(\mathbf{P}) &= d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}), \mathbf{X} \cdot (\mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \\ &\quad + d \langle \mathbf{X} \cdot (\mathbf{P}, \dots, \mathbf{P}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \\ &= w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P}).\end{aligned}\tag{51}$$

Hence the Riemannian gradient equals

$$\begin{aligned}\text{grad } f(\mathbf{P}) &= \text{ad}_{\mathbf{P}}^2(\nabla f(\mathbf{P})) \\ &= \text{ad}_{\mathbf{P}}(\mathbf{P}\nabla f(\mathbf{P}) - \nabla f(\mathbf{P})\mathbf{P}) \\ &= \mathbf{P}(\mathbf{P}\nabla f(\mathbf{P}) - \nabla f(\mathbf{P})\mathbf{P}) - (\mathbf{P}\nabla f(\mathbf{P}) - \nabla f(\mathbf{P})\mathbf{P})\mathbf{P} \\ &= \mathbf{P}(\nabla f(\mathbf{P})) - 2\mathbf{P}(\nabla f(\mathbf{P}))\mathbf{P} + (\nabla f(\mathbf{P}))\mathbf{P}\end{aligned}$$

Substituting (51) into the above equation,

$$\begin{aligned}\text{grad } f(\mathbf{P}) &= \mathbf{P}(w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P})) \\ &\quad - 2\mathbf{P}(w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P}))\mathbf{P} + (w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P}))\mathbf{P} \\ &= \mathbf{P}w(\mathbf{P})(\mathbf{I}_n - \mathbf{P}) + (\mathbf{I}_n - \mathbf{P})w(\mathbf{P})\mathbf{P} = \text{sym}\left(2(\mathbf{I}_n - \mathbf{P})w(\mathbf{P})\mathbf{P}\right).\end{aligned}\tag{52}$$

Next, we relate the gradients of $f(\mathbf{P})$ and $F(\mathbf{Q})$. The Euclidean gradient of F is

$$\begin{aligned}\nabla F(\mathbf{Q}) &= 2d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{Q}, \dots, \mathbf{Q}) \rangle_{-1} \\ &= 2d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \mathbf{Q} \\ &= 2d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \mathbf{Q} = 2w(\mathbf{P})\mathbf{Q}.\end{aligned}\tag{53}$$

Therefore,

$$(\mathbf{I}_n - \mathbf{Q}\mathbf{Q}^\top) \nabla F(\mathbf{Q})\mathbf{Q}^\top = 2(\mathbf{I}_n - \mathbf{P})w(\mathbf{P})\mathbf{Q}\mathbf{Q}^\top = 2(\mathbf{I}_n - \mathbf{P})w(\mathbf{P})\mathbf{P}.$$

By (52), we have

$$\text{grad } f(\mathbf{P}) = \text{sym}\left((\mathbf{I}_n - \mathbf{Q}\mathbf{Q}^\top) \nabla F(\mathbf{Q})\mathbf{Q}^\top\right).$$

The proof of Proposition 6.3 is complete. \square

C Proof of Lemma 6.4

Proof. We derive the update on \mathbf{P}_t (21) based on the iterate \mathbf{Q}_t (20). We decompose the update direction $\gamma_t \nabla F(\mathbf{Q}_t)$ into two parts respectively in $\text{colspan}(\mathbf{Q}_t)$ and $\text{colspan}(\mathbf{Q}_t^\perp)$, i.e.

$$\gamma_t \nabla F(\mathbf{Q}_t) = \gamma_t \mathbf{Q}_t \mathbf{Q}_t^\top \nabla F(\mathbf{Q}_t) + \gamma_t (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t).$$

We approximate $\mathbf{Q}_{t+1} = \text{orth}(\mathbf{Q}_t + \gamma_t \nabla F(\mathbf{Q}_t))$ by the first-order Taylor expansion of orth centered at $\mathbf{Q}_t + \gamma_t \mathbf{Q}_t \mathbf{Q}_t^\top \nabla F(\mathbf{Q}_t)$:

$$\begin{aligned}\mathbf{Q}_{t+1} &= \text{orth}\left((\mathbf{Q}_t + \gamma_t \mathbf{Q}_t \mathbf{Q}_t^\top \nabla F(\mathbf{Q}_t)) + \gamma_t (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t)\right) \\ &= \widetilde{\mathbf{Q}}_t + \text{Dorth}(\widetilde{\mathbf{Q}}_t)[\gamma_t (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t)] \\ &\quad + \mathcal{O}(\gamma_t^2 \|(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t)\|^2),\end{aligned}$$

where

$$\widetilde{\mathbf{Q}}_t \mathbf{R} = \mathbf{Q}_t + \gamma_t \mathbf{Q}_t \mathbf{Q}_t^\top \nabla F(\mathbf{Q}_t) \in \mathbb{R}^{n \times r}\tag{54}$$

denotes the QR factorization. (Note this quantity is full-rank according to C3 in Section 6.2 and $\text{orth} : \text{St}(n, r) \rightarrow \text{St}(n, r)$ is C^∞ at full-rank inputs [66].)

By the derivative formula of the QR factorization given in [68, Algorithm 1] (originally [66]),

$$\begin{aligned}\mathbf{Q}_{t+1} &= \widetilde{\mathbf{Q}}_t + \gamma_t (\mathbf{I}_n - \widetilde{\mathbf{Q}}_t \widetilde{\mathbf{Q}}_t^\top) (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \\ &\quad + \widetilde{\mathbf{Q}}_t s\left(\gamma_t \widetilde{\mathbf{Q}}_t^\top (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1}\right) \\ &\quad + \mathcal{O}(\gamma_t^2 \|(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t)\|^2) \\ &= \widetilde{\mathbf{Q}}_t + \gamma_t (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} + \mathcal{O}(\gamma_t^2 \|(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t)\|^2).\end{aligned}\tag{55}$$

To see the second equality, note $\mathbf{I}_n - \widetilde{\mathbf{Q}}_t \widetilde{\mathbf{Q}}_t^\top = \mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top$, because $\widetilde{\mathbf{Q}}_t$ and \mathbf{Q}_t are both the orthonormal bases for $\text{colspan}(\mathbf{Q}_t)$. Moreover, given the linear operator $s : \mathbb{R}^{r \times r} \rightarrow \mathbb{R}^{r \times r}$ ⁵ the term $s\left(\gamma_t \widetilde{\mathbf{Q}}_t^\top (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1}\right) = \mathbf{0}$ as $\widetilde{\mathbf{Q}}_t^\top (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) = \mathbf{0}$.

