HHS Public Access

Author manuscript

J Comput Graph Stat. Author manuscript; available in PMC 2024 January 01.

Published in final edited form as:

J Comput Graph Stat. 2023 ; 32(2): 366–377. doi:10.1080/10618600.2022.2115500.

Fast Multilevel Functional Principal Component Analysis

Erjia Cui[*,1],

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, 615 N. Wolfe Street, Baltimore, MD 21205

Ruonan Li[*],

Department of Statistics, North Carolina State University, 2311 Stinson Dr, Raleigh, NC 27607

Ciprian M. Crainiceanu,

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, 615 N. Wolfe Street, Baltimore, MD 21205

Luo Xiao

Department of Statistics, North Carolina State University, 2311 Stinson Dr, Raleigh, NC 27607

Abstract

We introduce fast multilevel functional principal component analysis (fast MFPCA), which scales up to high dimensional functional data measured at multiple visits. The new approach is orders of magnitude faster than and achieves comparable estimation accuracy with the original MFPCA (Di et al., 2009). Methods are motivated by the National Health and Nutritional Examination Survey (NHANES), which contains minute-level physical activity information of more than 10000 participants over multiple days and 1440 observations per day. While MFPCA takes more than five days to analyze these data, fast MFPCA takes less than five minutes. A theoretical study of the proposed method is also provided. The associated function `mfpca.face()` is available in the `R` package `refund`.

Keywords

multilevel models; functional principal component analysis; mixed model equations

1 Introduction

Functional data measured at multiple visits have become increasingly common. A standard technique for analyzing such data is multilevel functional principal component analysis

[1] ecui1@jhmi.edu.

[*] indicates co-first authors

Conflict of Interest: Dr. Crainiceanu is consulting with Bayer, Johnson and Johnson, and Cytel on methods development for wearable devices in clinical trials. The details of the contracts are disclosed through the Johns Hopkins University eDisclose system and have no direct or apparent relationship with the current paper.

SUPPLEMENTARY MATERIALS

fastMFPCA supp.pdf Technical details of the method, proof of the theorem, and additional simulation and application results. (.pdf file)

code.zip The `R` function `mfpca.face()` and the code for simulation. (.zip file)

(MFPCA) (Di et al., 2009), which provides a decomposition of the observed data into within- and between-subject variation. The MFPCA model generalizes traditional measurement error and multilevel models to the case when the basic measurement unit is a function.

The motivating data is a large physical activity dataset from the National Health and Nutrition Examination Survey (NHANES), a study conducted in two-year waves by the United States Centers for Disease Control and Prevention (CDC). Each study participant in the 2003–2004 and 2005–2006 waves was asked to wear a hip-worn physical activity monitor (PAM) for seven consecutive days. Acceleration data were publicly released as minute-level activity counts (AC), a proprietary measure of physical activity intensity. For quality control purposes, some days were excluded from the analysis; see Section 6 for exclusion criteria. Figure 1 displays the physical activity profiles of three randomly selected NHANES study participants (left, middle and right panels). The number of available days varies by study participant with a maximum of 7. For example, the data for the study participant shown in the left panels contains only 6 days, while the data for the study participant shown in the middle panels contains only 5 days. Within a column, each row shows the minute-level AC of one day from midnight to midnight, where the title for each panel indicates the corresponding day of the week. The dataset has 12802 study participants and 65777 days in total, with 1440 observations per day for a total of 94718880 minute-level observations.

The NHANES dataset is an example of large-scale multilevel high-dimensional functional data. For subject i on day j of the week, the physical activity intensity value at minute $s \in \mathcal{S}$ can be denoted as $Y_{ij}(s)$, where $\mathcal{S}$ is the time interval from midnight to midnight. Functional principal component analysis (FPCA) is a popular approach in functional data analysis (Ramsay and Silverman, 2005). Some early work, including Ramsay and Dalzell (1991); Silverman et al. (1996); Yao et al. (2005), focused on single-level analysis. For multilevel functional data, multilevel functional principal component analysis (MFPCA) (Di et al., 2009) provides an explicit decomposition of the within- and between-subject variation in the functional space. MFPCA is designed to analyze functional data with two levels of functional variation. This is the simplest model in the rapidly expanding family of multilevel functional mixed effects models, including multilevel functional models (Aston et al., 2010; Brumback and Rice, 1998; Chen and Müller, 2012; Gaynanova et al., 2022; Goldsmith et al., 2015; Li et al., 2015; Morris et al., 2011; Morris and Carroll, 2006; Serban and Jiang, 2012; Xu et al., 2018), longitudinal functional models (Boland et al., 2022; Cui et al., 2021b; Greven et al., 2010; Li et al., 2022; Park and Staicu, 2015; Scheffler et al., 2020; Shamshoian et al., 2022; Zipunnikov et al., 2014), spatial functional models (Li et al., 2021; Zhang et al., 2016), and structured functional models (Scheipl et al., 2015; Shou et al., 2015). MFPCA is also related to but distinct from multivariate functional data (Berrendero et al., 2011; Chiou et al., 2014; Happ and Greven, 2018; Kowal et al., 2017; Wong et al., 2019).

Applying MFPCA to a dataset with over 10000 study participants and over 1000 observations per function remains computationally challenging. The current implementation of MFPCA is slow for high dimensional functional data, as the number of computations

is proportional to the *cube* of the number of observations per function. The problem is that the current MFPCA requires: (1) the construction, smoothing and eigendecomposition of covariance matrices with the dimension equal to the number of observations per function; and (2) the score prediction which relies on the inversion of multiple high dimensional covariance matrices. The fast covariance estimation in Xiao et al. (2016), referred to as FACE, addressed these problems for single-level functional data. Indeed, FACE, implemented in the `fpca.face()` function of the `refund R` package (Goldsmith et al., 2020), requires only minutes to smooth covariance matrices of dimension 100000. Here we provide methods that substantially accelerate MFPCA by extending methods inspired by FACE.

Therefore, we propose fast multilevel functional principal component analysis (fast MFPCA) and implement it in the `mfpca.face()` function of the `refund R` package. The fast MFPCA approach improves MFPCA by: (1) constructing transformed functional data instead of calculating the method of moments estimators of covariance matrices; (2) obtaining level-specific eigendecompositions by extending FACE to multilevel functional data, which avoids using high dimensional covariance matrices; and (3) predicting principal component scores based on mixed model equations (MME). Using the combination of these ideas, the fast MFPCA scales up *linearly* with the number of observations per function and is orders of magnitude faster than MFPCA. For example, fast MFPCA took less than 5 minutes to fit the NHANES data compared with MFPCA, which took more than 5 days.

The rest of the paper is organized as follows. We review MFPCA in Section 2 and introduce the fast MFPCA approach in Section 3. A theoretical study of the proposed method is provided in Section 4. A simulation study is conducted in Section 5 to compare the computation time and accuracy of our new approach with existing methods. We discuss the NHANES application results in Section 6 and conclude with a discussion in Section 7.

2 Multilevel Functional Principal Component Analysis

We briefly review the multilevel functional data model proposed in Di et al. (2009). Denote by $Y_{ij}(s)$ the observed data for subject $i = 1, \ldots, I$ at visit $j = 1, \ldots, J_i$ and location $s \in \{s_1, \ldots, s_L\} \in \mathcal{S}$, where $\mathcal{S}$ is a compact domain. Each function has L observations at the same set of time points and we focus on the case when L is relatively large. Denote by $n = \sum_{i=1}^{I} J_i$ the total number of visits. Consider a functional ANOVA model with measurement error: $Y_{ij}(s) = X_{ij}(s) + \epsilon_{ij}(s) = \mu(s) + \eta_j(s) + Z_i(s) + W_{ij}(s) + \epsilon_{ij}(s)$, where $\mu(s)$ is the population mean function, $\eta_j(s)$ is the jth visit-specific shift from $\mu(s)$, $Z_i(s)$ is the random subject-specific mean deviation for the ith subject, $W_{ij}(s)$ is the random jth visit-specific deviation from $Z_i(s)$, and $\epsilon_{ij}(s)$ is a white noise with variance σ^2. The random functions $Z_i(s)$ and $W_{ij}(s)$ are mutually independent zero mean processes with covariance functions $K_B(s, t) = \text{cov}\{Z_i(s), Z_i(t)\}$ and $K_W(s, t) = \text{cov}\{W_{ij}(s), W_{ij}(t)\}$, respectively.

Algorithm 1

MFPCA

1. Estimate mean functions $\mu(s)$ and $\eta_j(s)$ by applying univariate smoothers to observed data under working independence assumption and subtract them from observed data.

2. Construct method of moment (MoM) estimators of the total covariance $K_T(s, t)$ and between-subject covariance $K_B(s, t)$, denoted by $\widehat{K}_T(s, t)$ and $\widehat{K}_B(s, t)$, respectively.

3. Smooth $\widehat{K}_T(s, t)$ and $\widehat{K}_B(s, t)$ using bivariate smoothing, leading to two smooth estimates, denoted by $\widetilde{K}_T(s, t)$ and $\widetilde{K}_B(s, t)$, respectively. Let $\widetilde{K}_W(s, t) = \widetilde{K}_T(s, t) - \widetilde{K}_B(s, t)$.

4. Conduct eigenanalysis on discretized $\widetilde{K}_B(s, t)$ and $\widetilde{K}_W(s, t)$ matrices.

5. Estimate error variance σ^2 by $\hat{\sigma}^2 = \int_{\mathcal{S}} \{\widehat{K}_T(s, s) - \widetilde{K}_T(s, s)\} ds$.

