Probabilistic PCA from Heteroscedastic Signals: Geometric Framework and Application to Clustering

Antoine Collas, Florent Bouchard, Arnaud Breloy, Guillaume Ginolhac, Chengfang Ren, Jean-Philippe Ovarlez

***Abstract*—This paper studies a statistical model for heteroscedastic (*i.e.*, power fluctuating) signals embedded in white Gaussian noise. Using the Riemannian geometry theory, we propose an unified approach to tackle several problems related to this model. The first axis of contribution concerns parameters (signal subspace and power factors) estimation, for which we derive intrinsic Cramér-Rao bounds and propose a flexible Riemannian optimization algorithmic framework in order to compute the maximum likelihood estimator (as well as other cost functions involving the parameters). Interestingly, the obtained bounds are in closed forms and interpretable in terms of problem's dimensions and SNR. The second axis of contribution concerns the problem of clustering data assuming a mixture of heteroscedastic signals model, for which we generalize the Euclidean *K-means*++ to the considered Riemannian parameter space. We propose an application of the resulting clustering algorithm on the *Indian Pines* segmentation problem benchmark.**

***Index Terms*—Covariance Matrices, Probabilistic PCA, Heteroscedastic data, Robust Estimation, Riemannian Optimization, Clustering**

I. Introduction

Principal Component Analysis (PCA) [1] is a standard tool used in signal processing and machine learning literature for dimensional reduction and statistical interpretation. In this scope, Probabilistic PCA (PPCA) refers to a reformulation of PCA as a parametric estimation problem. This approach was proposed in [2], which considered a model of white Gaussian noise (WGN) plus a linear mapping of a low-dimensional centered Gaussian latent space with unit variance (the signal contribution). The maximum likelihood estimate (MLE) of the signal subspace basis corresponds to the sample covariance matrix's (SCM) first principal eigenvectors.

Leveraging the statistical formulation of PPCA allows going beyond Gaussian models. For example, the two independent contributions (either signal or noise) can be generalized to the distribution of Compound Gaussian (CG). CGs represent a family of elliptical distributions (cf. review in [3]) that encompasses numerous standard heavy-tailed models, such as the multivariate t-distribution. Its stochastic representation involves a Gaussian vector multiplied by an independent random power factor referred to as *texture*. In order to be robust to various underlying distributions, this parameter is often assumed to be unknown deterministic, which yields the so-called scaled Gaussian model [4], also referred to as *heteroscedastic* (HS) [5]. In this scope [6]–[8] considered HS distributions for the signal component to perform robust PCA for non-Gaussian signals. Conversely, [5] considered Gaussian signals embedded in white CG noise to model data where some samples are noisier than others. Alternatively, [9] uses a t-distribution to model both of the contributions. Finally, [10] considered a mixture of three components to account for potential outliers (the thirds contribution being orthogonal to the signal subspace).

In the following, we will focus on HS plus WGN model [6]–[8] which is interpreted as impulsive signals (power variation across samples) plus thermal noise due to electronics. A common relaxation of this model is to assume that eigenvalues of the (low-rank) signal covariance matrix are identical as in [11], [12]. Indeed, this hypothesis is relevant since we still estimate the power variations which contain, the information of the eigenvalues. Moreover, [6], [10], [13] showed that neglecting the differences between eigenvalues does not harm the accuracy of subspace estimation while allowing for a more meaningful statistical interpretation [11].

Yet, the previous studies still left some unanswered issues: first, the algorithms in [11], [12] are dedicated bloc-coordinate descent type. Thus, they can be limited in practice, as they offer no generalization to on-line (or stochastic) settings. It would then be relevant for the estimation problem to be cast in a more generic optimization framework that can account for the parameter structure (*e.g.*, subspaces, vectors with strictly positive values). Second, the MLE of the considered model is the solution of a nonconvex problem with no guarantee for global optimality. Thus, it would be interesting to derive performance bound in order to assess for various algorithms performance. Such bound is not trivial for these models because structured parameters require accounting for specific constraints, as well as for the use of relevant distances as error measure (*e.g.* to ensure for some invariance). Finally, one can inquire if the features of such statistical model can be meaningfully leveraged in machine learning tasks such as clustering.

Therefore, this paper conducts a study of the HS plus WGN model [11], [12] through the prism of Riemannian geometry, as this this theoretical framework allows us to propose a unified view to tackle the aforementioned questions. The contributions concern the following directions:
i) Riemannian optimization framework for model features: HS plus WGN model involves parameters that are textures (power factors) and a low-rank subspace. Endowing this parameter

A. Collas, C. Ren are with SONDRA, CentraleSuplec, Universit Paris-Saclay. F. Bouchard is with CNRS, L2S, CentraleSuplec, Universit Paris-Saclay. G. Ginolhac is with LISTIC (EA3703), University Savoie Mont Blanc. A. Breloy is with LEME (EA4416), University Paris Nanterre. J-P Ovarlez is with SONDRA, CentraleSuplec, Universit Paris-Saclay and DEMR, ONERA. Part of this work was supported by ANR-ASTRID MARGARITA (ANR-17-ASTR-0015).

space with a Riemannian metric yields a Riemannian manifold, which can be leveraged in an optimization framework [14]. In this context, we consider the model's Fisher information metric (FIM). We then obtain several essential tools (tangent space, Riemannian gradient, retraction) from established results on the Grassman manifold [15]. These tools are then used to propose algorithms in order to compute the MLE, as well as the Riemannian means used in clustering algorithms (cf. next points). We notably propose a Riemannian stochastic gradient descent algorithm [16] suited to large datasets (or online settings [17]).
ii) Performance bounds: We show that the FIM of the considered model (and its corresponding Riemannian distance) permits to derive closed forms and decoupled intrinsic Cramér-Rao lower bound (iCRLB) for the model's parameters. These lower bounds represent partial extensions of [7] (Euclidean CRLB in the case of colored signals) to the iCRLB framework of [18]. Interestingly, the proposed approach offers a new interpretable result regarding problem dimensions and signal-to-noise ratio (SNR). Then, we assess the performance of different estimation algorithms numerically. We show that both the proposed estimation algorithm and the previously established block-coordinate algorithm [12] are statistically efficient for the signal subspace estimation. In a low SNR scenario, they also both outperform subspace estimated by singular value decomposition (SVD) in terms of MSE.
iii) Applications to clustering: we propose a Riemannian clustering algorithm for data following the HS plus WGN model. Indeed, the use of the Riemannian geometry of statistical features in order to classify batches of samples has already demonstrated its merits; see *e.g.* [19], [20] for such methods based on covariance matrices. Here, we extend such methodology to the considered statistical model using the principle of K-means++ [21], which optimizes the within-cluster sum of squares (WCSS) iteratively. Replacing the Euclidean distance by a Riemannian one allows for this clustering algorithm to takes into account the geometrical constraints of the parameter space (invariance properties of subspaces and positivity of powers), which is shown to improve the clustering performance on the hyperspectral image Indian Pines benchmark [22]. We also show that this replacement still preserves the upper bound on the expectation of the WCSS in [21], which guarantees (in expectation) a clustering close to the optimal one.

This paper is organized as follows. Section II presents the statistical model and the parameter space as a manifold. Section III presents a Riemannian geometry for this manifold, and essential tools driven from two possible metrics. Section IV presents results related to parameter estimation (MLE algorithms based on Riemannian optimization and iCRLBs). Section V presents a clustering algorithm (Riemannian *K-means++*) adapted to the considered parameter manifold. Numerical results are presented in Section VI. Appendix A contains the technical proofs.

II. Model and parameter space

A. Heteroscedastic signal model

Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be a data set of p-dimensional complex vectors. We consider a k-dimensional linear signal representation embedded in white Gaussian noise, i.e. the model:

$$\boldsymbol{x} \overset{d}{=} \boldsymbol{U}\,\boldsymbol{g} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{g} \in \mathbb{C}^k$ is the signal of interest, $\boldsymbol{n} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p)$ is a white Gaussian noise, and $\boldsymbol{U} \in \mathrm{St}_{p,k}$ is an orthonormal basis of the signal subspace, where

$$\mathrm{St}_{p,k} = \left\{ \boldsymbol{U} \in \mathbb{C}^{p\times k} : \boldsymbol{U}^H \boldsymbol{U} = \boldsymbol{I}_k \right\}, \tag{2}$$

denotes the complex Stiefel manifold. In array-processing literature it is classically assumed that $\boldsymbol{g}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, which yields a low-rank structured Gaussian model, also referred to as the (Gaussian) Probabilistic PCA (PPCA) model in [2]. Note that these models often rely on the unconstrained identification $\boldsymbol{x} \overset{d}{=} \boldsymbol{W}\,\tilde{\boldsymbol{g}} + \boldsymbol{n}$, with $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$ and $\tilde{\boldsymbol{g}}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. However, using $\boldsymbol{U} \in \mathrm{St}_{p,k}$ is here more coherent with later developments.

In order to model heavy-tailed signals (e.g., outliers or power discrepancies), several works [6]–[8], [11] considered generalizing the Gaussian PPCA to Compound Gaussian (CG) distributions [3]. Such signal model yields

$$\boldsymbol{x}_i | \tau_i \overset{d}{=} \sqrt{\tau_i}\,\boldsymbol{U}\,\boldsymbol{g}_i + \boldsymbol{n}_i, \tag{3}$$

where $\boldsymbol{g}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and $\tau_i \in \mathbb{R}^+$ is a random power factor referred to as texture, which is statistically independent of $\boldsymbol{g}_i$. Starting from this representation, we make the following additional assumptions:
• *Known noise floor*: The variance σ^2 is considered known. If σ^2 is unknown in practice, it can be accurately pre-estimated by averaging lowest eigenvalues of the SCM [2]. The hypothesis of known σ^2 simplifies the exposition and does not change significantly the performance in practice when compared to a joint estimation scheme (see *e.g.* [23]). Without loss of generality, such assumption allows us to set $\sigma^2 = 1$.
• *Unknown deterministic textures*: In order to provide a model that is robust to any underlying CG distribution, it is often assumed that the textures $\{\tau_i\}_{i=1}^n$ are unknown deterministic rather than assigning it a pre-determined probability density function [6]–[8]. Such distribution is then referred to as scaled Gaussian model [4] or heteroscedastic signals [5].
• *Isotropic signal*: We consider the relaxation from [11], [12], assuming that the eigenvalues of the signal covariance matrix are identical, i.e., $\boldsymbol{g}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \sigma_s \boldsymbol{I}_k)$. In conjunction with the unknown deterministic textures assumption, this allows the change of variable $\tilde{\tau}_i = \sigma_s \tau_i$, and thus setting $\sigma_s = 1$ without loss of generality. While apparently not realistic, this hypothesis is still representative since the average signal power information is accounted for by the texture parameters. Moreover, [6], [10], [13] showed that neglecting the differences between eigenvalues does not harm the accuracy of subspace estimation while allowing for a more meaningful statistical interpretation [11].

Finally, we have the data $\{\boldsymbol{x}_i\}_{i=1}^n$ distributed as in (3) where $\boldsymbol{g}_i \sim \mathbb{C}\mathcal{N}(\mathbf{0}, \boldsymbol{I}_k)$ and $\boldsymbol{n}_i \sim \mathbb{C}\mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$. The unknown model parameters are the textures $\{\tau_i\}_{i=1}^p$ (denoted by the vector $\boldsymbol{\tau} \in (\mathbb{R}^{++})^n$) and the signal subspace, represented by a basis $\boldsymbol{U} \in \mathrm{St}_{p,k}$. The following section will recast this parameter space as a manifold. This reformulation will then allow us to leverage tools from the Riemannian geometry in order to derive distances, intrinsic Cramér-Rao Bounds and optimization methods with a unified view.

B. Manifold approach to the parameter space

Due to their specific geometrical structure, the parameters $(\boldsymbol{U}, \boldsymbol{\tau})$ of model (3) can be embedded into the product manifold $\overline{\mathcal{M}}_{p,k,n} = \mathrm{St}_{p,k} \times (\mathbb{R}^{++})^n$. With this model, from $\overline{\mathcal{M}}_{p,k,n}$, the scaled covariance matrix in $\mathcal{H}_p^{++}$ of sample $\boldsymbol{x}_i$ is obtained through the function

$$\begin{array}{rccc} \bar{\psi}_i: & \overline{\mathcal{M}}_{p,k,n} & \to & \mathcal{H}_p^{++} \\ & (\boldsymbol{U}, \boldsymbol{\tau}) & \mapsto & \boldsymbol{I}_p + \tau_i \boldsymbol{U}\boldsymbol{U}^H. \end{array} \tag{4}$$

It follows that the negative log-likelihood corresponding to model (3) is given, for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, by

$$\bar{L}(\bar{\theta}) = \sum_i \log \left|\bar{\psi}_i(\bar{\theta})\right| + \boldsymbol{x}_i^H \left(\bar{\psi}_i(\bar{\theta})\right)^{-1} \boldsymbol{x}_i. \tag{5}$$

The model (3) is ambiguous since the representation by the basis $\boldsymbol{U}$ is invariant by rotation: for all $\boldsymbol{O} \in \mathcal{U}_k$ (where $\mathcal{U}_k$ is the unitary group of degree k), $(\boldsymbol{UO}, \boldsymbol{\tau})$ is equivalent to $(\boldsymbol{U}, \boldsymbol{\tau})$, *i.e.*, it yields the same scaled covariance matrices in $\mathcal{H}_p^{++}$. The consequence is that the manifold $\overline{\mathcal{M}}_{p,k,n}$ is not optimal with respect to the model of interest. In terms of optimization, for instance for maximum likelihood estimation, it is possible to exploit $\overline{\mathcal{M}}_{p,k,n}$ directly but it is advantageous to take into account the invariance. Moreover, to measure estimation errors or perform geometrical classification and clustering, employing a distance function onto $\overline{\mathcal{M}}_{p,k,n}$ is not ideal: the distance between two equivalent points is not equal to zero. It thus appears very attractive to take this invariance into account.