From (55),

$$\begin{aligned} \mathbf{P}_{t+1} - \mathbf{P}_t &= \mathbf{Q}_{t+1} \mathbf{Q}_{t+1}^\top - \widetilde{\mathbf{Q}}_t \widetilde{\mathbf{Q}}_t^\top \\ &= 2\gamma_t \text{sym} \left((\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top \right) \\ &\quad + \mathcal{O} \left(\gamma_t^2 \left\| (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \right\|^2 \right) \\ &\quad + \mathcal{O} \left(\gamma_t^2 \left\| \text{sym} \left((\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{Q}_t^\top \right) \right\|^2 \right). \end{aligned} \tag{56}$$

Here the second line refers to the symmetric term

$$\gamma_t^2 (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \mathbf{R}^{-\top} \nabla F(\mathbf{Q}_t)^\top (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top),$$

whose norm is less than $\gamma_t^2 \left\| (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \right\|^2$.

We transform the right-hand side of (56) from an expression in terms of \mathbf{Q} to one with \mathbf{P} .

For the first component of (56), we separate $(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top$ into two parts:

$$\text{Factor 1} := (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{Q}_t^\top, \quad \text{Factor 2} := \mathbf{Q}_t \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top,$$

using $\mathbf{Q}_t^\top \mathbf{Q}_t = \mathbf{I}_r$. By (53),

$$\text{Factor 1} = 2(\mathbf{I}_n - \mathbf{P}_t) w(\mathbf{P}_t) \mathbf{Q}_t \mathbf{Q}_t^\top = 2(\mathbf{I}_n - \mathbf{P}_t) w(\mathbf{P}_t) \mathbf{P}_t. \tag{57}$$

Also,

$$\text{Factor 2} = \mathbf{Q}_t \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top = (\mathbf{Q}_t^\top)^\dagger \mathbf{R}^{-1} (\widetilde{\mathbf{Q}}_t)^\dagger = \left(\widetilde{\mathbf{Q}}_t \mathbf{R} \mathbf{Q}_t^\top \right)^\dagger. \tag{58}$$

by the fact that pseudoinversion satisfies $\mathbf{X}^\dagger \mathbf{Y}^\dagger = (\mathbf{Y}\mathbf{X})^\dagger$ if \mathbf{X} and \mathbf{Y} respectively have orthonormal rows and columns. Inserting (54) gives

$$\text{Factor 2} = (\mathbf{Q}_t \mathbf{Q}_t^\top + \gamma_t \mathbf{Q}_t \mathbf{Q}_t^\top \nabla F(\mathbf{Q}_t) \mathbf{Q}_t^\top)^\dagger = (\mathbf{P}_t + 2\gamma_t \mathbf{P}_t w(\mathbf{P}_t) \mathbf{P}_t)^\dagger = \alpha_t (\mathbf{P}_t)^\dagger,$$

where the last equality is by (53) and the definition of α_t (28). Moreover,

$$\text{Factor 1}^\top \times \text{Factor 2} = \mathbf{Q}_t \nabla F(\mathbf{Q}_t)^\top \underbrace{(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \mathbf{Q}_t}_{\mathbf{0}} \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top = \mathbf{0}.$$

Combining (57) and (58), $(\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top$ equals:

$$\begin{aligned} \text{Factor 1} \times \text{Factor 2} &= 2(\mathbf{I}_n - \mathbf{P}_t) w(\mathbf{P}_t) \mathbf{P}_t \alpha_t (\mathbf{P}_t)^\dagger \\ &= (\text{Factor 1} + \text{Factor 1}^\top) \times \text{Factor 2} = 2 \text{sym} \left(2(\mathbf{I}_n - \mathbf{P}_t) w(\mathbf{P}_t) \mathbf{P}_t \right) \alpha_t (\mathbf{P}_t)^\dagger \\ &= 2 \text{grad } f(\mathbf{P}_t) \alpha_t (\mathbf{P}_t)^\dagger. \end{aligned} \tag{59}$$

The formula of $\text{grad } f$ (52) is used in the last equality. Therefore, the first component of (56) reads

$$2\gamma_t \text{sym} \left((\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top \right) = 4\gamma_t \text{sym} \left(\text{grad } f(\mathbf{P}_t) \alpha_t (\mathbf{P}_t)^\dagger \right). \tag{60}$$

As for second component of (56), note

$$\begin{aligned} \left\| (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \right\|^2 &= \left\| (\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{R}^{-1} \widetilde{\mathbf{Q}}_t^\top \right\|^2 \\ &= 4 \left\| \text{grad } f(\mathbf{P}_t) \alpha_t (\mathbf{P}_t)^\dagger \right\|^2. \end{aligned}$$

⁵The operator s is a composition of element-wise multiplication and skew-symmetrization. See [68] Algorithm 1] for details.

