

# On the use of the Gram matrix for multivariate functional principal components analysis

Steven Golovkine<sup>a,\*</sup>, Edward Gunning<sup>b</sup>, Andrew J. Simpkin<sup>c</sup>, Norma Bargary<sup>a</sup>

<sup>a</sup>MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland

<sup>b</sup>Department of Biostatistics and Epidemiology, University of Pennsylvania, USA

<sup>c</sup>School of Mathematical and Statistical Sciences, University of Galway, Ireland

---

## Abstract

Dimension reduction is crucial in functional data analysis (FDA). The key tool to reduce the dimension of the data is functional principal component analysis. Existing approaches for functional principal component analysis usually involve the diagonalization of the covariance operator. With the increasing size and complexity of functional datasets, estimating the covariance operator has become more challenging. Therefore, there is a growing need for efficient methodologies to estimate the eigenelements. Using the duality of the space of observations and the space of functional features, we propose to use the inner-product between the curves to estimate the eigenelements of multivariate and multidimensional functional datasets. The relationship between the eigenelements of the covariance operator and those of the inner-product matrix is established. We explore the application of these methodologies in several FDA settings and provide general guidance on their usability.

**Keywords:** Dimension reduction, Functional data analysis, Functional principal components, Multivariate functional data.

**2020 MSC:** Primary 62R10

---

## 1. Introduction

Functional data analysis (FDA) is a statistical methodology for analyzing data that can be characterized as functions. These functions could represent measurements taken over time or space, such as temperature readings over a yearly period or spatial patterns of disease occurrence. The goal of FDA is to extract meaningful information from these functions and to model their behavior. See, e.g., [21, 26, 33, 39] for some references on FDA.

Functional principal component analysis (FPCA) is an extension of principal component analysis (PCA, a commonly used tool for dimension reduction in multivariate data) to functional data. FPCA was introduced by [24] and [8], based on the Karhunen-Loèv decomposition, independently developed by [27, 29] and [23]. Since then, FPCA has become a prevalent tool in FDA due to its ability to convert infinite-dimensional functional data into finite-dimensional vectors of random scores. These scores are a countable sequence of uncorrelated random variables that can be truncated to a finite vector in practical applications. By applying multivariate data analysis tools to these random scores, FPCA can achieve the goal of dimension reduction while assuming mild assumptions about the underlying stochastic process. FPCA is usually used as a preprocessing step to feed, e.g., regression and classification models. Recently, FPCA has been extended to multivariate functional data, which are data that consist of multiple functions that are observed simultaneously. This extension is referred to as multivariate functional principal component analysis (MFPCA). As for FPCA, a key benefit of MFPCA is that it allows one to identify and visualize the main sources of variation in the multivariate functional data. This can be useful in different applications, such as identifying patterns of movements in sport biomechanics [40], analyzing changes in brain activity in neuroscience [38], or comparing countries' competitiveness in economics [28].

---

\*Corresponding author. Email address: [steven.golovkine@ul.ie](mailto:steven.golovkine@ul.ie)

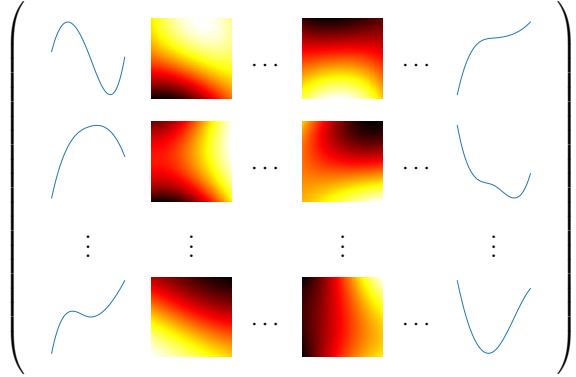
In MFPCA, we seek to decompose the covariance structure of the multivariate functional data into a set of orthogonal basis functions, named the principal components, which capture the main sources of variation in the data. There are multiple approaches to estimate the principal components of a multivariate functional dataset. [33] combine the multivariate curves into one big curve and then perform a usual FPCA via an eigendecomposition of the covariance structure. This methodology can only be run for data that are defined on the same unidimensional domain, that exhibit similar amounts of variability and are measured in the same units. [22] propose to represent each feature of the multivariate function separately using a basis function expansion. This results in a different set of coefficients for each univariate curve. The eigendecomposition is then run on the matrix of stacked coefficients. To consider the normalization issue of [33], [22] and [6] propose to normalize the data by the standard deviation of the curves at each of the sampling points. [18] extend the estimation of multivariate principal components to functional data defined on different dimensional domains. Their estimation procedure is based on carrying out FPCA on each univariate feature, and then using a weighted combination of the resulting principal components to obtain the multivariate eigencomponents. Finally, [3] develop a different method to estimate the eigencomponents as they perform a principal components analysis for each sampling time point.

The key motivation of this paper is to investigate the duality between rows and columns of a data matrix to estimate the eigencomponents of a multivariate functional dataset. The duality between rows and columns of a data matrix is a fundamental concept in classical multivariate statistics [11, 36]. A data matrix typically represents a set of observations of multiple features, each row corresponds to an individual observation and each column corresponds to an individual feature. The duality between rows and columns refers to the fact that many statistical methodologies can be conducted either on the rows or the columns of the data matrix, and the results will be related to each other. For example, the eigenvalues obtained from a PCA run on the rows of the data matrix are the same as the ones obtained from a PCA run on the columns of the matrix. The choice of method to use, based on criteria such as computational time or data storage needs, is thus left to the user. This concept has been widely studied in multivariate statistics (see, e.g., [19, 31]). In the context of functional data, this principle has received limited attention despite being mentioned in the seminal paper of FDA [34]. [33] briefly commented on it in a concluding remark of Chapter 8, while [25] and [2] utilized it to compute principal components for dense univariate functional data. [5] also employ it to gain computational advantage when univariate functional data are sampled on a very dense grid. To the best of our knowledge, however, there is no available literature on its application to multivariate functional data that are observed on different dimensional domains. Our aim is therefore to investigate this duality for multivariate functional data observed on different dimensional domains and provide guidelines on which method to use in different cases.

The remainder of the paper is organized as follows. In Section 2, we define multivariate functional data with components that are observed on possibly different domains. In Section 3, we develop the duality between the observations' space and the functional components' space. The relationship between the eigencomponents of the covariance operator of the multivariate functional datasets and the eigencomponents of the inner-product matrix between the observations is derived in Section 4. Extensive simulations are given in Section 5. We also provide guidelines on which method to use with respect to different data characteristics. We present an application related to sports science data from the National Basketball Association (NBA) in Section 6. The paper concludes with a discussion and an outlook in Section 7.

## 2. Model

The structure of the data we consider, referred to as *multivariate functional data*, is similar to that presented in [18]. The data consist of independent trajectories of a vector-valued stochastic process  $X = (X^{(1)}, \dots, X^{(P)})^\top$ ,  $P \geq 1$ . (Here and in the following, for any matrix  $A$ ,  $A^\top$  denotes its transpose.) For each  $1 \leq p \leq P$ , let  $\mathcal{T}_p$  be a rectangle in some Euclidean space  $\mathbb{R}^{d_p}$  with  $d_p \geq 1$ , e.g.,  $\mathcal{T}_p = [0, 1]^{d_p}$ . Each coordinate, or feature,  $X^{(p)} : \mathcal{T}_p \rightarrow \mathbb{R}$  is assumed to belong to  $\mathcal{L}^2(\mathcal{T}_p)$ , the Hilbert space of square-integrable real-valued functions defined on  $\mathcal{T}_p$ , having the usual inner product that we denote by  $\langle \cdot, \cdot \rangle$ , and  $\|\cdot\|$  the associated norm. Thus  $X$  is a stochastic process indexed by  $\mathbf{t} = (t_1, \dots, t_P)$  belonging to the  $P$ -fold Cartesian product  $\mathcal{T} := \mathcal{T}_1 \times \dots \times \mathcal{T}_P$  and taking values in the  $P$ -fold Cartesian product space  $\mathcal{H} := \mathcal{L}^2(\mathcal{T}_1) \times \dots \times \mathcal{L}^2(\mathcal{T}_P)$ .



**Fig. 1:** Functional data matrix, adapted from [3].

We consider the function  $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$ ,

$$\langle\!\langle f, g \rangle\!\rangle := \sum_{p=1}^P \langle f^{(p)}, g^{(p)} \rangle = \sum_{p=1}^P \int_{\mathcal{T}_p} f^{(p)}(t_p) g^{(p)}(t_p) dt_p, \quad f, g \in \mathcal{H}.$$

$\mathcal{H}$  is a Hilbert space with respect to the inner product  $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ [18]. We denote by  $|\cdot|$ , the norm induced by  $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ . Let  $\mu : \mathcal{T} \rightarrow \mathcal{H}$  denote the mean function of the process  $X$ ,  $\mu(\mathbf{t}) := \mathbb{E}(X(\mathbf{t}))$ ,  $\mathbf{t} \in \mathcal{T}$ . Let  $C$  denote the  $P \times P$  matrix-valued covariance function which, for  $\mathbf{s}, \mathbf{t} \in \mathcal{T}$ , is defined as

$$C(\mathbf{s}, \mathbf{t}) := \mathbb{E}((X(\mathbf{s}) - \mu(\mathbf{s}))(X(\mathbf{t}) - \mu(\mathbf{t}))^\top), \quad \mathbf{s}, \mathbf{t} \in \mathcal{T}.$$

More precisely, for  $1 \leq p, q \leq P$ , the  $(p, q)$ th entry of the matrix  $C(\mathbf{s}, \mathbf{t})$  is the covariance function between the  $p$ th and the  $q$ th features of the process  $X$ :

$$C_{p,q}(s_p, t_q) := \mathbb{E}((X^{(p)}(s_p) - \mu^{(p)}(s_p))(X^{(q)}(t_q) - \mu^{(q)}(t_q))^\top), \quad s_p \in \mathcal{T}_p, t_q \in \mathcal{T}_q.$$

Let  $\Gamma : \mathcal{H} \rightarrow \mathcal{H}$  denote the covariance operator of  $X$ , defined as an integral operator with kernel  $C$ . That is, for  $f \in \mathcal{H}$  and  $\mathbf{t} \in \mathcal{T}$ , the  $p$ th feature of  $\Gamma f(\mathbf{t})$  is given by

$$(\Gamma f)^{(p)}(t_p) := \langle\!\langle C_{p,\cdot}(t_p, \cdot), f(\cdot) \rangle\!\rangle = \langle\!\langle C_{\cdot,p}(\cdot, t_p), f(\cdot) \rangle\!\rangle, \quad t_p \in \mathcal{T}_p.$$

Let us consider a set of  $N$  independent multivariate curves  $\mathcal{X} = \{X_n\}_{1 \leq n \leq N}$  generated as a random sample of the  $P$ -dimensional stochastic process  $X$  with continuous trajectories. The data can be viewed as a table with  $N$  rows and  $P$  columns where each entry is a curve, potentially on a multidimensional domain (see Figure 1). Each row of this matrix represents an observation; while each column represents a functional feature. At the intersection of row  $n$  and column  $p$ , we thus have  $X_n^{(p)}$  which is the curve that concerns the (functional) feature  $p$  for the individual  $n$ . For  $n \in \{1, \dots, N\}$ , each observation  $n$  is attributed the weight  $\pi_n$  such that  $\sum_n \pi_n = 1$ , e.g.,  $\pi_n = 1/N$ . For the set  $\mathcal{X}$ , the inner-product matrix, also called the Gram matrix,  $\mathbf{M}$  is defined as a matrix of size  $N \times N$  with entries

$$\mathbf{M}_{nn'} = \sqrt{\pi_n \pi_{n'}} \langle\!\langle X_n - \mu, X_{n'} - \mu \rangle\!\rangle, \quad n, n' = 1, \dots, N. \quad (1)$$

This matrix is symmetric, positive definite, and interpretable as a proximity matrix, each entry being the similarity between the weighted observations.

This setting can be generalized to incorporate a general weighting scheme, as [6] or [18]. We consider the function  $\langle\!\langle \cdot, \cdot \rangle\!\rangle_w : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$ ,

$$\langle\!\langle f, g \rangle\!\rangle_w := \sum_{p=1}^P w_p \langle f^{(p)}, g^{(p)} \rangle = \sum_{p=1}^P w_p \int_{\mathcal{T}_p} f^{(p)}(t_p) g^{(p)}(t_p) dt_p, \quad f, g \in \mathcal{H}.$$

Let  $\Gamma_w : \mathcal{H} \rightarrow \mathcal{H}$  denote the weighted covariance operator of  $X$ , defined as an integral operator with kernel  $C$ . That is, for  $f \in \mathcal{H}$  and  $\mathbf{t} \in \mathcal{T}$ , the  $p$ th feature of  $\Gamma_w f(\mathbf{t})$  is given by

$$(\Gamma_w f)^{(p)}(t_p) := \langle\langle C_{p,\cdot}(t_p, \cdot), f(\cdot) \rangle\rangle_w = \langle\langle C_{\cdot,p}(\cdot, t_p), f(\cdot) \rangle\rangle_w, \quad t_p \in \mathcal{T}_p.$$

Similarly, we define the weighted inner-product matrix  $\mathbf{M}_w$ , the matrix of size  $N \times N$  with entries

$$\mathbf{M}_{w,nn'} = \sqrt{\pi_n \pi_{n'}} \langle\langle X_n - \mu, X_{n'} - \mu \rangle\rangle_w, \quad n, n' = 1, \dots, N.$$

**Remark 1.** The observation weights  $\pi_n$  and the feature weights  $w_p$  are not the same and should not be confused. The observation weights are used to give more (or less) importance to specific observations, while the feature weights control for different levels of variation for the features. A discussion on the feature weights is provided in Section 3.4.