Fortunately, it is possible to naturally handle this rotation invariance from a geometrical perspective. It is achieved by considering the Grassmann manifold $\mathrm{Gr}_{p,k}$, which is the set of all k-dimensional subspaces of $\mathbb{C}^p$. The Grassmann manifold can be identified to the quotient manifold [14], [15], [24]

$$\mathrm{Gr}_{p,k} = \{\{\boldsymbol{UO} : \boldsymbol{O} \in \mathcal{U}_k\} : \boldsymbol{U} \in \mathrm{St}_{p,k}\}. \tag{6}$$

From there, to optimally embed the parameters of model (3), we construct the manifold $\mathcal{M}_{p,k,n} = \mathrm{Gr}_{p,k} \times (\mathbb{R}^{++})^n$. This manifold can be viewed as a quotient manifold of $\overline{\mathcal{M}}_{p,k,n}$, *i.e.*, it can be defined as

$$\mathcal{M}_{p,k,n} = \{\pi(\bar{\theta}) : \bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}\}, \tag{7}$$

where, for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, the equivalence class is defined as

$$\pi(\bar{\theta}) = \{(\boldsymbol{UO}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\}. \tag{8}$$

Functions $\bar{\psi}_i$ defined onto $\overline{\mathcal{M}}_{p,k,n}$ induce functions ψ_i onto $\mathcal{M}_{p,k,n}$, *i.e.* $\bar{\psi}_i(\bar{\theta}) = \psi_i(\pi(\bar{\theta}))$. Thus, $\boldsymbol{x}_i$ in (3) is drawn as $\boldsymbol{x}_i \sim \mathbb{C}\mathcal{N}(\mathbf{0}, \psi_i(\theta))$. It follows that the log-likelihood $\bar{L}$ in (5) defined onto $\overline{\mathcal{M}}_{p,k,n}$ can also be defined onto $\mathcal{M}_{p,k,n}$ by using functions ψ_i instead of $\bar{\psi}_i$. This log-likelihood function is denoted L in the following.

Besides acknowledging the model invariances, considering $\mathcal{M}_{p,k,n}$ as a manifold allows for advantageously exploiting Riemannian geometry, *i.e.*, the geometries of $\mathcal{M}_{p,k,n}$ induced by Riemannian metrics. In particular for signal processing applications, it can be leveraged for:

1) Estimation: the Riemannian optimization framework can be employed to compute maximum likelihood estimators (Section IV-A) and Riemannian means (Section V) in various practical scenarios.
2) Performance measuring: the Riemannian distance naturally defines an error measure, which can then be bounded using the framework of intrinsic Cramér-Rao bound [18]. This point will be detailed in Section IV-B.
3) Machine learning: the Riemannian distance can also be exploited to cluster and classify various data which follow model (3), which will be further discussed in Section V.

In order to achieve these, different geometrical objects are needed. Section III will introduce these tools conditionally to the choice of the Riemannian metric.

III. Geometry of $\mathcal{M}_{p,k,n}$

Various choices of Riemannian geometries are available for $\mathcal{M}_{p,k,n}$, entirely depending on the choice of the Riemannian metric. Among different possibilities, one is optimal with respect to the considered statistical model: the Fisher information metric [25]. Indeed, it is derived from the log-likelihood function of the distribution at hand and thus perfectly captures the particularities of the model. However, the geometry induced by the Fisher information metric is often hard to fully leverage. One has therefore to compromise and define an alternate geometry (induced by a metric as close as possible to the Fisher one) in order to obtain tractable expressions for the needed geometrical tools.

In this section, we first provide an introduction on $\mathcal{M}_{p,k,n}$ viewed as a Riemannian quotient manifold in Section III-A. We then study the Fisher information metric of likelihood (5) and derive the geometrical objects needed for Riemannian optimization in Section III-B. However, required objects related to Riemannian distances cannot be obtained in closed-form. An alternate geometry using a decoupled metric (close to the Fisher one) is thus proposed in order to achieve these in Section III-C. The obtained results are summarized in Table I.

A. $\mathcal{M}_{p,k,n}$ as a Riemannian quotient manifold

Since $\mathrm{Gr}_{p,k}$ is a quotient manifold of $\mathrm{St}_{p,k}$ with respect to the action of $\mathcal{U}_k$ [15], $\mathcal{M}_{p,k,n} = \mathrm{Gr}_{p,k} \times (\mathbb{R}^{++})^n$ is a quotient of $\overline{\mathcal{M}}_{p,k,n} = \mathrm{St}_{p,k} \times (\mathbb{R}^{++})^n$. To handle elements of $\mathcal{M}_{p,k,n}$, which are equivalence classes $\{(\boldsymbol{UO}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\}$, one usually exploits the canonical projection $\pi : \overline{\mathcal{M}}_{p,k,n} \to \mathcal{M}_{p,k,n}$ in (8). Equivalence classes are obtained through π as $\{(\boldsymbol{UO}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\} = \pi^{-1}(\pi(\boldsymbol{U}, \boldsymbol{\tau}))$ and each element $\theta \in \mathcal{M}_{p,k,n}$ can be represented by any $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ such that $\theta = \pi(\bar{\theta})$. In general, geometrical objects on $\mathcal{M}_{p,k,n}$ can

		Tools for Riemmanian optimization		Tools for Riemmanian distances			
Metric	Horizontal space $\mathcal{H}_{\bar{\theta}}$	Riemannian gradient	Retraction	Orthonormal basis of $\mathcal{H}_{\bar{\theta}}$	Distance	Exp.	Log.
Fisher information metric (12)	(13)	(15) (Prop. 2 for L)	(16)	$\sim$	x	x	x
Decoupled metric (17)	(13)	$\sim$	$\sim$	Prop. 5	(18)-(19)	(21)	(20)

TABLE I: Summary of the geometric tools (and their intended use) obtained for $\mathcal{M}_{p,k,n}$. Symbol $\sim$ means that it is not provided in this paper but that it could be easily derived; and symbol x means that it is complicated to find and remains unknown.

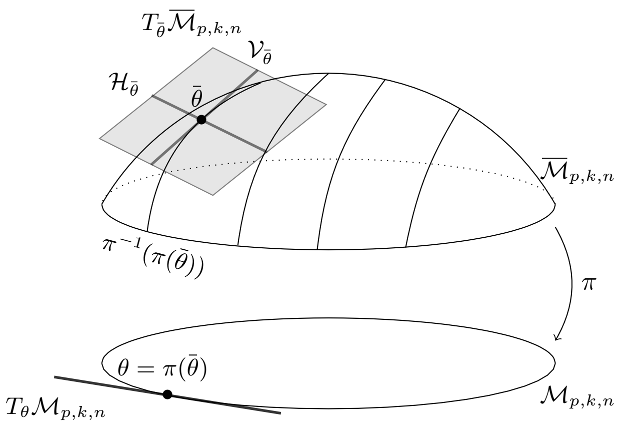

$T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$
$\mathcal{V}_{\bar{\theta}}$
$\mathcal{H}_{\bar{\theta}}$
$\bar{\theta}$
$\overline{\mathcal{M}}_{p,k,n}$
$\pi^{-1}(\pi(\bar{\theta}))$
π
$\theta = \pi(\bar{\theta})$
$T_{\theta}\mathcal{M}_{p,k,n}$
$\mathcal{M}_{p,k,n}$

Fig. 1: Illustration of the quotient $\mathcal{M}_{p,k,n}$ represented by elements of $\overline{\mathcal{M}}_{p,k,n}$. The set of all representations of $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ is the equivalence class $\pi^{-1}(\pi(\bar{\theta})) \subset \overline{\mathcal{M}}_{p,k,n}$. The tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ can be decomposed into the vertical space $\mathcal{V}_{\bar{\theta}} = T_{\boldsymbol{U}}\pi^{-1}(\pi(\bar{\theta}))$ and its orthogonal complement, the horizontal space $\mathcal{H}_{\bar{\theta}}$, which provides proper representatives for tangent vectors in $T_{\theta}\mathcal{M}_{p,k,n}$.

be represented by objects on $\overline{\mathcal{M}}_{p,k,n}$. A schematic illustration of the quotient manifold is provided in Figure 1.

The tangent space $T_{\theta}\mathcal{M}_{p,k,n}$ of $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ can be represented by a subspace of the tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$. First, we note that

$$\begin{aligned} T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n} &= T_{\boldsymbol{U}}\mathrm{St}_{p,k} \times T_{\boldsymbol{\tau}}(\mathbb{R}^{++})^n \\ &= \{(\boldsymbol{\xi_U}, \boldsymbol{\xi_\tau}) \in \mathbb{C}^{p\times k} \times \mathbb{R}^n : \boldsymbol{U}^H\boldsymbol{\xi_U} + \boldsymbol{\xi_U}^H\boldsymbol{U} = \boldsymbol{0}\}. \end{aligned} \tag{9}$$

thanks to $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ being a product manifold, and standard results on $\mathrm{St}_{p,k}$ and $(\mathbb{R}^{++})^n$ respectively. The tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ can now be decomposed into two complementary subspaces: the vertical and horizontal subspaces [14]. The vertical space is defined as the tangent space $T_{\bar{\theta}}\pi^{-1}(\pi(\bar{\theta}))$ of the equivalence class $\pi^{-1}(\pi(\bar{\theta}))$ at $\bar{\theta}$. In the case of $\mathcal{M}_{p,k,n}$, the vertical space at $\bar{\theta}$ is

$$\mathcal{V}_{\bar{\theta}} = \{(\boldsymbol{U}\boldsymbol{A}, \boldsymbol{0}) : \boldsymbol{A} \in \mathcal{H}_k^{\perp}\}, \tag{10}$$

where $\mathcal{H}_k^{\perp} = \{\boldsymbol{A} \in \mathbb{C}^{k\times k} : \boldsymbol{A}^H = -\boldsymbol{A}\}$ is the set of $k \times k$ skew-Hermitian matrices. The orthogonal complement of the vertical space $\mathcal{V}_{\bar{\theta}}$ is the horizontal space $\mathcal{H}_{\bar{\theta}}$, which provides proper representations of the tangent vectors in $T_{\theta}\mathcal{M}_{p,k,n}$: there is a one-to-one correspondance between elements of these two spaces. Note that the notion of orthogonal complement is conditioned by the choice of an inner product $\langle\cdot,\cdot\rangle_{\bar{\theta}}$ defined on $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, which will also turn $\mathcal{M}_{p,k,n}$ into a Riemannian manifold.

Indeed, a Riemannian manifold is a manifold endowed with a Riemannian metric (inner product defined for every tangent space). In the case of a Riemannian quotient manifold, such metric can be represented by a metric on $\overline{\mathcal{M}}_{p,k,n}$, i.e., an inner product $\langle\cdot,\cdot\rangle_{\bar{\theta}}$ defined for $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ at each point $\bar{\theta}$. Still, for $\mathcal{M}_{p,k,n}$ to be properly defined as a Riemannian quotient manifold, this metric on $\overline{\mathcal{M}}_{p,k,n}$ has to be invariant along each equivalence class. In our case, for all $\boldsymbol{O} \in \mathcal{U}_k$, $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, $\bar{\xi} = (\boldsymbol{\xi_U}, \boldsymbol{\xi_\tau})$ and $\bar{\eta} = (\boldsymbol{\eta_U}, \boldsymbol{\eta_\tau})$ in $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, we must have

$$\langle\bar{\xi}, \bar{\eta}\rangle_{\bar{\theta}} = \langle(\boldsymbol{\xi_U}\boldsymbol{O}, \boldsymbol{\xi_\tau}), (\boldsymbol{\eta_U}\boldsymbol{O}, \boldsymbol{\eta_\tau})\rangle_{(\boldsymbol{U}\boldsymbol{O},\boldsymbol{\tau})}. \tag{11}$$

The choice of such Riemannian metric on $\overline{\mathcal{M}}_{p,k,n}$ will then induce a specific geometry (and corresponding theoretical tools) for this space.

B. Fisher information metric: geometry for optimization

First, we consider the geometry resulting from the Fisher information metric of corresponding to likelihood (5) on $\overline{\mathcal{M}}_{p,k,n}$. Since the statistical model is invariant along equivalence classes, the corresponding Fisher metric satisfies (11). It thus induces a Riemannian metric onto $\mathcal{M}_{p,k,n}$. To do so, we first derive this metric in Proposition 1.

Proposition 1 (Fisher information metric)**.** *The Fisher information metric at $\bar{\theta}$ corresponding to the negative likelihood* (5) *is, for all $\bar{\xi}$, $\bar{\eta} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$,*

$$\begin{aligned} \langle\bar{\xi}, \bar{\eta}\rangle_{\bar{\theta}}^{\mathrm{FIM}} &= 2\, n\, c_{\boldsymbol{\tau}}\, \mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{\xi_U}^H \boldsymbol{\eta_U}\right)\right) \\ &\quad + k\left(\boldsymbol{\xi_\tau} \odot (\boldsymbol{1} + \boldsymbol{\tau})^{\odot -1}\right)^T \left(\boldsymbol{\eta_\tau} \odot (\boldsymbol{1} + \boldsymbol{\tau})^{\odot -1}\right), \end{aligned} \tag{12}$$

where $c_{\boldsymbol{\tau}} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \dfrac{\tau_i^2}{1+\tau_i}$.