The orthogonality of $\widetilde{\mathbf{Q}}_t$, which preserves norms, gives the first equality. The second is from (59).
The third component of (56) is dealt with directly by (27),

$$\left\| \text{sym} \left((\mathbf{I}_n - \mathbf{Q}_t \mathbf{Q}_t^\top) \nabla F(\mathbf{Q}_t) \mathbf{Q}_t^\top \right) \right\|^2 = \|\text{grad } f(\mathbf{P}_t)\|^2.$$

Thus, (56) consists of the main component (60) and the residuals

$\mathcal{O} \left(\gamma_t^2 \|\text{grad } f(\mathbf{P}_t) \alpha_t(\mathbf{P}_t)^\dagger\|^2 \right)$, $\mathcal{O} \left(\gamma_t^2 \|\text{grad } f(\mathbf{P}_t)\|^2 \right)$ as needed in (29). This completes the proof of Lemma 6.4 \square

D Proof of Lemma 6.5

Proof. Consider the step size $\gamma_t > 0$ and projector $\mathbf{P} \in \text{Gr}(n, r)$. Definition (28) reads

$$\alpha_t(\mathbf{P}) = \mathbf{P} + 2\gamma_t \mathbf{P} w(\mathbf{P}) \mathbf{P} = \mathbf{P} (\mathbf{I}_n + 2\gamma_t w(\mathbf{P})) \mathbf{P}. \quad (61)$$

So $\text{colspan}(\alpha_t(\mathbf{P})) \subseteq \text{colspan}(\mathbf{P})$. Also for any vector $\mathbf{x} \in \text{colspan}(\mathbf{P})$,

$$\mathbf{x}^\top \alpha_t(\mathbf{P}) \mathbf{x} = \mathbf{x}^\top \mathbf{P} (\mathbf{I}_n + 2\gamma_t w(\mathbf{P})) \mathbf{P} \mathbf{x} = \mathbf{x}^\top (\mathbf{I}_n + 2\gamma_t w(\mathbf{P})) \mathbf{x} \geq \|\mathbf{x}\|^2,$$

as $\mathbf{P} \mathbf{x} = \mathbf{x}$ and $\mathbf{I}_n + 2\gamma_t w(\mathbf{P}) \succeq \mathbf{I}_n$. It follows $\text{colspan}(\alpha_t(\mathbf{P})) = \text{colspan}(\mathbf{P})$, and $\alpha_t(\mathbf{P})$ has only r non-zero singular values, each greater than 1. By properties of pseudoinverses,

$$\alpha_t(\mathbf{P})^\dagger \alpha_t(\mathbf{P}) = \text{Proj}_{\text{range}(\alpha_t(\mathbf{P}))} = \mathbf{P}, \quad (62)$$

and $\alpha_t(\mathbf{P})^\dagger$ has only r non-zero singular values, each less than 1. Hence

$$\|\alpha_t(\mathbf{P})^\dagger\| \leq \sqrt{r}. \quad (63)$$

Constrain the step size by

$$0 < \gamma_t \leq \Gamma_1 := \frac{1}{4\sqrt{r} \sup_{\mathbf{P} \in \text{Gr}(n, r)} \|\mathbf{P} w(\mathbf{P}) \mathbf{P}\|}, \quad (64)$$

where $\sup_{\mathbf{P} \in \text{Gr}(n, r)} \|\mathbf{P} w(\mathbf{P}) \mathbf{P}\|$ is finite because the function $\mathbf{P} \mapsto \mathbf{P} w(\mathbf{P}) \mathbf{P}$ is continuous and $\text{Gr}(n, r)$ is compact. (We exclude the trivial case $\mathbf{P} w(\mathbf{P}) \mathbf{P} \equiv 0$ when $\mathfrak{X} = 0$.) From (62),

$$\mathbf{P} - \alpha_t(\mathbf{P})^\dagger = \alpha_t(\mathbf{P})^\dagger \alpha_t(\mathbf{P}) - \alpha_t(\mathbf{P})^\dagger \mathbf{P} = -\alpha_t(\mathbf{P})^\dagger (\mathbf{P} - \alpha_t(\mathbf{P})). \quad (65)$$

Then by (63) and (64),

$$\begin{aligned} \|\mathbf{P} - \alpha_t(\mathbf{P})^\dagger\| &= \|-\alpha_t(\mathbf{P})^\dagger (\mathbf{P} - \alpha_t(\mathbf{P}))\| \\ &\leq \|\alpha_t(\mathbf{P})^\dagger\| \|\mathbf{P} - \alpha_t(\mathbf{P})\| \\ &\leq 2\gamma_t \sqrt{r} \|\mathbf{P} w(\mathbf{P}) \mathbf{P}\| \leq \frac{1}{2}. \end{aligned} \quad (66)$$

We now compare $\|\text{grad } f(\mathbf{P})\|$ and $\|\text{grad } f(\mathbf{P}) \alpha_t(\mathbf{P})^\dagger\|$. By (27),

$$\text{grad } f(\mathbf{P}) \mathbf{P} = (\mathbf{I}_n - \mathbf{P}) w(\mathbf{P}) \mathbf{P}.$$

Inserting the above equation into (27),

$$\text{grad } f(\mathbf{P}) = \text{sym} \left(2(\mathbf{I}_n - \mathbf{P}) w(\mathbf{P}) \mathbf{P} \right) = \text{sym} \left(2 \text{grad } f(\mathbf{P}) \mathbf{P} \right).$$

Since

$$\langle \text{grad } f(\mathbf{P}) \mathbf{P}, \mathbf{P} \text{grad } f(\mathbf{P}) \rangle = \text{trace} \left((\mathbf{I}_n - \mathbf{P}) w(\mathbf{P}) \underbrace{\mathbf{P} (\mathbf{I}_n - \mathbf{P})}_{\mathbf{0}} w(\mathbf{P}) \mathbf{P} \right) = 0,$$

we have

$$\|\text{grad } f(\mathbf{P})\| = \sqrt{2} \|\text{grad } f(\mathbf{P})\mathbf{P}\|. \quad (67)$$

Then due to triangle inequality and (65),

$$\begin{aligned} \|\text{grad } f(\mathbf{P})\| &\leq \sqrt{2} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\| + \sqrt{2} \|\text{grad } f(\mathbf{P}) (\mathbf{P} - \alpha_t(\mathbf{P})^\dagger)\| \\ &= \sqrt{2} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\| + \sqrt{2} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger (\mathbf{P} - \alpha_t(\mathbf{P})^\dagger)\| \\ &\leq \sqrt{2} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\| (1 + \|\mathbf{P} - \alpha_t(\mathbf{P})^\dagger\|) \\ &= \frac{3\sqrt{2}}{2} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\|. \end{aligned}$$

Inserting (66) shows the last inequality. Thus,

$$\|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\| \geq \frac{\sqrt{2}}{3} \|\text{grad } f(\mathbf{P})\|.$$