### 2.1. Inference

If we observe the set  $\mathcal{X}$ , the ideal estimators of the mean and covariance function are

$$\tilde{\mu}(\mathbf{t}) = \sum_{n=1}^N \pi_n X_n(\mathbf{t}), \quad \text{and} \quad \tilde{C}(\mathbf{s}, \mathbf{t}) = \sum_{n=1}^N \pi_n \{X_n(\mathbf{s}) - \tilde{\mu}(\mathbf{s})\} \{X_n(\mathbf{t}) - \tilde{\mu}(\mathbf{t})\}^\top, \quad \mathbf{s}, \mathbf{t} \in \mathcal{T}.$$

Similarly, one could estimate the inner-product  $\mathbf{M}$  by replacing the mean by their estimators. The resulting matrix  $\tilde{\mathbf{M}}$  has entries

$$\tilde{\mathbf{M}}_{nn'} = \sqrt{\pi_n \pi_{n'}} \sum_{p=1}^P \int_{\mathcal{T}_p} \{X_n^{(p)}(t_p) - \tilde{\mu}^{(p)}(t_p)\} \{X_{n'}^{(p)}(t_p) - \tilde{\mu}^{(p)}(t_p)\} dt_p, \quad n, n' = 1, \dots, N.$$

In many applications, the elements of the set  $\mathcal{X}$  are observed with error and on a finite grid of points in  $\mathcal{T}$ . For each  $1 \leq n \leq N$ , and given a vector of positive integers  $\mathbf{M}_n = (\mathbf{M}_n^{(1)}, \dots, \mathbf{M}_n^{(P)})$ , let  $\mathbf{T}_{n,\mathbf{m}} = (T_{n,m_1}^{(1)}, \dots, T_{n,m_P}^{(P)})$ ,  $1 \leq m_p \leq M_n^{(p)}$ ,  $1 \leq p \leq P$ , be the random observation times for the curve  $X_n$ . These times are obtained as independent realizations of a random variable  $\mathbf{T}$  taking values in  $\mathcal{T}$ . The vectors  $\mathbf{M}_1, \dots, \mathbf{M}_N$  represent an independent sample of an integer-valued random vector  $\mathbf{M}$  with known expectation. We assume that the realizations of  $X$ ,  $\mathbf{M}$  and  $\mathbf{T}$  are mutually independent. The observations associated with an observation  $X_n$  consist of the pairs  $(Y_{n,\mathbf{m}}, \mathbf{T}_{n,\mathbf{m}}) \in \mathbb{R}^P \times \mathcal{T}$ , where  $\mathbf{m} = (m_1, \dots, m_P)$ ,  $1 \leq m_p \leq M_n^{(p)}$ ,  $1 \leq p \leq P$  and  $Y_{n,\mathbf{m}}$  is defined as

$$Y_{n,\mathbf{m}} = X_n(\mathbf{T}_{n,\mathbf{m}}) + \varepsilon_{n,\mathbf{m}}, \quad 1 \leq n \leq N, \tag{2}$$

with the  $\varepsilon_{n,\mathbf{m}}$  being independent realizations of a centered error random vector  $\varepsilon \in \mathbb{R}^P$  with finite variance  $\sigma^2 \mathbf{1}_P \in \mathbb{R}^{P \times P}$  where  $\sigma^2 = (\sigma_1^2, \dots, \sigma_P^2) \in \mathbb{R}^P$ .

Let  $\widehat{X}_n$  be a suitable estimator of  $X_n$  applied to the pairs  $(Y_{n,\mathbf{m}}, \mathbf{T}_{n,\mathbf{m}})$ , such as P-splines [e.g. 10] or local polynomials [e.g. 12]. We define the estimator of the mean function as

$$\widehat{\mu}(\mathbf{t}) = \sum_{n=1}^N \pi_n \widehat{X}_n(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}.$$

Concerning the estimation of the covariance of the  $p$ th feature, we distinguish the diagonal from the non-diagonal points. The estimation of the covariance function of the non-diagonal points is defined as

$$\widehat{C}_{p,p}(s_p, t_p) = \sum_{n=1}^N \pi_n \left( \{\widehat{X}_n^{(p)}(s_p) - \widehat{\mu}^{(p)}(s_p)\} \{\widehat{X}_n^{(p)}(t_p) - \widehat{\mu}^{(p)}(t_p)\} \right), \quad s_p \neq t_p, \quad s_p, t_p \in \mathcal{T}_p. \tag{3}$$

The variance function  $C_{p,p}(s_p, t_p)$  induces a singularity when estimating the covariance function  $C_{p,p}(\cdot, \cdot)$  [see 42, 43]. We define the estimator of the diagonal of the covariance as

$$\widehat{C}_{p,p}(s_p, s_p) = \sum_{n=1}^N \pi_n \{\widehat{X}_n^{(p)}(s_p) - \widehat{\mu}^{(p)}(s_p)\}^2 - \sigma_p^2, \quad s_p \in \mathcal{T}_p. \tag{4}$$

This singularity does not appear in the estimation of the cross-covariance function. We define the estimator of the cross-covariance function between the  $p$ th and  $q$ th features by

$$\widehat{C}_{p,q}(s_p, t_q) = \sum_{n=1}^N \pi_n \left( \{\widehat{X}_n^{(p)}(s_p) - \widehat{\mu}^{(p)}(s_p)\} \{\widehat{X}_n^{(q)}(t_q) - \widehat{\mu}^{(q)}(t_q)\} \right), \quad s_p \in \mathcal{T}_p, t_q \in \mathcal{T}_q.$$

The estimator of the inner-product matrix can be defined by replacing the curves by their estimators. Following [2] and [15], we define the estimator of the inner-product matrix as

$$\widehat{\mathbf{M}}_{nn'} = \sqrt{\pi_n \pi_{n'}} \sum_{p=1}^P \int_{\mathcal{T}_p} \{\widehat{X}_n^{(p)}(t_p) - \widehat{\mu}^{(p)}(t_p)\} \{\widehat{X}_{n'}^{(p)}(t_p) - \widehat{\mu}^{(p)}(t_p)\} dt_p, \quad n, n' = 1, \dots, N.$$

As for the covariance estimation, the model implies a bias on the diagonal term of the inner-product matrix. A correction for the diagonal of the Gram matrix is given by

$$\widehat{\mathbf{M}}_{nn} = \pi_n \sum_{p=1}^P \left\{ \int_{\mathcal{T}_p} \{\widehat{X}_n^{(p)}(t_p) - \widehat{\mu}^{(p)}(t_p)\}^2 dt_p - \sigma_p^2 \right\}, \quad n = 1, \dots, N. \quad (5)$$

In practice, the variances  $\sigma_p^2$  in equations (4) and (5) are unknown and have to be estimated. For the estimation, we refer the reader to [17] and [16].

### 3. On the geometry of multivariate functional data

#### 3.1. Duality diagram

The distinction between the space of rows of a matrix as a sample from a population and the space of columns as the fixed variables on which the observations were measured has been explained in [20] and [7] for multivariate data. We propose to define a duality diagram in the context of multivariate functional data. Consider the data matrix defined by the set  $X$ . We define an operator  $L_X : \mathcal{H} \rightarrow \mathbb{R}^N$  by

$$L_X : f \mapsto \begin{pmatrix} \sqrt{\pi_1} \langle\langle X_1 - \mu, f \rangle\rangle \\ \vdots \\ \sqrt{\pi_N} \langle\langle X_N - \mu, f \rangle\rangle \end{pmatrix}.$$

Using the linearity of the inner-product  $\langle\langle \cdot, \cdot \rangle\rangle$  and vectors, the operator  $L_X$  is linear. Define the operator  $L_X^* : \mathbb{R}^N \rightarrow \mathcal{H}$  as

$$L_X^* : u \mapsto \begin{pmatrix} \sum_{n=1}^N \sqrt{\pi_n} u_n \{X_n^{(1)}(t_1) - \mu^{(1)}(t_1)\} \\ \vdots \\ \sum_{n=1}^N \sqrt{\pi_n} u_n \{X_n^{(P)}(t_P) - \mu^{(P)}(t_P)\} \end{pmatrix}.$$

**Lemma 1.**  $L_X^*$  is the adjoint operator of the linear operator  $L_X$ .

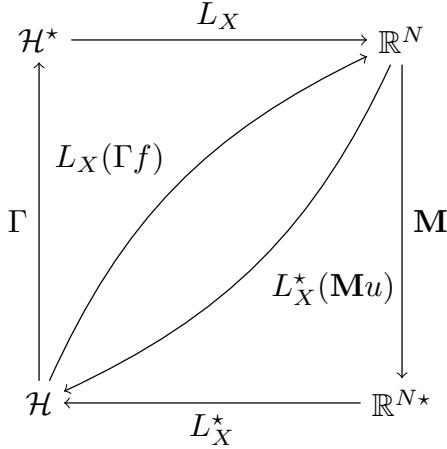
The proof of Lemma 1 is given in Appendix 8. As an adjoint operator,  $L_X^*$  is a linear operator. The operator  $\Gamma$  and the matrix  $\mathbf{M}$  define geometries in  $\mathcal{H}$  and  $\mathbb{R}^N$ , respectively, through

$$\langle\langle f, g \rangle\rangle_\Gamma = \langle\langle f, \Gamma g \rangle\rangle, \quad f, g \in \mathcal{H}, \quad \text{and} \quad (u, v)_\mathbf{M} = u^\top \mathbf{M} v, \quad u, v \in \mathbb{R}^N.$$

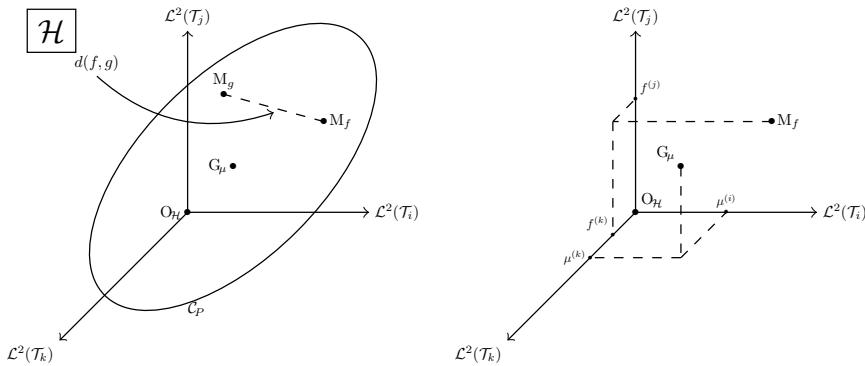
We denote by  $\|\cdot\|_\Gamma$  and  $(\cdot, \cdot)_\mathbf{M}$  their associated norms. Using the definition of adjoint operators  $L_X$  and  $L_X^*$ , we have that

$$(L_X(f), u)_\mathbf{M} = \langle\langle f, L_X^*(u) \rangle\rangle_\Gamma, \quad \text{for all } f \in \mathcal{H}, u \in \mathbb{R}^N.$$

These relationships can be expressed as a duality diagram, see Figure 2. The triplet  $(X, \Gamma, \mathbf{M})$  defines a (multivariate) functional data analysis framework. One consequence of this transition between spaces is that the eigencomponents of  $X$  can be estimated equivalently using either the covariance operator  $\Gamma$  or the Gram matrix  $\mathbf{M}$ . The relationship between the eigencomponents of  $\Gamma$  and the eigencomponents of  $\mathbf{M}$  are derived in Section 4.



**Fig. 2:** Duality diagram between the spaces  $\mathcal{H}$  and  $\mathbb{R}^N$ . The operator  $L_X$  and its adjoint  $L_X^*$  are linear operators. The covariance operator  $\Gamma$  and the matrix  $\mathbf{M}$  define geometries in  $\mathcal{H}$  and  $\mathbb{R}^N$  respectively. The space  $\mathcal{H}^*$  (resp.  $\mathbb{R}^{N*}$ ) is the dual space of  $\mathcal{H}$  (resp.  $\mathbb{R}^N$ ).



**Fig. 3:** Left: Cloud of observations. Right: Projection of the points onto the elements of  $\mathcal{H}$ . The observation  $f$  (resp.  $g$ ) is identified by the point  $M_f$  (resp.  $M_g$ ) in the cloud  $C_P$ . The point  $G_\mu$  is the center of gravity of  $C_P$  and the point  $O_H$  is the origin of the space  $\mathcal{H}$ .

**Remark 2.** In general, in order to avoid confusion, inner-products and norms in function space,  $\mathcal{H}$  or  $L^2(\mathcal{T})$ , will be referred with angle brackets,  $\langle \cdot, \cdot \rangle$ , while inner-products and norms in coordinate space,  $\mathbb{R}^N$ , will be referred with round brackets,  $(\cdot, \cdot)$ .

**Remark 3.** We present here the duality diagram for the linear integral operator  $\Gamma$  with kernel  $C$ . It is however possible to define duality diagrams for more general linear integral operators defined with a continuous symmetric positive definite function as kernel (see [14] and [41] for discussions on possible integral operators to represent univariate functional data).

### 3.2. Cloud of individuals

Given an element  $f \in \mathcal{H}$ , let  $\{f^{(p)}(t_p), t_p \in \mathcal{T}_p, p = 1, \dots, P\}$  be the features set of the element. We identify this set as the point  $M_f$  in the space  $\mathcal{H}$ . The space  $\mathcal{H}$  is referred to as the observation space. The cloud of points that represents the set of observations  $X$  in  $\mathcal{H}$  is denoted by  $C_P$ . Let  $G_\mu$  be the centre of gravity of the cloud  $C_P$ . In the space  $\mathcal{H}$ , its coordinates are given by  $\{\mu^{(p)}(t_p), t_p \in \mathcal{T}_p, p = 1, \dots, P\}$ . If the features are centered, the origin  $O_H$  of the axes in  $\mathcal{H}$  coincides with  $G_\mu$ .

Let  $f$  and  $g$  be two elements in  $\mathcal{H}$  and denote by  $M_f$  and  $M_g$  their associated points in  $C_P$  (see Figure 3). The most

natural distance between these observations is based on the usual inner product in  $\mathcal{H}$ ,  $\langle\langle \cdot, \cdot \rangle\rangle$ , and is defined as

$$d^2(\mathbf{M}_f, \mathbf{M}_g) = \|f - g\|^2 = \sum_{p=1}^P \int_{\mathcal{T}_p} \{f^{(p)}(t_p) - g^{(p)}(t_p)\}^2 dt_p.$$

This distance measures how different the observations are, and thus gives one characterization of the shape of the cloud  $C_P$ . Another description of this shape is to consider the distance between each observation and  $\mathbf{G}_\mu$ , the center of the cloud. Let  $f$  be an element of  $\mathcal{H}$ , associated to the point  $\mathbf{M}_f$ , and  $\mu$  the element of  $\mathcal{H}$  related to  $\mathbf{G}_\mu$ , the distance between  $f$  and  $\mu$  is given by

$$d^2(\mathbf{M}_f, \mathbf{G}_\mu) = \|f - \mu\|^2 = \sum_{p=1}^P \int_{\mathcal{T}_p} \{f^{(p)}(t_p) - \mu^{(p)}(t_p)\}^2 dt_p.$$

Given the set  $\mathcal{X}$ , the total inertia of  $C_P$ , with respect to  $\mathbf{G}_\mu$  and the distance  $d$ , is given by

$$\sum_{n=1}^N \pi_n d^2(\mathbf{M}_n, \mathbf{G}_\mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(\mathbf{M}_i, \mathbf{M}_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \text{Var } X^{(p)}(t_p) dt_p. \quad (6)$$

The duality diagram, however, allows us to define another suitable distance to characterize the shape of the cloud  $C_P$ . We thus define

$$d_\Gamma^2(\mathbf{M}_f, \mathbf{M}_g) = \|f - g\|_\Gamma^2.$$

The utilization of the distance measure  $d_\Gamma$ , which accounts for the variability among all the features within the functional data, corresponds to a Mahalanobis-type distance framework for multivariate functional data [see 4, 30]. Given the set  $\mathcal{X}$ , the total inertia of  $C_P$ , with respect to  $\mathbf{G}_\mu$  and the distance  $d_\Gamma$ , is given by

$$\sum_{n=1}^N \pi_n d_\Gamma^2(\mathbf{M}_n, \mathbf{G}_\mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d_\Gamma^2(\mathbf{M}_i, \mathbf{M}_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p \cdot(t_p, \cdot)\|^2 dt_p. \quad (7)$$

The derivation of these equalities are given in Appendix 8.