Proof. See Appendix A. □

The part of the Fisher metric in the above proposition which is related to $\boldsymbol{U}$, *i.e.*, the part that depends on components $\boldsymbol{\xi_U}$ and $\boldsymbol{\eta_U}$, is equal to the classical metric on Grassmann [14], [15], [24], up to the factor $2nc_{\boldsymbol{\tau}}$. We can also note that this factor does not affect the classical definition of the horizontal space of the Grassmann manifold. This directly yields that the horizontal space $\mathcal{H}_{\bar{\theta}}$ in $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ associated with the metric of Proposition 1 is

$$\mathcal{H}_{\bar{\theta}} = \{(\boldsymbol{\xi_U}, \boldsymbol{\xi_\tau}) \in \mathbb{C}^{p\times k} \times \mathbb{R}^n : \boldsymbol{U}^H\boldsymbol{\xi_U} = \boldsymbol{0}\}. \tag{13}$$

Unfortunately, the geometry of $\mathcal{M}_{p,k,n}$ associated with the Fisher information metric of Proposition 1 is complicated to fully characterize. In particular, finding the geodesics of $\mathcal{M}_{p,k,n}$ (curves of minimal length between two points in

$\mathcal{M}_{p,k,n}$) is very hard because of the factor $c_{\boldsymbol{\tau}}$ in the metric. In this part, we will focus on the use of the Fisher information metric in the framework of Riemannian optimization [14]. Alternate tractable geometric tools regarding geodesics and distance measurements (Riemannian exponential and logarithm mapping, Riemannian distance), will be obtained from a decoupled metric in Section III-C.

We will consider optimization problems of the form

$$\underset{\theta \in \mathcal{M}_{p,k,n}}{\text{minimize}} \;\; f(\theta) \tag{14}$$

for a cost function $f : \mathcal{M}_{p,k,n} \to \mathbb{R}$, induced by $\bar{f} : \overline{\mathcal{M}}_{p,k,n} \to \mathbb{R}$ invariant along equivalence classes (*i.e.*, $\bar{f} = f \circ \pi$). In order to perform first order Riemannian optimization algorithms, we essentially need two tools: the Riemannian gradient and a retraction (operator transforming tangent vectors into points onto the manifold) [14].

The Riemannian gradient $\operatorname{grad} f(\theta)$ of f at $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ is represented by the Riemannian gradient $\operatorname{grad} \bar{f}(\bar{\theta})$ of $\bar{f}$ at $\bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$. By definition, the gradient is the only tangent vector in $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ satisfying

$$\forall\ \bar{\xi} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n},\;\; \mathrm{D}\,\bar{f}(\bar{\theta})[\bar{\xi}] = \langle \operatorname{grad} \bar{f}(\bar{\theta}), \bar{\xi} \rangle_{\bar{\theta}}^{\mathrm{FIM}}. \tag{15}$$

Note that this vector always belongs to the horizontal space $\mathcal{H}_{\bar{\theta}}$ due to the invariance of $\bar{f}$ along equivalence classes. In upcoming sections, this definition of the Riemannian gradient will then be used to construct descent direction depending on the considered cost function and optimization algorithm.

To obtain a point on $\mathcal{M}_{p,k,n}$ from a descent direction (vector in $\mathcal{H}_{\bar{\theta}}$) one needs a retraction, i.e., an operator $R_\theta : T_\theta\mathcal{M}_{p,k,n} \to \mathcal{M}_{p,k,n}$ which maps tangent vectors onto the manifold. Such retraction on $\mathcal{M}_{p,k,n}$ can be obtained by a retraction on $\overline{\mathcal{M}}_{p,k,n}$ (denoted $\bar{R}_{\bar{\theta}} : T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n} \to \overline{\mathcal{M}}_{p,k,n}$) using the relation $R_\theta(\xi) = \pi(\bar{R}_{\bar{\theta}}(\bar{\xi}))$. This requires two conditions

1) $\bar{R}_{\bar{\theta}}$ is a proper retraction on $\overline{\mathcal{M}}_{p,k,n}$: $\forall\ \bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, $\bar{R}_{\bar{\theta}}(0) = \bar{\theta}$ and $\mathrm{D}\,\bar{R}_{\bar{\theta}}(0)[\bar{\xi}] = \bar{\xi}$.
2) The induced retraction on $\mathcal{M}_{p,k,n}$ invariant along the equivalence classes: in our case, this translates into $\pi(\bar{R}_{\bar{\theta}}(\bar{\xi})) = \pi(\bar{R}_{(\boldsymbol{U}\boldsymbol{O},\boldsymbol{\tau})}((\boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{O}, \boldsymbol{\xi}_{\boldsymbol{\tau}}))$, for all $\boldsymbol{O} \in \mathcal{U}_k$, $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$.

Notice that the notion of retraction does not depend on the choice of the metric, so several options are generally available. In this paper, we consider the following retraction from classical results on $\mathrm{St}_{p,k}$ [26] and $(\mathbb{R}^{++})^n$. This retracion defined on $\overline{\mathcal{M}}_{p,k,n}$ for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathcal{H}_{\bar{\theta}}$ as

$$\bar{R}_{\bar{\theta}}(\bar{\xi}) = \left(\boldsymbol{X}\boldsymbol{Y}^H, \boldsymbol{\tau} + \boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{1}{2}\boldsymbol{\tau}^{\odot -1}\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \right), \tag{16}$$

where $\boldsymbol{U} + \boldsymbol{\xi}_{\boldsymbol{U}} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}^H$ is the thin SVD. Notice that for the part that concerns $\boldsymbol{\tau}$, we have a second degree polynom in $\boldsymbol{\xi}_{\boldsymbol{\tau}}$ with a negative discriminant, thus the resulting vector contains strictly positive numbers. It can be checked that the two conditions are satisfied, and this option was chosen for its numerical stability.

C. Decoupled metric: geometry for distances

Riemannian distances can be used either for performance assessment, or in machine learning algorithms (e.g. for clustering). Their interest can notably be their natural invariances with respect to the manifold and/or metric of interest. These distances are obtained by measuring the length of geodesics, which generalize straight lines onto manifolds while taking into account the curvature induced by the metric and geometric constraints. Unfortunately the Riemannian distance induced by the Fisher information metric of Proposition 1 cannot be obtained in closed-form. To overcome this difficulty, we propose to use a decoupled metric from the following definition.

Definition 1 (Decoupled metric)**.** *The Riemannian metric* $\langle \cdot, \cdot \rangle_{\cdot}$ *is defined, for all* $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$*,* $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}})$ *and* $\bar{\eta} = (\boldsymbol{\eta}_{\boldsymbol{U}}, \boldsymbol{\eta}_{\boldsymbol{\tau}}) \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$*, as*

$$\begin{aligned} \langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}} = \alpha \mathfrak{Re} \left(\operatorname{Tr} \left(\boldsymbol{\xi}_{\boldsymbol{U}}^H \boldsymbol{\eta}_{\boldsymbol{U}} \right) \right) \\ + \beta \left(\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1} \right)^T \left(\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1} \right), \end{aligned} \tag{17}$$

where $\alpha, \beta > 0$.

Notice that the decoupled metric has a structure similar to the Fisher information metric in Proposition A: it consists in a scaled combination of standard metrics on $\mathrm{Gr}_{p,k}$ [14], [15], [24] and $(\mathbb{R}^{++})^n$ [27]. The main difference is that the weights α and β remain constant in the decoupled metric, which will yield a geometry from well-known results. Another particular interest is that the flexibility regarding this factors allows emphasizing a parameter (subspace spanned by $\boldsymbol{U}$ or textures $\boldsymbol{\tau}$) in the considered geometry. This is notably interesting for clustering applications (see Section V) where we want to control the importance of each feature.

First, one can check that the horizontal space at $\bar{\theta}$ in $\overline{\mathcal{M}}_{p,k,n}$ for the Riemannian metric in Definition 1 is the same as the one given in (13) corresponding to the Fisher information metric of Proposition 1. It is thus also denoted $\mathcal{H}_{\bar{\theta}}$ in the following.

Second, we can deduce several geometric tools from classical results about $\mathrm{Gr}_{p,k}$ in [14], [15], [24] and $(\mathbb{R}^{++})^n$ in [27]. The squared Riemannian distance between $\theta_1 = \pi(\bar{\theta}_1)$ and $\theta_2 = \pi(\bar{\theta}_2)$ in $\mathcal{M}_{p,k,n}$ is given by

$$d^2_{\mathcal{M}_{p,k,n}}(\theta_1, \theta_2) = \alpha d^2_{\mathrm{Gr}_{p,k}}(\boldsymbol{U}_1, \boldsymbol{U}_2) + \beta d^2_{(\mathbb{R}^{++})^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2), \tag{18}$$

where $d^2_{\mathrm{Gr}_{p,k}}$ and $d^2_{(\mathbb{R}^{++})^n}$ are the squared Riemannian distances of $\mathrm{Gr}_{p,k}$ and $(\mathbb{R}^{++})^n$, respectively. They are defined as

$$\begin{aligned} d^2_{\mathrm{Gr}_{p,k}}(\boldsymbol{U}_1, \boldsymbol{U}_2) &= \|\boldsymbol{\Theta}\|_2^2, \\ d^2_{(\mathbb{R}^{++})^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) &= \|\log(\boldsymbol{\tau}_1) - \log(\boldsymbol{\tau}_2)\|_2^2, \end{aligned} \tag{19}$$

where $\boldsymbol{\Theta}$ is obtained from the SVD $\boldsymbol{U}_1^H\boldsymbol{U}_2 = \boldsymbol{O}_1 \cos(\boldsymbol{\Theta})\boldsymbol{O}_2^H$. An additional tool linked to the Riemannian distance is the Riemannian logarithm mapping. Given a reference point $\theta_1 = \pi(\bar{\theta}_1)$ and a second point $\theta_2 = \pi(\bar{\theta}_2)$ both in $\mathcal{M}_{p,k,n}$, the Riemannian logarithm mapping is an operator that provides a vector of $T_{\theta_1}\mathcal{M}_{p,k,n}$ that points towards θ_2 and whose squared norm with respect to the metric in (17) is $d^2_{\mathcal{M}_{p,k,n}}(\theta_1, \theta_2)$

(as defined in (18)). Here, the representation in $\mathcal{H}_{\bar{\theta}_1}$ of the Riemannian logarithm mapping on $\mathcal{M}_{p,k,n}$ is

$$\begin{aligned}
&\log_{\bar{\theta}_1}(\bar{\theta}_2) = \left(\log^{\mathrm{Gr}_{p,k}}_{\boldsymbol{U}_1}(\boldsymbol{U}_2),\ \log^{(\mathbb{R}^{++})^n}_{\boldsymbol{\tau}_1}(\boldsymbol{\tau}_2)\right),\\
&\log^{\mathrm{Gr}_{p,k}}_{\boldsymbol{U}_1}(\boldsymbol{U}_2) = \boldsymbol{X}\boldsymbol{\Theta}\boldsymbol{Y}^H, \qquad (20)\\
&\log^{(\mathbb{R}^{++})^n}_{\boldsymbol{\tau}_1}(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1 \odot \log(\boldsymbol{\tau}_1^{\odot -1} \odot \boldsymbol{\tau}_2),
\end{aligned}$$

where $\boldsymbol{X}\boldsymbol{\Theta}\boldsymbol{Y}^H$ is defined through the SVD $(\boldsymbol{I}_p - \boldsymbol{U}_1\boldsymbol{U}_1^H)\boldsymbol{U}_2(\boldsymbol{U}_1^H\boldsymbol{U}_2)^{-1} = \boldsymbol{X}\tan(\boldsymbol{\Theta})\boldsymbol{Y}^H$ Conversely, the inverse of this application is the Riemannian exponential mapping on $\mathcal{M}_{p,k,n}$, whose representation in $\overline{\mathcal{M}}_{p,k,n}$, for $\bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathcal{H}_{\bar{\theta}}$ is given by

$$\begin{aligned}
&\exp_{\bar{\theta}}(\bar{\xi}) = \left(\exp^{\mathrm{Gr}_{p,k}}_{\boldsymbol{U}}(\boldsymbol{\xi}_{\boldsymbol{U}}),\ \exp^{(\mathbb{R}^{++})^n}_{\boldsymbol{\tau}}(\boldsymbol{\xi}_{\boldsymbol{\tau}})\right),\\
&\exp^{\mathrm{Gr}_{p,k}}_{\boldsymbol{U}}(\boldsymbol{\xi}_{\boldsymbol{U}}) = \boldsymbol{U}\boldsymbol{Y}\cos(\boldsymbol{\Sigma}) + \boldsymbol{X}\sin(\boldsymbol{\Sigma}), \qquad (21)\\
&\exp^{(\mathbb{R}^{++})^n}_{\boldsymbol{\tau}}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = \boldsymbol{\tau} \odot \exp(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}),
\end{aligned}$$

where $\boldsymbol{\xi}_{\boldsymbol{U}} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}^H$ is the SVD such that $\boldsymbol{X} \in \mathbb{C}^{p\times k}$ and $\boldsymbol{\Sigma},\ \boldsymbol{Y} \in \mathbb{C}^{k\times k}$. These operators provide mappings between the manifold and its tangent space, which will notably be instrumental in in Section IV-B to define an estimation error vector, and in Section V in order to define Riemannian means.

IV. Parameter estimation

A. MLE with Riemannian optimization

In this section, we cast the MLE as an optimization problem on $\mathcal{M}_{p,k,n}$, i.e. we seek to solve:

$$\theta^\star = \underset{\theta \in \mathcal{M}_{p,k,n}}{\arg\min}\ L(\theta), \qquad (22)$$

where $L : \mathcal{M}_{p,k,n} \to \mathbb{R}$ is the negative log-likelihood defined in (5). To solve this estimation problem, a block coordinate descent (BCD) has been proposed in [12]. Here, we present an alternative algorithm leveraging the information geometry presented in Section III-B.

A first alternative is to use a Riemannian gradient descent (RGD) [14]. An iteration of this algorithm consists in computing the gradient of L and then retracting minus the gradient multiplied by a step size. Given the iterate $\theta^{(t)}$ represented by $\bar{\theta}^{(t)}$, the RGD algorithm yields

$$\bar{\theta}^{(t+1)} = \bar{R}_{\bar{\theta}^{(t)}}\left(-\nu_t \operatorname{grad}\bar{L}(\bar{\theta}^{(t)})\right), \qquad (23)$$

where ν_t is a step size, $\operatorname{grad}\bar{L}(\bar{\theta}^{(t)})$ is a representative of the Riemannian gradient associated to the Fisher information metric of Proposition 3, and $\bar{R}_{\bar{\theta}^{(t)}}$ is the retraction defined in (16). Hence, it also corresponds to the so-called natural gradient as defined in [28], which regained interest due to its link with second order optimization methods [29].