On the other hand, as $\alpha_t(\mathbf{P})^\dagger = \mathbf{P}\alpha_t(\mathbf{P})^\dagger$, it holds

$$\begin{aligned} \|\text{grad } f(\mathbf{P})\alpha_t(\mathbf{P})^\dagger\| &\leq \|\text{grad } f(\mathbf{P})\mathbf{P}\| \|\alpha_t(\mathbf{P})^\dagger\| \\ &\leq \frac{\sqrt{2}r}{2} \|\text{grad } f(\mathbf{P})\|. \end{aligned}$$

The results (67) and (63) give the last derivation. Combining the above two inequalities completes the proof of Lemma 6.5. \square

E Proof of Proposition 6.6

Proof. Let $\mathbf{P} \in \text{Gr}(n, r)$ and $\Delta\mathbf{P} \in \text{T}_{\mathbf{P}} \text{Gr}(n, r)$. We first show the computation of the Riemannian Hessian. Applying [27, Equation (2.109)],

$$\begin{aligned} &\text{Hess } f(\mathbf{P})[\Delta\mathbf{P}] \\ &= \text{ad}_{\mathbf{P}}^2 (\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}]) - \text{ad}_{\mathbf{P}} \text{ad}_{\nabla f(\mathbf{P})}(\Delta\mathbf{P}) \\ &= \text{ad}_{\mathbf{P}} \left(\mathbf{P}\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}] - (\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} - \nabla f(\mathbf{P})\Delta\mathbf{P} + \Delta\mathbf{P}\nabla f(\mathbf{P}) \right) \\ &= \mathbf{P}(\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}])(\mathbf{I}_n - \mathbf{P}) + (\mathbf{I}_n - \mathbf{P})(\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} \\ &\quad - \mathbf{P}\nabla f(\mathbf{P})\Delta\mathbf{P} + \mathbf{P}\Delta\mathbf{P}\nabla f(\mathbf{P}) + \nabla f(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}\nabla f(\mathbf{P})\mathbf{P} \\ &= 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P})(\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} + \nabla f(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}\nabla f(\mathbf{P})\mathbf{P} \right). \end{aligned} \quad (68)$$

Recall (51) that $\nabla f(\mathbf{P}) = w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P})$. We differentiate $\nabla f(\mathbf{P})$ to obtain the Euclidean Hessian of f :

$$\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}] = (Dw(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} + w(\mathbf{P})\Delta\mathbf{P} + \Delta\mathbf{P}w(\mathbf{P}) + \mathbf{P}(Dw(\mathbf{P})[\Delta\mathbf{P}]), \quad (69)$$

where the differential is

$$\begin{aligned} &Dw(\mathbf{P})[\Delta\mathbf{P}] \\ &= 2d(d-1) \text{sym} \left(\langle \mathbf{x} \cdot (\mathbf{I}_n, \Delta\mathbf{P}, \mathbf{P}, \dots, \mathbf{P}), \mathbf{x} \cdot (\mathbf{I}_n, \mathbf{P}, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \right) \\ &= d(d-1) \langle \mathbf{x} \cdot (\mathbf{I}_n, (\Delta\mathbf{P})\mathbf{P} + \mathbf{P}\Delta\mathbf{P}, \mathbf{P}, \dots, \mathbf{P}), \mathbf{x} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1} \\ &= d(d-1) \langle \mathbf{x} \cdot (\mathbf{I}_n, \Delta\mathbf{P}, \mathbf{P}, \dots, \mathbf{P}), \mathbf{x} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{P}, \dots, \mathbf{P}) \rangle_{-1}. \end{aligned} \quad (70)$$

The last equality follows from $(\Delta\mathbf{P})\mathbf{P} + \mathbf{P}\Delta\mathbf{P} = \Delta\mathbf{P}$ in (24).

For the Riemannian Hessian (68), we first consider the term involving $\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}]$:

$$\begin{aligned} &(\mathbf{I}_n - \mathbf{P})(\nabla^2 f(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} \\ &= (\mathbf{I}_n - \mathbf{P})(Dw(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P} + (\mathbf{I}_n - \mathbf{P})w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} + (\mathbf{I}_n - \mathbf{P})\Delta\mathbf{P}w(\mathbf{P})\mathbf{P} \\ &= (\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}] + (\mathbf{I}_n - \mathbf{P})w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} + (\mathbf{I}_n - \mathbf{P})\Delta\mathbf{P}w(\mathbf{P})\mathbf{P}. \end{aligned} \quad (71)$$

The last equality is due to $v(\mathbf{P})[\Delta\mathbf{P}] = (Dw(\mathbf{P})[\Delta\mathbf{P}])\mathbf{P}$ by (40) and (70).

We then compute the remaining terms in (68), which involve $\nabla f(\mathbf{P})$:

$$\begin{aligned} & \nabla f(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}\nabla f(\mathbf{P})\mathbf{P} \\ &= (w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P}))(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}(w(\mathbf{P})\mathbf{P} + \mathbf{P}w(\mathbf{P}))\mathbf{P} \\ &= \mathbf{P}w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}w(\mathbf{P})\mathbf{P} - (\Delta\mathbf{P})\mathbf{P}w(\mathbf{P})\mathbf{P}. \end{aligned} \quad (72)$$

In the last equality, we eliminate the term that includes $\mathbf{P}(\Delta\mathbf{P})\mathbf{P}$, since this is $\mathbf{0}$ due to (24).

We add (71) and (72) to obtain

$$\begin{aligned} & \text{Hess } f(\mathbf{P})[\Delta\mathbf{P}] \\ &= 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}] + (\mathbf{I}_n - \mathbf{P} + \mathbf{P})w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} \right. \\ & \quad \left. + (\mathbf{I}_n - \mathbf{P} - \mathbf{I}_n)(\Delta\mathbf{P})w(\mathbf{P})\mathbf{P} - (\Delta\mathbf{P})\mathbf{P}w(\mathbf{P})\mathbf{P} \right) \\ &= 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}] + w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - (\mathbf{P}\Delta\mathbf{P} + (\Delta\mathbf{P})\mathbf{P})w(\mathbf{P})\mathbf{P} \right) \\ &= 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}] + w(\mathbf{P})(\Delta\mathbf{P})\mathbf{P} - \Delta\mathbf{P}w(\mathbf{P})\mathbf{P} \right). \end{aligned}$$

This is the formula for the Riemannian Hessian in (41).