6. Predict scores using best linear unbiased prediction (BLUP).

Let $K_T(s, t) := \text{cov}\{X_{ij}(s), X_{ij}(t)\} = K_B(s, t) + K_W(s, t)$ be the total variance of the smooth functional data. Suppose that the between-subject covariance function K_B admits the eigendecomposition $K_B(s, t) = \sum_{k \geq 1} \lambda_k^{(1)} \phi_k(s) \phi_k(t)$, where $\lambda_1^{(1)} \geq \lambda_2^{(1)} \geq \cdots \geq 0$ are eigenvalues with associated orthonormal eigenfunctions $\phi_k(s)$, that is, $\int_{\mathcal{S}} \phi_{k_1}(s) \phi_{k_2}(s) ds = \delta_{\{k_1 = k_2\}}$ for any pair (k_1, k_2). Here $\delta_{\{\cdot\}}$ is an indicator function which is equal to 1 if the statement is true and 0 otherwise. Then the random function $Z_i(s)$ can be written as $Z_i(s) = \sum_{k \geq 1} \xi_{ik} \phi_k(s)$, where ξ_{ik} are scores with zero mean and variance $\lambda_k^{(1)}$ and are mutually uncorrelated. Similarly, suppose that the within-subject covariance function K_W also has an eigendecomposition $K_W(s, t) = \sum_{k \geq 1} \lambda_k^{(2)} \psi_k(s) \psi_k(t)$, where $\lambda_1^{(2)} \geq \lambda_2^{(2)} \geq \cdots \geq 0$ are the eigenvalues with associated orthonormal eigenfunctions $\psi_k(s)$. Then $W_{ij}(s) = \sum_{k \geq 1} \zeta_{ijk} \psi_k(s)$, where ζ_{ijk} are mutually uncorrelated scores with zero mean and variance $\lambda_k^{(2)}$.

The primary goal of MFPCA is to reduce the functional data to two sets of uncorrelated scores: level-1 scores ξ_{ik_1} and level-2 scores ζ_{ijk_2}. The eigenfunctions are also useful for understanding the variation patterns in functional data. The traditional MFPCA methods in Di et al. (2009) are summarized in Algorithm 1.

3 Fast MFPCA

MFPCA slows down substantially when the number of observations per function increases. Indeed, constructing the $L \times L$ sample covariance matrices (step 2 of Algorithm 1) requires $O(IL^2)$ computations, where I is the total number of functions. Next, standard bivariate smoothing of $L \times L$ covariance matrices (step 3 of Algorithm 1) requires $O(L^3)$ computations. Furthermore, the eigenanalysis of $L \times L$ matrices (step 4 of Algorithm 1) also requires $O(L^3)$ computations. Finally, the score prediction (step 6 of Algorithm 1) requires inversion of covariance matrices of size $L \times L$, which require $O(L^3)$ computations.

To deal with the computational challenges of traditional MFPCA, we propose the fast MFPCA approach, which differs from traditional MFPCA in three aspects. First, we construct transformed functional data for which the underlying smooth curves have the desired covariance operators. This construction takes only $O(IL)$ computations compared

to $O(IL^2)$ computations for the method of moment (MoM) sample covariance estimators. Second, we apply the fast covariance estimation method (FACE, Xiao et al. (2016)) to the transformed data to estimate the covariance operators. FACE avoids the direct calculation of MoM and eigenanalysis of empirical of $L \times L$ covariance matrices. The computational complexity of FACE is $O(ILc)$, where c is the number of B-spline bases functions used for smoothing and is much smaller than I and L. Finally, we predict the principal component scores using mixed model equations (MME), which is computationally efficient because the number of eigenfunctions is much smaller than the number of observations per curve.

To the best of our knowledge, this is the first time FACE is extended and applied to general multilevel functional data. Moreover, we are not aware of any literature using MME for score prediction in functional data analysis. This idea combination reduces the computational complexity of MFPCA from from $O(IL^2 + L^3)$ to $O(ILc)$.

3.1 FACE and Eigenanalysis

The fast covariance estimation method (FACE) in Xiao et al. (2016) is a bivariate smoothing method based on the tensor-product splines. FACE is computationally fast as it scales up linearly with the number of functions and the number of observations per function. In addition, the eigenanalysis via FACE avoids computationally expensive eigendecompositions of large covariance matrices. Below, we provide the technical details of FACE, which inspired the functional data transformation described in Section 3.2.

Let $\mathbf{Y}$ be an $L \times I$ data matrix with each column corresponding to one observed single-level function (centered and scaled) evaluated at the time points $\{s_1, \ldots, s_L\}$. Let $\widehat{\mathbf{K}} = I^{-1}\mathbf{Y}\mathbf{Y}^T$ be the sample covariance matrix estimator. Denote by $\mathbf{B}(s) = [B_1(s), \ldots, B_c(s)]^T$ the $c \times 1$ dimensional vector of c cubic B-spline basis evaluated at s. Let $\mathbf{B} = [\mathbf{B}(s_1), \ldots, \mathbf{B}(s_L)]^T$ be the $L \times c$ design matrix, where each row corresponds to a sampling point and each column corresponds to a spline basis. An $L \times L$ smoother matrix is constructed as $\mathbf{S} = \mathbf{B}(\mathbf{B}^T\mathbf{B}/L + \lambda\mathbf{P})^{-1}\mathbf{B}^T/L$, where $\mathbf{P}$ is the qth order penalty matrix in P-splines (Eilers and Marx, 1996) and λ is the smoothing parameter; see Section S.1 in the supplementary material for more details. FACE uses $\widetilde{\mathbf{K}} = \mathbf{S}\widehat{\mathbf{K}}\mathbf{S}$ as the smooth estimator of the covariance. Thus, the (s, t) entry of the covariance estimator is $\widetilde{K}(s, t) = \mathbf{B}^T(s)\mathbf{\Theta}B(t)$, where $\mathbf{\Theta} = (\mathbf{B}^T\mathbf{B}/L + \lambda\mathbf{P})^{-1}(\mathbf{B}^T\widehat{\mathbf{K}}\mathbf{B}/L^2)(\mathbf{B}^T\mathbf{B}/L + \lambda\mathbf{P})^{-1}$ is a $c \times c$ symmetric and positive semi-definite matrix. FACE does not directly calculate $\widetilde{\mathbf{K}}$ and only computes the $c \times c$ coefficient matrix $\mathbf{\Theta}$, which can be written as $\mathbf{F}\mathbf{F}^T$ with $\mathbf{F} = (L\sqrt{I})^{-1}(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{P})^{-1}\mathbf{B}^T\mathbf{Y}$. Notice that $\mathbf{F}$ is of dimension $c \times I$ and its calculation requires $O(ILc)$ computations. Because c is the number of spline functions, which is much smaller than L and I, the calculation of $\mathbf{F}$ is much faster than direct bivariate smoothing of a covariance operator. The main idea is to use the decomposition of the sample covariance $\widehat{\mathbf{K}}$ combined with the property that the smoothed covariance $\widetilde{\mathbf{K}}$ matrix is low rank. FACE selects the smoothing parameter λ by minimizing the pooled generalized cross validation (PGCV) (Xiao et al., 2013), which is fast because it relies on univariate functional smoothing to control bivariate covariance smoothing.

The tensor product spline form of $\widetilde{K}(s, t)$ can be used to reduce the computational complexity of eigenfunctions and eigenvalues estimation procedure. Indeed, let $\mathbf{G} = \int \mathbf{B}(s)\mathbf{B}^T(s)ds$, which can be constructed to be positive definite. Take the eigendecomposition $\mathbf{G}^{1/2}\mathbf{\Theta}\mathbf{G}^{1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_C]$ is an orthonormal matrix and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_C)$ is a diagonal matrix with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_C \geq 0$. Then, $\mathbf{U}_k^T\mathbf{G}^{-1/2}\mathbf{B}(s)$ is the estimated kth eigenfunction corresponding to eigenvalue λ_k. In contrast, the direct approach is to conduct a spectral decomposition of the covariance matrix evaluated on a dense grid; see, for example, Yao et al. (2005). This method is computationally expensive, especially in high dimensions.

3.2 Transformed Functional Data

To simplify the notation introduced in Section 2, let $\widetilde{Y}_{ij}(s) = Y_{ij}(s) - \mu(s) - \eta_j(s)$ be the demeaned observed data, where $\mu(s)$ and $\eta_j(s)$ will be replaced by their estimates in applications. Recall that the data are observed on a regular grid $\{s_1, ..., s_L\}$. Define $\widetilde{\mathbf{Y}}_{ij} = \left[\widetilde{Y}_{ij}(s_1), \ldots, \widetilde{Y}_{ij}(s_L)\right]^T$ and $\widetilde{\mathbf{Y}} = \left[\widetilde{\mathbf{Y}}_{11}, \ldots, \widetilde{\mathbf{Y}}_{1J_1}, \ldots, \widetilde{\mathbf{Y}}_{I1}, \ldots, \widetilde{\mathbf{Y}}_{IJ_I}\right] \in \mathbb{R}^{L \times n}$. Denote by $n = \sum_{i=1}^{I} J_i$, the total number of curves, and define $n_I = \sum_{i=1}^{I} J_i(J_i - 1)$.

As suggested in Shou et al. (2015), MoM estimators of covariance matrices in structured functional data often take the "sandwich" form $n^{-1}\widetilde{\mathbf{Y}}\mathbf{H}\widetilde{\mathbf{Y}}^T$, where $\mathbf{H}$ are design-specific matrices. For example, for the MoM estimator of total covariance, $\mathbf{H}$ is the $n \times n$ identity matrix. Therefore, FACE can be applied to the transformed data $\widetilde{\mathbf{Y}}\mathbf{H}^{1/2}$ to smooth the covariance whenever $\mathbf{H}$ is positive semi-definite. However, there are two potential issues with this approach. First, the sample covariance matrix for the between-subject covariance $\widehat{\mathbf{K}}_B$ and the corresponding $\mathbf{H}$ matrix are not positive semi-definite. Therefore, FACE is not directly applicable. One solution was proposed by Xiao et al. (2016), who truncated negative eigenvalues to zero. The second issue the computation of $\mathbf{H}^{1/2}$, which is of dimension $n \times n$. When the total number of curves, n, is large, calculating $\mathbf{H}^{1/2}$ can become challenging.

To address the first problem, we smooth the total and within-subject covariance, which are positive semi-definite. The between-subject covariance is then estimated by the difference between total and within-subject covariance. To address the second problem, we provide an analytic form for the transformed functional data. We provide the technical details below.

As in Di et al. (2009), an empirical estimator of the between-subject covariance is given by the $L \times L$ matrix $\widehat{\mathbf{K}}_B = n_I^{-1}\widetilde{\mathbf{Y}}\mathbf{H}_B\widetilde{\mathbf{Y}}^T$, where $\mathbf{H}_B = \text{blockdiag}(\mathbf{1}_{J_1}\mathbf{1}_{J_1}^T - \mathbf{I}_{J_1}, \ldots, \mathbf{1}_{J_I}\mathbf{1}_{J_I}^T - \mathbf{I}_{J_I})$, $\mathbf{I}_J$ is the identity matrix of size J and $\mathbf{1}_J = (1, 1, \ldots, 1)_J^T$. As each block matrix $\mathbf{1}_{J_i}\mathbf{1}_{J_i}^T - \mathbf{I}_{J_i}$ has only two different eigenvalues, $J_i - 1$ (with geometric multiplicity 1) and -1 (with geometric multiplicity $J_i - 1$), $\mathbf{H}_B$ is not a positive semi-definite matrix. Because most eigenvalues are equal to -1, this may be the reason why the estimation performance was found to be sub-optimal when trimming negative eigenvalues of $\mathbf{H}_B$ (Xiao et al., 2016). We next focus on the total and within-subject covariance.