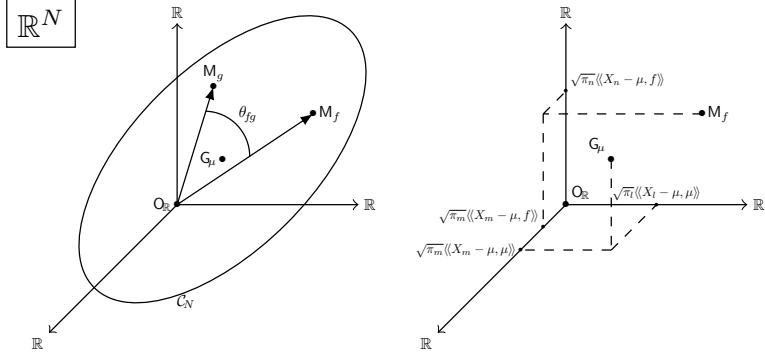
**Remark 4.** These results have the same interpretation as for multivariate scalar data. This is also the multivariate analogue of the relation between variance and sum of squared differences known for univariate functional data. If the features are reduced beforehand, such that  $\int_{\mathcal{T}_p} \text{Var } X^{(p)}(t_p) dt_p = 1$  for the distance  $d$  or  $\int_{\mathcal{T}_p} \|C_p \cdot(t_p, \cdot)\|^2 dt_p = 1$  for the distance  $d_\Gamma$ , the total inertia of the cloud  $C_P$  is equal to the number of components  $P$ . We are, in general, not interested by the total inertia but how this variance is spread among the features.

### 3.3. Cloud of features

Given an element  $f \in \mathcal{H}$ , let  $L_X(f) = \{\sqrt{\pi_n} \langle\langle X_n - \mu, f \rangle\rangle, n = 1, \dots, N\}$  be the set of projections of  $f$  onto the centered observations. We identify this set as the point  $\mathbf{M}_f$  in the space  $\mathbb{R}^N$ . The space  $\mathbb{R}^N$  is referred to as the features' space. The cloud of points that represents the set of observations in  $\mathbb{R}^N$  is denoted by  $C_N$ . Let  $\mathbf{G}_\mu$  be the centre of gravity of the cloud  $C_N$ . In the space  $\mathbb{R}^N$ , its coordinates are given by  $L_X(\mu) = \{\sqrt{\pi_n} \langle\langle X_n - \mu, \mu \rangle\rangle, n = 1, \dots, N\}$ . If the data are centered, the origin  $\mathbf{O}_{\mathbb{R}}$  of the axes in  $\mathbb{R}^N$  coincides with  $\mathbf{G}_\mu$ .

We consider the usual inner-product in  $\mathbb{R}^N$ , such that for all  $u, v \in \mathbb{R}^N$ ,  $(u, v) = u^\top v$ , associated with the norm  $(\cdot)$ . Let  $f$  and  $g$  be two elements in  $\mathcal{H}$  and denote by  $\mathbf{M}_f$  and  $\mathbf{M}_g$  their associated points in  $C_N$  (see Figure 4). The distance between  $\mathbf{M}_f$  and  $\mathbf{M}_g$  is thus defined as

$$d^2(\mathbf{M}_f, \mathbf{M}_g) = ((L_X(f) - L_X(g)))^2 = \sum_{n=1}^N \pi_n \langle\langle X_n - \mu, f - g \rangle\rangle^2.$$



**Fig. 4:** Left: Cloud of features. Right: Projection of the points on the elements of  $\mathbb{R}^N$ . The observation  $f$  (resp.  $g$ ) is identified by the point  $M_f$  (resp.  $M_g$ ) in the cloud  $C_N$ . The point  $G_\mu$  is the center of gravity of  $C_N$  and the point  $O_R$  is the origin of the space  $\mathbb{R}^N$ .

Similarly to the cloud of individuals, this distance characterizes the shape of the cloud  $C_N$  and we also have access to this characterization through the distance with the center of gravity  $G_\mu$ . Let  $f$  be an element of  $\mathcal{H}$ , associated to the point  $M_f$ , and  $\mu$  the element of  $\mathcal{H}$  related to  $G_\mu$ , the distance between  $M_f$  and  $\mu$  is given by

$$d^2(M_f, G_\mu) = ((L_X(f) - L_X(\mu))^2 = \sum_{n=1}^N \pi_n \langle X_n - \mu, f - \mu \rangle^2.$$

Given the set  $\mathcal{X}$ , the total inertia of  $C_N$ , with respect to  $G_\mu$  and the distance  $d$ , is given by

$$\sum_{n=1}^N \pi_n d^2(M_n, G_\mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(M_i, M_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p. \quad (8)$$

Using the distances induced by the duality diagram, the total inertia of the cloud  $C_N$  is thus equal to the total inertia of the cloud  $C_P$ . This property highlights the duality between the spaces  $\mathcal{H}$  and  $\mathbb{R}^N$ . To further emphasize this duality, the cosine of the angle  $\theta_{fg}$  formed by the two points  $M_f$  and  $M_g$  is equal to their correlation coefficient and can be written

$$\cos(\theta_{fg}) = \frac{(L_X(f), L_X(g))}{((L_X(f)) (L_X(g)))} = \frac{\langle f, g \rangle_\Gamma}{\|f\|_\Gamma \|g\|_\Gamma}.$$

The derivation of these equalities are given in Appendix 8.

**Remark 5.** Although each axis of the space does not directly represent the features, but rather the projection of an element of  $\mathcal{H}$  onto the elements of the set  $\mathcal{X}$ , we refer to this space as the features' space. We use this terminology to highlight the similarity between multivariate functional data analysis and traditional multivariate data analysis, as well as to emphasize the dimensionality of this space.

### 3.4. On centering and reducing

For conducting an MFPCA, the features are usually assumed centred [18]. [32] give a complete overview of centering in the context of FDA. Here, we comment on the geometric interpretation of centering in this context and compare with the multivariate scalar case. We focus on the usual centering in FDA, namely  $X_n^{(p)}(t_p) - \mu^{(p)}(t_p)$ ,  $t_p \in \mathcal{T}_p$  (referred to *object centering* in [32]). The geometric interpretation of object centering is the same if we refer to the observation space  $\mathcal{H}$  or the feature space  $\mathbb{R}^N$ . Within the space  $\mathcal{H}$  (resp.  $\mathbb{R}^N$ ), centering is interpreted as translating the centre of gravity of the clouds,  $G_\mu$  (resp.  $G_\mu$ ), to the origin point  $O_H$  (resp.  $O_R$ ) of the space  $\mathcal{H}$  (resp.  $\mathbb{R}^N$ ). This transformation, being a translation, does not change the shape of the cloud  $C_P$  (resp.  $C_N$ ). The interpretation is the same as for the centering in the multivariate scalar data context within their observation space.

Concerning the standardization of the data, there are two main proposals in the literature. [18] propose to consider the weights

$$w_p = \left( \int_{\mathcal{T}_p} \text{Var } X^{(p)}(t_p) dt_p \right)^{-1}.$$

This standardization is coherent with the derivation of the total inertia of the observation space using the usual distance in  $\mathcal{H}$ . [6] and [22] propose the weighting function

$$w_p(t_p) = (\text{Var } X^{(p)}(t_p))^{-1}, \quad t_p \in \mathcal{T}_p.$$

This corresponds to a standardization of the curves by the standard deviation of the component at each sampling point. The standard deviation curve is estimated as the square root of the diagonal of the covariance function estimates, obtained using a local linear smoother of the pooled data. For each functional feature  $p$ , this standardization mimics the standardization used for principal components analysis if the number of (scalar) features is infinite. Considering the duality diagram and the total inertia of the clouds with respect to the distance  $d_\Gamma$ , we propose to use the weights

$$w_p = \left( \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p \right)^{-1}.$$

The total inertia of  $C_P$  (and  $C_N$ ) will thus be equal to the number of components. In every case, we may consider the rescaled elements  $\tilde{X}^{(p)} = w_p^{1/2} \{X^{(p)} - \mu^{(p)}\}$  in place of the elements  $X^{(p)}$ .

#### 4. Multivariate functional principal components analysis

Assuming that the covariance operator  $\Gamma$  is a compact positive operator on  $\mathcal{H}$  and using the results in [18], and the theory of Hilbert-Schmidt operators, e.g., [35], there exists a complete orthonormal basis  $\Phi = \{\phi_k\}_{k \geq 1} \subset \mathcal{H}$  associated to a set of real numbers  $\{\lambda_k\}_{k \geq 1}$  such that  $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$  that satisfy

$$\Gamma \phi_k = \lambda_k \phi_k, \quad \text{and} \quad \lambda_k \rightarrow 0 \quad \text{as} \quad k \rightarrow \infty. \quad (9)$$

The set  $\{\lambda_k\}_{k \geq 1}$  contains the eigenvalues of the covariance operator  $\Gamma$  and  $\Phi$  contains the associated eigenfunctions. Using the multivariate Karhunen-Loëve theorem [18], we obtain the decomposition

$$X(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{k=1}^{\infty} c_k \phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T} \quad (10)$$

where  $c_k = \langle\langle X - \mu, \phi_k \rangle\rangle$  are the projections of the centered curves onto the eigenfunctions. We have that  $\mathbb{E}(c_k) = 0$ ,  $\mathbb{E}(c_k^2) = \lambda_k$  and  $\mathbb{E}(c_k c_{k'}) = 0$  for  $k \neq k'$ . Note that the coefficients  $c_k$  are scalar random variables while the multivariate functions  $\phi_k$  are vectors of functions. Let us call  $\Phi$  the multivariate functional principal component analysis basis. In practice, we use a truncated version of the Karhunen-Loëve expansion (10) as the eigenvalues  $\lambda_k$ , and hence the contribution of  $c_k$  to (10), becomes negligible as  $k$  goes to infinity. Let

$$X_{[K]}(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{k=1}^K c_k \phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}, \quad K \geq 1, \quad (11)$$

be the truncated Karhunen-Loëve expansion of the process  $X$  and

$$X_{[K_p]}^{(p)}(t_p) = \mu^{(p)}(t_p) + \sum_{k=1}^{K_p} c_k^{(p)} \varphi_k^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad K_p \geq 1, \quad 1 \leq p \leq P, \quad (12)$$

be the truncated Karhunen-Loëve expansion of the  $p$ th feature of the process  $X$ . For each  $p$ , the function  $\mu^{(p)}$  is the  $p$ th feature of the multivariate mean function  $\mu$  and the set  $\{\varphi_k^{(p)}\}_{1 \leq k \leq K_p}$  is a basis of univariate functions in  $L^2(\mathcal{T}_p)$ , whose elements are not the components of the multivariate functions  $\phi_k$ . In (12), the coefficients  $c_k^{(p)}$  are the projection of the centered curve  $X^{(p)}$  onto the eigenfunctions  $\varphi_k^{(p)}$  and are not (directly) related to the coefficients  $c_k$  in (11).

#### 4.1. Diagonalization of the covariance operator

The estimation of the eigencomponents of the covariance  $\Gamma$  by its diagonalization is derived in [18] for a general class of multivariate functional data defined on different dimensional domains. They give a direct relationship between the truncated representation (12) of the single elements  $X^{(p)}$  and the truncated representation (11) of the multivariate functional data  $X$ .

We recall here how to estimate the eigencomponents. Following Happ and Greven [18, Prop. 5], the multivariate components for  $X$  are estimated by a weighted combination of the univariate components computed from each  $X^{(p)}$ . First, we perform a univariate FPCA on each of the features of  $X$  separately. For a feature  $X^{(p)}$ , the eigenfunctions and eigenvectors are computed using a matrix decomposition of the estimated covariance  $C_{p,p}$  from (3). This results in a set of eigenfunctions  $\{\varphi_k^{(p)}\}_{1 \leq k \leq K_p}$  associated with a set of eigenvalues  $\{\lambda_k^{(p)}\}_{1 \leq k \leq K_p}$  for a given truncation integer  $K_p$ . Then, the univariate scores for a realization  $X_n^{(p)}$  of  $X^{(p)}$  are given by  $\mathbf{c}_{nk}^{(p)} = \langle X_n^{(p)}, \varphi_k^{(p)} \rangle$ ,  $1 \leq k \leq K_p$ . These scores might be estimated by numerical integration for example. Considering  $K_+ := \sum_{p=1}^P K_p$ , we then define the matrix  $\mathcal{Z} \in \mathbb{R}^{N \times K_+}$ , where on each row we concatenate the scores obtained for the  $P$  features of the  $n$ th observation:  $(\mathbf{c}_{n1}^{(1)}, \dots, \mathbf{c}_{nK_1}^{(1)}, \dots, \mathbf{c}_{n1}^{(P)}, \dots, \mathbf{c}_{nK_P}^{(P)})$ . An estimation of the covariance of the matrix  $\mathcal{Z}$  is given by  $\mathbf{Z} = (N-1)^{-1} \mathcal{Z}^\top \mathcal{Z}$ . An eigenanalysis of the matrix  $\mathbf{Z}$  is carried out to estimate the eigenvectors  $\mathbf{v}_k$  and eigenvalues  $\lambda_k$ . Finally, the multivariate eigenfunctions are estimated as a linear combination of the univariate eigenfunctions using

$$\phi_k^{(p)}(t_p) = \sum_{l=1}^{K_p} [\mathbf{v}_k]_l^{(p)} \varphi_l^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad 1 \leq k \leq K_+, \quad 1 \leq p \leq P,$$

where  $[\mathbf{v}_k]_l^{(p)}$  denotes the  $l$ th entry of the  $p$ th block of the vector  $\mathbf{v}_k$ . The multivariate scores are estimated as

$$c_{nk} = \mathcal{Z}_{n,:} \mathbf{v}_k, \quad 1 \leq n \leq N, \quad 1 \leq k \leq K_+,$$

where  $\mathcal{Z}_{n,:}$  is the  $n$ th row of the matrix  $\mathcal{Z}$ . We refer the reader to [18] for the derivation of the eigencomponents of the covariance operator if the curves are expanded in a general basis of functions.

#### 4.2. Diagonalization of the inner product matrix

We use the duality relation between row and column spaces of a data matrix to estimate the eigencomponents of the covariance operator. Consider the inner-product matrix  $\mathbf{M}$ , with entries defined in (1) and assuming that all observations are equally weighted, i.e., for all  $n = 1, \dots, N$ ,  $\pi_n = 1/N$ . Let  $\{l_k\}_{1 \leq k \leq N}$  such that  $l_1 \geq \dots \geq l_N \geq 0$  be the set of eigenvalues and  $\{\mathbf{u}_k\}_{1 \leq k \leq N}$  be the set of eigenvectors of the matrix  $\mathbf{M}$ . The relationship between all nonzero eigenvalues of the covariance operator  $\Gamma$  and the eigenvalues of  $\mathbf{M}$  is given by

$$\lambda_k = l_k, \quad k = 1, 2, \dots, N, \tag{13}$$

while the relationship between the multivariate eigenfunctions of the covariance operator  $\Gamma$  and the orthonormal eigenvectors of  $\mathbf{M}$  is given by

$$\phi_k(\mathbf{t}) = \frac{1}{\sqrt{N l_k}} \sum_{n=1}^N [\mathbf{u}_k]_n \{X_n(\mathbf{t}) - \mu(\mathbf{t})\}, \quad \mathbf{t} \in \mathcal{T}, \quad k = 1, 2, \dots, N, \tag{14}$$

where  $[\mathbf{u}_k]_n$  is the  $n$ th entry of the vector  $\mathbf{u}_k$ . The scores are then computed as the inner-product between the multivariate curves and the multivariate eigenfunctions and are given by

$$c_{nk} = \sqrt{N l_k} [\mathbf{u}_k]_n, \quad n = 1, 2, \dots, N, \quad k = 1, 2, \dots, N. \tag{15}$$

The derivations of these equalities are given in Appendix ?? in a slightly more general framework where the observation weights  $\pi_n$  are not equal. These results can be extended in a natural way if all the curves are expanded in a general basis of functions, see Section ?? in the Supplementary Material.