Here, we propose a more flexible approach following the recent works [30], [31]: we derive a Riemannian stochastic gradient descent (R-SGD) on $\mathcal{M}_{p,k,n}$. The R-SGD is a Riemannian optimization algorithm that computes the gradient of the function to minimize only on a subset A of all measured signals $\{\boldsymbol{x}_i\}_{i=1}^n$. Hence, contrary to the BCD or the RGD, this algorithm can be used on large scale datasets and the cost of an iteration can be modulated according to the computing capacity. Since the number of samples A can be chosen arbitrarily set, this algorithm also encompasses the "plain" R-SGD ($A = \{\boldsymbol{x}_i\}$) and the classical RGD [14] ($A = \{\boldsymbol{x}_i\}_{i=1}^n$). Additionally, the R-SGD will be shown to have a lower complexity (per iteration) than the BCD.

In order to derive the R-SGD, the negative log-likelihood L defined on $\mathcal{M}_{p,k,n}$ is rewritten

$$L(\theta) = \sum_{i=1}^n L_i(\theta), \qquad (24)$$

where L_i is the negative log-likelihood defined on the sample $\boldsymbol{x}_i$. Hence, the same notation applies to the negative log-likelihood (5) defined on $\overline{\mathcal{M}}_{p,k,n}$: $\bar{L}(\bar{\theta}) = \sum_{i=1}^n \bar{L}_i(\bar{\theta})$. In short, given the actual iterate $\theta^{(t)}$, an iteration of R-SGD proceeds in three steps: *(i)* a set A of samples is randomly drawn from $\{\boldsymbol{x}_i\}_{i=1}^n$, *(ii)* then the gradient of $\sum_{\boldsymbol{x}_i \in A} L_i(\theta^{(t)})$ is computed, *(iii)* finally a new iterate is given by retracting minus the gradient times a step size. Since a retraction on $\mathcal{M}_{p,k,n}$ is provided in Section III-B, the only remaining element to provide is the Riemannian gradient of $L_i(\theta)$. This gradient is given in the following proposition:

Proposition 2 (Riemannian gradient)**.** *Given $\theta = \pi(\boldsymbol{U}, \boldsymbol{\tau}) \in \mathcal{M}_{p,k,n}$ represented by $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, the representative in $\mathcal{H}_{\boldsymbol{U}} \times T_{\boldsymbol{\tau}}(\mathbb{R}^{++})^n$ of the Riemannian gradient of L_i at θ is*

$$\operatorname{grad}\bar{L}_i(\bar{\theta}) = (\boldsymbol{G}_{\boldsymbol{U}}, \boldsymbol{G}_{\boldsymbol{\tau}})$$

where

$$\boldsymbol{G}_{\boldsymbol{U}} = -\frac{\tau_i}{nc_{\boldsymbol{\tau}}(1+\tau_i)}(\boldsymbol{I}_p - \boldsymbol{U}\boldsymbol{U}^H)\boldsymbol{x}_i\boldsymbol{x}_i^H\boldsymbol{U},$$

and the j^{th} element of $\boldsymbol{G}_{\boldsymbol{\tau}}$ is

$$(\boldsymbol{G}_{\boldsymbol{\tau}})_j = \begin{cases} 1 + \tau_i - \frac{1}{k}\boldsymbol{x}_i^H\boldsymbol{U}\boldsymbol{U}^H\boldsymbol{x}_i & \textit{for} \quad j = i \\ 0 \quad \textit{otherwise}. \end{cases}$$

Proof. See appendix B. □

Following from this gradient, the resulting R-SGD on $\mathcal{M}_{p,k,n}$ is detailed in the box Algorithm 1. Concerning the computation of the step size, several options exist. When the gradient is computed on all data, *i.e* $A = \{\boldsymbol{x}_i\}_{i=1}^n$, a line search (*e.g.* [14, 4.2]) is recommended. When the gradient is computed on a subset of all data, a step size proportional to $1/t$, where t is the number of iterations, can be used as in [28].

By rearranging the operations of $\boldsymbol{G}_{\boldsymbol{U}}$ in Proposition 2, the computational complexity of the gradient of $\sum_{\boldsymbol{x}_i \in A} L_i(\theta)$ is $\mathcal{O}(mpk+n)$, where m the number of samples in A. In practice, $c_{\boldsymbol{\tau}}$ can be approximated using only the textures associated with the samples in A, *i.e.* $c_{\boldsymbol{\tau}} \approx \frac{1}{m}\sum_{\boldsymbol{x}_i \in A} \frac{\tau_i^2}{1+\tau_i}$. Hence, the complexity of the gradient becomes $\mathcal{O}(mpk)$. Then, the complexity of the retraction (16) is $\mathcal{O}(pk^2 + m)$, as we only retract the non-zero elements of the gradient $\boldsymbol{G}_{\boldsymbol{\tau}}$ from Proposition 2. Hence, the total complexity of each iteration of Algorithm 1 is $\mathcal{O}(mpk + pk^2)$, which is much lower than the $\mathcal{O}(np^2 + p^3)$ of the BCD in [12] (which involves the SVD of the scaled SCM at each step).

Algorithm 1: Riemannian stochastic gradient descent

```
Input  : Initial iterate $\bar{\theta}^{(1)} \in \overline{\mathcal{M}}_{p,k,n}$.
Output: Sequence of iterates $\{\bar{\theta}^{(t)}\}$.
$t = 1$
while no convergence do
    Randomly draw a subset $A \subset \{\boldsymbol{x}_i\}_{i=1}^n$ and set
      $\bar{\xi}^{(t)} = \sum_{x_i \in A} \operatorname{grad} \bar{L}_i(\bar{\theta}^{(t)})$
    Compute a step size $\nu_t$ and set
      $\bar{\theta}^{(t+1)} = \bar{R}_{\bar{\theta}^{(t)}}(-\nu_t \bar{\xi}^{(t)})$
    $t = t + 1$
end
```

B. Intrinsic Cramér-Rao bounds

Obtaining performance bounds for the model in (3) is a complex issue, notably because the signal subspace is represented by a point in $\mathrm{Gr}_{p,k}$. A first approach was proposed in [7] for the model $\boldsymbol{x}_i \sim \mathcal{CN}(\boldsymbol{0}, \tau_i \boldsymbol{G}\boldsymbol{G}^H + \boldsymbol{I})$, where $\boldsymbol{G} \in \mathbb{C}^{p\times k}$ is a lower-triangular matrix with positive diagonal elements. Such parameterization is carefully chosen in order to obtain a minimal and essentially unconstrained parametrization of the low-rank signal covariance matrix. This allows obtaining the standard Cramér-Rao inequality for the parameter $\boldsymbol{g} = \mathrm{vec}(\boldsymbol{G})$. In a second step, the signal subspace is represented by the orthogonal projection matrix $\boldsymbol{\Pi} = \boldsymbol{G}(\boldsymbol{G}^H\boldsymbol{G})^{-1}\boldsymbol{G}^H$ and the CRB for $\boldsymbol{\pi} = \mathrm{vec}(\boldsymbol{\Pi})$ is obtained as

$$\mathrm{CRB}(\boldsymbol{\pi}) = \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{g}^T}\mathrm{CRB}(\mathbf{g})\frac{\partial \boldsymbol{\pi}^T}{\partial \mathbf{g}} \;\Rightarrow\; \mathbb{E}\left[||\boldsymbol{\Pi} - \hat{\boldsymbol{\Pi}}||_F^2\right] \geq \mathrm{Tr}\left\{\mathrm{CRB}(\boldsymbol{\pi})\right\} \tag{25}$$

thus enabling to assess approximately the minimum distance between the estimated and the true signal subspace. Another option could have been to start with the constrained parameterization $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{D}^{1/2}$ and to directly handle the orthonormality constraints $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}_k$ with the the theory of constrained CRLBs [32]–[35] to obtain $\mathrm{CRB}(\mathrm{vec}(\boldsymbol{U}))$, then deriving the same result as in (25) from $\boldsymbol{\pi} = \mathrm{vec}(\boldsymbol{U}\boldsymbol{U}^H)$. This method is expected to yield the same result as in [7] from a different path of derivations.

While the obtained inequality in (25) allows for an analysis with numerical experiments, it still lacks some interpretable closed-form. In the following, we will directly treat the signal subspace as a point in $\mathrm{Gr}_{p,k}$[1] and rely on the intrinsic CRLB theory from [18], [37]. The interest is twofold: first it will yield a simple and interpretable closed form for the bound on the subspace estimation. Second, this bound will be obtained for natural distance on $\mathrm{Gr}_{p,k}$ in (19), which is expected to better reflect breakdown points at low sample support (cf. [18] for an example regarding covariance matrix estimation).

Intrinsic (i.e., manifold oriented) versions of the Cramér-Rao inequality have been established [18] and extended to quotient manifolds in [37]. The main difference compared to the classical CRLBs is that the parameter θ is treated as being in a Riemmanian manifold endowed by an arbitrary chosen "error" metric. The estimation error is thus measured using the Riemannian distance d that emanated from this error metric. The obtained inequality is of the form

$$\boldsymbol{C} \succcurlyeq \boldsymbol{F}^{-1} + \text{ curvature terms,} \tag{26}$$

where $\boldsymbol{C}$ is the covariance matrix of the error vector (defined as the Riemannian logarithm mapping $\log_\theta(\hat{\theta})$, which is induced by the error metric), and $\boldsymbol{F}^{-1}$ is the inverse of the Fisher information matrix (which depends on both the chosen metric and the Fisher information metric). Neglecting the curvature terms and taking the trace of (26) yields the inequality $\mathbb{E}\left[d^2(\theta, \hat{\theta})\right] \geq \mathrm{Tr}(\boldsymbol{F}^{-1})$ for an unbiased estimator $\hat{\theta}$, which will be here our primary interest.

In our context, we consider $\mathcal{M}_{p,k,n}$ endowed with the decoupled metric in (17) (Definition 1) in order to bound the error measure defined by $d^2_{\mathcal{M}_{p,k,n}}$ as in (18). For the sake of exposition, the obtained results are directly reported in the two following propositions, while the technical details are let in the Appendix C.

Proposition 3 (Fisher information matrix)**.** *The Fisher information matrix $\boldsymbol{F}_\theta$ on $\mathcal{M}_{p,k,n}$ admits the structure*

$$\boldsymbol{F}_\theta = \begin{pmatrix} \boldsymbol{F}_{\boldsymbol{U}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{\boldsymbol{\tau}} \end{pmatrix},$$

with the blocks $\boldsymbol{F}_{\boldsymbol{U}} = 2\,\alpha^{-1}\,n\,c_{\boldsymbol{\tau}}\,\boldsymbol{I}_{2(p-k)k}$, and $\boldsymbol{F}_{\boldsymbol{\tau}} = \beta^{-1}\,k\,\mathrm{diag}\left(\boldsymbol{\tau}^{\odot 2} \odot (\boldsymbol{1} + \boldsymbol{\tau})^{\odot -2}\right)$, and where $\mathrm{diag}(\cdot)$ *returns the diagonal matrix formed with the elements of its argument.*

Proof. See Appendix C. □

Proposition 4 (iCRLB)**.** *Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be a sample set following the model in* (3)*. Let $\hat{\theta}$ be an estimate of $\theta \in \mathcal{M}_{p,k,n}$ for the model. The estimation error defined by $d^2_{\mathcal{M}_{p,k,n}}$ as in* (18) *is bounded as*

$$\mathbb{E}[d^2_{\mathcal{M}_{p,k,n}}(\hat{\theta}, \theta)] \geq \alpha\,\mathrm{CRB}_{\boldsymbol{U}} + \beta\,\mathrm{CRB}_{\boldsymbol{\tau}}. \tag{27}$$

where

$$\mathrm{CRB}_{\boldsymbol{U}} = \frac{(p-k)\,k}{n\,c_{\boldsymbol{\tau}}} \quad \text{and} \quad \mathrm{CRB}_{\boldsymbol{\tau}} = \frac{1}{k}\sum_{i=1}^{n}\frac{(1+\tau_i)^2}{\tau_i^2}.$$

Furthermore, two iCRLB, on $\mathrm{Gr}_{p,k}$ *and* $(\mathbb{R}^{++})^n$ *respectively, are given by*

$$\mathbb{E}[d^2_{\mathrm{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}), \pi(\boldsymbol{U}))] \geq \mathrm{CRB}_{\boldsymbol{U}}, \tag{28}$$

$$\mathbb{E}[d^2_{(\mathbb{R}^{++})^n}(\hat{\boldsymbol{\tau}}, \boldsymbol{\tau})] \geq \mathrm{CRB}_{\boldsymbol{\tau}}. \tag{29}$$

Proof. See Appendix C. □

Notice that the problem of estimating a subspace should not depend on its basis $\boldsymbol{U}$, as two estimates $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{U}}\boldsymbol{Q}$ yield the same subspace estimate (but would yield different MSEs for the basis $\boldsymbol{U}$). The obtained bound on $d^2_{\mathrm{Gr}_{p,k}}$ satisfies this property. Furthermore, Proposition 4 shows that the subspace estimation problem for model (3) does not depend on the underlying subspace itself, but rather only on its dimension and the SNR, which is theoretically appealing. Conversely,

[1] Note that we consider the case of equal eigenvalues, but this restriction has been carefully motivated in the model introduction section. The extension to the general case could be considered using recent derivations from [36] but this complex issue goes far beyond the scope of the paper.

the euclidean CRLBs in [7], bounding the MSE on $\boldsymbol{U}\boldsymbol{U}^H$ (orthogonal projector) as in (25) does not exhibit such direct interpretability. Finally, in the specific case of data following a Gaussian low-rank (spiked) model for which $\tau_i = \text{SNR}$ so that $\boldsymbol{x} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \text{SNR} \times \boldsymbol{U}\boldsymbol{U}^H + \boldsymbol{I}_p)$, we retrieve the iCRLB of [18, Eq.145], *i.e.*,

$$\mathbb{E}[d^2_{\text{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}), \pi(\boldsymbol{U}))] \geq \frac{(p-k)\, k\, (1+\text{SNR})}{n\, \text{SNR}^2}. \tag{30}$$

V. Riemannian clustering

In this section, we apply the statistical model developed in Section II with its Riemannian geometry $\mathcal{M}_{p,k,n}$, presented in Section III-C, to clustering problems. More specifically, we assume that we have M batches $\boldsymbol{X}_i$ (*e.g.* sets of local pixels of an image, EEG epochs of measurements, *etc*). Each $\boldsymbol{X}_i \in \mathbb{C}^{p \times n}$ is a column-wise concatenation of n observations $\boldsymbol{x}_i \in \mathbb{C}^p$ defined in Section II. Furthermore, each batch $\boldsymbol{X}_i$ belongs to an unknown class $y \in [\![1, K]\!]$.