For the total covariance, let $\bar{J} = n^{-1}\sum_i J_i$ be the average number of visits per subject. The MoM estimator of the total covariance, $K_T(s, t)$, is $\widehat{K}_T(s, t) = \sum_{i=1}^{n}\sum_{j=1}^{J_i} w_i \widetilde{Y}_{ij}(s)\widetilde{Y}_{ij}(t)$, where $w_i > 0$ are weights that satisfy the constraint $\sum_i J_i w_i = 1$. In Di et al. (2009), the same weight is used for each visit, which means $w_i = 1/(n\bar{J})$, though other weights could be used. For example, by setting $w_i = 1/(IJ_i)$, equal weights are assigned to study participants instead of visits. The matrix format of $\widehat{K}_T(s, t)$ is $\widehat{\mathbf{K}}_T = \sum_{i=1}^{I}\sum_{j=1}^{J_i} w_i \widetilde{\mathbf{Y}}_{ij}\widetilde{\mathbf{Y}}_{ij}^T$. The key insight is that $\widehat{\mathbf{K}}_T$ is the sample covariance of the transformed functional data $\left\{\sqrt{nw_i}\widetilde{\mathbf{Y}}_{ij},\ 1 \le j \le J_i,\ 1 \le i \le I\right\}$. Therefore, smoothing $\widehat{\mathbf{K}}_T$ can be achieved by applying FACE to the transformed functional data.

For the within-subject covariance, notice that $K_W(s,t) = E\left\{\widetilde{Y}_{ij}(s) - \widetilde{Y}_{ik}(s)\right\}\left\{\widetilde{Y}_{ij}(t) - \widetilde{Y}_{ik}(t)\right\}^T/2$ if $j \neq k$. Let $v_i \ge 0$ be weights such that $\sum_{i=1}^{I} J_i(J_i - 1)v_i = 1$. An estimator of $K_W(s, t)$ is $\widehat{\mathbf{K}}_W = \sum_{i=1}^{I} v_i/2\sum_{j\neq k}\left(\widetilde{\mathbf{Y}}_{ij} - \widetilde{\mathbf{Y}}_{ik}\right)\left(\widetilde{\mathbf{Y}}_{ij} - \widetilde{\mathbf{Y}}_{ik}\right)^T$. The constraint on the weights ensures that when functional data are observed without random noise, $\sigma^2 = 0$, $\widehat{\mathbf{K}}_W$ is an unbiased estimator of K_W. If the same weight is used for each visit, then $v_i = n_I^{-1}$. If the same weight is used for each participant, then $v_i = \{(\sum_{l=1}^{I}\delta_{\{J_l \ge 2\}})J_i(J_i - 1)\}^{-1}$ if $J_i \ge 2$ and 0 otherwise. Let $\overline{\mathbf{Y}}_{i\cdot} = J_i^{-1}(\sum_{j=1}^{J_i}\widetilde{\mathbf{Y}}_{ij})$. It can be shown that $\widehat{\mathbf{K}}_W = \sum_{i=1}^{I}\sum_{j=1}^{J_i} v_i J_i\left(\widetilde{\mathbf{Y}}_{ij} - \overline{\mathbf{Y}}_{i\cdot}\right)\left(\widetilde{\mathbf{Y}}_{ij} - \overline{\mathbf{Y}}_{i\cdot}\right)^T$, which is the sample covariance of the transformed data $\left\{\sqrt{nv_i J_i}\left(\widetilde{\mathbf{Y}}_{ij} - \overline{\mathbf{Y}}_{i\cdot}\right),\ 1 \le j \le J_i,\ 1 \le i \le I\right\}$. Therefore, smoothing of $\widehat{\mathbf{K}}_W$ can be achieved by applying FACE to the above transformed data.

The construction of transformed functional data for the total and within-subject covariance require $O(nL)$ operations. This is a critical difference from the traditional MoM estimators, which require $O(nL^2)$ operations.

3.3 Multilevel FACE

We apply FACE to the transformed functional data for the total covariance and within-subject covariance, respectively, and obtain the smooth estimates. The corresponding eigenfunctions are obtained as described in the FACE approach for univarate functional data. Let $\widetilde{K}_T(s, t) = \mathbf{B}^T(s)\mathbf{\Theta}_T\mathbf{B}(t)$ be the estimate of $K_T(s, t)$ and $\widetilde{K}_W(s, t) = \mathbf{B}^T(s)\Theta_W\mathbf{B}(t)$ be the estimate of $K_W(s, t)$. Here $\mathbf{\Theta}_T$ and $\mathbf{\Theta}_W$ are both $c \times c$ positive semi-definite matrices obtained from FACE. The between-subject covariance $K_B(s, t)$ is estimated by $\widetilde{K}_B(s, t) = \mathbf{B}(s)^T\mathbf{\Theta}_B\mathbf{B}(t)$, where $\mathbf{\Theta}_B = \mathbf{\Theta}_T - \mathbf{\Theta}_W$. To ensure that $\widetilde{K}_B(s,t)$ is positive semi-definite, an eigendecomposition of $\mathbf{\Theta}_B$ is taken and the eigenvectors associated with negative eigenvalues are discarded. For details, see Section S.2 of the supplementary material.

3.4 Score Prediction via Mixed Model Equations

Predicting the principal component scores via best linear unbiased prediction (BLUP) requires the inversion of matrices that are of dimension equal to the number of observations per curve, L. Here we propose a novel solution based on mixed model equations (MME), which further reduces the computational complexity.

Assuming the level-1 eigenfunctions $\phi_k(s)$ and level-2 eigenfunctions $\psi_k(s)$ are known, the multilevel functional model becomes the mixed effects model

$$\widetilde{Y}_{ij}(s) = \sum_{k_1 \geq 1} \xi_{ik_1}\phi_{k_1}(s) + \sum_{k_2 \geq 1} \zeta_{ijk_2}\psi_{k_2}(s) + \epsilon_{ij}(s), \tag{1}$$

where ξ_{ik_1}, ζ_{ijk_2} are uncorrelated scores that are uncorrelated with the $\epsilon_{ij}(s)$. After obtaining $\phi_k(s)$ and $\psi_k(s)$ by multilevel FACE in Section 3.3, Equation (1) can be approximated by

$$\widetilde{Y}_{ij}(s) = \sum_{k_1=1}^{N_1} \xi_{ik_1}\phi_{k_1}(s) + \sum_{k_2=1}^{N_2} \zeta_{ijk_2}\psi_{k_2}(s) + \epsilon_{ij}(s),$$

where we have retained N_1 level-1 scores and N_2 level-2 scores.

Define $M_i = J_iL$. Let $\widetilde{\mathbf{Y}}_{ij} = \left[\widetilde{Y}_{ij}(s_1), \ldots, \widetilde{Y}_{ij}(s_L)\right]^T$, $\boldsymbol{\xi}_i = [\xi_{i1}, \ldots, \xi_{iN_1}]^T$, $\boldsymbol{\zeta}_{ij} = [\zeta_{ij1}, \ldots, \zeta_{ijN_2}]^T$, $\boldsymbol{\phi}_{k_1} = \left[\phi_{k_1}(s_1), \ldots, \phi_{k_1}(s_L)\right]^T$, $\boldsymbol{\Psi}_{k_2} = \left[\psi_{k_2}(s_1), \ldots, \psi_{k_2}(s_L)\right]^T$, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N_1}] \in \mathbb{R}^{L \times N_1}$, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{N_2}] \in \mathbb{R}^{L \times N_2}$, and $\boldsymbol{\epsilon}_{ij} = [\epsilon_{ij1}, \ldots, \epsilon_{ijL}]^T$. Then $\widetilde{\mathbf{Y}}_{ij} = \boldsymbol{\Phi}\boldsymbol{\xi}_i + \boldsymbol{\Psi}\boldsymbol{\zeta}_{ij} + \epsilon_{ij}$. We further define $\boldsymbol{\Phi}_i = \mathbf{1}_{J_i} \otimes \boldsymbol{\Phi} \in \mathbb{R}^{M_i \times N_1}$, $\boldsymbol{\Psi}_i = \mathbf{I}_{J_i} \otimes \boldsymbol{\Psi} \in \mathbb{R}^{M_i \times (J_iN_2)}$, $\boldsymbol{\zeta}_i = [\boldsymbol{\zeta}_{i1}^T, \ldots, \boldsymbol{\zeta}_{iJ_i}^T]^T \in \mathbb{R}^{J_iN_2}$, $\widetilde{\mathbf{Y}}_i = [\widetilde{\mathbf{Y}}_{i1}^T, \ldots, \widetilde{\mathbf{Y}}_{iJ_i}^T]^T \in \mathbb{R}^{M_i}$, and $\epsilon_i = [\epsilon_{i1}^T, \ldots, \epsilon_{iJ_i}^T]^T \in \mathbb{R}^{M_i}$. The covariance matrix is $\boldsymbol{\Lambda}_1 = \mathrm{diag}(\lambda_1^{(1)}, \ldots, \lambda_{N_1}^{(1)})$ for $\boldsymbol{\xi}_i$, $\boldsymbol{\Lambda}_2 = \mathrm{diag}(\lambda_1^{(2)}, \ldots, \lambda_{N_2}^{(2)})$ for $\boldsymbol{\zeta}_{ij}$, and $\sigma^2\mathbf{I}_{M_i}$ for $\boldsymbol{\epsilon}_i$. We then have the matrix form of the mixed effects model

$$\begin{aligned} &\widetilde{\mathbf{Y}}_i = \boldsymbol{\Phi}_i\boldsymbol{\xi}_i + \boldsymbol{\Psi}_i\boldsymbol{\zeta}_i + \epsilon_i, \\ &\mathrm{E}\begin{pmatrix}\boldsymbol{\xi}_i \\ \boldsymbol{\zeta}_i \\ \epsilon_i\end{pmatrix} = \begin{pmatrix}\mathbf{0}_{N_1} \\ \mathbf{0}_{J_iN_2} \\ \mathbf{0}_{M_i}\end{pmatrix}, \ \mathrm{Cov}\begin{pmatrix}\boldsymbol{\xi}_i \\ \boldsymbol{\zeta}_i \\ \epsilon_i\end{pmatrix} = \begin{pmatrix}\boldsymbol{\Lambda}_1 & 0 & 0 \\ 0 & \mathbf{I}_{J_i} \otimes \boldsymbol{\Lambda}_2 & 0 \\ 0 & 0 & \sigma^2\mathbf{I}_{M_i}\end{pmatrix}. \end{aligned} \tag{2}$$