Next, we compute the differential $D\phi(\mathbf{P})[\Delta\mathbf{P}]$. This is the term that is linear in $\Delta\mathbf{P}$ in the Taylor expansion of the difference:

$$\phi(\mathbf{P} + \Delta\mathbf{P}) - \phi(\mathbf{P}) = \Phi(\mathbf{Q} + \Delta\mathbf{Q})\Phi(\mathbf{Q} + \Delta\mathbf{Q})^\top - \Phi(\mathbf{Q})\Phi(\mathbf{Q})^\top. \quad (73)$$

So we start with $\Phi(\mathbf{Q} + \Delta\mathbf{Q})$ and $\Phi(\mathbf{Q})$. From the first-order Taylor expansion of orth ,

$$\begin{aligned} \Phi(\mathbf{Q} + \Delta\mathbf{Q}) &= \text{orth}(\mathbf{Q} + \Delta\mathbf{Q} + \gamma\nabla F(\mathbf{Q} + \Delta\mathbf{Q})) \\ &= \text{orth}\left(\mathbf{Q} + \gamma\nabla F(\mathbf{Q}) + \Delta\mathbf{Q} + \gamma(\nabla F(\mathbf{Q} + \Delta\mathbf{Q}) - \nabla F(\mathbf{Q}))\right) \\ &= \text{orth}\left(\mathbf{Q} + \gamma\nabla F(\mathbf{Q}) + \Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}] + \mathcal{O}(\|\Delta\mathbf{Q}\|^2)\right) \\ &= \text{orth}(\mathbf{Q} + \gamma\nabla F(\mathbf{Q})) \\ & \quad + D\text{orth}(\mathbf{Q} + \gamma\nabla F(\mathbf{Q}))[\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}] + \mathcal{O}(\|\Delta\mathbf{Q}\|^2)]. \end{aligned}$$

Write $\Phi(\mathbf{Q})\mathbf{R}$ for the QR decomposition of $\mathbf{Q} + \gamma\nabla F(\mathbf{Q})$. We again apply the differentiation formula for orth in (68, Algorithm 1) and get

$$\begin{aligned} \Phi(\mathbf{Q} + \Delta\mathbf{Q}) &= \Phi(\mathbf{Q}) + (\mathbf{I}_n - \Phi(\mathbf{Q})\Phi(\mathbf{Q})^\top)(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1} \\ & \quad + \Phi(\mathbf{Q})\mathbf{S} + \mathcal{O}(\|\Delta\mathbf{Q}\|^2). \end{aligned} \quad (74)$$

Here $\mathbf{S} \in \mathbb{R}^{r \times r}$ is a skew-symmetric matrix that is linear in $\Delta\mathbf{Q}$. The Euclidean gradient $\nabla F(\mathbf{Q})$ is given by (53), and the Euclidean Hessian of F is calculated as

$$\begin{aligned} & \nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}] \\ &= 2d \langle \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}) \rangle_{-1} \Delta\mathbf{Q} \\ & \quad + 2d(d-1) \langle \mathbf{X} \cdot (\mathbf{I}_n, (\Delta\mathbf{Q})\mathbf{Q} + \mathbf{Q}\Delta\mathbf{Q}, \mathbf{Q}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}) \rangle_{-1} \mathbf{Q} \\ &= 2w(\mathbf{P})\Delta\mathbf{Q} + 2d(d-1) \langle \mathbf{X} \cdot (\mathbf{I}_n, \Delta\mathbf{P}, \mathbf{Q}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}) \rangle_{-1} \mathbf{Q}, \end{aligned} \quad (75)$$

due to (26) and $\Delta\mathbf{P} = (\Delta\mathbf{Q})\mathbf{Q}^\top + \mathbf{Q}(\Delta\mathbf{Q})^\top$.

We substitute (74) into (73). We only keep the terms that are linear in $\Delta\mathbf{Q}$ and get

$$\begin{aligned} D\phi(\mathbf{P})[\Delta\mathbf{P}] &= 2 \text{sym} \left((\mathbf{I}_n - \Phi(\mathbf{Q})\Phi(\mathbf{Q})^\top)(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right. \\ & \quad \left. + (\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ &= 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ & \quad + 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ &= 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ & \quad + 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ &= 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \\ & \quad + 2 \text{sym} \left((\mathbf{I}_n - \phi(\mathbf{P}))(\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}])\mathbf{R}^{-1}\Phi(\mathbf{Q})^\top \right) \end{aligned} \quad (76)$$

For the first equality, we use $\Phi(\mathbf{Q})\mathbf{S}\Phi(\mathbf{Q})^\top + \Phi(\mathbf{Q})\mathbf{S}^\top\Phi(\mathbf{Q})^\top = \mathbf{0}$ and the fact \mathbf{S} is linear in $\Delta\mathbf{Q}$

It remains to transform the differential expression (76) in $\mathbf{Q}, \Delta\mathbf{Q}$ to be in terms of $\mathbf{P}, \Delta\mathbf{P}$. To do this, we split (76) into two factors:

$$\begin{aligned}\text{Factor 1} &:= (\mathbf{I}_n - \phi(\mathbf{P})) (\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}]) \mathbf{Q}^\top, \\ \text{Factor 2} &:= \mathbf{Q} (\mathbf{Q} + \gamma\nabla F(\mathbf{Q}))^\dagger.\end{aligned}$$

We first tackle Factor 2. As \mathbf{Q}^\top and $\mathbf{Q} + \gamma\nabla F(\mathbf{Q})$ respectively have orthonormal rows and columns, we have

$$\mathbf{Q} (\mathbf{Q} + \gamma\nabla F(\mathbf{Q}))^\dagger = (\mathbf{Q}^\top)^\dagger (\mathbf{Q} + \gamma\nabla F(\mathbf{Q}))^\dagger = ((\mathbf{Q} + \gamma\nabla F(\mathbf{Q})) \mathbf{Q}^\top)^\dagger.$$