The use of statistical descriptors is a classical procedure in machine learning as they are often more discriminative than raw data (see *e.g.* [19], [20]). Hence, we begin by estimating a descriptor $\theta_i \in \mathcal{M}_{p,k,n}$ of the batch $\boldsymbol{X}_i$ following Section IV-A. Then, the aim is to partition the descriptors $\{\theta_i\}_{i=1}^M$ in $S = \{S_1, S_2, \cdots, S_K\}$. Thus, we get a partition of the original batches $\{\boldsymbol{X}_i\}_{i=1}^M$.

Each parameter θ_i is represented by a couple, *i.e.* $\theta_i = \pi(\boldsymbol{U}_i, \boldsymbol{\tau}_i)$. Our contribution is to cluster both components (subspace and power) in a unified manner, leveraging the geometry of $\mathcal{M}_{p,k,n}$ featured in Section III-C. This section is focused on the application of a *K-means++* [21] on $\mathcal{M}_{p,k,n}$ with the tools developed earlier. However, the proposed methodology is flexible: *(i)* descriptors θ_i can be replaced by other statistical estimates with their associated Riemannian geometries, *(ii)* many Euclidean based clustering algorithms can be transformed to Riemannian ones (replacing distances and means by their Riemannian counterparts).

A. Distance and mean computations

Most clustering algorithms, including *K-means++* [21], rely on distance and mean computations. Since θ_i lies on a Riemannian manifold we first need to define distance and mean computations other than simple Euclidean ones.

A natural choice is the use of the distance $d_{\mathcal{M}_{p,k,n}}$ defined in (18). In the context of clustering, the distance on $\text{Gr}_{p,k}$ and the one on $(\mathbb{R}^{++})^n$ do not necessarily have the same amplitude or the same ability to discriminate. Thus, the parameters α, β of the metric of Definition 1 are to be chosen carefully. We propose a 2-step strategy to select α, β: (*i*) correction of the scale effect and (*ii*) choice of a trade-off between the distances on $\text{Gr}_{p,k}$ and $(\mathbb{R}^{++})^n$. To correct the scale effect we propose to normalize the squared distances by their mean values on the samples $\{\theta_i\}_{i=1}^M$. Then, a trade-off can be made between the two distances. More precisely, $\forall \gamma \in [0,1]$, we define

$$\begin{aligned} \alpha &= \frac{1-\gamma}{\frac{1}{M^2}\sum_{q,l \in [\![1,M]\!]} d^2_{\text{Gr}_{p,k}}(\boldsymbol{U}_q, \boldsymbol{U}_l)}, \\ \beta &= \frac{\gamma}{\frac{1}{M^2}\sum_{q,l \in [\![1,M]\!]} d^2_{(\mathbb{R}^{++})^n}(\boldsymbol{\tau}_q, \boldsymbol{\tau}_l)}. \end{aligned} \tag{31}$$

It remains to define a mean computation algorithm on a set of parameters S_j. In [38], the mean of a set of points on a Riemannian manifold is defined as the minimizer of the variance of this set. Let $m = \#S_j$, the variance V of S_j at $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ is defined as,

$$V(\bar{\theta}) = \frac{1}{m} \sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(\theta, \theta_i). \tag{32}$$

The mean $c = \pi(\bar{c}) \in \mathcal{M}_{p,k,n}$ of the set of points S_j is obtained from the minimization of the variance,

$$\bar{c} = \underset{\theta \in \mathcal{M}_{p,k,n}}{\arg\min}\ \frac{1}{2} V(\bar{\theta}). \tag{33}$$

Denoting $\bar{c} = (\boldsymbol{U}, \boldsymbol{\tau})$, one can check that the mean $\boldsymbol{\tau}$ corresponding to the distance $d_{(\mathbb{R}^{++})^n}$ is simply the geometric mean

$$\boldsymbol{\tau} = \left(\prod_{\theta_i \in S_j}^{\odot} \boldsymbol{\tau}_i \right)^{\odot\, 1/m}, \tag{34}$$

where $\prod^{\odot}$ is the elementwise product. Similarly, the mean corresponding to distance $d_{\text{Gr}_{p,k}}$ is well-known [24]. Unfortunately, no closed form is known to compute it. It is obtained through the following iterative procedure: given $\boldsymbol{U}^{(t)}$, the iterate $\boldsymbol{U}^{(t+1)}$ is obtained with

$$\boldsymbol{U}^{(t+1)} = \exp^{\text{Gr}_{p,k}}_{\boldsymbol{U}^{(t)}} \left(\frac{\nu_t}{m} \sum_{\theta_i \in S_j} \log^{\text{Gr}_{p,k}}_{\boldsymbol{U}^{(t)}}(\boldsymbol{U}_i) \right), \tag{35}$$

where ν_t is the step size which can be computed thanks to a line search [14]. Since we get one mean per class, in the rest of the paper, the mean of S_j is noted c_j.

B. *K-means++* on $\mathcal{M}_{p,k,n}$

With the distance and mean computation algorithms explained above, we use a *K-means++* on $\mathcal{M}_{p,k,n}$ to partition $\{\theta_i\}_{i=1}^M$ in S (and thus partition $\{\boldsymbol{X}_i\}_{i=1}^M$).

Instead of choosing class centers c_j uniformly at random from $\{\theta_i\}_{i=1}^M$, *K-means++* initializes them by recursively choosing a new center θ_i with probability $\frac{D(\theta_i)^2}{\sum_{\theta_j} D(\theta_j)^2}$ [21]. Here, $D(\theta_i)$ denotes the distance $d_{\mathcal{M}_{p,k,n}}$ from θ_i to the closest center among those already chosen.

Once these class centers are initialized, *K-means++* on $\mathcal{M}_{p,k,n}$ iteratively applies two steps [21]:

(i) **Assignment step**: each θ_i is assigned to the cluster S_j whose center c_j is the closest using the distance $d_{\mathcal{M}_{p,k,n}}$,

(ii) **Update step**: each new class center c_j is computed using (34) and (35).

Once terminated, *K-means++* on $\mathcal{M}_{p,k,n}$ outputs the partition S. Intuitively, *K-means++* finds clusters S_j whose points $\theta_i \in S_j$ are close to each other using the distance $d_{\mathcal{M}_{p,k,n}}$.

C. Theoretical properties

To analyze the performance of *K-means++* on $\mathcal{M}_{p,k,n}$, we begin by defining the within-cluster sum of squares (WCSS),

$$\phi(S) = \sum_{j=1}^{K} \sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(c_j, \theta_i). \tag{36}$$

A "good" clustering algorithm finds a partition whose associated ϕ is close to the minimum ϕ_{OPT} of the WCSS (36). In the Euclidean case, [21] establishes that the Euclidean WCSS of a partition produced by *K-means++* is upper bounded with respect to $\phi^{\epsilon}_{\text{OPT}}$ (minimum of the Euclidian WCSS). This property is central to *K-means++* since it is proven that a plain *K-means* [39] cannot admit such a bound. Moreover, this bound is true from the initialization. As stated in [40], this result in the Euclidean case holds for any distance (thus for $d_{\mathcal{M}_{p,k,n}}$) and does not rely on the mean computation. Hence, the WCSS (36) of the *K-means++* initialization on $\mathcal{M}_{p,k,n}$ satisfies

$$\mathbb{E}[\phi] \leq 8(\ln K + 2)\phi_{\text{OPT}} \tag{37}$$

where the expectation is taken with respect to the seeding procedure.

Moreover, "**Assignment step**" and "**Update step**" from Algorithm 2 decrease WCSS (36). Indeed, the "**Assignment step**" directly decreases the WCSS (36) by assigning points $\{\theta_i\}_{i=1}^M$ to the closest centers. Furthermore, we defined, in (33), the mean of S_j as the minimizer of the variance. It follows that $\forall S_j \in S$,

$$\sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(\theta^{(t)}, \theta_i) \geq \sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(\theta^{(t+1)}, \theta_i), \tag{38}$$

where $\theta^{(t)}$ and $\theta^{(t+1)}$ are the means taken before and after the "**Update step**" respectively. Hence, the "**Update step**" decreases the WCSS (36). This implies that the final clustering returned by *K-means++* on $\mathcal{M}_{p,k,n}$ satisfies (37).

However, this clustering is not necessarily a global minimum of WCSS (36). Hence, a standard practice is to run the algorithm several times with different initializations and then to keep the clustering with the lowest inertia (36). *K-means++* on $\mathcal{M}_{p,k,n}$ with the strategy of several initializations is presented in Algorithm 2.

VI. Numerical experiments

A. Simulations

This section illustrates the performance of the Algorithm 1 as well as the Cramér-Rao bounds developed in section IV. The covariance matrix of the simulated data follows the model $\boldsymbol{\Sigma}_i = \boldsymbol{I}_p + \tau_i \boldsymbol{U}\boldsymbol{U}^H$. The basis $\boldsymbol{U}$ is a random matrix in $\text{St}_{p,k}$. The textures τ_i are randomly drawn from a Log-normal$(-\frac{s^2}{2}, s^2)$ multiplied by the desired SNR. Hence, we get $\mathbb{E}[\tau_i] = \text{SNR}$. The shape parameter s^2 controls the heterogeneity of the textures: the higher the s^2, the greater the heterogeneity. We generate sets $\{\boldsymbol{x}_i\}_{i=1}^n$, with $n \in [\![10, 1000]\!]$, from the zero mean complex Gaussian multivariate distribution with covariance $\boldsymbol{\Sigma}_i$. For each value of n, N sets $\{\boldsymbol{x}_i\}_{i=1}^n$ are simulated and estimators $\hat{\boldsymbol{U}}, \hat{\boldsymbol{\tau}}$ are computed in each case.

Algorithm 2: *K-means++* on $\mathcal{M}_{p,k,n}$

```
Input  : A set $\{\theta_i\}_{i=1}^M \subset \mathcal{M}_{p,k,n}$ to partition, a
         number of clusters $K$ and a number of
         initializations $l$.
Output: Best partition $S^\star$.
$\phi^\star \leftarrow +\infty$
for $1$ to $l$ do
    # Initialization
    Take one center $c_1$, chosen uniformly at random
      from $\{\theta_i\}_{i=1}^M$.
    while $\#\{c_i\} < K$ do
        Take a new center $c_j$, choosing $\theta_i \in \{\theta_i\}_{i=1}^M$
          with probability $\frac{D(\theta_i)^2}{\sum_{\theta_m} D(\theta_m)^2}$.
    end
    # K-means
    while no convergence do
        Assignment step: $\forall i \in [\![1, M]\!]$ assign $\theta_i$ to the
          cluster $S_j$ with the nearest $c_j$, $j \in [\![1, K]\!]$.
        Update step: Calculate new centers $c_j$ of
          clusters $S_j$, $\forall j \in [\![1, K]\!]$, using (34) and (35).
    end
    Compute $\phi(S)$ with (36).
    if $\phi(S) < \phi^\star$ then
        $S^\star \leftarrow S$
        $\phi^\star \leftarrow \phi(S)$
    end
end
```

Here are the considered estimators in the simulations:

1) SCM: the k first principal eigenvectors of the SCM of $\{\boldsymbol{x}_i\}_{i=1}^n$ are concatenated to get $\boldsymbol{U}^{\text{SCM}}$.
2) BCD: the MLE estimate is done using BCD algorithm on $\{\boldsymbol{x}_i\}_{i=1}^n$ [12] . The estimators are denoted $\boldsymbol{U}^{\text{BCD}}$ and $\boldsymbol{\tau}^{\text{BCD}}$.
3) RGD: Algorithm 1 is performed using all samples at each iteration, *i.e.* $A = \{\boldsymbol{x}_i\}_{i=1}^n$. Pymanopt library [41] (builds upon the Manopt library [42]) achieves this optimization. The estimators are denoted $\boldsymbol{U}^{\text{RGD}}$ and $\boldsymbol{\tau}^{\text{RGD}}$.

To measure the subspace estimation performance of the considered estimators, we compute the mean squared error (MSE) between the estimators $\hat{\boldsymbol{U}} \in \{\boldsymbol{U}^{\text{SCM}}, \boldsymbol{U}^{\text{BCD}}, \boldsymbol{U}^{\text{RGD}}\}$ and the real parameter $\boldsymbol{U}$. We compute the MSE as the mean squared distance on $\text{Gr}_{p,k}$ (19) between estimated parameters $\hat{\boldsymbol{U}}$ and real parameter $\boldsymbol{U}$. Texture estimation performance is also assessed. The MSE is computed between the estimators $\hat{\boldsymbol{\tau}} \in \{\boldsymbol{\tau}^{\text{BCD}}, \boldsymbol{\tau}^{\text{RGD}}\}$ and real parameter $\boldsymbol{\tau}$ as the mean squared distance on $(\mathbb{R}^{++})^n$ (19).