It follows that the BLUP of $\boldsymbol{\xi}_i$ and $\boldsymbol{\zeta}_i$ is

$$\begin{pmatrix}\hat{\boldsymbol{\xi}}_i \\ \hat{\boldsymbol{\zeta}}_i\end{pmatrix} = \begin{pmatrix}\boldsymbol{\Lambda}_1\boldsymbol{\Phi}_i^T \\ (\mathbf{I}_{J_i} \otimes \boldsymbol{\Lambda}_2)\boldsymbol{\Psi}_i^T\end{pmatrix}\left\{\mathrm{cov}(\widetilde{\mathbf{Y}}_i)\right\}^{-1}\widetilde{\mathbf{Y}}_i, \tag{3}$$

where $\mathrm{cov}\left(\widetilde{\mathbf{Y}}_i\right) = \boldsymbol{\Phi}_i\boldsymbol{\Lambda}_1\boldsymbol{\Phi}_i^T + \boldsymbol{\Psi}_i(\mathbf{I}_{J_i} \otimes \boldsymbol{\Lambda}_2)\boldsymbol{\Psi}_i^T + \sigma^2\mathbf{I}_{M_i} \in \mathbb{R}^{M_i \times M_i}$.

When M_i is large, implementing equation (3) is difficult as the inverse of $\mathrm{cov}\left(\widetilde{\mathbf{Y}}_i\right)$ requires $O(M_i^3)$ calculations. In practice, it takes more than a day to predict scores when $I = 1000$, $J_i = J = 3$, $L = 1000$ using existing methods. The mixed model equations (MME) (Henderson, 1973) can be used to re-express the scores as

$$\begin{pmatrix}\hat{\boldsymbol{\xi}}_i \\ \hat{\boldsymbol{\zeta}}_i\end{pmatrix} = \begin{pmatrix}\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i + \sigma^2\boldsymbol{\Lambda}_1^{-1} & \boldsymbol{\Phi}_i^T\boldsymbol{\Psi}_i \\ \boldsymbol{\Psi}_i^T\boldsymbol{\Phi}_i & \boldsymbol{\Psi}_i^T\boldsymbol{\Psi}_i + \sigma^2\mathbf{I}_{J_i} \otimes \boldsymbol{\Lambda}_2^{-1}\end{pmatrix}^{-1}\begin{pmatrix}\boldsymbol{\Phi}_i^T\widetilde{\mathbf{Y}}_i \\ \boldsymbol{\Psi}_i^T\widetilde{\mathbf{Y}}_i\end{pmatrix}, \tag{4}$$

where the dimension of the matrix that is inverted is $(N_1 + J_i N_2)$, which is usually much smaller than M_i. As a result, the total computational time using equation (4) is reduced to $O(M_I(N_1 + J_i N_2)^2 + (N_1 + J_i N_2)^2)$, which is linear in L since $M_i = J_i L$. In addition, the matrix inverse can be computed using block-wise calculations. The equivalence of BLUP and random effects solutions in MME was shown in Henderson (1963).

3.5 Fast MFPCA Algorithm

We summarize the steps of the fast MFPCA method in Algorithm 2. When compared with Algorithm 1, there are major differences in Steps 2–4 and 6. Steps 2–4 of fast MFPCA avoid computations that involve construction, smoothing and eigendecomposition of high dimensional covariance matrices. Step 6 of fast MFPCA uses a faster approach to the prediction of scores. The algorithm was implemented in the function `mfpca.face()` and released in the `R` package `refund`.

Algorithm 2

fast MFPCA

1. Estimate mean functions $\mu(s)$ and $\eta_j(s)$ and subtract them from the observed data.
2. Apply FACE to the transformed functional data $\left\{\sqrt{nw_i}\widetilde{\mathbf{Y}}_{ij},\ 1 \le j \le J_i,\ 1 \le i \le I\right\}$ for total covariance.
3. Apply FACE to the transformed functional data $\left\{\sqrt{nv_i J_i}\left(\widetilde{\mathbf{Y}}_{ij} - \overline{\mathbf{Y}}_{i\cdot}\right), 1 \le j \le J_i, 1 \le i \le I\right\}$ for within-subject covariance and obtain within-subject eigenfunctions/eigenvalues.
4. Calculate between-subject covariance from the difference between total and within-subject covariance and extract between-subject eigenfunctions/eigenvalues.
5. Estimate error variance σ^2 by $\hat{\sigma}^2 = \int_{\mathcal{S}}\{\widehat{K}_T(s,\ s) - \widetilde{K}_T(s,\ s)\}ds$.
6. Estimate scores by MME in equation (4).

3.6 Incomplete Data

As in the physical activity data, some data might be missing. Xiao et al. (2016) proposed an iterative approach for single-level functional data which consists of: (1) initializing the missing data by imputation from any smoother; (2) applying FACE to the data and impute missing data by their BLUP; (3) iterating step 2 until reaching convergence. They reported convergence usually within 10 iterations. As we rely on FACE, which was designed to deal with missing data, the problem is solved automatically in our approach. The MME for score prediction in (4) can be easily modified to work with observed data only and hence the details are omitted. We have found that this method works well for incomplete data; see simulation results in Section 5.

4 Asymptotic Theory

We establish the L_2 convergence rate of the proposed fast MFPCA method for estimating the between- and within-subject covariance functions and show that a parametric convergence rate can be achieved when functional data are densely observed. To simplify theoretical analysis, we assume that $\mu(s)$ and $\eta_j(s)$ are known and that the study participants have the same number of visits, $J_i = J$. In this case the weights are $w_i = (nJ)^{-1}$ for estimating the

total covariance function and $v_I = \{nJ(J-1)\}^{-1}$ for estimating the within-subject covariance function. Finally, we use the same smoothing parameter λ for both $\widetilde{K}_T$ and $\widetilde{K}_W$, the smooth estimates from fast MFPCA.

We introduce some notation. The little o and big O notation are with respect to the number of study participants I and we allow the number of observations per curve L to increase with I. For two scalars a and b, let $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. For a bivariate continuous function g over $\mathcal{S}^2$ let $\|g\|_{L_2}$ be its L_2 norm. For an integer $p \geq 2$, let $\mathscr{C}^p(\mathcal{S}^2)$ be the class of bivariate functions such that if $K \in \mathscr{C}^p(\mathcal{S}^2)$, then for any $0 \leq j \leq p$, $\partial^p K(s, t)/\partial s^j \partial t^{p-j}$ is continuous in $\mathcal{S}^2$.

Assumption 1

(a). The random functions Z_i are independent across the subjects with zero-mean function and the same covariance function $K_B(s, t)$; (b). The random functions W_{ij} are independent across i and j with zero-mean function and the same covariance function $K_W(s, t)$; (c). The random errors $\epsilon_{ij\ell} = \epsilon_{ij}(s_\ell)$ are independent across i, j and ℓ with zero-mean and the same variance $\sigma_\epsilon^2 < \infty$; (d). The random functions Z_i, $\{W_{i1}, \ldots, W_{iJ_i}\}$, and the random errors $\{\epsilon_{ij1}, \ldots, \epsilon_{ijL}\}$ are mutually independent across the subjects.

Assumption 2

$\sup_{s \in \mathcal{S}} \mathbb{E}\left[Z_i^4(s)\right] < \infty$; $\sup_{s \in \mathcal{S}} \mathbb{E}[W_{ij}^4(s)] < \infty$; $\mathbb{E}[\epsilon_{ij\ell}^4] < \infty$.

Assumption 3

a. *(a). $L \geq I^{\delta_1}$ for some constant $\delta_1 > 0$; (b). $c \geq I^{\delta_2}$ for some constant $\delta_2 > 0$ and $c = o(I)$; (c). There exists a sufficiently small constant δ_3 such that $c \leq \delta_3 L$; (d). $\lambda = o\left(I^{-2q\delta_1}\right)$.*

Recall that m is the order of spline functions, c is the number of spline functions, and q is the order of smoothness penalty. Let $h = c^{-1}$ and $h_e = h \vee \lambda^{1/(2q)}$, and the latter is the effective bandwidth for penalized splines (Xiao, 2019). The proof for the theorem below is given in Section S.3 of the supplementary material.

Theorem 1

Suppose that Assumptions 1 – 3 hold. If $K_B \in \mathscr{C}^p(\mathcal{S}^2)$ and $K_W \in \mathscr{C}^p(\mathcal{S}^2)$ with $q \leq p \wedge m$, then

$$\mathbb{E}(\|\widetilde{K}_B - K_B\|_{L_2}^2) = O(L^{-2}h_e^{-1}) + O(h^{2m}) + o(h^{2p}) + O(\lambda^2 h_e^{-2q}) + O(I^{-1}),$$

$$\mathbb{E}(\|\widetilde{K}_W - K_W\|_{L_2}^2) = O(h^{2m}) + o(h^{2p}) + O(\lambda^2 h_e^{-2q}) + O(I^{-1}).$$

Except for the term $O(L^{-2}h_e^{-1})$, the derived rate in Theorem 1 is the same as those in Theorem 4.1 in Xiao (2020), which considers covariance function estimation using penalized splines for functional data with a fixed common design. The convergence rate for

estimating the between-subject covariance contains the term $O\left(L^{-2}h_e^{-1}\right)$, which is due to the bias in the MoM estimator of the total covariance, which involves the extra variance, σ^2, along the main diagonal. The convergence rate for estimating the within-subject covariance does not have the term $O\left(L^{-2}h_e^{-1}\right)$ because the empirical estimate $\widehat{K}_W$ is an unbiased estimate of K_W.