### 4.3. Computational complexity

We describe the time complexity for the computation of the MFPCA algorithm using the covariance operator and the Gram matrix. Considering the observation of  $N$  curves with  $P$  features, we assume that all observations of feature  $p$  are sampled on a common grid of  $M_p$  points. For  $a \in \mathbb{N}$ , let  $M^a = \sum_p M_p^a$ . Let  $K$  be the number of multivariate eigenfunctions to estimate. For the estimation of the eigencomponents using the covariance operator, we have  $K \leq K_+$ . While  $K$  has the same interpretation for both the eigendecomposition of the covariance operator and the eigendecomposition of the inner product matrix, in the latter case, it is not computed as the summation over the univariate elements, but rather as the number of components needed to achieve a certain amount of variance explained. Here, we also assume that the curves are perfectly observed, and thus no smoothing step is included in the expression of the time complexity. Note that the smoothing step will often have the same impact on complexity between the approaches as the smoothing is a preprocessing step.

To estimate the time complexity, we count the number of elementary operations performed, considering a fixed execution time for each. Worst-case time complexity is considered. We first give the time complexity for the estimation of the eigencomponents using the covariance operator by explaining the time complexity of each individual step (see [18] and Section 4.1). For each feature  $p$ , the time complexity of the estimation of the covariance matrix is  $O(NM_p^2)$ , of the eigendecomposition of the matrix is  $O(M_p^3)$  and of the univariate scores is  $O(NM_pK_p)$ . Therefore, the total time complexity is the sum over the  $p$  univariate time complexities. The covariance matrix  $\mathbf{Z}$  of the stacked univariate scores  $\mathcal{Z}$  is then computed with a time complexity of  $O(NK_+^2)$ , because the dimension of the matrix  $\mathcal{Z}$  is  $N \times K_+$ . The eigendecomposition of the matrix  $\mathbf{Z}$  has a time complexity of  $O(K_+^3)$ . The final step is to compute the multivariate eigenfunctions and scores. For the estimation of the  $K \leq K_+$  multivariate eigenfunctions, the time complexity is  $O(K \sum_p M_p K_p)$  and for the estimation of the scores, the time complexity is  $O(NK^2)$ . Gathering all the results, the final complexity of the estimation of the eigencomponents using the eigendecomposition of the covariance operator is

$$O\left(NM^2 + M^3 + N \sum_{p=1}^P M_p K_p + NK_+^2 + K_+^3 + K \sum_{p=1}^P M_p K_p + NK^2\right).$$

We now consider the time complexity of the estimation of the eigencomponents using the eigendecomposition of the inner product matrix (see Section 4.2). The inner product between two curves can be estimated in  $O(M^1)$ . Since there are  $N^2$  terms in the matrix, the time complexity for the computation of the inner product matrix is then  $O(N^2M^1)$ . The eigendecomposition of this matrix has a time complexity of  $O(N^3)$ . For the multivariate eigenfunctions, the time complexity is  $O(KNP)$  and is  $O(KN)$  for the multivariate scores. Gathering all the results, the final complexity of the estimation of eigencomponents using the eigendecomposition of the inner product matrix is

$$O\left(N^2M^1 + N^3 + KNP + KN\right).$$

The number of components  $K$  to estimate is usually small compared to the number of curves  $N$  or to the total number of sampling points  $M^1$ . Both time complexities can then be reduced to  $O(NM^2 + M^3)$  for the diagonalization of the covariance operator and to  $O(N^2M^1 + N^3)$  using the Gram matrix. If the number of observations is large compared to the total number of sampling points, it thus seems preferable to use the covariance operator to estimate the eigencomponents, while if the total number of sampling points is large compare to the number of observations, the use of the Gram matrix seems better. Note that the number of features  $P$  does not have much impact on the computational complexity, in the sense that the important part is the total number of sampling points. One component with 1000 sampling points will have the same computational complexity as 100 components with 10 sampling points. These results are confirmed in the simulation (see Section 5.2).

**Remark 6.** We can use singular values decomposition (SVD) in both cases to make the algorithm faster as it allows to compute only the first  $K$  eigenfunctions. In practice, this might be important as the maximum number of non-zero eigenvalues is the minimum between the number of observations and the number of sampling points.

## 5. Empirical analysis

Using simulated data, we compare the estimation of the eigencomponents and the reconstruction of the curves using the diagonalization of the covariance operator, (a) based on univariate FPCA as well as (b) P-splines basis

expansion, and (c) the diagonalization of the Gram matrix. These methods will be referred as (a) (Tensor) PCA, (b) 2D/1D B-Splines and (c) Gram respectively. For the diagonalization of the covariance operator, we consider the methodology of [18]. In the case of MFPCA based on the expansion of each univariate feature into univariate principal components ((Tensor) PCA), the eigencomposition of the 1-dimensional curves is performed using univariate FPCA and the eigendecomposition of the 2-dimensional curves is calculated using the Functional Canonical Polyadic-Tensor Power Algorithm (FCP-TPA) for regularized tensor decomposition [1]. In the case of MFPCA based on P-splines basis expansion (2D/1D B-Splines), data are expanded in B-splines basis with suitable smoothness penalty [9].

The results of the simulation are compared using computation times (CT), the integrated squared error (ISE) for the multivariate eigenfunctions, the relative squared error (RSE) for the eigenvalues and the mean relative squared error (MRSE) for the reconstructed data. Let  $\phi_k$  be the true eigenfunction and  $\widehat{\phi}_k$  the estimated eigenfunction defined on  $\mathcal{T}$ . We then define the ISE as

$$\text{ISE}(\phi_k, \widehat{\phi}_k) = \left\| \phi_k - \widehat{\phi}_k \right\|^2 = \sum_{p=1}^P \int_{\mathcal{T}_p} \{ \phi_k^{(p)}(t_p) - \widehat{\phi}_k^{(p)}(t_p) \}^2 dt_p, \quad k = 1, \dots, K. \quad (16)$$

Let  $\lambda = \{\lambda_1, \dots, \lambda_K\}$  be the set of true eigenvalues and  $\widehat{\lambda} = \{\widehat{\lambda}_1, \dots, \widehat{\lambda}_K\}$  be the set of estimated eigenvalues. We then define the RSE as

$$\text{RSE}(\lambda_k, \widehat{\lambda}_k) = (\lambda_k - \widehat{\lambda}_k)^2 / \lambda_k^2, \quad k = 1, \dots, K. \quad (17)$$

Let  $X$  be the set of true data and  $\widehat{X}$  be the set of reconstructed data. We define the MISE of the reconstructed data as

$$\text{MRSE}(X, \widehat{X}) = \frac{1}{N} \sum_{n=1}^N \left\| X_n - \widehat{X}_n \right\|^2 = \frac{1}{N} \sum_{n=1}^N \sum_{p=1}^P \int_{\mathcal{T}_p} \{ X_n^{(p)}(t_p) - \widehat{X}_n^{(p)}(t_p) \}^2 dt_p. \quad (18)$$

Each integral is approximated by the trapezoidal rule with an equidistant grid.

### 5.1. Simulation experiments

The simulation setting is based on the *Setting 3* of the simulation in [18]. The data generating process is based on a truncated version of the Karhunen-Loève decomposition. We simulate multivariate functional data with  $P = 2$  components. For the first component, we generate an orthonormal basis  $\{\phi_k^{(1)}\}_{1 \leq k \leq K}$  of  $\mathcal{L}^2(\mathcal{T}_1)$  on an interval  $\mathcal{T}_1 = [0, 1] \times [0, 0.5]$  as the tensor product of the first Fourier basis functions:

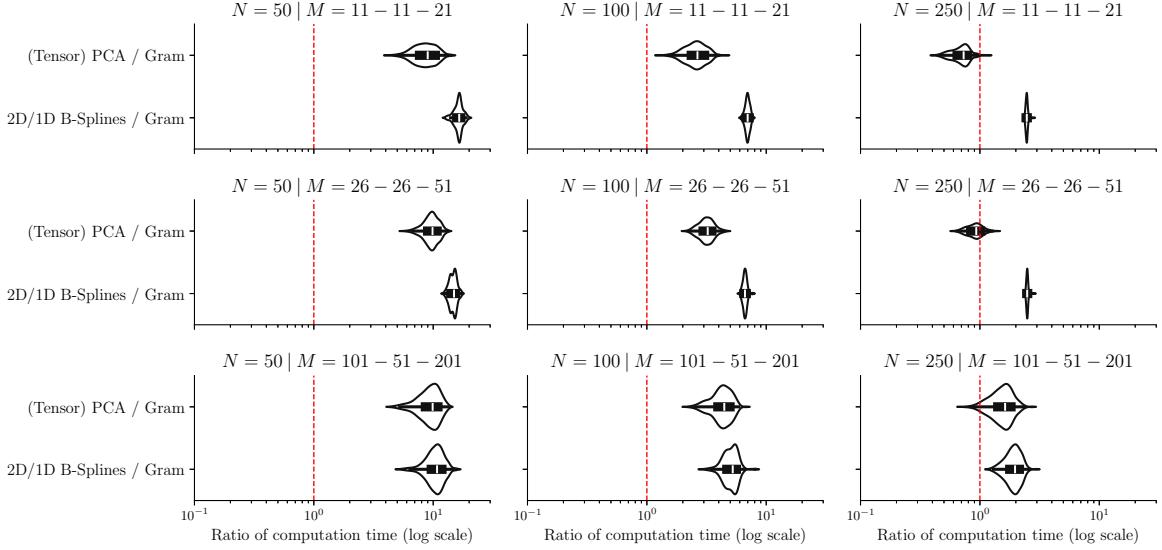
$$\phi_k^{(1)}(s, t) = \psi_l(s) \otimes \psi_m(t), \quad s \in [0, 1] \text{ and } t \in [0, 0.5], \quad k = 1, \dots, K,$$

where  $\psi_l$  and  $\psi_m$  are elements of the Fourier basis. For the second component, we generate an orthonormal Legendre basis  $\{\phi_k^{(2)}\}_{1 \leq k \leq K}$  of  $\mathcal{L}^2(\mathcal{T}_2)$  on an interval  $\mathcal{T}_2 = [-1, 1]$ . To ensure orthonormality, the basis  $\phi_k^{(1)}$  and  $\phi_k^{(2)}$  are weighted by random factors  $\alpha^{1/2}$  and  $(1 - \alpha)^{1/2}$ , respectively, where  $\alpha \sim \mathcal{U}(0.2, 0.8)$ . Each curve is then simulated using the truncated multivariate Karhunen-Loève expansion (11):

$$X(\mathbf{t}) = \sum_{k=1}^K c_k \phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T} := \mathcal{T}_1 \times \mathcal{T}_2,$$

where  $\phi_k = (\phi_k^{(1)}, \phi_k^{(2)})^\top$  and the scores  $c_k$  are sampled as random normal variables with mean 0 and variance  $\lambda_k$ . The eigenvalues  $\lambda_k$  are defined with an exponential decrease,  $\lambda_k = \exp(-(k+1)/2)$ . We simulate, for each replication of the simulation,  $N = 50, 100$  and  $250$  observations. The first component is sampled on a regular grid of  $M^{(1)} = 11 \times 11, 26 \times 26$  and  $101 \times 51$  sampling points. The second component is sampled on a regular grid of  $M^{(2)} = 21, 51$  and  $201$  sampling points. We set  $K = 25$ . We also consider data with measurement errors. In that case, we observe data from (2), where  $\sigma^2 = (0.25, 0.25)$ . Finally, we study sparse data as in [18], with medium (50% – 70% missing) and high (90% – 95% missing) sparsity in both components. For this setting, no noise is added to the data.

The (Tensor) PCA method is based on the univariate estimation of  $K^{(1)} = 20$  eigenimages and  $K^{(2)} = 15$  eigenfunctions. The estimation of the eigenimages is performed with the FCP-TPA where the smoothing parameters are chosen via cross-validation in  $[10^{-5}, 10^5]$  for dense, sparse and noisy data. In the case of sparse data, we also



**Fig. 5:** Ratio of computation time in the dense case between the (Tensor) PCA and 2D/1D B-Splines methods and the Gram method.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).

linearly interpolate the two-dimensional observations to get data that are regularly sampled to run the FCP-TPA. The estimation of the eigenfunctions is done using the PACE algorithm [42] with P-splines to smooth the mean and covariance functions. Penalties are chosen using cross-validation [9].

For the 2D/1D B-splines method,  $X^{(1)}$  is expanded in tensor products of  $K^{(1)} = 13 \times 13$  B-splines and  $X^{(2)}$  is expanded in  $K^{(2)} = 13$  B-splines. For dense data, no penalty is involved, while for sparse and noisy data, penalties are chosen using cross-validation.

Finally, the Gram method is based on the observed data points in the dense case, on the linear interpolation of the data in the sparse case [2] and on the P-splines smoothing of the data in the noisy case.

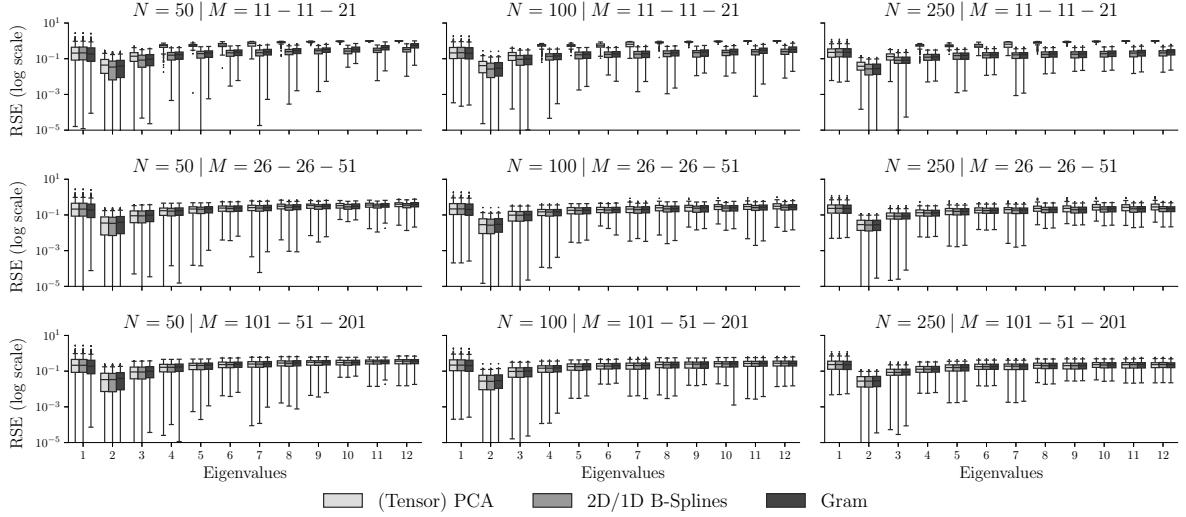
## 5.2. Simulation results

We compared MFPCA results of the different methods in terms of their CT, estimation of eigenvalues (RSE), estimation of eigenfunctions (ISE), and reconstruction of curves (MRSE). We fix the number of retained components to be 12 for each simulation scenario. Each experiment is repeated 200 times. The results are presented below.