The subspace estimation performance is studied for two different s^2 along two SNR in Figure 2. Firstly, we observe that our proposed estimation algorithm performs identically to the block coordinate algorithm [12] in every scenario. Also, both estimators are statistically efficient, *i.e.* reach the lower bound (28) when n is sufficiently large. Finally, in the case of a low SNR (*i.e.*, SNR = 1), the block coordinate descent and our Riemannian gradient descent outperform the projected SCM regardless of texture heterogeneity.

#	Class	Number of samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	718
4	Corn	229
5	Grass-pasture	438
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	943
11	Soybean-mintill	2371
12	Soybean-clean	577
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	290
16	Stone-Steel-Towers	93
	Total	9859

TABLE II: *Indian Pines* [22] classes.

Figure 3 presents the texture estimation error as a function of SNR with two different s^2. Firstly, our proposed estimation algorithm performs identically to the block coordinate algorithm [12]. Interestingly, the rate of convergence of the estimation error in the case of low heterogeneity, *i.e.* $s^2 = 2$, is much faster than in the case of high heterogeneity, *i.e.* $s^2 = 4$. Moreover, both estimators reach the lower bound (29) for a high enough SNR.

A final simulation is conducted on high dimensional data. In Section IV, we recalled that the complexity of the BCD grows linearly with the number of data n and quadratically with the dimension p of the data. Hence, the BCD is no longer practicable when both n and p get large. However, in Section IV, we showed that the R-SGD proposed in Algorithm 1 has a constant complexity for the number of data and linear for the dimension of the data. Figure 4 illustrates this situation with $n \in [\![10^3, 10^4]\!]$, $p = 10^4$ and $k = 10$ (dimensions for which the iteration of BCD cannot be computed on the tested setup). This shows the efficiency of the proposed R-SGD.

B. Clustering: application to image segmentation

To illustrate the interest of the Riemannian geometry $\mathcal{M}_{p,k,n}$ and of the parameters of the statistical model (3) used as descriptors, we apply the Algorithm 2 to a hyperspectral image segmentation problem. We cluster a 145×145 pixels hyperspectral image called *Indian Pines* [22]. This image consists of $p = 200$ spectral reflectance bands in the wavelength range $0.42.5\,\mu m$. The Figure 5 shows the ground truth and divides the image in 16 classes (see Table II for details).

After centering the image by subtracting the global mean, a sliding window of size $w \times w$ is applied to the image. One descriptor θ_i is estimated using the $n = w^2$ observations in each window denoted $\boldsymbol{X}_i \in \mathbb{R}^{p \times n}$. Thus, we get a set of descriptors $\{\theta_i\}$ to cluster using a *K-means++* [21].

We compare the descriptors of the considered statistical model (HS+WGN) with different descriptors and geometries. Due to the data's high dimensionality, some methods require a PCA on the whole image as a preprocessing. Then, we keep only the k first components. We begin by presenting these different methods:

1) "center pixel": we extract the center vector of the window. *K-means++* cluster these pixels using the Euclidean metric (*i.e.*, classical inner product). It amounts to cluster directly the image using a classical *K-means++*.
2) "mean pixel": we average the pixels inside the window. Then *K-means++* cluster these means using the Euclidean metric.
3) "SCM": we compute the SCM using pixels inside the window. *K-means++* cluster these matrices using the Riemannian geometry of symmetric positive definite matrices S_p^{++} (see [43]–[45]).

Next, we present the different methods that take into account this high dimensionality. Therefore, we do not use any dimensional reduction preprocessing.

1) "subspace SCM": the k first eigenvectors of the SCM are retained. Then, they are clustered using a *K-means++* on $\text{Gr}_{p,k}$.
2) "robust subspace $\gamma = 0$": our method. Subspaces and textures are estimated following statistical model (3). Only the subspaces are clustered using a *K-means++* on $\text{Gr}_{p,k}$. σ^2 is pre-estimated using the $p - k$ lowest eigenvalues of the SCM.
3) "robust subspace $\gamma > 0$": our method. Subspaces and textures are estimated following statistical model (3). The textures and subspaces are clustered using a *K-means++* on $\mathcal{M}_{p,k,n}$ as explained in Section V and detailed in Algorithm 2. σ^2 is pre-estimated using the $p - k$ lowest eigenvalues of the SCM.

Because *Indian Pines* [22] has 16 classes, we set the number of clusters K to 16. Furthermore, we set $k = 5$. Indeed, from Figure 6, we observe that the first 5 principal eigenvectors of the SCM calculated on *Indian Pines* [22] contain more than 95% of the total variance. Since we use an unsupervised algorithm, the output classes are not necessarily the same as the ground truth. Hence, a KuhnMunkres algorithm is applied to the segmentation to recover ground truths classes. Furthermore, we do 10 different initializations (parameter l in Algorithm 2) and keep the clustering with the lowest inertia (36). To measure the variability of the results, each *K-means++* is run 10 times. The averaged Overall Accuracy (OA), as well as the averaged mean Intersection over Union (mIoU), are reported with their standard deviations (std) in Table III.

First of all, the methods based on non-Euclidean geometries all surpass the other methods ("center pixel" and "mean pixel") by at least 8.9% in terms of averaged Overall Accuracy. This proves the interest in using Riemannian geometries other than the simple Euclidean one. Secondly, "robust subspace, $\gamma = 0$" slightly exceeds "subspace SCM" which shows the interest of robust estimation of subspaces. Thirdly, "robust subspace" with $\gamma = 0.1$ outperform "robust subspace $\gamma = 0$" by nearly 4%. Finally, our method "robust subspace $\gamma = 0.1$" outperforms the strong baseline "SCM" by 2.8% in terms of Overall Accuracy. However, "SCM" performs better in terms of mIoU, by nearly 2%, compared to "robust subspace, $\gamma = 0.1$". This means "SCM" better classifies classes with small number of samples.

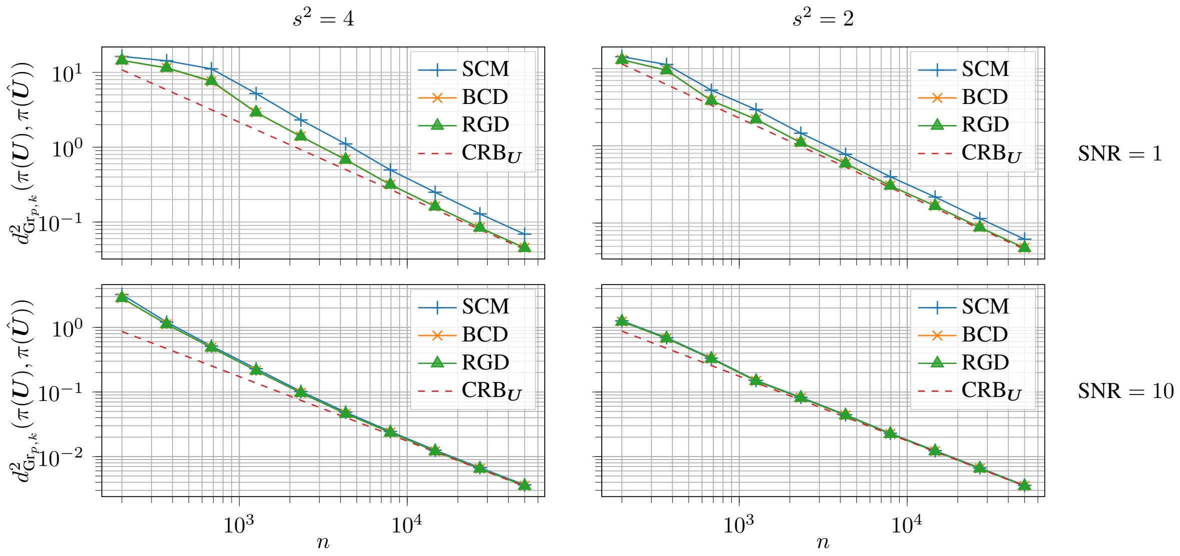


Fig. 2: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}$ ($p = 100$ and $k = 20$) with respect to the number of samples n for the three considered estimators. The textures are generated with $s^2 = 4$ (left part), $s^2 = 2$ (right part), SNR $= 1$ (upper part), SNR $= 10$ (lower part).

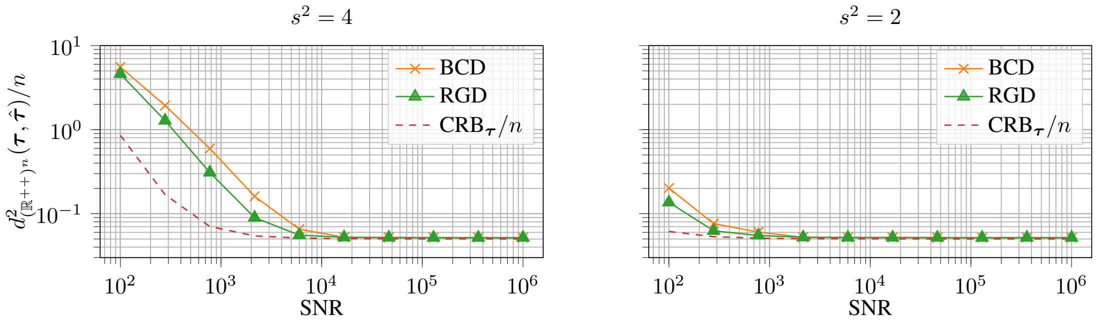


Fig. 3: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}$ ($n = 10^4$, $p = 100$ and $k = 20$) with respect to the SNR for the BCD and RGD estimators. The textures are generated with $s^2 = 4$ (left) and $s^2 = 2$ (right).

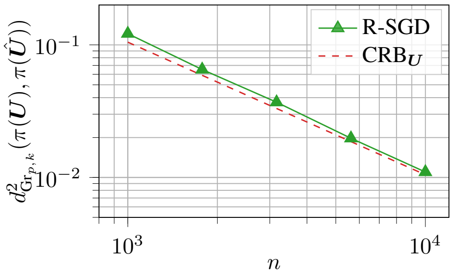


Fig. 4: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}$ ($p = 10^4$ and $k = 10$) with respect to the number of samples n for the R-SGD estimator. 150 samples are used for each computation of the gradient. The textures are generated with $s^2 = 2$ and SNR $= 10^3$.

As mentioned in Section V, a trade-off must be made between the subspaces' distance and textures' distance. A hyperparameter $\gamma \in [0, 1]$ realizes this trade-off. Figure 7 shows that our method "robust subspace" outperforms the "SCM" when we emphasis the $\mathrm{Gr}_{p,k}$ distance. Figure 7 illustrates that our method works for an interval of γ and therefore does not need a critical choice to maximize Overall Accuracy. However, to maximize mIoU, the smaller γ the better.

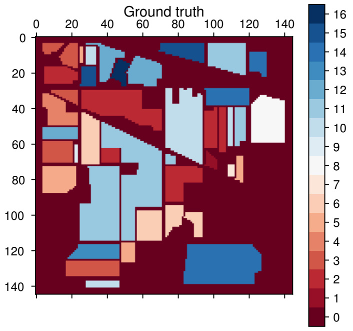


Fig. 5: Ground truth of image *Indian Pines* [22]. The background (no class available) is represented by class 0.

Figure 9 presents the segmentations of 4 methods: "center pixel", "SCM", "robust subspace $\gamma = 0$" and "robust subspace $\gamma = 0.1$". The segmentations are those with the lowest inertia (36) for each method. We note a significant improvement occurs on class 14 (lower right part) between baseline "SCM"

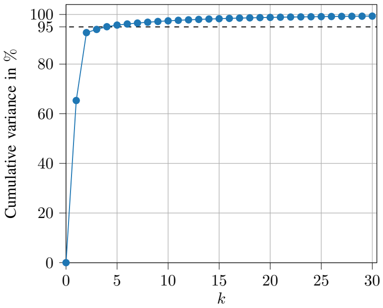


Fig. 6: Cumulative variance, *i.e.* $\sum_{i=1}^{k}\lambda_i/\sum_{i=1}^{p}\lambda_i$, with respect to $k \in [\![1,30]\!]$. $\{\lambda_i\}_{i=1}^{p}$ are the eigenvalues in descending order of the SCM computed with all pixels of *Indian Pines* [22]. Only the first 30 eigenvalues out of $p = 200$ are plotted. We notice that the first 5 principle eigenvectors contain more than 95% of the cumulative variance.

PCA	Descriptor	OA ± std	mIoU ± std
Yes	center pixel	32.66 ± 0.84	18.30 ± 0.82
	mean pixel	34.02 ± 0.48	20.17 ± 2.00
	SCM	45.08 ± 1.58	$\mathbf{29.95 \pm 1.87}$
No	subspace SCM	42.95 ± 0.71	27.06 ± 0.76
	robust subspace, $\gamma = 0$	43.93 ± 0.93	28.11 ± 0.63
	robust subspace, $\gamma = 0.1$	$\mathbf{47.89 \pm 2.67}$	28.00 ± 1.49

TABLE III: Performance of the different descriptors on *Indian Pines* [22] with $w = 7$ and $k = 5$.

in Figure 9b and our method "robust subspace $\gamma = 0.1$" in Figure 9d. Also, the textures help to better cluster classes 8 and 14, see Figure 9c versus 9d.

Finally, our method "robust subspace $\gamma = 0.1$" converges quickly, *i.e.* in less than 20 iterations (see Figure 8). Interestingly, the WCSS (36) decreases a lot in the first iterations and hence the *K-means++* can be stopped after few iterations to faster computation.