In both of the derived rates in Theorem 1, the term $O(h^{2m}) + o(h^{2p})$ is the approximation bias of spline functions, the term $O(\lambda^2 h_e^{-2q})$ is the shrinkage bias due to the smoothness penalty, and the term $O(I^{-1})$ is the variability of the estimate. To achieve a parametric rate of $O(I^{-1})$ for $\widetilde{K}_W$, a simple choice is to let $q = p$ and $h = O(I^{-1/(2p)})$. The second condition on h means that the method could achieve a parametric rate as long as the number of knots (or spline functions) is sufficiently large. As for $\widetilde{K}_B$, the parametric rate can also be achieved if the condition $L^{-2}h_e^{-1} = O\left(I^{-1}\right)$ also holds. This means that a sufficiently dense sampling design for each function is required. The additional condition seems reasonable for high-dimensional functional data. When L is small and the condition $L^{-2}h_e^{-1} = O\left(I^{-1}\right)$ becomes stringent, one could smooth the empirical estimate $\widehat{K}_T$ without its diagonal terms or we replace the diagonal terms by other estimates with negligible bias.

While these theoretical results are for multilevel functional data, the proofs can be applied to other functional data and the convergence rate of the FACE method for single-level functional data has also been derived; see Section S.3 of the supplement for more details.

5 Simulation Studies

We perform simulations to: (1) assess the computational improvement and scaling behavior of fast MFPCA; and (2) evaluate the estimation accuracy of the fast MFPCA method. The `mfpca.face()` function is provided in the supplementary material and published in the `refund` package. For the implementation of traditional MFPCA we use the `mfpca.sc()` function in the `refund` package. For fast MFPCA, we use the same weight for each visit (curve), the same as traditional MFPCA. Additional simulation results for fast MFPCA with equal weight per subject are given in Section S.4 of the supplementary material.

5.1 Simulation Settings

We assume that the functions are observed on an equally-spaced grid $\{s_1, ..., s_L\}$ of domain $\mathcal{S} = [0, 1]$ such that $s_l = l/L$ for $l = 1, \ldots, L$. We also consider scenarios with incomplete (missing) data, where the number of observed points per function is $T_{ij} = 0.5 \times L$. Denote by J the mean number of visits per subject. For each subject i, the number of visits J_i is either balanced ($J_i = J$) or unbalanced (J_i varies by subject). We use a similar simulation setting as in Di et al. (2009). For visit j of subject i consider the following model

$$Y_{ij}(s_l) = \sum_{k_1=1}^{4} \xi_{ik_1}\phi_{k_1}(s_l) + \sum_{k_2=1}^{4} \zeta_{ijk_2}\psi_{k_2}(s_l) + \epsilon_{ij}(s_l),$$

where $\xi_{ik_1} \sim \mathcal{N}\{0,\ \lambda_{k_1}^{(1)}\}$, $\zeta_{ijk_2} \sim \mathcal{N}\{0,\ \lambda_{k_2}^{(2)}\}$, $\epsilon_{ij}(s) \sim \mathcal{N}\left(0,\ \sigma^2\right)$. We assume that there are $N_1 = 4$ components at the between-subject level and $N_2 = 4$ components at the within-subject level. Higher ranks on both levels with higher frequency eigenfunctions were further evaluated for both methods and the results are shown in Section S.4 of the supplementary material. The true eigenvalues are $\lambda_{k_1}^{(1)} = 0.5^{k_1 - 1}$, $k_1 = 1, 2, 3, 4$ and $\lambda_{k_2}^{(2)} = 0.5^{k_2 - 1}$, $k_2 = 1, 2, 3, 4$. The true eigenfunctions are selected as

$$\text{Level 1: } \phi_{k_1}(s) = \left\{\sqrt{2}\sin(2\pi s),\ \sqrt{2}\cos(2\pi s),\ \sqrt{2}\sin(4\pi s),\ \sqrt{2}\cos(4\pi s)\right\}$$

$$\text{Level 2: } \psi_{k_2}(s) = \left\{1,\ \sqrt{3}(2s-1),\ \sqrt{5}\left(6s^2 - 6s + 1\right),\ \sqrt{7}\left(20s^3 - 30s^2 + 12s - 1\right)\right\}$$

The eigenfunctions within levels 1 and 2 are mutually orthogonal, but they are not orthogonal between levels. We fix $\sigma = 1$, as the difference on computation time is marginal for different noise levels. We consider the following sample size parameters: (1) number of subjects: $I \in \{100, 200, 1000, 5000\}$; (2) number of visits per subject: $J \in \{2, 4, 20, 100\}$. For unbalanced design, the number of visits J_i is drawn from Poisson(J) with a minimum of 1 visit for subject i; and (3) dimension of the functional domain: $L \in \{100, 200, 1000, 5000, 50000\}$. To reduce computational burden, we set the baseline as $\{I = 100, J = 2, L = 100\}$ and increase the sample size one at a time while fixing the others. For example, we fix $J = 2$, $L = 100$ and increase I from 100 to 5000. This gives a total of $2 \times 2 \times (4 + 3 + 4) = 44$ simulation scenarios. For each scenario we conduct 100 replications on a high performance computing cluster using 1 core per simulation. The computation time of fast MFPCA (`mfpca.face`) and MFPCA (`mfpca.sc`) is obtained under different scenarios. In addition, we derive the estimation accuracy of both methods by calculating MISE($\mathbf{Y}$), $\text{MISE}(\phi) = (N_1 L)^{-1} \|\hat{\phi} - \phi\|_F^2$ and $\text{MISE}(\boldsymbol{\psi}) = (N_2 L)^{-1} \|\widehat{\boldsymbol{\psi}} - \boldsymbol{\psi}\|_F^2$ for each simulation.

5.2 Simulation Results

Tables 1–3 provide the simulation results for different scenarios. For each table, we only increase one parameter in the order by *I, J, L,* while fixing the others at their baseline, as discussed in Section 5.1. Within each table, we show the computation time ("Time(s)"), MISE of $\mathbf{Y}$ ("MISE($\mathbf{Y}$)") and MISE of eigenfunctions ("MISE($\boldsymbol{\phi}$)", "MISE($\boldsymbol{\psi}$)") using both methods for complete and incomplete data for balanced and unbalanced designs. For fast MFPCA we display the results weighted by visits. The results weighted by subjects are in Section S.4 of the supplementary material.

When I is large, fast MFPCA achieves similar accuracy for eigenfunction estimation with MFPCA for a balanced design and performs slightly better when the data are unbalanced. From a computational perspective, both methods exhibit a linear increase in computation time with *I,* though fast MFPCA is still much faster than MFPCA. For example, for $I = 5000$ with complete data and unbalanced design fast MFPCA takes less than 6 seconds compared to 900 seconds for traditional MFPCA (Table 1). The computational advantage of fast MFPCA is more pronounced when J and L increase; see Table 2 and Table 3. For example, for complete data with an unbalanced design, MFPCA takes at least 3 hours when

$J = 20$ and more than a day when $J = 100$. In contrast, fast MFPCA takes 2.7 seconds for $J = 20$ and less than 8 seconds for $J = 100$. For large L, MFPCA slows down substantially and takes, for example, over 6 hours for an analysis of complete unbalanced data when $L = 1000$. In contrast, fast MFPCA takes fewer than 100 seconds for $L = 50000$ (MFPCA would simply not run on such large examples).

Figure 2 shows the estimated eigenvalues for the complete unbalanced data when $I = 1000$. True eigenvalues are shown as gray dashed lines, while results from 100 simulations are shown in red for fast MFPCA and in blue for MFPCA. The eigenvalue estimates of both levels are close to their nominal values, while the level-2 estimates have higher precision using both methods. For level-1 there is a slight bias for the third and fourth eigenvalues using both methods, while the first eigenvalue estimates appears to be more precise for fast FPCA. In general, the eigenvalue estimates are accurate for large datasets using both approaches.

Figure 3 shows the estimated eigenfunctions under the same simulation setting. The top two rows display estimates from fast MFPCA in red and the bottom two rows display estimates from MFPCA in blue. Within each panel, the black solid curves indicate the true eigenfunctions at each level. For level-1 eigenfunctions, both methods exhibit similar accuracy. For level-2 eigenfunctions, we observe a larger variability for MFPCA, especially on the third and fourth eigenfunctions. This higher accuracy of fast MFPCA is also reflected by its slightly smaller MISE($\boldsymbol{\phi}$) (0.0120 vs. 0.0146) and much smaller MISE($\boldsymbol{\psi}$) (0.0063 vs. 0.0176) shown in Table 1.

In summary, fast MFPCA achieves similar estimation accuracy with MFPCA under different simulation settings, while the computation is at least two orders of magnitude faster. For a dataset with a large number of visits per subject ($J \sim 100$) or very high dimensions of the functional domain ($L \sim 50000$), fast MFPCA helps reduce the total computation time from several days or longer to just a few minutes.

6 Application

Objective physical activity measured by accelerometers and its association with health outcomes is an active area of research (Cui et al., 2021a; Smirnova et al., 2020). The National Health and Nutrition Examination Survey (NHANES) is a study conducted by the United States Centers for Disease Control and Prevention (CDC) with the aim of assessing the health and nutritional status of the US population. It became a continuous program conducted in two-year waves since 1999. The NHANES study collected accelerometry data using hip-worn physical activity monitors (ActiGraph model AM-7164) in the 2003–2004 and 2005–2006 waves. Both waves share the same protocol, where each study participant was asked to wear the device for 7 consecutive days. Data were released by the National Center for Health Statistics (NCHS) as minute-level activity counts (AC), a proprietary measure of physical activity intensity. We use the processed accelerometry data as described in Leroux et al. (2019). To reduce the severe skewness of the original data, for this analysis the minute-level AC were transformed into LAC := log(1 + AC), as suggested by Varma et al. (2017, 2018). The 2003–2004 and 2005–2006 waves have a total of 14631 study

participants with accelerometry data. Days with less than 10 hours of estimated wear time or days that were deemed by NHANES to have poor quality data were excluded. The final dataset has 12802 study participants and 65777 participant-days, with 1440 observations per day. The average number of available days per study participant is 5.14.

For this analysis, we are interested in decomposing the variability of the minute-level accelerometry data at both study participant (level-1) and day of the week (level-2) levels. While the problem is stated in simple terms, applying the existing MFPCA method to this large dataset takes at least 5 days on a regular laptop (2.7GHz Dual-Core i5 Processor). In contrast, fast MFPCA took less than 5 minutes on the same laptop.