**Computational time.** To compare the computational time of the different methods, we measured the time taken for each method to complete the MFPCA for each simulated dataset in the dense and noiseless case. Figure 5 shows the kernel density estimates of the ratio of CT for each method across all sample sizes and number of sampling points. Except when we have many observations with few sampling points, the Gram method is faster than the other two methods. This result is coherent with the computational complexity derived in Section 4.3.

The shorter computational time of the diagonalization of the Gram matrix makes it a more efficient option for analyzing two and higher-dimensional functional datasets as the number of sampling points tend to grow faster in these situations. However, the computational time can still vary depending on the specific implementation of each method, the computational resources available, the complexity of the dataset (number of observations, number of sampling points, etc.) and whether a smoothing step needs to be included.

**Eigenvalue estimation.** To compare the estimation of the eigenvalues between the different methods, we calculated the RSE (17) between the estimated eigenvalues and the true eigenvalues for each simulated dataset and for the twelve estimated eigenvalues. Figure 6 shows the boxplots of the RSE for each method across all sample sizes and number of sampling points. We found that the three methods behave similarly for the settings with a moderately large to a large number of sampling points. When we observe only a few sampling points,  $M^{(1)} = 11 \times 11$  and  $M^{(2)} = 21$ , the quality of the estimation is approximately the same for the first three eigenvalues for all methods, but from the



**Fig. 6:** RSE for the estimated eigenvalues for each method in the dense case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).

fourth eigenvalues, the 2D/1D B-splines and the Gram methods give slightly better results than the (Tensor) PCA method. The results for the sparse and noisy cases are similar to the dense and noiseless case and are provided in the Appendix 9.1.

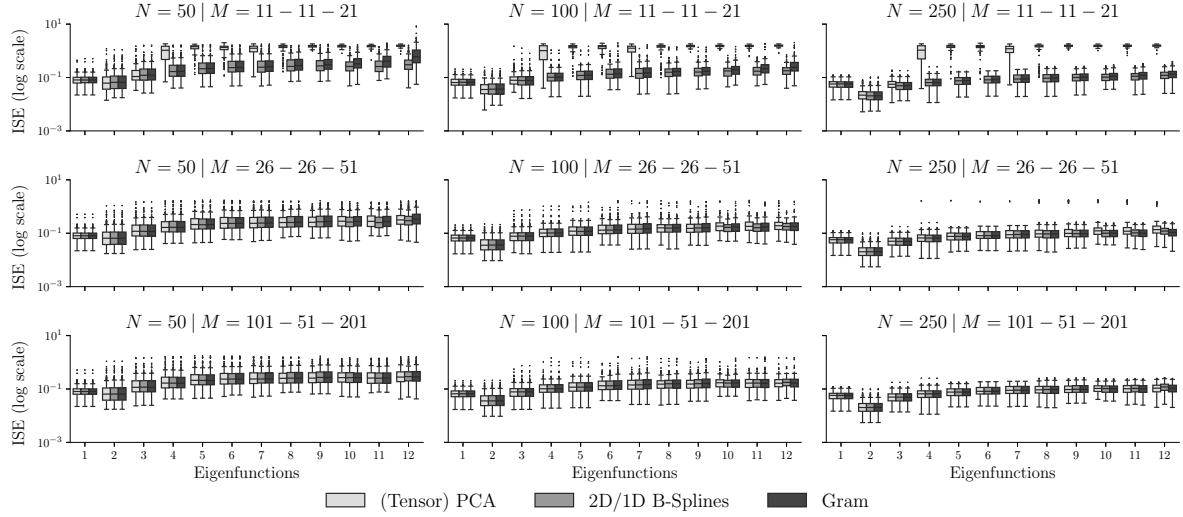
**Eigenfunction estimation.** To compare the estimation of the eigenfunctions between the different methods, we calculated the ISE (16) between the estimated eigenfunctions and the true eigenfunctions for each simulated dataset and for the twelve estimated eigenfunctions. Figure 7 shows the boxplots of the ISE for each method across all sample sizes and number of sampling points. The results are similar to those found when estimating the eigenvalues. When we observe only a few sampling points, the quality of the estimation of the eigenfunctions starts to decrease after the third component for the (Tensor) PCA method, while this is not the case for the other methods. The results for the sparse and noisy cases are similar to the dense and noiseless case and are provided in the Appendix 9.1.

**Curve reconstruction.** To compare the quality of the reconstruction of the curves between the different methods, we calculated the MRSE (18) between the reconstruction of the curves and the true curves for each simulated dataset. Figure 8 shows the boxplots of the MRSE for each method across all sample sizes and number of sampling points. Except in the case of few sampling points where the (Tensor) PCA method does not perform well (because of the poor estimation of the eigenvalues and eigenfunctions), all methods give nearly the same results for each setting. The results for the sparse and noisy cases are similar and are provided in the Appendix 9.1.

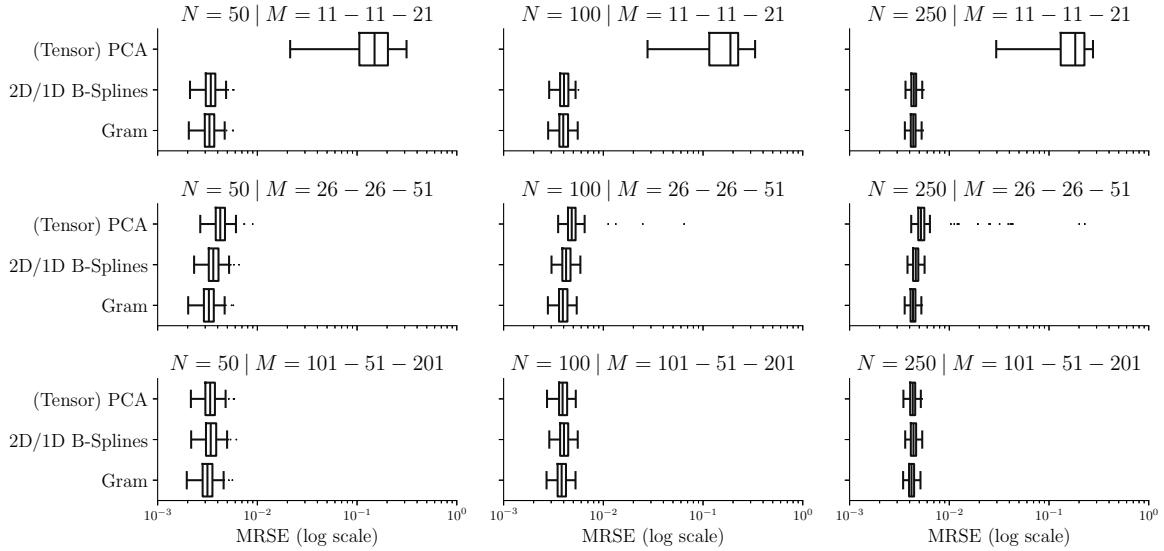
## 6. Application

We apply our methodology utilizing data to the National Basketball Association (NBA). To access comprehensive game data, we utilize the Python package `nba_api`<sup>1</sup>, which provides access to the APIs of `nba.com`. The dataset encompasses shot location data from all NBA games spanning the seasons between 2018 – 2019 and 2022 – 2023. Filtering the dataset, we focus solely on players who have made more than 1000 shots during this five-season period, resulting in a cohort of 131 players. These players accounted for a total of 493723 shots attempted, of which 234941 (47%) were successful. Subsequently, we exclude shots deemed impossible (e.g., out-of-bounds), leaving us with a dataset comprising 492621 shots (see Figure 9 for the shots chart of Stephen Curry). We remove all the shots that are close to the hoop (a square of  $2.7 \times 2.8$  meters around the hoop), as these shots will be the most common

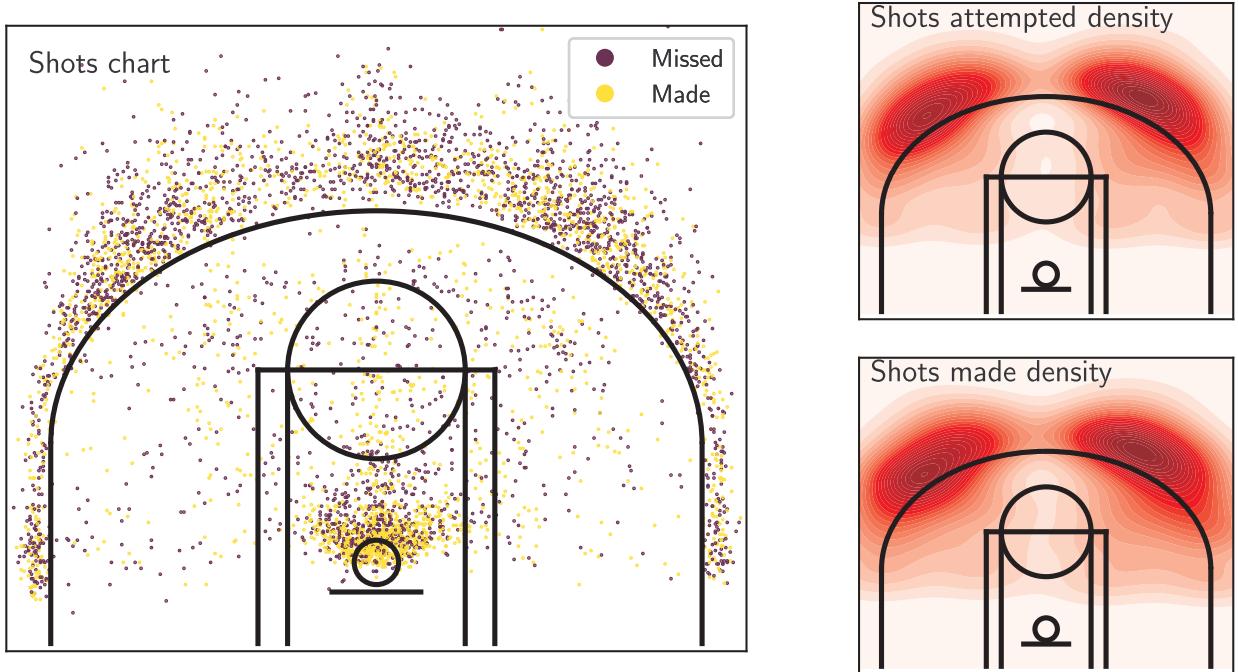
<sup>1</sup>[https://github.com/swar/nba\\_api/](https://github.com/swar/nba_api/)



**Fig. 7:** ISE for the estimated eigenfunctions for each method in the dense case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).



**Fig. 8:** MRSE for the reconstructed curves for each method in the dense case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).



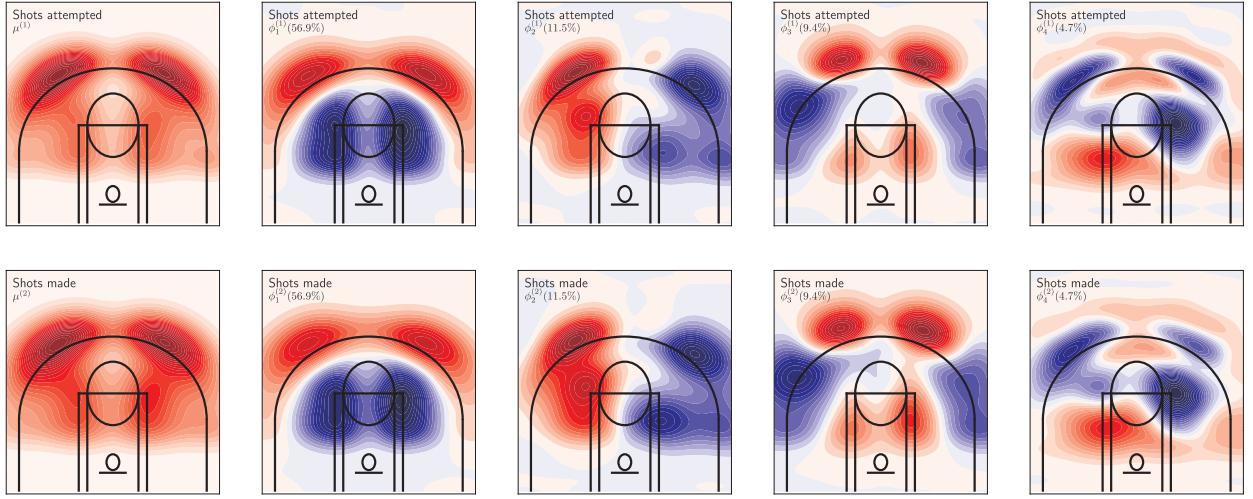
**Fig. 9:** Made/Missed shots chart (left) and the estimated densities (right) for Stephen Curry.

and their pattern is not interesting from a shooting behavior perspective, and players that have made fewer than 100 shots outside this restricted area during the five-season period. This results in a cohort of 119 players, with 186621 attempted shots and 71893 (38.5%) made shots.

To analyze shooting behavior, we employ a 2-dimensional kernel density estimation for both the attempted and made shots, utilizing Silverman's rule [37] for bandwidth estimation. The density estimation is conducted on a regularly spaced grid consisting of  $201 \times 201$  points (see Figure 9 for the estimated densities for Stephen Curry). We obtain observations of bivariate functional data (i.e.,  $P = 2$  functional features), where both features are defined on a two-dimensional rectangular domain, and observed on an equidistant grid of size  $201 \times 201$ .

We estimate the functional principal components using the Gram, (Tensor) PCA and 2D/1D B-Splines methods. For the Gram method, we directly use the estimated densities to estimate the eigencomponents. For the (Tensor) PCA method, the estimation of the multivariate eigenimages is based on the univariate estimation of 30 univariate eigenimages. The estimation of the (univariate) eigenimages is performed with the FCP-TPA where the smoothing parameters are chosen via cross-validation in  $[10^{-5}, 10^5]$ . For the 2D/1D B-splines method, the two components are expanded in tensor products of  $13 \times 13$  B-splines. We do not add a smoothing penalty in that case, as the smoothing step is performed in the kernel density estimation.