Inserting (53) into the above equation,

$$\text{Factor 2} = ((\mathbf{Q} + 2\gamma w(\mathbf{P})\mathbf{Q}) \mathbf{Q}^\top)^\dagger = (\mathbf{P} + 2\gamma w(\mathbf{P})\mathbf{P})^\dagger. \quad (77)$$

Next, we consider Factor 1. By (75), it equals

$$\begin{aligned} & (\mathbf{I}_n - \phi(\mathbf{P})) (\Delta\mathbf{Q} + \gamma\nabla^2 F(\mathbf{Q})[\Delta\mathbf{Q}]) \mathbf{Q}^\top \\ &= (\mathbf{I}_n - \phi(\mathbf{P})) (\mathbf{I}_n + 2\gamma w(\mathbf{P})) (\Delta\mathbf{Q}) \mathbf{Q}^\top \\ &+ (\mathbf{I}_n - \phi(\mathbf{P})) (2\gamma d(d-1) \langle \mathbf{X} \cdot (\mathbf{I}_n, \Delta\mathbf{P}, \mathbf{Q}, \dots, \mathbf{Q}), \mathbf{X} \cdot (\mathbf{I}_n, \mathbf{I}_n, \mathbf{Q}, \dots, \mathbf{Q}) \rangle_{-1} \mathbf{Q} \mathbf{Q}^\top) \\ &= (\mathbf{I}_n - \phi(\mathbf{P})) ((\mathbf{I}_n + 2\gamma w(\mathbf{P})) (\Delta\mathbf{P} - \mathbf{Q}(\Delta\mathbf{Q})^\top) + 2\gamma v(\mathbf{P})[\Delta\mathbf{P}]). \end{aligned} \quad (78)$$

The last equality is from the definition of $v(\mathbf{P})[\Delta\mathbf{P}]$ (40).

However, note that

$$\begin{aligned} & (\mathbf{I}_n - \phi(\mathbf{P})) (\mathbf{I}_n + 2\gamma w(\mathbf{P})) \mathbf{Q} (\Delta\mathbf{Q})^\top \\ &= (\mathbf{I}_n - \phi(\mathbf{P})) (\mathbf{Q} + 2\gamma w(\mathbf{P})\mathbf{Q}) (\Delta\mathbf{Q})^\top \\ &= (\mathbf{I}_n - \Phi(\mathbf{Q})\Phi(\mathbf{Q})^\top) (\mathbf{Q} + \gamma\nabla F(\mathbf{Q})) (\Delta\mathbf{Q})^\top = \mathbf{0}. \end{aligned}$$

Here, we replace $2w(\mathbf{P})\mathbf{Q}$ by $\nabla F(\mathbf{Q})$ by (53). Then from $\text{colspan}(\mathbf{Q} + \gamma\nabla F(\mathbf{Q}))$ equals $\text{colspan}(\Phi(\mathbf{Q}))$ by (20), it holds $(\mathbf{I}_n - \Phi(\mathbf{Q})\Phi(\mathbf{Q})^\top) (\mathbf{Q} + \gamma\nabla F(\mathbf{Q})) = \mathbf{0}$.

Returning to (78),

$$\begin{aligned}\text{Factor 1} &= (\mathbf{I}_n - \phi(\mathbf{P})) ((\mathbf{I}_n + 2\gamma w(\mathbf{P})) (\Delta\mathbf{P}) + 2\gamma v(\mathbf{P})[\Delta\mathbf{P}]) \\ &= (\mathbf{I}_n - \phi(\mathbf{P})) (\Delta\mathbf{P} + 2\gamma w(\mathbf{P})(\Delta\mathbf{P}) + 2\gamma v(\mathbf{P})[\Delta\mathbf{P}]). \end{aligned} \quad (79)$$

Putting (76), (77) and (79) together, we obtain the formula for the differential $D\phi(\mathbf{P})[\Delta\mathbf{P}]$ (42). The proof of Proposition 6.6 is complete. \square

F Proof of Lemma 6.7

Proof. Let $\mathbf{P} \in \text{Gr}(n, r)$ be a first-order critical point of f (22). The Riemannian Hessian $\text{Hess } f(\mathbf{P}) : \mathbb{T}_{\mathbf{P}} \text{Gr}(n, r) \rightarrow \mathbb{T}_{\mathbf{P}} \text{Gr}(n, r)$ is diagonalizable because it is self-adjoint. The dimension of the tangent space $\mathbb{T}_{\mathbf{P}} \text{Gr}(n, r)$ is $\dim(\text{Gr}(n, r)) = r(n-r)$. Let $\Delta\mathbf{P}_1, \dots, \Delta\mathbf{P}_{r(n-r)} \in \mathbb{T}_{\mathbf{P}} \text{Gr}(n, r)$ be a basis of eigenmatrices of $\text{Hess } f(\mathbf{P})$ with corresponding eigenvalues $\lambda_1, \dots, \lambda_{r(n-r)}$, i.e.

$$\text{Hess } f(\mathbf{P})[\Delta\mathbf{P}_i] = \lambda_i \Delta\mathbf{P}_i$$

for $i \in [r(n-r)]$. Assume there is at least one strictly positive eigenvalue of $\text{Hess } f(\mathbf{P})$, w.l.o.g. $\lambda_1 > 0$.

We analyze the eigenvalues of $D\phi(\mathbf{P})$ by expressing the differential with respect to $\{\Delta\mathbf{P}_i\}$. Under this basis, the differential is represented by the matrix $\mathbf{M} \in \mathbb{R}^{r(n-r) \times r(n-r)}$ with entries $m_{i,j}$ equaling to

$$\left\langle 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P}) (\Delta\mathbf{P}_i + 2\gamma w(\mathbf{P})\Delta\mathbf{P}_i + \gamma v(\mathbf{P})[\Delta\mathbf{P}_i]) (\mathbf{P} + 2\gamma w(\mathbf{P})\mathbf{P})^\dagger, \Delta\mathbf{P}_j \right), \Delta\mathbf{P}_j \right\rangle, \quad (80)$$

for $i, j \in [r(n-r)]$. Here we apply the formula (42) and the fact that $\mathbf{P} = \phi(\mathbf{P})$ as critical points of f are fixed points of ϕ by C3 in Section 6.2. The eigenvalues of the operator $D\phi(\mathbf{P})$ are the same as the eigenvalues of the matrix \mathbf{M} .