Figure 4 displays the estimated overall mean function $\mu(s)$ and the mean function for each day of the week $\mu(s) + \eta_j(s)$, $j = 1, \ldots, 7$. The weekend curves are shown as dashed lines, while the weekday curves are shown as dotted lines. The overall mean function exhibits a clear circadian rhythm. In addition, there are distinguishable weekend-weekday patterns, as the physical activity intensity is higher than average on Friday and Saturday nights and lower than average on Saturday and Sunday mornings. These results provide visual evidence of a weekend effect in the NHANES cohort.

We identify 22 level-1 principal components and 31 level-2 principal components using the pre-specified percentage of variance explained (PVE) with a value of 0.99 at both levels. The total explained between-subject variance is 1.02. The total explained within-subject variance is 1.78, which is nearly twice that of the between-subject variance. The proportion of variability explained by level 1 is 0.36, defined as $\sum_{k_1=1}^{\infty} \lambda_{k_1}^{(1)} / (\sum_{k_1=1}^{\infty} \lambda_{k_1}^{(1)} + \sum_{k_2=1}^{\infty} \lambda_{k_2}^{(2)})$ in Di et al. (2009). Figure 5a shows the first three estimated level-1 eigenfunctions $\phi_{k_1}^{(1)}(s)$ of the physical activity data, which explain 78.4% of the total variability. The first eigenfunction is negative at night and positive during the day, suggesting that study participants with positive scores on this component will have less activity at night and more activity during the day. The second eigenfunction is only negative during the morning (5am to 12pm), suggesting that study participants with positive scores on this component will have less activity in the morning and more activity during the rest of a day. Study participants with positive scores on the third component have less activity during working hours (10am to 6pm) and more activity at all other times of a day.

At level 2 the first 3 components explain only 35.8% of the level 2 variability. Figure 5b shows the first three estimated level-2 eigenfunctions $\psi_{k_2}(s)$. The interpretation is different, as level-2 characterizes within-subject behavior. Specifically, days of the week with positive scores on the first principal component correspond to lower physical activity at night and sharply higher in the morning compared to the average activity of the individual. Similarly, days of the week with positive scores on the second principal component correspond to lower activity during the morning and higher during the rest of the day compared to the average activity of the individual.

7 Discussion

We propose fast MFPCA, which solves the major computational bottlenecks of the traditional MFPCA (Di et al., 2009), enabling it to be used on much larger and higher dimensional data sets. For example, the NHANES dataset contains minute-level physical activity information of more than 10000 study participants over multiple days. While applying MFPCA on such dataset takes more than 5 days on a regular laptop, the proposed fast MFPCA takes less than 5 minutes. The substantial computational improvement is due both to the development of new methods and to their careful coding. Simulation results show that fast MFPCA achieves similar estimation accuracy with MFPCA, while the computation times are at least two orders of magnitude faster.

In this paper we only considered a dense design for functional data, the most common scenario. However, the extension of FACE Xiao et al. (2018) to sparse designs suggests possible extensions to multilevel sparse functional data (Di et al., 2014). Such extensions will be studied in future work.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

ACKNOWLEDGEMENTS

We thank the editor, the associate editor, and two reviewers for their careful reading of the original manuscript and for their comments that significantly improved the paper. This work was supported by the National Institute of Neurological Disorders and Stroke under Award Number R01 NS060910, R01 NS091307 and R01 NS112303; and National Institute on Aging under Award Number R01 AG064803.

References

Aston JA, Chiou J-M, and Evans JP (2010). Linguistic pitch analysis using functional principal component mixed effect models. Journal of the Royal Statistical Society: Series C (Applied Statistics), 59(2):297–317.

Berrendero JR, Justel A, and Svarc M (2011). Principal components for multivariate functional data. Computational Statistics & Data Analysis, 55(9):2619–2634.

Boland J, Telesca D, Sugar C, Jeste S, Goldbeck C, and Senturk D (2022). A study of longitudinal trends in time-frequency transformations of EEG data during a learning experiment. Computational Statistics & Data Analysis, 167:107367. [PubMed: 35663825]

Brumback BA and Rice JA (1998). Smoothing spline models for the analysis of nested and crossed samples of curves. Journal of the American Statistical Association, 93(443):961–976.

Chen K and Müller H-G (2012). Modeling repeated functional observations. Journal of the American Statistical Association, 107(500):1599–1609.

Chiou J-M, Chen Y-T, and Yang Y-F (2014). Multivariate functional principal component analysis: A normalization approach. Statistica Sinica, 24(4):1571–1596.

Cui E, Crainiceanu CM, and Leroux A (2021a). Additive functional Cox model. Journal of Computational and Graphical Statistics, 30(3):780–793. [PubMed: 34898969]

Cui E, Leroux A, Smirnova E, and Crainiceanu CM (2021b). Fast univariate inference for longitudinal functional models. Journal of Computational and Graphical Statistics, 31(1):1–12.

Di C, Crainiceanu CM, Caffo BS, and Punjabi NM (2009). Multilevel functional principal component analysis. The Annals of Applied Statistics, 3(1):458. [PubMed: 20221415]

Di C, Crainiceanu CM, and Jank WS (2014). Multilevel sparse functional principal component analysis. Stat, 3(1):126–143. [PubMed: 24872597]

Eilers P and Marx B (1996). Flexible smoothing with b-splines and penalties (with Discussion). Statistical Science, 11:89–121.

Gaynanova I, Punjabi N, and Crainiceanu C (2022). Modeling continuous glucose monitoring (CGM) data during sleep. Biostatistics, 23(1):223–239. [PubMed: 32443145]

Goldsmith J, Scheipl F, Huang L, Wrobel J, Di C, Gellar J, Harezlak J, McLean MW, Swihart B, Xiao L, Crainiceanu C, and Reiss PT (2020). refund: Regression with functional data. R package version 0.1–23.

Goldsmith J, Zipunnikov V, and Schrack J (2015). Generalized multilevel function-on-scalar regression and principal component analysis. Biometrics, 71(2):344–353. [PubMed: 25620473]

Greven S, Crainiceanu C, Caffo B, and Reich DS (2010). Longitudinal functional principal component analysis. Electronic Journal of Statistics, 4:1022–1054. [PubMed: 21743825]

Happ C and Greven S (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. Journal of the American Statistical Association, 113(522):649–659.

Henderson CR (1963). Selection index and expected genetic advance. Statisitical Genetics and Plant Breeding, 982:140–163.

Henderson CR (1973). Sire evaluation and genetic trends. Journal of Animal Science, 1973(Symposium):10–41.

Kowal DR, Matteson DS, and Ruppert D (2017). A Bayesian multivariate functional dynamic linear model. Journal of the American Statistical Association, 112(518):733–744.

Leroux A, Di J, Smirnova E, Mcguffey EJ, Cao Q, Bayatmokhtari E, Tabacu L, Zipunnikov V, Urbanek JK, and Crainiceanu C (2019). Organizing and analyzing the activity data in nhanes. Statistics in Biosciences, 11(2):262–287. [PubMed: 32047572]

Li H, Kozey Keadle S, Staudenmayer J, Assaad H, Huang JZ, and Carroll RJ (2015). Methods to assess an exercise intervention trial based on 3-level functional data. Biostatistics, 16(4):754–771. [PubMed: 25987650]

Li R, Xiao L, Smirnova E, Cui E, Leroux A, and Crainiceanu CM (2022). Fixed-effects inference and tests of correlation for longitudinal functional data. Statistics in Medicine.

Li Y, Nguyen DV, Banerjee S, Rhee CM, Kalantar-Zadeh K, Kürüm E, and Şentürk D (2021). Multilevel modeling of spatially nested functional data: Spatiotemporal patterns of hospitalization rates in the us dialysis population. Statistics in Medicine, 40(17):3937–3952. [PubMed: 33902165]

Morris JS, Baladandayuthapani V, Herrick RC, Sanna P, and Gutstein H (2011). Automated analysis of quantitative image data using isomorphic functional mixed models, with application to proteomics data. The Annals of Applied Statistics, 5(2A):894. [PubMed: 22408711]

Morris JS and Carroll RJ (2006). Wavelet-based functional mixed models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(2):179–199. [PubMed: 19759841]

Park SY and Staicu A-M (2015). Longitudinal functional data analysis. Stat, 4(1):212–226. [PubMed: 26594358]

Ramsay J and Silverman B (2005). Functional data analysis. Springer, New York.

Ramsay JO and Dalzell C (1991). Some tools for functional data analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 53(3):539–561.

Scheffler A, Telesca D, Li Q, Sugar CA, Distefano C, Jeste S, and Şentürk D (2020). Hybrid principal components analysis for region-referenced longitudinal functional EEG data. Biostatistics, 21(1):139–157. [PubMed: 30084925]

Scheipl F, Staicu A-M, and Greven S (2015). Functional additive mixed models. Journal of Computational and Graphical Statistics, 24(2):477–501. [PubMed: 26347592]

Serban N and Jiang H (2012). Multilevel functional clustering analysis. Biometrics, 68(3):805–814. [PubMed: 22313290]

Shamshoian J, Şentürk D, Jeste S, and Telesca D (2022). Bayesian analysis of longitudinal and multidimensional functional data. Biostatistics, 23(2):558–573. [PubMed: 33017019]

Shou H, Zipunnikov V, Crainiceanu CM, and Greven S (2015). Structured functional principal component analysis. Biometrics, 71(1):247–257. [PubMed: 25327216]

Silverman BW et al. (1996). Smoothed functional principal components analysis by choice of norm. The Annals of Statistics, 24(1):1–24.

Smirnova E, Leroux A, Cao Q, Tabacu L, Zipunnikov V, Crainiceanu C, and Urbanek JK (2020). The predictive performance of objective measures of physical activity derived from accelerometry data for 5-year all-cause mortality in older adults: National health and nutritional examination survey 2003–2006. The Journals of Gerontology: Series A, 75(9):1779–1785.

Varma VR, Dey D, Leroux A, Di J, Urbanek J, Xiao L, and Zipunnikov V (2017). Re-evaluating the effect of age on physical activity over the lifespan. Preventive Medicine, 101:102–108. [PubMed: 28579498]

Varma VR, Dey D, Leroux A, Di J, Urbanek J, Xiao L, and Zipunnikov V (2018). Total volume of physical activity: TAC, TLAC or TAC(λ). Preventive Medicine, 106:233–235. [PubMed: 29080825]

Wong RK, Li Y, and Zhu Z (2019). Partially linear functional additive models for multivariate functional data. Journal of the American Statistical Association, 114(525):406–418.