VII. Conclusion

This paper proposed to study the information geometry of heteroscedastic signals embedded in WGN. This geometric approach offered a unified framework in order to $i)$ derive new optimization algorithm based on Riemannian stochastic gradient descent; $ii)$ obtain iCRLBs (error bounds driven by a Riemannian distance) with interesting interpretations; $iii)$ propose a new Riemannian clustering algorithm based on the model features, which was applied it to a hyperspectral image to illustrate the interest of the approach.

Appendix

A. Proof of Proposition 1

By definition of the Fisher information metric [18],

$$\langle \bar\xi, \bar\eta\rangle^{\mathrm{FIM}}_{\bar\theta} = \mathbb{E}[\mathrm{D}\,\bar L(\bar\theta)[\bar\xi]\,\mathrm{D}\,\bar L(\bar\theta)[\bar\eta]] = -\mathbb{E}[\mathrm{D}^2\,\bar L(\bar\theta)[\bar\xi,\bar\eta]]$$

$\bar L$ defined in (5) can be written as

$$\bar L(\bar\theta) = \sum_{i=1}^{n} L^g_{\boldsymbol{x}_i}(\bar\psi_i(\bar\theta)),$$

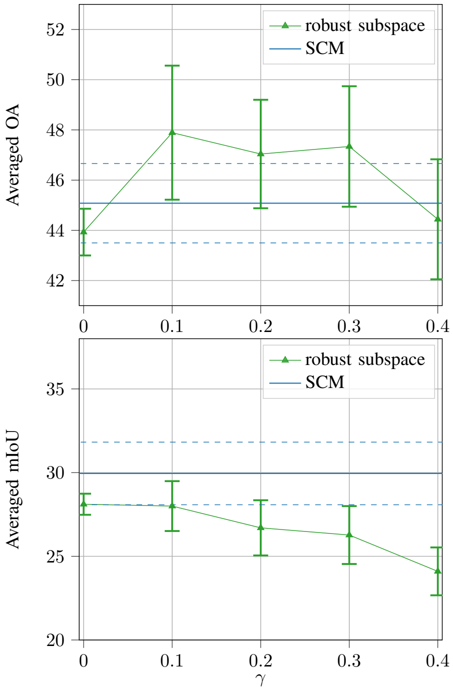


Fig. 7: Overall accuracy and mIoU of our method "robust subspace" with respect to parameter γ on *Indian Pines* [22] with $w = 7$ and $k = 5$. Mean performance are reported with their standard deviations (with error bars for "robust subspace" and in dashed blue lines for "SCM").

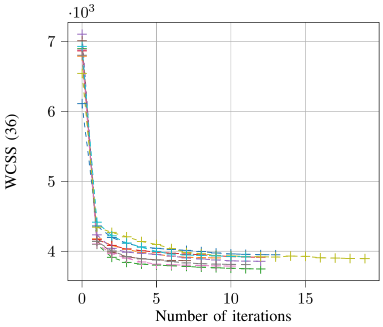
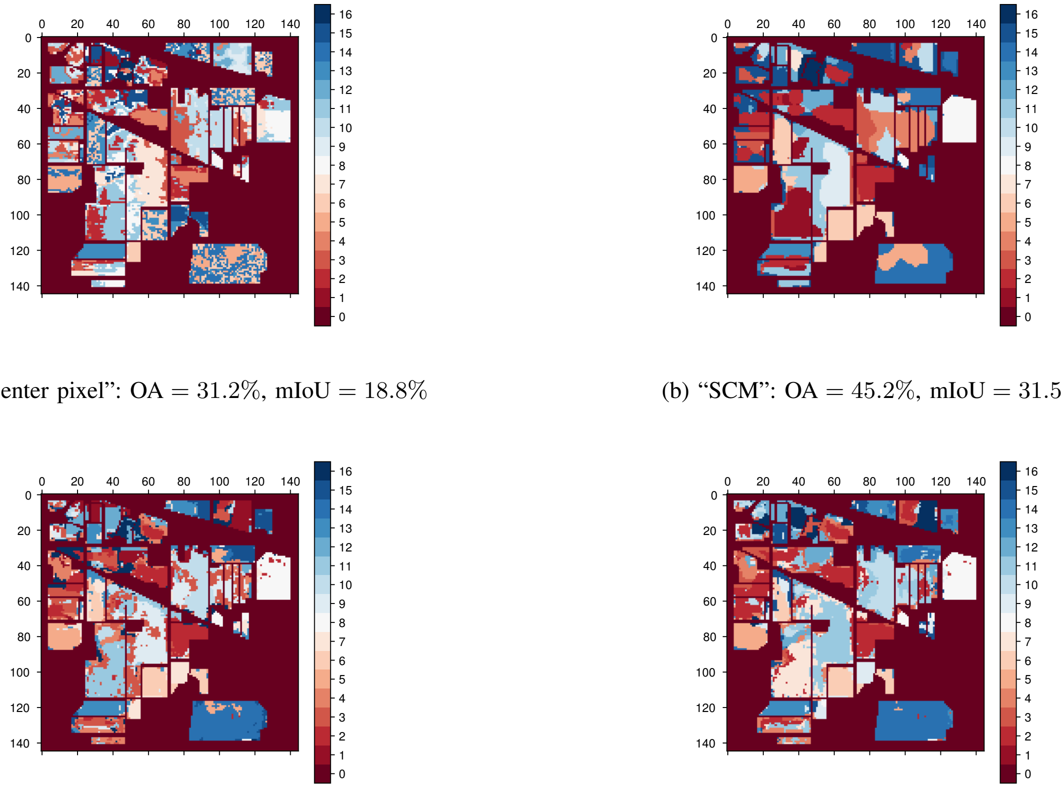


Fig. 8: WCSS (36) with respect to the number of iterations of *K-means++* [21] for "robust subspace" $\gamma = 0.1$ corresponding to Figure 9d. The curves correspond to the 10 initializations.

where $L^g_{\boldsymbol{x}}(\boldsymbol{\Sigma}) = \log|\boldsymbol{\Sigma}| + \boldsymbol{x}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ is the negative Gaussian log-likelihood on $\mathcal{H}_p^{++}$. Thus, following the reasoning of [36, Proposition 6] and [27, Proposition 3.1], one can show

$$\langle \bar\xi, \bar\eta\rangle^{\mathrm{FIM}}_{\bar\theta} = \sum_{i=1}^{n} \langle \mathrm{D}\,\bar\psi_i(\bar\theta)[\bar\xi], \mathrm{D}\,\bar\psi_i(\bar\theta)[\bar\eta]\rangle^{\mathrm{FIM},g}_{\bar\psi_i(\bar\theta)}, \qquad (39)$$

where $\langle \boldsymbol{\xi}_{\boldsymbol{\Sigma}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}}\rangle^{\mathrm{FIM},g}_{\boldsymbol{\Sigma}} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})$ is the Fisher information metric of the Gaussian distribution on $\mathcal{H}_p^{++}$; see *e.g.* [18]. The definition (4) of $\bar\psi_i(\bar\theta)$ and $\mathrm{D}\,\bar\psi_i(\bar\theta)[\bar\xi] =$



(a) “center pixel”: OA $= 31.2\%$, mIoU $= 18.8\%$

(b) “SCM”: OA $= 45.2\%$, mIoU $= 31.5\%$

(c) “robust subspace $\gamma = 0$”: OA $= 43.3\%$, mIoU $= 27.3\%$

(d) “robust subspace $\gamma = 0.1$”: OA $= 47.2\%$, mIoU $= 29.3\%$

Fig. 9: *Indian Pines* [22] segmentation results achieved using 4 methods: “center pixel”, “SCM”, “robust subspace” $\gamma = 0$ and “robust subspace” $\gamma = 0.1$ ($w = 7$ and $k = 5$ for all methods). These segmentations are those with the lowest WCSS computed with their respective distances.

$\tau_i(\boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H) + (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i\boldsymbol{U}\boldsymbol{U}^H$ yields

$$\begin{aligned}
&\langle \mathrm{D}\,\bar{\psi}_i(\bar{\theta})[\bar{\xi}], \mathrm{D}\,\bar{\psi}_i(\bar{\theta})[\bar{\eta}]\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} = \\
&\quad (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i(\boldsymbol{\eta}_{\boldsymbol{\tau}})_i\langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} \\
&\quad + (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i\,\tau_i\langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} \\
&\quad + \tau_i\,(\boldsymbol{\eta}_{\boldsymbol{\tau}})_i\langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} \\
&\quad + (\tau_i)^2\,\langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})}
\end{aligned} \tag{40}$$

Then we compute each term separately:

$$\langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} = \frac{k}{(1+\tau_i)^2} \tag{41}$$

$$\langle \boldsymbol{U}\,\boldsymbol{U}^H, \boldsymbol{U}\,\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\,\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} = 0 \tag{42}$$

$$\langle \boldsymbol{U}\,\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\,\boldsymbol{U}^H, \boldsymbol{U}\,\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} = 0 \tag{43}$$

$$\begin{aligned}
&\langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H\rangle^{\mathrm{FIM},g}_{\bar{\psi}_i(\bar{\theta})} = \\
&\frac{2}{1+\tau_i}\mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{\xi}_{\boldsymbol{U}}^H\boldsymbol{\eta}_{\boldsymbol{U}}\right)\right)
\end{aligned} \tag{44}$$

The Fisher information metric stated in Proposition 1 is obtained by combining eqs. (39) to (44).

B. *Proof of Proposition 2*

Since $\mathrm{Gr}_{p,k}$ is a quotient manifold of $\mathrm{St}_{p,k}$, $\mathrm{grad}\,L_i(\theta)$ is represented by $\mathrm{grad}\,\bar{L}_i(\bar{\theta}) \in \mathcal{H}_{\boldsymbol{U}} \times T_{\boldsymbol{\tau}}(\mathbb{R}^{++})^n$. By definition, $\forall\bar{\xi} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, $\mathrm{D}\,\bar{L}_i(\bar{\theta})[\bar{\xi}] = \langle \mathrm{grad}\,\bar{L}_i(\bar{\theta}), \bar{\xi}\rangle^{\mathrm{FIM}}_{\bar{\theta}}$ [14]. Notice that $\left|\bar{\psi}_i(\bar{\theta})\right| = (1+\tau_i)^k$ and $(\bar{\psi}_i(\bar{\theta}))^{-1} = \boldsymbol{I}_p - \dfrac{\tau_i}{1+\tau_i}\boldsymbol{U}\,\boldsymbol{U}^H$ (Woodbury formula). It follows that

$$\begin{aligned}
\mathrm{D}\,\bar{L}_i(\bar{\theta})[\bar{\xi}] &= -2\frac{\tau_i}{1+\tau_i}\mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{x}_i\boldsymbol{x}_i^H\boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H\right)\right) \\
&\qquad + \frac{k\,(1+\tau_i) - \boldsymbol{x}_i^H\,\boldsymbol{U}\,\boldsymbol{U}^H\,\boldsymbol{x}_i}{(1+\tau_i)^2}\,(\boldsymbol{\xi}_{\boldsymbol{\tau}})_i \\
&= 2nc_{\boldsymbol{\tau}}\langle -\frac{\tau_i}{nc_{\boldsymbol{\tau}}(1+\tau_i)}\,\boldsymbol{x}_i\,\boldsymbol{x}_i^H\,\boldsymbol{U}, \boldsymbol{\xi}_{\boldsymbol{U}}\rangle^{\mathrm{St}_{p,k}}_{\boldsymbol{U}} \\
&\qquad + \langle \boldsymbol{a}, \boldsymbol{\xi}_{\boldsymbol{\tau}}\rangle^{(\mathbb{R}^{++})^n}_{\boldsymbol{\tau}}
\end{aligned}$$

where $\boldsymbol{a} \in \mathbb{R}^n$ is a vector such that

$$\boldsymbol{a}_j = \begin{cases} 1 + \tau_i - \frac{1}{k}\boldsymbol{x}_i^H\boldsymbol{U}\boldsymbol{U}^H\boldsymbol{x}_i & \text{for} \quad j = i \\ 0 \quad \text{otherwise.} \end{cases}$$

To obtain the Riemannian gradient $\mathrm{grad}\,\bar{L}(\bar{\theta})$ by identification, it remains to project $-\frac{\tau_i}{nc_{\boldsymbol{\tau}}(1+\tau_i)}\boldsymbol{x}_i\,\boldsymbol{x}_i^H\,\boldsymbol{U}$ onto $\mathcal{H}_{\boldsymbol{U}}$ with $P_{\boldsymbol{U}}^{\mathrm{Gr}_{p,k}}(\boldsymbol{\xi}_{\boldsymbol{U}}) = \left(\boldsymbol{I}_p - \boldsymbol{U}\,\boldsymbol{U}^H\right)\boldsymbol{\xi}_{\boldsymbol{U}}$ [14], which is enough to conclude.

C. Proof of Proposition 3 and 4

In this section we derive the elements of the generic iCRLB inequality (26) for the estimation problem of $\theta \in \mathcal{M}_{p,k,n}$ (and data model in (3)) when the chosen error metric is (17). Following from [18] the estimation error between θ and $\hat{\theta}$ is characterized by $\log_\theta(\hat{\theta})$, i.e., the Riemannian logarithm mapping induced by the error metric (which is defined in (20)). Recall that this object corresponds to a vector of $T_\theta\mathcal{M}_{p,k,n}$ that points towards $\hat{\theta}$ and whose norm with respect to the error metric is $d^2_{\mathcal{M}_{p,k,n}}(\theta, \hat{\theta})$ as defined in (18). Hence we directly have $\mathrm{Tr}(\boldsymbol{C}) = \mathrm{Tr}(\mathbb{E}[\log_\theta(\hat{\theta})\log_\theta(\hat{\theta})^H]) = \mathbb{E}[d^2_{\mathcal{M}_{p,k,n}}(\theta, \hat{\theta})]$ by definition. Yet, we still we need to select a proper system of coordinates of the tangent space $T_\theta\mathcal{M}_{p,k,n}$ so that the entries of $\mathbf{F}^{-1}$ can be actually obtained: $\mathcal{M}_{p,k,n}$ being a quotient manifold, there are two solutions in order to represent this object. The first one is to simply consider coordinates of $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ without restrictions. The resulting Fisher information matrix will then be singular, but its pseudo-inverse still yields the desired inequality [37]. The second option, which will be chosen here, is to consider only coordinates in the horizontal space $\mathcal{H}_{\bar{\theta}}$, which is given in our case in (13).