We next compute the matrix \mathbf{M} . Due to the rank constraint on \mathbf{P} , we utilize its block structure to facilitate the computation. W.l.o.g., we suppose the rank- r orthogonal projection \mathbf{P} is:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0}_{r \times (n-r)} \\ \mathbf{0}_{(n-r) \times r} & \mathbf{0}_{(n-r) \times (n-r)} \end{bmatrix} \in \text{Gr}(n, r),$$

by choosing an appropriate for \mathbb{R}^n .

From (52), $\text{grad} f(\mathbf{P}) = \text{sym} \left(2(\mathbf{I}_n - \mathbf{P})w(\mathbf{P})\mathbf{P} \right) = \mathbf{0}$, and (24), $\Delta\mathbf{P} = (\Delta\mathbf{P})\mathbf{P} + \mathbf{P}\Delta\mathbf{P}$ for all $\Delta\mathbf{P} \in \mathbb{T}_{\mathbf{P}} \text{Gr}(n, r)$, we obtain block matrices for $w(\mathbf{P})$ and $\{\Delta\mathbf{P}_i\}$. Namely there are PSD matrices $\mathbf{A} \in \mathbb{S}^2(\mathbb{R}^r)$, $\mathbf{C} \in \mathbb{S}^2(\mathbb{R}^{n-r})$ and matrices $\mathbf{H}_i \in \mathbb{R}^{(n-r) \times r}$ for $i \in [r(n-r)]$ such that

$$w(\mathbf{P}) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \in \mathbb{S}^2(\mathbb{R}^n), \quad \Delta\mathbf{P}_i = \begin{bmatrix} \mathbf{0} & \mathbf{H}_i^\top \\ \mathbf{H}_i & \mathbf{0} \end{bmatrix} \in \mathbb{T}_{\mathbf{P}} \text{Gr}(n, r). \quad (81)$$

Since $\Delta\mathbf{P}_i$ are eigenmatrices of $\text{Hess} f(\mathbf{P})$, by (41) we see

$$\begin{aligned} & 2 \text{sym} \left((\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}_i] \right) \\ &= \text{Hess} f(\mathbf{P})[(\Delta\mathbf{P}_i)] - 2 \text{sym} \left(w(\mathbf{P})(\Delta\mathbf{P}_i)\mathbf{P} - \Delta\mathbf{P}_i w(\mathbf{P})\mathbf{P} \right) \\ &= \begin{bmatrix} \mathbf{0} & \lambda_i \mathbf{H}_i^\top - \mathbf{H}_i^\top \mathbf{C} + \mathbf{A} \mathbf{H}_i^\top \\ \lambda_i \mathbf{H}_i - \mathbf{C} \mathbf{H}_i + \mathbf{H}_i \mathbf{A} & \mathbf{0} \end{bmatrix}. \end{aligned}$$

It follows that

$$(\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}_i] = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \lambda_i \mathbf{H}_i - \mathbf{C} \mathbf{H}_i + \mathbf{H}_i \mathbf{A} & \mathbf{0} \end{bmatrix},$$

because $(\mathbf{I}_n - \mathbf{P})v(\mathbf{P})[\Delta\mathbf{P}_i]$ (40) is in the format $(\mathbf{I}_n - \mathbf{P})(\cdot)\mathbf{P}$ so only its left bottom block is nonzero.

To get $m_{i,j}$ (80), we compute

$$\begin{aligned} & (\mathbf{I}_n - \mathbf{P})(\Delta\mathbf{P}_i + 2\gamma w(\mathbf{P})\Delta\mathbf{P}_i + 2\gamma v(\mathbf{P})[\Delta\mathbf{P}_i]) \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-r} \end{bmatrix} \begin{bmatrix} \mathbf{I}_r + 2\gamma \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-r} + 2\gamma \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{H}_i^\top \\ \mathbf{H}_i & \mathbf{0} \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ 2\gamma(\lambda_i \mathbf{H}_i - \mathbf{C} \mathbf{H}_i + \mathbf{H}_i \mathbf{A}) & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ (1 + 2\gamma \lambda_i) \mathbf{H}_i + 2\gamma \mathbf{H}_i \mathbf{A} & \mathbf{0} \end{bmatrix}. \end{aligned}$$

Also,

$$(\mathbf{P} + 2\gamma w(\mathbf{P})\mathbf{P})^\dagger = \left(\begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} 2\gamma \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right)^\dagger = \begin{bmatrix} (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Here we note that since \mathbf{A} is PSD and $\gamma > 0$, the inverse $(\mathbf{I}_r + 2\gamma \mathbf{A})^{-1}$ exists and is symmetric.

Thus the left component in the inner product (80) is

$$\begin{aligned} & 2 \text{sym} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ (1 + 2\gamma \lambda_i) \mathbf{H}_i + 2\gamma \mathbf{H}_i \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{H}_i^\top + 2\gamma \lambda_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} \mathbf{H}_i^\top \\ \mathbf{H}_i + 2\gamma \lambda_i \mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} & \mathbf{0} \end{bmatrix} \end{aligned}$$

Finally (80) reads $m_{i,j}$ equals to

$$\begin{aligned} & \left\langle \begin{bmatrix} \mathbf{0} & \mathbf{H}_i^\top + 2\gamma \lambda_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} \mathbf{H}_i^\top \\ \mathbf{H}_i + 2\gamma \lambda_i \mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1} & \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{H}_j^\top \\ \mathbf{H}_j & \mathbf{0} \end{bmatrix} \right\rangle \\ &= 2 \langle \mathbf{H}_i + 2\gamma \lambda_i \mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-1}, \mathbf{H}_j \rangle \\ &= \begin{cases} 4\gamma \lambda_i \left\| \mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}} \right\|^2 + 1 & i = j \\ 4\gamma \lambda_i \left\langle \mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}}, \mathbf{H}_j (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}} \right\rangle & i \neq j, \end{cases} \end{aligned} \quad (82)$$

where we use

$$\langle \mathbf{H}_i, \mathbf{H}_j \rangle = \frac{1}{2} \langle \Delta \mathbf{P}_i, \Delta \mathbf{P}_j \rangle = \begin{cases} \frac{1}{2}, & i = j \\ 0, & i \neq j. \end{cases}$$