Xiao L (2019). Asymptotic theory of penalized splines. Electronic Journal of Statistics, 13(1):747–794.

Xiao L (2020). Asymptotic properties of penalized splines for functional data. Bernoulli, 26(4):2847–2875.

Xiao L, Li C, Checkley W, and Crainiceanu C (2018). Fast covariance estimation for sparse functional data. Statistics and Computing, 28(3):511–522. [PubMed: 29449762]

Xiao L, Li Y, and Ruppert D (2013). Fast bivariate p-splines: the sandwich smoother. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(3):577–599.

Xiao L, Zipunnikov V, Ruppert D, and Crainiceanu C (2016). Fast covariance estimation for high-dimensional functional data. Statistics and Computing, 26(1):409–421. [PubMed: 26903705]

Xu Y, Li Y, and Nettleton D (2018). Nested hierarchical functional data modeling and inference for the analysis of functional plant phenotypes. Journal of the American Statistical Association, 113(522):593–606.

Yao F, Müller H-G, and Wang J-L (2005). Functional data analysis for sparse longitudinal data. Journal of the American Statistical Association, 100(470):577–590.

Zhang L, Baladandayuthapani V, Zhu H, Baggerly KA, Majewski T, Czerniak BA, and Morris JS (2016). Functional CAR models for large spatially correlated functional datasets. Journal of the American Statistical Association, 111(514):772–786. [PubMed: 28018013]

Zipunnikov V, Greven S, Shou H, Caffo B, Reich DS, and Crainiceanu C (2014). Longitudinal high-dimensional principal components analysis with application to diffusion tensor imaging of multiple sclerosis. The Annals of Applied Statistics, 8(4):2175–2202. [PubMed: 25663955]

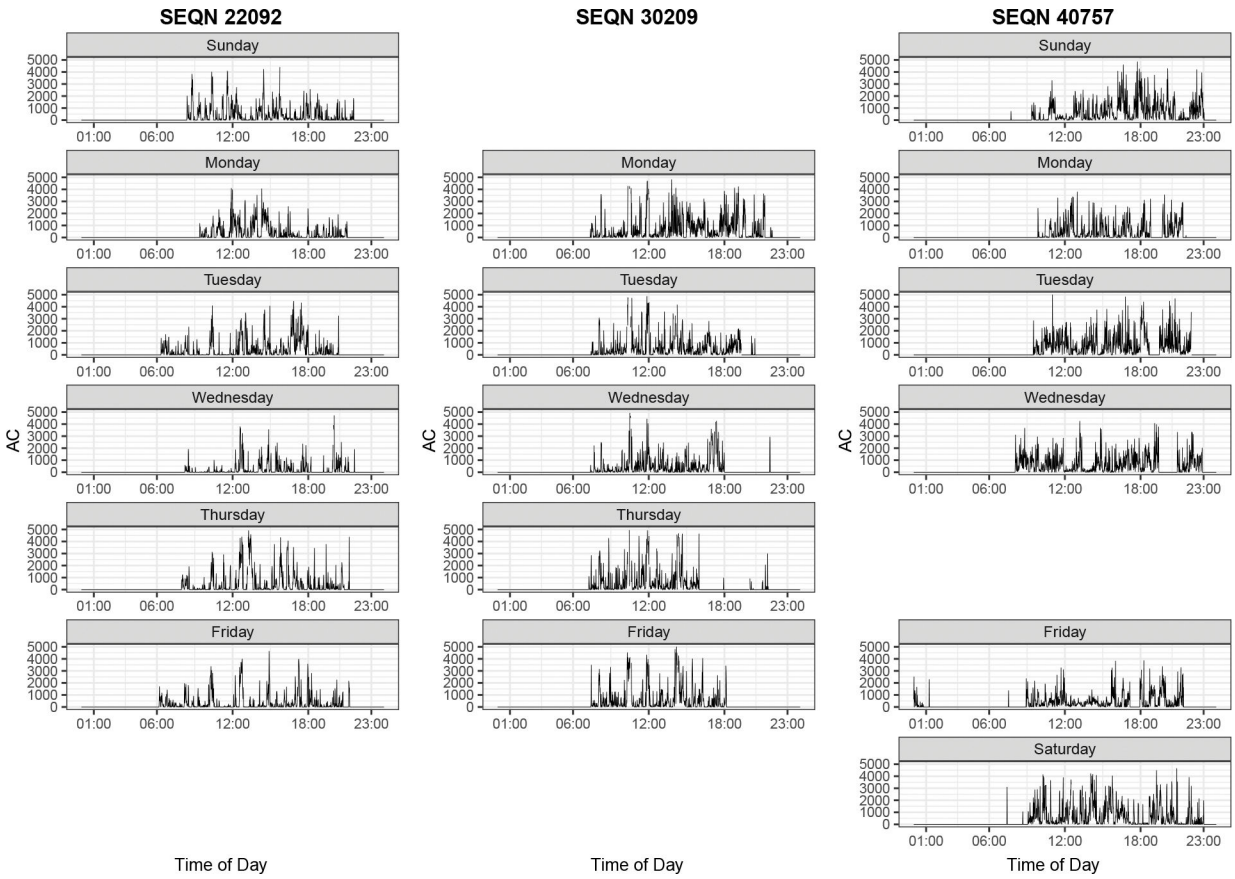


Figure 1:

Physical activity profiles of three NHANES study participants over available days. Each study participant is uniquely identified by the SEQN number. Left column: SEQN 22092. Middle column: SEQN 30209. Right column: SEQN 40757. Within each column, each row displays the minute-level AC of one day from midnight to midnight, titled by day of the week from Sunday (top row) to Saturday (bottom row).

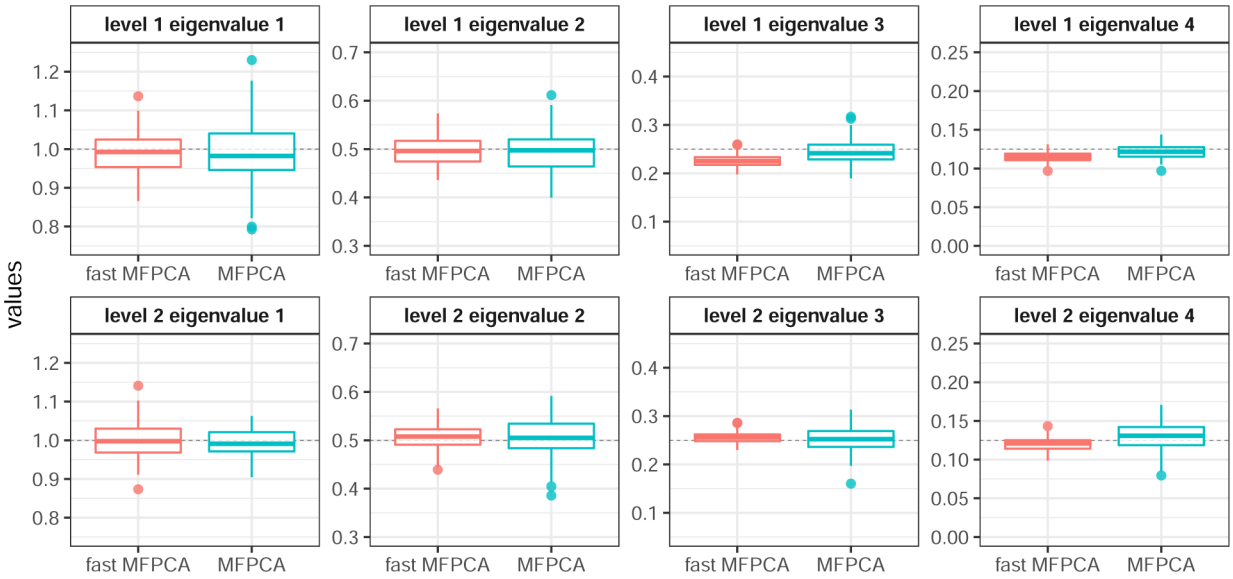


Figure 2:

Boxplots of estimated eigenvalues from 100 replications when the data are complete with $I = 1000$, $J = 2$, $L = 100$ under unbalanced design for level-1 (first row) and level-2 (second row). True eigenvalues are shown as gray dashed lines, fast MFPCA are shown in red while MFPCA are shown in blue.