Two ingredients are thus needed to establish the Fisher information matrix as in (26):

(i) The Fisher information metric $\langle\cdot,\cdot\rangle^{\mathrm{FIM}}_{\bar{\theta}}$, which was given in Proposition 1.

(ii) A basis of the horizontal space $\mathcal{H}_{\bar{\theta}}$ in (13) that is orthonormal with respect to the error metric (i.e., the decoupled metric in (17)), which is given in the following proposition.

Proposition 5 (Orthonormal basis)**.** *Given $\bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$, an orthonormal basis of the horizontal space $\mathcal{H}_{\bar{\theta}}$ defined in* (13) *with respect to the Riemannian metric of Definition 1 is*

$$\{e^q_{\bar{\theta}}\}_{1\leq q\leq 2(p-k)k+n} = B_{\boldsymbol{U}} \cup B_{\boldsymbol{\tau}},$$

with

$$B_{\boldsymbol{U}} = \bigcup_{\substack{1\leq i\leq p-k\\1\leq j\leq k}} \left\{\left(\alpha^{-\frac{1}{2}}\boldsymbol{U}_\perp\boldsymbol{K_{ij}}, \mathbf{0}\right), \left(\alpha^{-\frac{1}{2}} i\,\boldsymbol{U}_\perp\boldsymbol{K_{ij}}, \mathbf{0}\right)\right\},$$

$$B_{\boldsymbol{\tau}} = \bigcup_{1\leq i\leq n} \{(\mathbf{0}, \beta^{-\frac{1}{2}}\tau_i\boldsymbol{e_i})\},$$

where $\boldsymbol{U}_\perp \in \mathrm{St}_{p,p-k}$ such that $\boldsymbol{U}^H\boldsymbol{U}_\perp = \mathbf{0}$; $\boldsymbol{K_{ij}} \in \mathbb{R}^{(p-k)\times k}$: its ij^{th} element is 1*, zeros elsewhere; and $\boldsymbol{e_i} \in \mathbb{R}^n$: its i^{th} element is* 1*, zero elsewhere.*

Proof. As $\{e^q_{\bar{\theta}}\}$ contains the right amount of elements, it suffices to show that, $\forall q, l \in [\![1, 2(p-k)k+n]\!]$ such that $q \neq l$, we have $\langle e^q_{\bar{\theta}}, e^l_{\bar{\theta}}\rangle^{\overline{\mathcal{M}}_{p,k,n}}_{\bar{\theta}} = 0$ and $\langle e^q_{\bar{\theta}}, e^q_{\bar{\theta}}\rangle^{\overline{\mathcal{M}}_{p,k,n}}_{\bar{\theta}} = 1$. This can easily be checked by calculation. □

Using this system of coordinates, the ql^{th} element of the Fisher information matrix $\boldsymbol{F}_\theta$ is then represented by

$$(\boldsymbol{F}_\theta)_{ql} = \langle e^q_{\bar{\theta}}, e^l_{\bar{\theta}}\rangle^{\mathrm{FIM}}_{\bar{\theta}}. \tag{45}$$

Remarkably, $\boldsymbol{F}_\theta$ will turn to be diagonal which enables us to obtain closed forms iCRLB on $\mathcal{M}_{p,k,n}$, $\mathrm{Gr}_{p,k}$ and $(\mathbb{R}^{++})^n$ respectively. To show that $\boldsymbol{F}_\theta$ is block diagonal, it suffices to notice that there are no crossed terms between tangent vectors of $\boldsymbol{U}$ and $\boldsymbol{\tau}$ in the Fisher information metric of Proposition 3. Computing the elements of $\boldsymbol{F_U}$ yields

$$\langle(\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathrm{FIM}}_{\bar{\theta}} = \begin{cases} 2\,\alpha^{-1}\,n\,c_{\boldsymbol{\tau}} & \text{if } ij = l\,m \\ 0 & \text{otherwise}\end{cases}$$

$$\langle(\alpha^{-\frac{1}{2}} i\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}} i\,\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathrm{FIM}}_{\bar{\theta}} = \begin{cases} 2\alpha^{-1}\,n\,c_{\boldsymbol{\tau}} & \text{if } ij = l\,m \\ 0 & \text{otherwise}\end{cases}$$

$$\langle(\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}} i\,\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathrm{FIM}}_{\bar{\theta}} = 0$$

Hence, $\boldsymbol{F_U} = 2\alpha^{-1}\,n\,c_{\boldsymbol{\tau}}\,\boldsymbol{I}_{2\,(p-k)\,k}$. Computing the elements of $\boldsymbol{F_\tau}$ yields

$$\begin{aligned}\langle(\mathbf{0}, \beta^{-\frac{1}{2}}\tau_i\boldsymbol{e_i}), (\mathbf{0}, \beta^{-\frac{1}{2}}\tau_j\boldsymbol{e_j})\rangle^{\mathrm{FIM}}_{\bar{\theta}} &= \beta^{-1}\,k\,\frac{\tau_i\,\tau_j}{(1+\tau_i)\,(1+\tau_j)}\boldsymbol{e_i}^T\boldsymbol{e_j} \\ &= \begin{cases}\beta^{-1}\,k\,\dfrac{\tau_i^2}{(1+\tau_i)^2} & \text{if } i = j \\ 0 & \text{otherwise}\end{cases}\end{aligned}$$

Hence, $\boldsymbol{F_\tau} = \beta^{-1}k\,\mathrm{diag}\left(\boldsymbol{\tau}^{\odot 2}\odot(\mathbf{1}+\boldsymbol{\tau})^{\odot -2}\right)$, which concludes the part concerning the proof of Proposition 3.

Finally, we note that

$$\mathrm{Tr}\left(\boldsymbol{F_U}^{-1}\right) = \frac{\alpha(p-k)\,k}{n\,c_{\boldsymbol{\tau}}} \text{ and } \mathrm{Tr}\left(\boldsymbol{F_\tau}^{-1}\right) = \frac{\beta}{k}\sum_{i=1}^n\frac{(1+\tau_i)^2}{\tau_i^2}.$$

Furthermore, we get,

$$\mathrm{Tr}\left(\boldsymbol{F}_\theta^{-1}\right) = \frac{\alpha(p-k)\,k}{n\,c_{\boldsymbol{\tau}}} + \frac{\beta}{k}\sum_{i=1}^n\frac{(1+\tau_i)^2}{\tau_i^2}.$$

It follows that the error of an unbiased estimator $\hat{\theta}$ of the true parameter θ in $\mathcal{M}_{p,k,n}$ admits the iCRLB

$$\mathbb{E}[d^2_{\mathcal{M}_{p,k,n}}(\hat{\theta}, \theta)] \geq \mathrm{Tr}\left(\boldsymbol{F}_\theta^{-1}\right) \tag{46}$$

if we neglect the curvature terms when applying Theorem 2 of [18]. Since $\boldsymbol{F}_{\bar{\theta}}$ is block-diagonal we also get two separated iCRLB for the parameters on $\mathrm{Gr}_{p,k}$ and $(\mathbb{R}^{++})^n$ respectively, i.e.:

$$\mathbb{E}[d^2_{\mathrm{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}), \pi(\boldsymbol{U}))] \geq \alpha^{-1}\,\mathrm{Tr}(\boldsymbol{F_U}^{-1}) = \frac{(p-k)k}{nc_{\boldsymbol{\tau}}}, \tag{47}$$

$$\mathbb{E}[d^2_{(\mathbb{R}^{++})^n}(\hat{\boldsymbol{\tau}}, \boldsymbol{\tau})] \geq \beta^{-1}\,\mathrm{Tr}(\boldsymbol{F_\tau}^{-1}) = \frac{1}{k}\sum_{i=1}^n\frac{(1+\tau_i)^2}{\tau_i^2}. \tag{48}$$

This concludes the proof of Proposition 4.

References

[1] I. Jolliffe, *Principal Component Analysis*, Springer, 2011.

[2] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[3] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, "Complex elliptically symmetric distributions: Survey, new results and applications," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5597–5625, 2012.

[4] Ami Wiesel, "Regularized covariance estimation in scaled gaussian models," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2011, pp. 309–312.

[5] D. Hong, L. Balzano, and J. A. Fessler, "Probabilistic PCA for heteroscedastic data," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2019, pp. 26–30.

[6] A. Breloy, G. Ginolhac, F. Pascal, and P. Forster, "Clutter subspace estimation in low rank heterogeneous noise context," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2173–2182, 2015.

[7] O. Besson, "Bounds for a mixture of low-rank compound-gaussian and white gaussian noises," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5723–5732, 2016.

[8] Y. Sun, A. Breloy, P. Babu, D. P. Palomar, F. Pascal, and G. Ginolhac, "Low-complexity algorithms for low rank clutter parameters estimation in radar systems," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 1986–1998, 2016.

[9] T. Chen, E. Martin, and G. Montague, "Robust probabilistic PCA with missing data and contribution analysis for outlier detection," *Computational Statistics & Data Analysis*, vol. 53, no. 10, pp. 3706 – 3716, 2009.

[10] A. Breloy, Y. Sun, P. Babu, G. Ginolhac, D. P. Palomar, and F. Pascal, "A robust signal subspace estimator," in *2016 IEEE Statistical Signal Processing Workshop (SSP)*, 2016, pp. 1–4.

[11] R. S. Raghavan, "Statistical interpretation of a data adaptive clutter subspace estimation algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1370–1384, 2012.

[12] A. Breloy, L. Le Magoarou, G. Ginolhac, F. Pascal, and P. Forster, "Maximum likelihood estimation of clutter subspace in non homogeneous noise context," in *21st European Signal Processing Conference (EUSIPCO 2013)*, Sep. 2013, pp. 1–5.

[13] R. Ben Abdallah, A. Breloy, M.N. El Korso, and D. Lautru, "Bayesian signal subspace estimation with compound gaussian sources," *Signal Processing*, vol. 167, pp. 107310, 2020.

[14] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, USA, 2008.

[15] A. Edelman, T.A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[16] H. Zhang, S. J. Reddi, and S. Sra, "Riemannian svrg: Fast stochastic optimization on riemannian manifolds," in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016.

[17] J. Zhou and S. Said, "Fast, asymptotically efficient, recursive estimation in a Riemannian manifold," *Entropy*, vol. 21, no. 10, 2019.

[18] S. T. Smith, "Covariance, subspace, and intrinsic Cramér-rao bounds," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1610–1630, May 2005.

[19] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[20] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass braincomputer interface classification by riemannian geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2012.

[21] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, USA, 2007, SODA '07, p. 10271035, Society for Industrial and Applied Mathematics.

[22] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," Sep 2015.

[23] A. Mian, A. Collas, A. Breloy, G. Ginolhac, and J-P. Ovarlez, "Robust low-rank change detection for multivariate sar image time series," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3545–3556, 2020.

[24] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematica*, vol. 80, no. 2, pp. 199–220, 2004.

[25] S. Amari, *Information geometry and its applications*, vol. 194, Springer, 2016.

[26] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 635–650, 2002.

[27] F. Bouchard, A. Mian, J. Zhou, S. Said, G. Ginolhac, and Y. Berthoumieu, "Riemannian geometry for compound Gaussian distributions: Application to recursive change detection," *Signal Processing*, vol. 176, pp. 107716, 2020.

[28] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[29] J. Martens, "New insights and perspectives on the natural gradient method," *Journal of Machine Learning Research*, vol. 21, no. 146, pp. 1–76, 2020.

[30] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[31] R. Hosseini and S. Sra, "An alternative to em for gaussian mixture models: batch and stochastic riemannian optimization," *Math. Program.*, vol. 181, pp. 187–223, 2020.

[32] J. D. Gorman and A. Hero, "Lower bounds for parametric estimation with constraints," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1285–1301, 1990.

[33] P. Stoica and B. C. Ng, "On the Cramér-rao bound under parametric constraints," *IEEE Signal Processing Letters*, vol. 5, no. 7, pp. 177–179, 1998.

[34] T. J. Moore, R. J. Kozick, and B. M. Sadler, "The constrained Cramér–rao bound from the perspective of fitting a model," *IEEE Signal Processing Letters*, vol. 14, no. 8, pp. 564–567, 2007.

[35] E. Nitzan, T. Routtenberg, and J. Tabrikian, "Cramér–rao bound for constrained parameter estimation using lehmann-unbiasedness," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 753–768, 2018.

[36] F. Bouchard, A. Breloy, G. Ginolhac, A. Renaux, and F. Pascal, "A Riemannian framework for low-rank structured elliptical models," *IEEE Transactions on Signal Processing*, vol. 69, no. 3, pp. 1185 – 1199, 2021.

[37] Nicolas Boumal, "On intrinsic Cramér-rao bounds for riemannian submanifolds and quotient manifolds," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1809–1821, 2013.

[38] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on Pure and Applied Mathematics*, vol. 30, no. 5, pp. 509–541, 1977.

[39] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[40] F. Nielsen and R. Nock, "Total jensen divergences: Definition, properties and k-means++ clustering," 2013.

[41] J. Townsend, N. Koep, and S. Weichwald, "Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 47554759, Jan. 2016.

[42] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014.

[43] L. T. Skovgaard, "A Riemannian geometry of the multivariate Normal model," *Scandinavian Journal of Statistics*, vol. 11, no. 4, pp. 211–223, 1984.

[44] R. Bhatia, *Positive Definite Matrices*, Princeton University Press, 2007.

[45] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *International Journal of computer vision*, vol. 66, no. 1, pp. 41–66, 2006.