We set

$$\mathbf{L} = \begin{bmatrix} \text{vec} \left(\mathbf{H}_1 (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}} \right)^\top \\ \vdots \\ \text{vec} \left(\mathbf{H}_{r(n-r)} (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}} \right)^\top \end{bmatrix} \in \mathbb{R}^{r(n-r) \times r(n-r)}, \quad (83)$$

$$\mathbf{A} = 4\gamma \text{diag}(\lambda_1, \dots, \lambda_{r(n-r)}) \in \mathbb{S}^2(\mathbb{R}^{r(n-r)}).$$

Here vec denotes the vectorization operator. Combining (82) and (83), we have a short-hand expression for \mathbf{M} :

$$\mathbf{M} = \mathbf{A} \mathbf{L} \mathbf{L}^\top + \mathbf{I}_{r(n-r)}.$$

The linear independence of $\{\mathbf{H}_i\}$ implies the linear independence of $\{\mathbf{H}_i (\mathbf{I}_r + 2\gamma \mathbf{A})^{-\frac{1}{2}}\}$. Then \mathbf{L} is invertible because its rows are linearly independent. Hence $\mathbf{A} \mathbf{L} \mathbf{L}^\top$ is similar to $\mathbf{L}^\top \mathbf{A} \mathbf{L} = \mathbf{L}^\top (\mathbf{A} \mathbf{L} \mathbf{L}^\top) \mathbf{L}^{-\top}$. As $\mathbf{L}^\top \mathbf{A} \mathbf{L}$ is symmetric, $\mathbf{A} \mathbf{L} \mathbf{L}^\top$ is diagonalizable with the same eigenvalues as $\mathbf{L}^\top \mathbf{A} \mathbf{L}$. Then \mathbf{A} and $\mathbf{L}^\top \mathbf{A} \mathbf{L}$ are congruent. So their numbers of positive eigenvalues match. But there is a strictly positive eigenvalue of \mathbf{A} , namely $4\gamma \lambda_1 > 0$. It follows that $\mathbf{M} = \mathbf{A} \mathbf{L} \mathbf{L}^\top + \mathbf{I}_{r(n-r)}$ has an eigenvalue strictly greater than 1. The proof of Lemma 6.7 is complete. \square

G Proof of Lemma 6.8

Proof. We prove Lemma 6.8 by showing that ϕ is C^∞ and $D\phi(\mathbf{P})$ is invertible for all $\mathbf{P} \in \text{Gr}(n, r)$, when γ is small enough.

First we claim that Φ is C^∞ on $\text{St}(n, r)$. It is a composite of orth and $\mathbf{Q} \mapsto \mathbf{Q} + \gamma \nabla F(\mathbf{Q})$. The latter is a polynomial map, only taking full-rank values by C3 of Section 6.2. Meanwhile, orth is C^∞ at full-rank inputs by [66]. So Φ is C^∞ . Consequently ϕ is C^∞ by [40, Theorem A.27 (a)], since $\mathbf{Q} \mapsto \mathbf{Q} \mathbf{Q}^\top$ is a quotient map from $\text{St}(n, r)$ to $\text{Gr}(n, r)$.

Next we show $D\Phi(\mathbf{Q})$ is invertible for all $\mathbf{Q} \in \text{St}(n, r)$. Define $h : \text{St}(n, r) \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ by

$$h(\mathbf{Q}, \gamma) := \sigma_{\min}(D\Phi(\mathbf{Q})) = \sigma_{\min}(D\text{orth}(\mathbf{Q} + \gamma \nabla F(\mathbf{Q}))),$$

where σ_{\min} denotes the smallest singular value. Here orth is C^∞ at full-rank inputs, $\Phi(\mathbf{Q})$ is full-rank for $\gamma \geq 0$ and $\Phi(\mathbf{Q})$ is polynomial in \mathbf{Q}, γ . It follows that $D\text{orth}(\mathbf{Q} + \gamma \nabla F(\mathbf{Q}))$ is jointly continuous in \mathbf{Q}, γ . As the smallest singular value is also continuous, h is jointly continuous in \mathbf{Q}, γ . When $\gamma = 0$, Φ becomes the identity map on $\text{St}(n, r)$, so $h(\cdot, 0) \equiv 1$. Because the Stiefel manifold is compact, there exists a constant $\Gamma^{**}(\mathbf{X}, r) > 0$ such that for all $0 < \gamma \leq \Gamma^{**}$ it holds $h(\cdot, \gamma) > 0$. W.l.o.g. assume Γ^{**} is less than Γ^* in Theorem 6.1. Then $D\Phi : T_{\mathbf{Q}} \text{St}(n, r) \rightarrow T_{\phi(\mathbf{Q})} \text{St}(n, r)$ is invertible for all $\mathbf{Q} \in \text{St}(n, r)$ when $0 < \gamma \leq \Gamma^{**}$.

To conclude, differentiate the update diagram Figure 10. By the chain rule, we get a commutative diagram of differentials between tangent spaces. The ‘‘north-then-west’’ path is a composition of surjections because $D\Phi(\mathbf{Q}_t)$ is surjective and $\mathbf{Q} \mapsto \mathbf{Q} \mathbf{Q}^\top$ is a quotient map. By commutativity, the ‘‘west-then-north’’ composition is surjective. So the second map there must be a surjection, i.e. $D\phi(\mathbf{P}_t) : T_{\mathbf{P}_t} \text{Gr}(n, r) \rightarrow T_{\mathbf{P}_{t+1}} \text{Gr}(n, r)$ is surjective. Therefore it is invertible, as the tangent spaces share the same dimension. The proof of Lemma 6.8 is complete. \square