Figure 10 shows the estimated mean surfaces and eigenfunctions for the Gram method. The functional principal components, as a representation of the deviation from the mean surface function, may take negative values, while densities can only take positive values. For presentation purposes, the functional principal components have been normalized between  $-1$  and  $1$ . Blue areas correspond to negative values of the functional components, and thus contribute negatively to the density, while red area corresponds to positive values of the functional components, and thus contribute positively to the density. The decomposition of the density of made shots is similar that of attempted shots. The obtained functional components can be explained as different shooting behavior. The first component, accounting for 56.9% of the variance explained, contrasts two-point and three-point shooting. A player with a positive score for  $\phi_1$  will tend to take/attempt shots more behind the three-point line, while if he has a negative score, he will prefer shooting within the two-points zone. For the second component (11.5% of the variance explained), the contrast is between the left and right sides of the court. Similarly, if a player has a positive (resp. negative) score for  $\phi_2$ , he will prefer to shoot on the right (resp. left) side of the court while looking at the hoop. The third component (9.4%



**Fig. 10:** The estimated mean surfaces (first column) and the estimated eigenfunctions (second to fifth columns) for the shots dataset using the **Gram** method.

of the variance explained) contrasts shooting from the wing with shooting in the axis of the hoop. Positive (resp. negative) score players will shot in the axis of the hoop (resp. from the wing). The fourth component (4.7% of the variance explained) is more difficult to explain from a shooting behavior perspective. It may be linked to players having a preferred shooting location so that most of their shots are taken from the same position. The estimation of the eigencomponents with the (Tensor) PCA and 2D/1D B-Splines methods are similar to the results with the Gram method and are thus provided in the Appendix 9.

## 7. Discussion and conclusion

MFPCA is a fundamental statistical tool for the analysis of multivariate functional data, which enables us to capture the variability in observations defined by multiple curves. In this paper, we have described the duality between rows and columns of a data matrix within the context of multivariate functional data. We have proposed to use this duality to estimate the eigencomponents of the covariance operator in multivariate functional datasets. By comparing the results of the three methods, we provide the researcher with guidelines for determining the most appropriate method within a range of functional data frameworks.

Overall, our simulations showed that the (Tensor) PCA, 2D/1D B-Splines and Gram methods give similar results in terms of the estimation of the eigenvalues, the eigenfunctions and reconstruction of the curves. Regarding the computation time, the use of the Gram matrix is faster in most cases. The only situation where the diagonalization of the covariance operator is quicker is when the number of observations is larger than the number of sampling points. In conclusion, regarding the reconstruction error and computational complexity, if the data are defined on multi-dimensional domains (images) or the number of sampling points is much higher than the number of observations, we advise using the Gram matrix. Another advantage of the Gram method is that it does not require basis expansion, such as the 2D/1D B-splines method, or estimate smoothing parameters, such as the (Tensor) PCA method when the data are observed without noise.

In practice, observations of (multivariate) functional data are often subject to noise. In this case, the diagonalization of the covariance operator seems preferable as the curves as to be smoothed beforehand to estimate the Gram matrix and yield similar results, except when the number of sampling is small where the Gram method is preferable. In the case of sparsely sampled functional data, we would advise to use the Gram method, using linear interpolation to estimate the inner-product matrix, as it does not require hyper-parameters estimation and gives slightly better results than the other methods.

Utilizing the Gram matrix enables the estimation of the number of components retained via the percentage of variance explained by the multivariate functional data, whereas the decomposition of the covariance operator necessitates

the specification of the percentage of variance accounted for by each individual univariate feature. Specifying the percentage of variance explained for each feature does not guarantee recovery of the nominal percentage of variance explained for the multivariate functional data [13]. Although we have not investigated the extent to which this might be important, the duality relation derived in this work provides a direct solution to the problem. In settings where the univariate variance-explained cutoffs fail to retain the correct percentage of variance explained in multivariate functional data, the Gram matrix approach may be preferred.

The open-source implementation can be accessed at <https://github.com/StevenGolovkine/FDApy>, while scripts to reproduce the simulations and data analysis are at [https://github.com/FAST-ULxNUIG/geom\\_mfpca](https://github.com/FAST-ULxNUIG/geom_mfpca).

## Technical details - Derivation of the eigencomponents

### 7.1. General framework

In this section, we calculate the relationships between the eigenelements of the covariance operator  $\Gamma$  and the ones of the Gram matrix  $\mathbf{M}$  of a functional dataset. We then prove the equalities (13), (14) and (15).

Using the Hilbert-Schmidt theorem, there exists a complete orthonormal basis of eigenvectors  $\{\mathbf{u}_k\}_{1 \leq k \leq N}$  of the inner-product matrix  $\mathbf{M}$  such that

$$\mathbf{M}\mathbf{u}_k = l_k \mathbf{u}_k. \quad (19)$$

Let  $X = (X_1 - \mu, \dots, X_N - \mu)^\top$  and denote  $\tilde{X} = \text{diag}\{\sqrt{\pi_1}, \dots, \sqrt{\pi_N}\}X$ , the matrix of weighted observations. Recall that, in the case of  $P$ -dimensional process, the realisations of the process  $X_n$ ,  $n = 1, \dots, N$  and  $\mu$  are vectors of functions of length  $P$ , and thus  $X$  (and  $\tilde{X}$ ) is a matrix of functions of size  $N \times P$ . By left multiplying Equation (19) by  $\tilde{X}^\top$ , we obtain

$$\tilde{X}^\top \mathbf{M}\mathbf{u}_k = l_k \tilde{X}^\top \mathbf{u}_k. \quad (20)$$

Expanding Equation (20), for each component  $p = 1, \dots, P$ , we have,

$$\sum_{i=1}^N \sum_{j=1}^N \pi_i \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot)\} \langle\langle X_i - \mu, X_j - \mu \rangle\rangle = l_k \sum_{n=1}^N \sqrt{\pi_n} [\mathbf{u}_k]_n \{X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot)\}. \quad (21)$$

Here and in the following, we note  $[a]_p$  the  $p$ th entry of the vector  $a$ . Starting from the left side of Equation (21), we get

$$\begin{aligned} [\tilde{X}^\top \mathbf{M}\mathbf{u}_k]_p &= \sum_{i=1}^N \sum_{j=1}^N \pi_i \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot)\} \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \\ &= \sum_{q=1}^P \int_{\mathcal{T}_q} \sum_{i=1}^N \pi_i \{X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot)\} \{X_i^{(q)}(s_q) - \mu^{(q)}(s_q)\} \sum_{j=1}^N \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_j^{(q)}(s_q) - \mu^{(q)}(s_q)\} ds_q \\ &= \sum_{q=1}^P \int_{\mathcal{T}_q} C_{p,q}(\cdot, s_q) \sum_{j=1}^N \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_j^{(q)}(s_q) - \mu^{(q)}(s_q)\} ds_q \\ &= \sum_{j=1}^N \langle\langle C_p(\cdot, \cdot), \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_j - \mu\} \rangle\rangle \\ &= \Gamma \left( \sum_{j=1}^N \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_j - \mu\} \right)^{(p)}(\cdot), \end{aligned} \quad (22)$$

and, starting from the right side of Equation (21),

$$[l_k \tilde{X}^\top \mathbf{u}_k]_p = l_k \sum_{n=1}^N \sqrt{\pi_n} [\mathbf{u}_k]_n \{X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot)\}. \quad (23)$$

From Equation (22) and Equation (23), we obtain

$$\Gamma \left( \sum_{j=1}^N \sqrt{\pi_j} [\mathbf{u}_k]_j \{X_j - \mu\} \right)^{(p)} = l_k \sum_{n=1}^N \sqrt{\pi_n} [\mathbf{u}_k]_n \{X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot)\}, \quad \text{for all } p = 1, \dots, P.$$

By identification in Equation (9), we find that, for each components  $p$ ,

$$\lambda_k = l_k \quad \text{and} \quad \phi_k^{(p)}(\cdot) = \sum_{n=1}^N \sqrt{\pi_n} [\mathbf{u}_k]_n \{X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot)\}, \quad k \geq 1. \quad (24)$$

For  $k \geq 1$ , the norm of the eigenfunction is computed as the following:

$$\begin{aligned} |\phi_k|^2 &= \sum_{i=1}^N \sum_{j=1}^N \sqrt{\pi_i \pi_j} [\mathbf{u}_k]_i [\mathbf{u}_k]_j \langle X_i - \mu, X_j - \mu \rangle = \sum_{i=1}^N [\mathbf{u}_k]_i \sum_{j=1}^N \mathbf{M}_{ij} [\mathbf{u}_k]_j \\ &= \sum_{i=1}^N [\mathbf{u}_k]_i l_k [\mathbf{u}_k]_i = l_k (\|\mathbf{u}_k\|)^2 = l_k. \end{aligned}$$

Therefore, in order to have an orthonormal basis of eigenfunctions, we normalise the eigenfunctions  $\phi_k$  from Equation (24) by  $1/\sqrt{l_k}$ . Concerning the estimation of the scores, for  $n = 1, \dots, N$ , for  $k \geq 1$ , we have

$$\begin{aligned} c_{nk} &= \langle X_n - \mu, \phi_k \rangle = \frac{1}{\sqrt{l_k}} \sum_{j=1}^N \sqrt{\pi_j} [\mathbf{u}_k]_j \langle X_n - \mu, X_j - \mu \rangle \\ &= \frac{1}{\sqrt{l_k \pi_n}} \sum_{j=1}^N [\mathbf{u}_k]_j \mathbf{M}_{nj} = \sqrt{\frac{l_k}{\pi_n}} [\mathbf{u}_k]_n. \end{aligned}$$

If we assume that the observations are equally weighted, i.e.,  $\pi_n = 1/N, n = 1, \dots, N$ , we get the equalities (13), (14) and (15).

## Acknowledgments

S. Golovkine, A. J. Simpkin and N. Bargary are partially supported by Science Foundation Ireland under Grant No. 19/FFP/7002 and co-funded under the European Regional Development Fund. E. Gunning is supported in part Science Foundation Ireland (Grant No. 18/CRT/6049) and co-funded under the European Regional Development Fund.

## Appendix

### 8. Derivation of the equalities

#### 8.1. Proof of Lemma 1

Using the definition of adjoint operators, we must prove that

$$(L_X(f), u)_{\mathbf{M}} = \langle\langle f, L_X^*(u) \rangle\rangle_{\Gamma}, \quad \text{for all } f \in \mathcal{H}, \quad u \in \mathbb{R}^N. \quad (25)$$

*Proof.* For all  $f \in \mathcal{H}, u \in \mathbb{R}^N$ , we have that

$$\begin{aligned}
(L_X(f), u)_{\mathbf{M}} &= L_X(f)^{\top} \mathbf{M} u = \sum_{i=1}^N \sum_{j=1}^N \pi_i \sqrt{\pi_j} u_j \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \langle\langle X_i - \mu, f \rangle\rangle, \\
\langle\langle f, L_X^*(u) \rangle\rangle_{\Gamma} &= \langle\langle \Gamma f, L_X^*(u) \rangle\rangle \\
&= \sum_{p=1}^P \int_{\mathcal{T}_p} (\Gamma f)^{(p)}(t_p) \left\{ \sum_{n=1}^N \sqrt{\pi_n} u_n (X_n^{(p)}(t_p) - \mu^{(p)}(t_p)) \right\} dt_p \\
&= \sum_{p=1}^P \int_{\mathcal{T}_p} \left\{ \sum_{q=1}^P \int_{\mathcal{T}_q} C_{p,q}(t_p, s_q) f^{(q)}(s_q) ds_q \right\} \left\{ \sum_{j=1}^N \sqrt{\pi_j} u_j (X_j^{(p)}(t_p) - \mu^{(p)}(t_p)) \right\} dt_p \\
&= \sum_{i=1}^N \sum_{j=1}^N \pi_i \sqrt{\pi_j} u_j \left\{ \sum_{p=1}^P \int_{\mathcal{T}_p} (X_i^{(p)}(t_p) - \mu^{(p)}(t_p)) (X_j^{(p)}(t_p) - \mu^{(p)}(t_p)) dt_p \right\} \times \\
&\quad \left\{ \sum_{q=1}^P \int_{\mathcal{T}_q} (X_i^{(q)}(s_q) - \mu^{(q)}(s_q)) f^{(q)}(s_q) ds_q \right\} \\
&= \sum_{i=1}^N \sum_{j=1}^N \pi_i \sqrt{\pi_j} u_j \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \langle\langle X_i - \mu, f \rangle\rangle.
\end{aligned}$$

So, the equality (25) is proved and we conclude that  $L_X^*$  is the adjoint operator of  $L_X$ .  $\square$

## 8.2. Derivation of the inertia of the clouds

Recall that

$$\text{Var}\{X^{(p)}(t_p)\} = \sum_{n=1}^N \pi_n \{X_n^{(p)}(t_p)\}^2 - \{\mu^{(p)}(t_p)\}^2 \quad \text{where} \quad \mu^{(p)}(t_p) = \sum_{n=1}^N \pi_n X_n^{(p)}(t_p), \quad t_p \in \mathcal{T}_p.$$

We first show that the total inertia of the cloud of individuals  $C_P$  using the distance  $d$  is given by equalities in (6):

$$\sum_{n=1}^N \pi_n d^2(\mathbf{M}_n, \mathbf{G}_{\mu}) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(\mathbf{M}_i, \mathbf{M}_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \text{Var} X^{(p)}(t_p) dt_p.$$

*Proof.*

$$\begin{aligned}
\sum_{n=1}^N \pi_n d^2(\mathbf{M}_n, \mathbf{G}_{\mu}) &= \sum_{n=1}^N \pi_n \sum_{p=1}^P \left| X_n^{(p)} - \mu^{(p)} \right|^2 \\
&= \sum_{p=1}^P \left( \sum_{n=1}^N \pi_n \left| X_n^{(p)} \right|^2 - \left| \mu^{(p)} \right|^2 \right) \\
&= \sum_{p=1}^P \int_{\mathcal{T}_p} \text{Var} X^{(p)}(t_p) dt_p, \\
\sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(\mathbf{M}_i, \mathbf{M}_j) &= \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j \sum_{p=1}^P \left| X_i^{(p)} - X_j^{(p)} \right|^2 \\
&= \sum_{p=1}^P \left( 2 \sum_{i=1}^N \pi_i \left| X_i^{(p)} \right|^2 - 2 \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j \langle X_i^{(p)}, X_j^{(p)} \rangle \right) \\
&= 2 \sum_{p=1}^P \int_{\mathcal{T}_p} \text{Var} X^{(p)}(t_p) dt_p.
\end{aligned}$$

The equalities in Equation (6) are shown.  $\square$

Next, we derive the total inertia of the cloud of individuals  $C_P$  using the distance  $d_\Gamma$ , given by equalities in (7):

$$\sum_{n=1}^N \pi_n d_\Gamma^2(\mathbf{M}_n, \mathbf{G}_\mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d_\Gamma^2(\mathbf{M}_i, \mathbf{M}_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p.$$

*Proof.*

$$\begin{aligned} \sum_{n=1}^N \pi_n d_\Gamma^2(\mathbf{M}_n, \mathbf{G}_\mu) &= \sum_{n=1}^N \pi_n \langle\langle X_n - \mu, X_n - \mu \rangle\rangle_\Gamma \\ &= \sum_{n=1}^N \pi_n \|X_n\|_\Gamma^2 - \|\mu\|_\Gamma^2 \\ &= \sum_{p=1}^P \left( \sum_{n=1}^N \pi_n \|X_n^{(p)}\|_\Gamma^2 - \|\mu^{(p)}\|_\Gamma^2 \right) \\ &= \sum_{p=1}^P \sum_{q=1}^P \int_{\mathcal{T}_p} \int_{\mathcal{T}_q} C_{p,q}(t_p, s_q) C_{p,q}(t_p, s_q) ds_q dt_p \\ &= \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p, \\ \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d_\Gamma^2(\mathbf{M}_i, \mathbf{M}_j) &= \sum_{i=1}^N \pi_i \|X_i\|_\Gamma^2 - \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j \langle\langle X_i, X_j \rangle\rangle_\Gamma \\ &= \sum_{i=1}^N \pi_i \|X_i\|_\Gamma^2 - \|\mu\|_\Gamma^2 \\ &= \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p. \end{aligned}$$

The equalities in Equation (7) are shown.  $\square$

Finally, we derive the inertia of the cloud of features  $C_N$  using the distance  $d$ , given by equalities in (8):

$$\sum_{n=1}^N \pi_n d^2(\mathbf{M}_n, \mathbf{G}_\mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(\mathbf{M}_i, \mathbf{M}_j) = \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(t_p, \cdot)\|^2 dt_p.$$

*Proof.*

$$\begin{aligned}
\sum_{n=1}^N \pi_n d^2(M_n, G_\mu) &= \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \\
&= \sum_{p=1}^P \sum_{q=1}^P \int_{\mathcal{T}_p} \int_{\mathcal{T}_q} C_{p,q}(t_p, s_q) C_{p,q}(t_p, s_q) ds_q dt_p \\
&= \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(\cdot, t_p)\|^2 dt_p, \\
\sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j d^2(M_i, M_j) &= \sum_{i=1}^N \sum_{j=1}^N \sum_{n=1}^N \pi_i \pi_j \pi_n \langle\langle X_n - \mu, X_i - X_j \rangle\rangle \langle\langle X_n - \mu, X_i - X_j \rangle\rangle \\
&= 2 \sum_{i=1}^N \sum_{j=1}^N \pi_i \pi_j \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \\
&= 2 \sum_{p=1}^P \int_{\mathcal{T}_p} \|C_p(\cdot, t_p)\|^2 dt_p.
\end{aligned}$$

The equalities in Equation (8) are shown.  $\square$

## 9. More results

### 9.1. Simulation

We present the results of the simulations in the sparse and noisy cases. Figures 11, 13 and 15 present the boxplots of the RSE, ISE and MRSE, respectively, for the noisy case. To generate noisy data, we consider the model (2) where  $\sigma^2 = 0.25$ . For the (Tensor) PCA method, we first smooth the 2-dimensional data using P-Splines smoothing and estimate a smooth version of the mean and covariance functions for the one-dimensional data using P-Splines smoothing. For the 2D/1D B-Splines and Gram methods, all the observations have been smoothed using P-Splines smoothing beforehand. In every case, the penalty involved in P-Splines smoothing has been estimated using cross-validation.

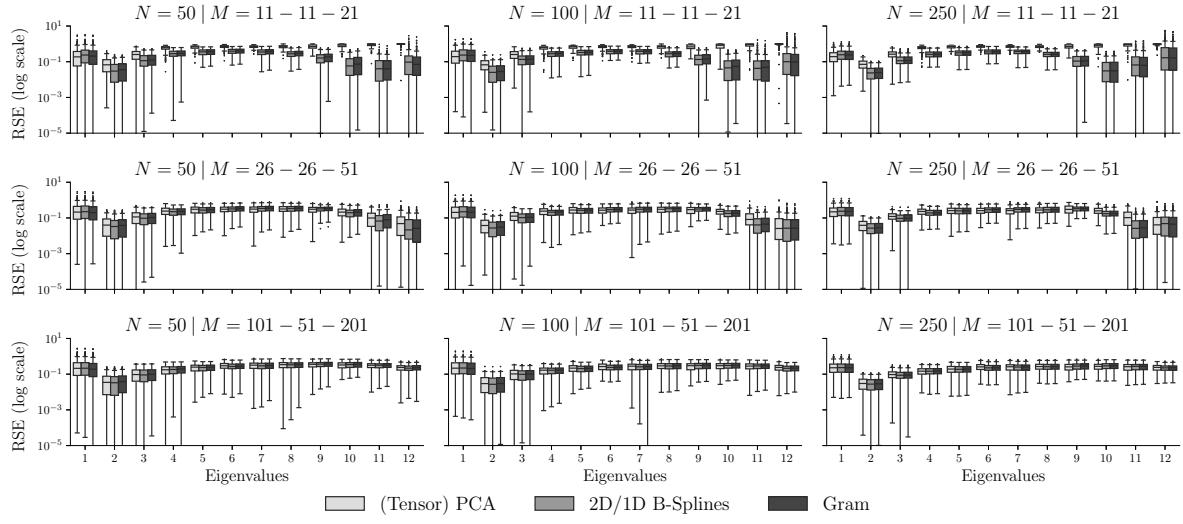
Figures 12, 14 and 16 present the boxplots of the RSE, ISE and MRSE, respectively, for the sparse case. In this case, we only consider  $N = 250$ ,  $M^{(1)} = 101 \times 51$ ,  $M^{(2)} = 201$  and no noise. We consider medium sparsity, where 50% – 70% of the sampling points have been removed and high sparsity, where 90% – 95% of the sampling points have been removed. For the (Tensor) PCA method, we first interpolate the 2-dimensional data to have the observations regularly sampled on a grid and estimate a smooth version of the mean and covariance functions for the one-dimensional data using P-Splines smoothing. For the 2D/1D B-Splines method, all the observations have been smoothed using P-splines smoothing with a penalty estimated by cross-validation. For the Gram method, the observations have been linearly interpolated to estimate the inner-product matrix.

### 9.2. Application

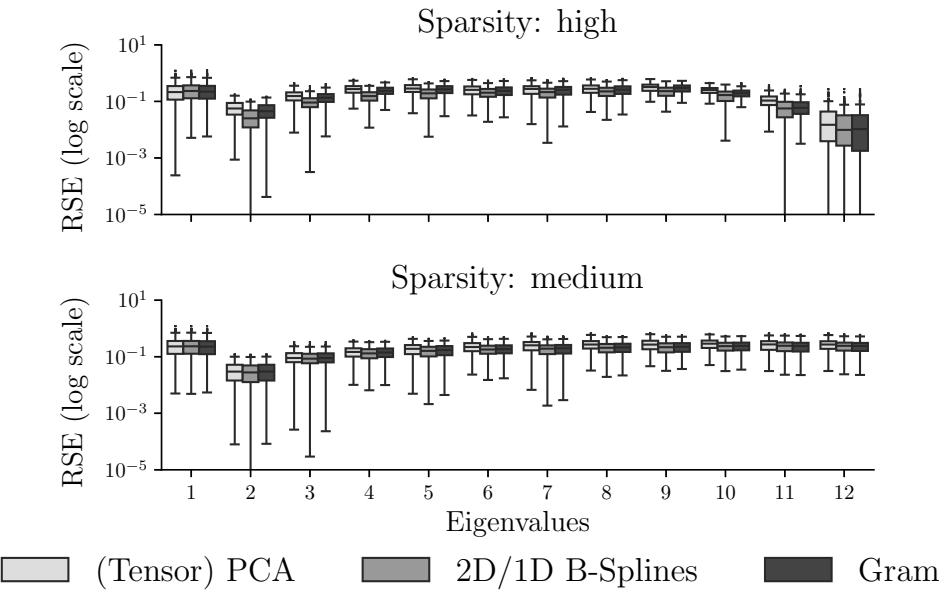
Here, we present the estimation of the eigencomponents of the shooting data using the (Tensor) PCA (see Figure 17) and 2D/1D B-Splines (see Figure 18) methods. The results are similar to those found using the Gram method for the eigenfunctions and the percentage of variance explained for each eigenfunctions.

## References

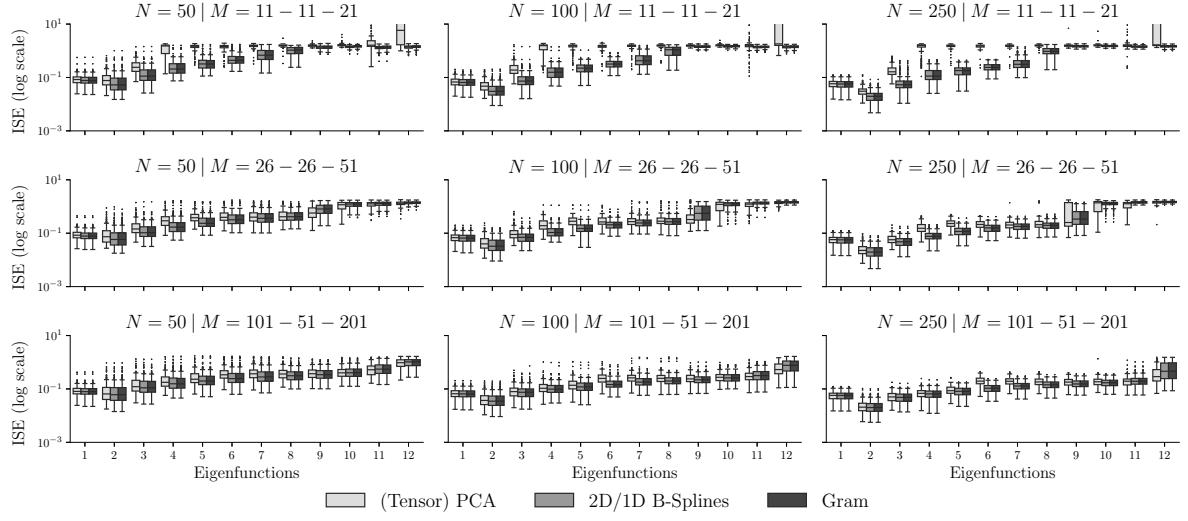
- [1] G. I. Allen, Multi-way functional principal components analysis, in: 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 220–223.
- [2] M. Benko, W. Härdle, A. Kneip, Common functional principal components, *The Annals of Statistics* 37 (2009) 1–34.



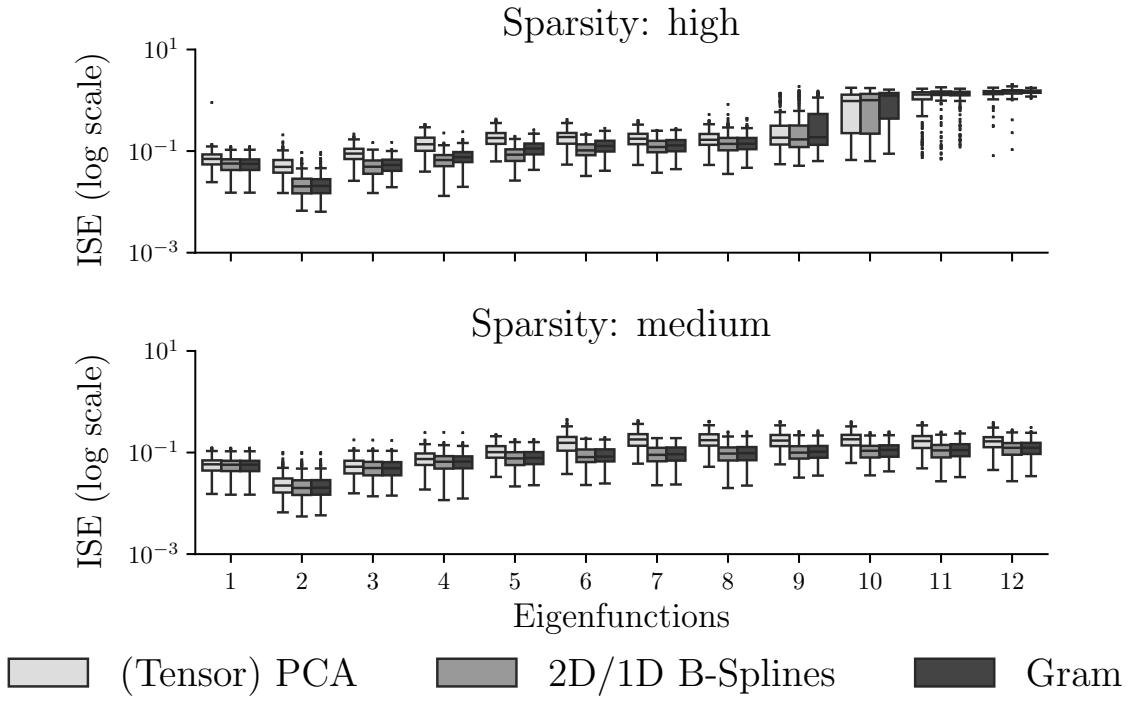
**Fig. 11:** RSE for the estimated eigenvalues for each method in the noisy case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).



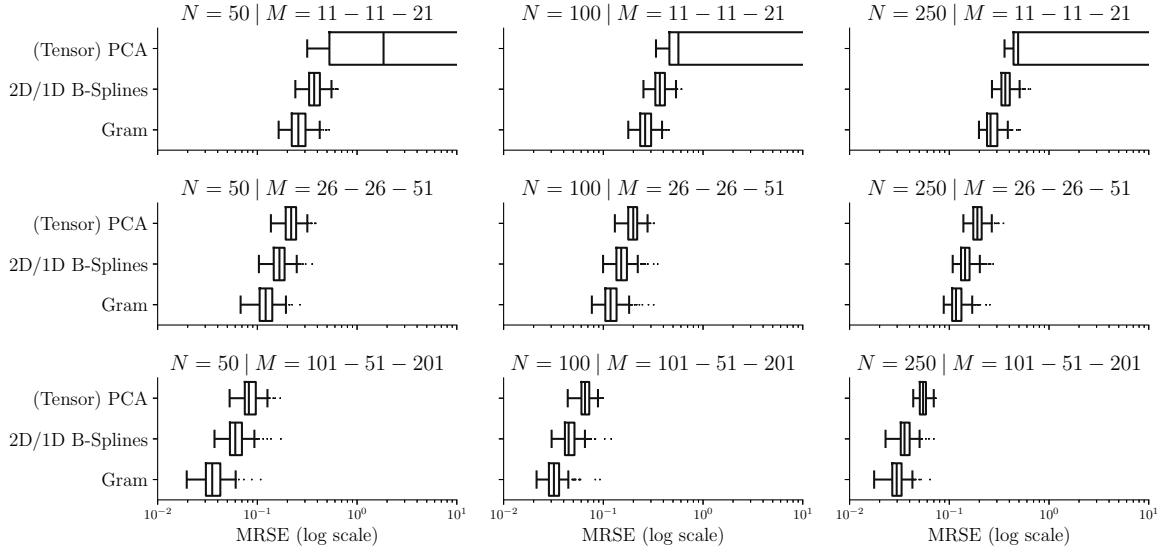
**Fig. 12:** RSE for the estimated eigenvalues for each method in the sparse case. We set  $N = 250$ ,  $M^{(1)} = 101 \times 51$  and  $M^{(2)} = 201$ . For the case of medium sparsity, we remove 50% – 70% of the sampling points and for the case of high sparsity, we remove 90% – 95% of the sampling points.



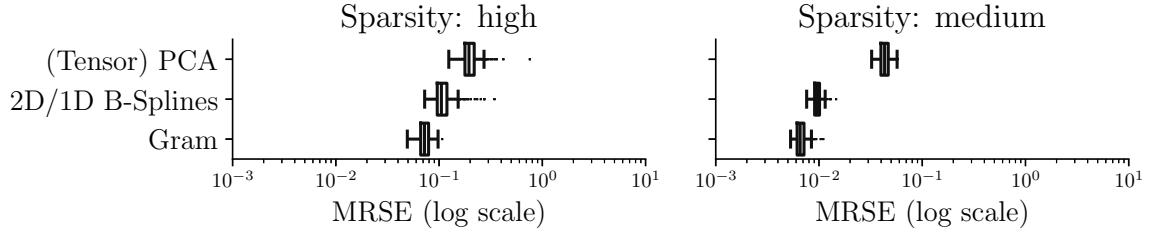
**Fig. 13:** ISE for the estimated eigenfunctions for each method in the noisy case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).



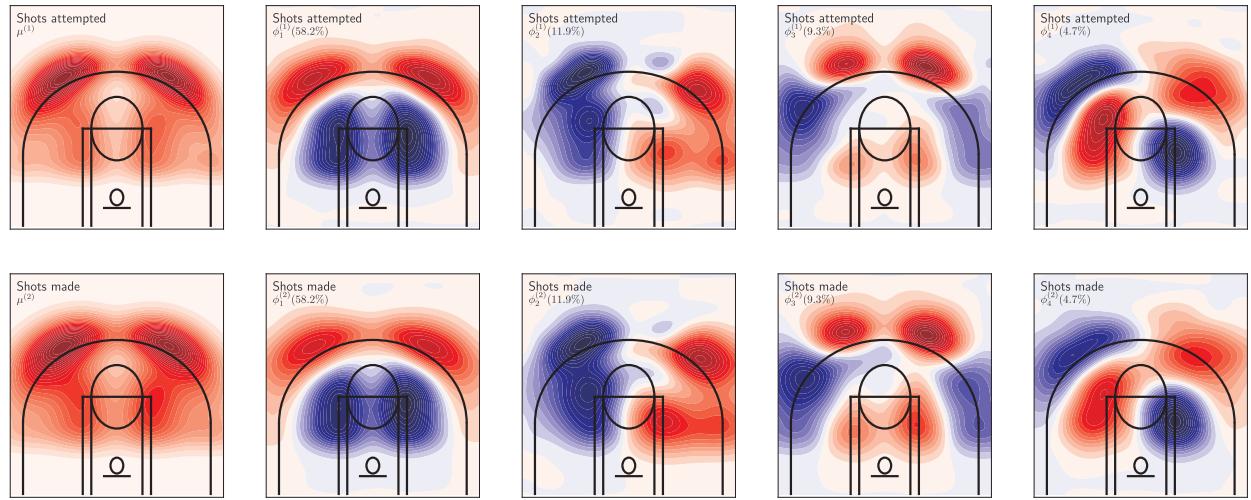
**Fig. 14:** ISE for the estimated eigenfunctions for each method in the sparse case. We set  $N = 250$ ,  $M^{(1)} = 101 \times 51$  and  $M^{(2)} = 201$ . For the case of medium sparsity, we remove 50% – 70% of the sampling points and for the case of high sparsity, we remove 90% – 95% of the sampling points.



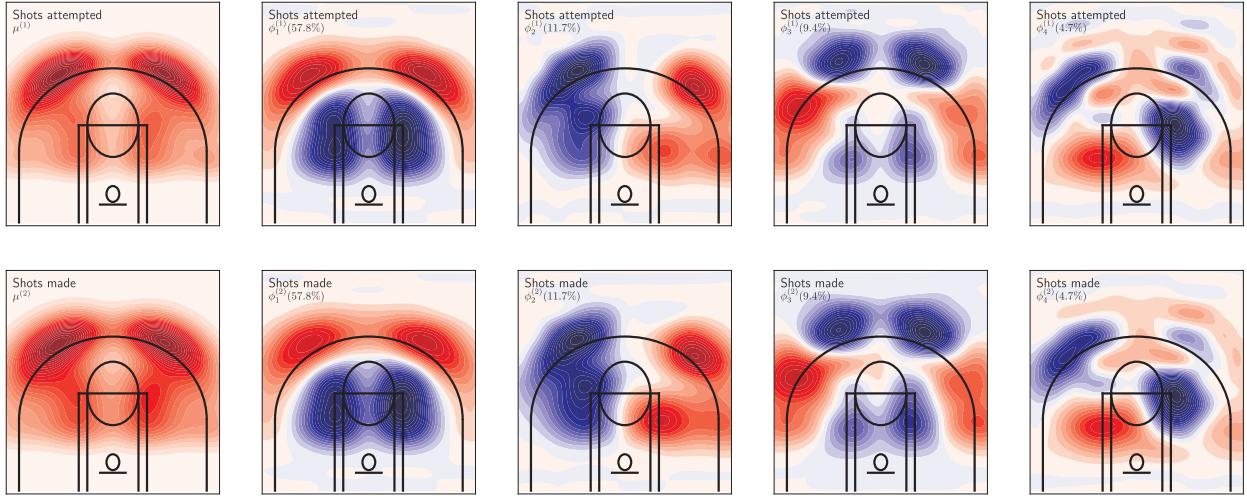
**Fig. 15:** MRSE for the reconstructed curves for each method in the noisy case.  $N$  is the number of observations,  $M$  is the number of sampling points per curve (the first two numbers are for the images and the last one is for the curves).



**Fig. 16:** MRSE for the reconstructed curves for each method in the sparse case. We set  $N = 250$ ,  $M^{(1)} = 101 \times 51$  and  $M^{(2)} = 201$ . For the case of medium sparsity, we remove 50% – 70% of the sampling points and for the case of high sparsity, we remove 90% – 95% of the sampling points.



**Fig. 17:** The estimated mean surfaces (first column) and the estimated eigenfunctions (second to fifth columns) for the shots dataset using the (Tensor) PCA method.



**Fig. 18:** The estimated mean surfaces (first column) and the estimated eigenfunctions (second to fifth columns) for the shots dataset using the 2D/1D B-splines method.

- [3] J. Berrendero, A. Justel, M. Svarc, Principal components for multivariate functional data, *Computational Statistics & Data Analysis* 55 (2011) 2619–2634.
- [4] J. R. Berrendero, B. Bueno-Larraz, A. Cuevas, On Mahalanobis Distance in Functional Settings, *Journal of Machine Learning Research* 21 (2020) 1–33.
- [5] K. Chen, X. Zhang, A. Petersen, H.-G. Müller, Quantifying Infinite-Dimensional Data: Functional Data Analysis in Action, *Statistics in Biosciences* 9 (2017) 582–604.
- [6] J.-M. Chiou, Y.-T. Chen, Y.-F. Yang, Multivariate Functional Principal Component Analysis: A Normalization Approach, *Statistica Sinica* 24 (2014) 1571–1596.
- [7] O. De la Cruz, S. Holmes, The Duality Diagram in Data Analysis: Examples of Modern Applications, *The annals of applied statistics* 5 (2011) 2266–2277.
- [8] J.-C. Deville, Méthodes statistiques et numériques de l’analyse harmonique, *Annales de l’inséé* (1974) 3–101.
- [9] P. H. C. Eilers, B. D. Marx, Flexible smoothing with B-splines and penalties, *Statistical Science* 11 (1996) 89–121.
- [10] P. H. C. Eilers, B. D. Marx, M. Durbán, Twenty years of P-splines, *SORT-Statistics and Operations Research Transactions* 39 (2015) 149–186.
- [11] B. Escoffier, Traitement simultané de variables qualitatives et quantitatives en analyse factorielle, *Cahiers de l’analyse des données* 4 (1979) 137–146.
- [12] J. Fan, I. Gijbels, Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66, CRC Press, 1996.
- [13] S. Golovkine, E. Gunning, A. J. Simpkin, N. Bargary, On the estimation of the number of components in multivariate functional principal component analysis, 2023.
- [14] J. González, A. Muñoz, Representing functional data in reproducing Kernel Hilbert Spaces with applications to clustering and classification, DES - Working Papers. Statistics and Econometrics. WS (2010).
- [15] M. Grith, H. Wagner, W. K. Härdle, A. Kneip, Functional Principal Component Analysis for Derivatives of Multivariate Curves, *Statistica Sinica* 28 (2018) 2469–2496.
- [16] P. Hall, J. W. Kay, D. M. Titterington, Asymptotically Optimal Difference-Based Estimation of Variance in Nonparametric Regression, *Biometrika* 77 (1990) 521–528.
- [17] P. Hall, J. S. Marron, On Variance Estimation in Nonparametric Regression, *Biometrika* 77 (1990) 415–419.
- [18] C. Happ, S. Greven, Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains, *Journal of the American Statistical Association* 113 (2018) 649–659.
- [19] W. K. Härdle, L. Simar, *Applied Multivariate Statistical Analysis*, Springer Nature, 2019.
- [20] S. Holmes, Multivariate data analysis: The French way, *Probability and Statistics: Essays in Honor of David A. Freedman* 2 (2008) 219–234.
- [21] L. Horváth, P. Kokoszka, *Inference for Functional Data with Applications*, Springer Series in Statistics, 2012.
- [22] J. Jacques, C. Preda, Model-based clustering for multivariate functional data, *Computational Statistics & Data Analysis* 71 (2014) 92–106.
- [23] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung., Ph.D. thesis, (Sana), Helsinki, 1947.
- [24] J. Kleffe, Principal components of random variables with values in a separable hilbert space, *Mathematische Operationsforschung und Statistik* 4 (1973) 391–406.
- [25] A. Kneip, K. J. Utikal, Inference for Density Families Using Functional Principal Component Analysis, *Journal of the American Statistical Association* 96 (2001) 519–532.
- [26] P. Kokoszka, H. Oja, B. Park, L. Sangalli, Special issue on functional data analysis, *Econometrics and Statistics* 1 (2017) 99–100.
- [27] D. D. Kosambi, Statistics in function space, *Journal of the Indian Mathematical Society* 7 (1943) 76–88.
- [28] M. Krzyśko, P. Nijkamp, W. Ratajczak, W. Wołyński, Multidimensional economic indicators and multivariate functional principal component analysis (MFPCA) in a comparative study of countries’ competitiveness, *Journal of Geographical Systems* 24 (2022) 49–65.

- [29] M. Loève, Sur les fonctions aléatoires stationnaires de second ordre, *La Revue Scientifique*, 5. Série 83 (1945) 297–303.
- [30] A. Martino, A. Ghiglietti, F. Ieva, A. M. Paganoni, A k-means procedure based on a Mahalanobis type distance for clustering multivariate functional data, *Statistical Methods & Applications* 28 (2019) 301–322.
- [31] J. Pagès, *Multiple Factor Analysis by Example Using R*, CRC Press, 2014.
- [32] J. Prothero, J. Hannig, J. S. Marron, New Perspectives on Centering, *The New England Journal of Statistics in Data Science* (2023) 1–21.
- [33] J. Ramsay, B. W. Silverman, *Functional Data Analysis*, Springer Science & Business Media, 2005.
- [34] J. O. Ramsay, When the data are functions, *Psychometrika* 47 (1982) 379–396.
- [35] M. Reed, B. Simon, *Methods of Modern Mathematical Physics: Functional Analysis*, Academic Press, 1980.
- [36] G. Saporta, Simultaneous Analysis of Qualitative and Quantitative Data, *Atti 35° Riunione Scientifica della Società Italiana di Statistica* (1990) pp. 63–72.
- [37] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, CRC Press, 1986.
- [38] J. Song, K. Kim, Sparse multivariate functional principal component analysis, *Stat* 11 (2022) e435.
- [39] J.-L. Wang, J.-M. Chiou, H.-G. Müller, Functional Data Analysis, *Annual Review of Statistics and Its Application* 3 (2016) 257–295.
- [40] J. Warmenhoven, S. Cobley, C. Draper, A. Harrison, N. Bargary, R. Smith, Bivariate functional principal components analysis: Considerations for use with multivariate movement signatures in sports biomechanics, *Sports Biomechanics* 18 (2019) 10–27.
- [41] R. K. W. Wong, X. Zhang, Nonparametric operator-regularized covariance function estimation for functional data, *Computational Statistics & Data Analysis* 131 (2019) 131–144.
- [42] F. Yao, H.-G. Müller, J.-L. Wang, Functional Data Analysis for Sparse Longitudinal Data, *Journal of the American Statistical Association* 100 (2005) 577–590.
- [43] X. Zhang, J.-L. Wang, From sparse to dense functional data and beyond, *The Annals of Statistics* 44 (2016) 2281–2321.

**Supplementary material:** We provide insights for when the data are already decomposed in a basis, e.g. Fourier or polynomials. In particular, we explain how to perform MFPCA as described in Section 4 in the main text.

## Stylistic guidelines

- The length of an JMVA article is 15-25 pages. There exists a special track for young researchers where somewhat shorter contributions can be accepted (see below).
- Number only those equations that are referred to in the text.
- Enumerations should list the first and last element only, i.e., write  $\{1, \dots, n\}$  (NOT  $\{1, 2, \dots, n\}$ ). Also do not use  $\dots$  in enumerations. Write  $i \in \{1, \dots, n\}$  instead of  $i = 1, \dots, n$ .
- Respect the following priority rules for fences  $\{\{\}\}$  unless the fences have a special meaning, i.e., write  $EX(t)$  (some authors use  $E[X(t)]$  which is acceptable). If you refer to the set  $1, \dots, n$ , don't write  $(1, \dots, n)$ , because in the latter two contexts,  $\{\}$  and  $\()$  have conventional meanings.
- Use  ${}^T$  for transposition (NOT  $'$ ,  ${}^T$ ,  $\tau$  or  $\dagger$ ) and  $\ln$  for  $\log$ , unless you mean  $\log_{10}$ .
- Symbols like sup, inf, max, min, E, Var, Cov, Corr, diag, trace (or tr), etc. should be in Roman characters (NOT in italics).
- Use  $\Pr$  for probability, so that it prints "Pr" in Roman characters.
- Avoid the symbol "l", always use  $\ell$  if you need an "ell".
- "i.e." and "e.g." should be preceded and followed by a comma: see, e.g.,
- Vectors should usually be interpreted as columns vectors (if it is not a coordinate free presentation).
- Bold upper and lower cases can be used for matrices and vectors, respectively.
- "cf." means "compare to", NOT "refer to". If you mean "refer to", write "see".
- Avoid in-line fractions; if you need one, write them in the form  $a/b$  rather than  $\frac{a}{b}$ .
- Concerning the references and the bibliography: Papers should be referred to by [] or by AUTHOR [number] where appropriate; in lists of references, order papers in increasing order, i.e., write [2, 5, 8], not [2, 8, 5], even if 2 did not appear first chronologically (use "cite", e.g., [? ] or [? ? ]); "et al." should be in Roman characters, NOT in italic; the bibliography should list the papers in alphabetical order and should be numbered.
- When referring to the software R, always use R, even in references. Packages should be typeset as ...