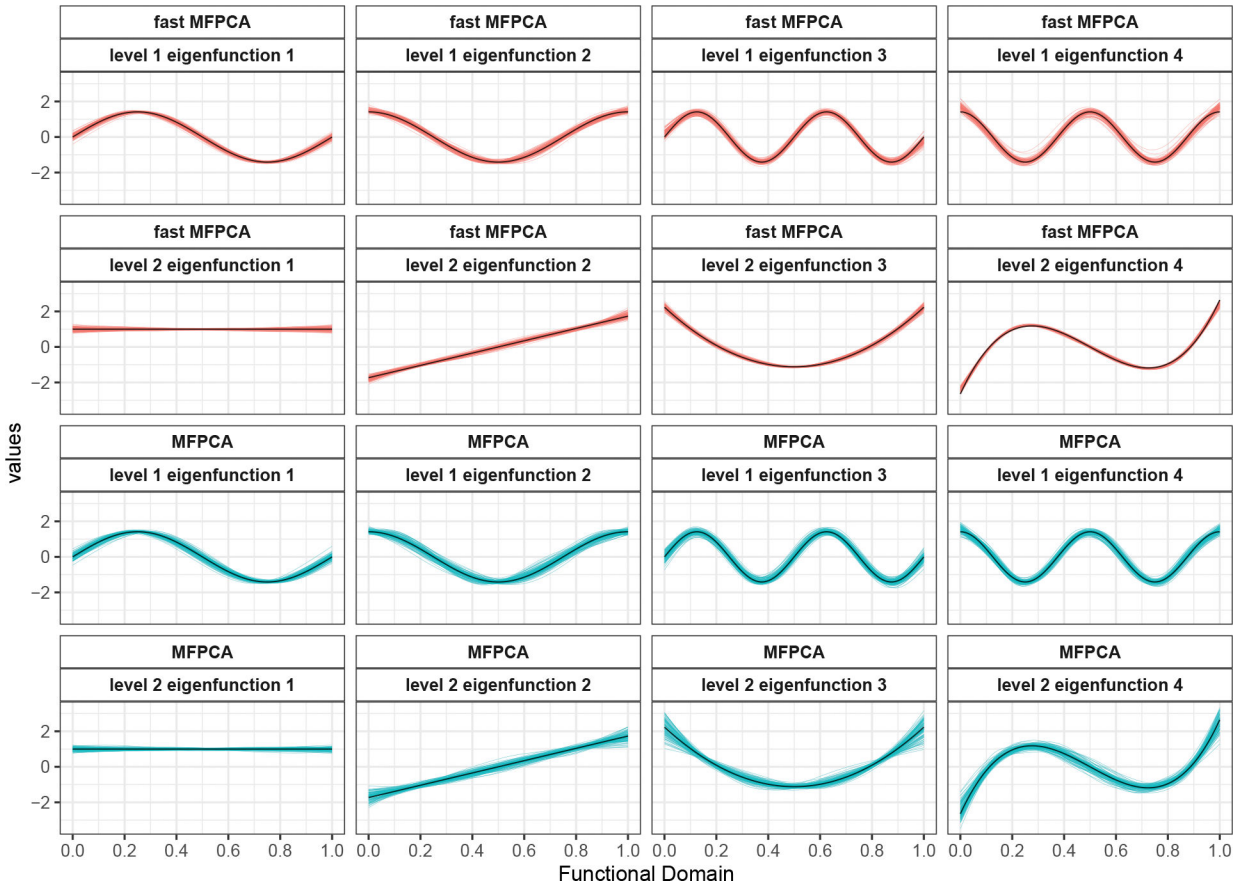


Figure 3:

Estimated eigenfunctions for fast MFPCA (top two rows) and MFPCA (bottom two rows) when the data are complete with $I = 1000$, $J = 2$, $L = 100$ with unbalanced design. Within each model, the top row displays level-1 estimates and the bottom row displays level-2 estimates. Black lines: true eigenfunction; red lines: 100 fast MFPCA estimates; blue lines: 100 MFPCA estimates.

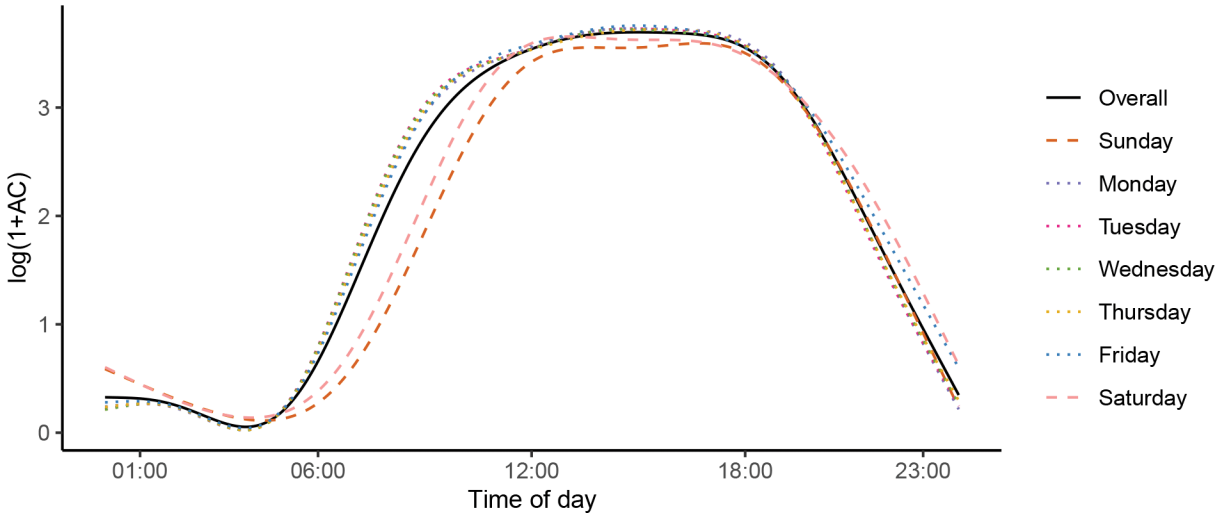
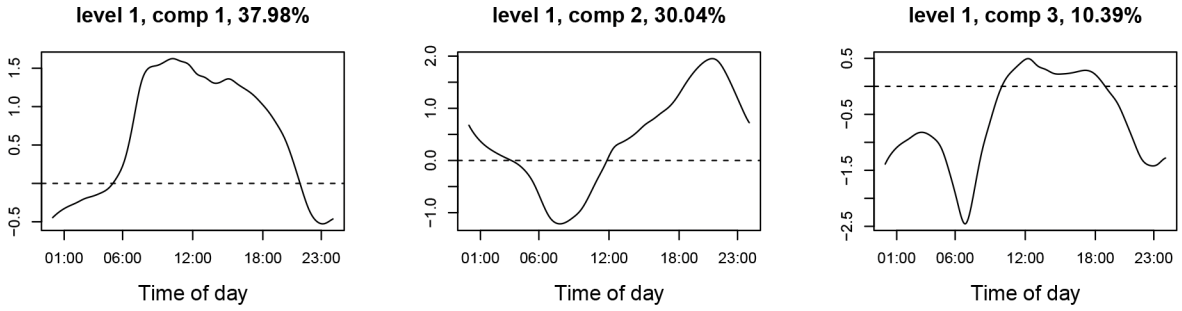
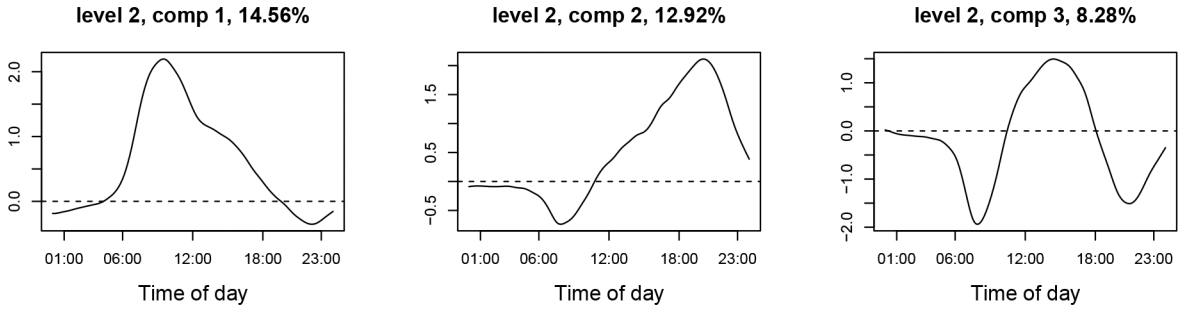


Figure 4:

Estimated overall mean function $\mu(s)$ and day-of-the-week-specific mean function $\mu(s) + \eta_j(s)$ in the NHANES dataset using fast MFPCA. Overall mean curve: black solid line; weekend days means: dashed lines; weekday mean curves: dotted lines.



(a) The first three estimated level-1 eigenfunctions.



(b) The first three estimated level-2 eigenfunctions.

Figure 5:
The top three estimated level-1 (first row) and level-2 (second row) eigenfunctions from the NHANES dataset using fast MFPCA. The proportion of variability explained in each principal component within each level is shown on the title of each panel.

Table 1:

Simulation results for different I when $J = 2$ and $L = 100$. The computation time ("Time(s)"), MISE of $\mathbf{Y}$ ("MISE($\mathbf{Y}$)") and eigenfunctions ("MISE($\boldsymbol{\phi}$)", "MISE($\boldsymbol{\psi}$)") reported in the table are median values across 100 replications.

Design	I	Method	Balanced				Unbalanced			
			Time(s)	MISE(Y)	MISE($\boldsymbol{\phi}$)	MISE($\boldsymbol{\psi}$)	Time(s)	MISE(Y)	MISE($\boldsymbol{\phi}$)	MISE($\boldsymbol{\psi}$)
Complete	100	fast MFPCA	**1.81**	0.9450	0.0781	0.0319	**1.74**	0.9450	0.1203	0.0416
		MFPCA	**14.82**	0.9451	0.0797	0.0315	**21.51**	0.9571	0.1930	0.1286
	200	fast MFPCA	**1.70**	0.9469	0.0413	0.0182	**1.88**	0.9465	0.0469	0.0229
		MFPCA	**25.42**	0.9474	0.0377	0.0171	**48.90**	0.9530	0.0780	0.0698
	1000	fast MFPCA	**2.20**	0.9505	0.0093	0.0075	**2.07**	0.9524	0.0120	0.0063
		MFPCA	**109.15**	0.9501	0.0073	0.0042	**262.76**	0.9525	0.0146	0.0176
	5000	fast MFPCA	**4.60**	0.9506	0.0034	0.0043	**5.31**	0.9516	0.0037	0.0046
		MFPCA	**540.84**	0.9503	0.0019	0.0008	**943.56**	0.9514	0.0036	0.0033
Incomplete	100	fast MFPCA	**2.16**	0.9003	0.0942	0.0348	**2.03**	0.9010	0.1570	0.0461
		MFPCA	**3.58**	0.8921	0.0999	0.0614	**5.00**	0.9089	0.2496	0.1644
	200	fast MFPCA	**2.34**	0.9028	0.0554	0.0198	**2.46**	0.9056	0.0671	0.0278
		MFPCA	**4.80**	0.8950	0.0489	0.0303	**8.34**	0.9045	0.0917	0.0857
	1000	fast MFPCA	**5.48**	0.9087	0.0230	0.0081	**5.83**	0.9074	0.0246	0.0074
		MFPCA	**13.19**	0.9037	0.0102	0.0064	**29.46**	0.9055	0.0179	0.0194
	5000	fast MFPCA	**19.99**	0.9077	0.0147	0.0048	**22.21**	0.9086	0.0150	0.0051
		MFPCA	**58.19**	0.9041	0.0022	0.0012	**134.49**	0.9066	0.0043	0.0037

Table 2:

Simulation results for different J when $I = 100$ and $L = 100$. The computation time ("Time(s)"), MISE of $\mathbf{Y}$ ("MISE($\mathbf{Y}$)") and eigenfunctions ("MISE(ϕ)", "MISE(ψ)") reported in the table are median values across 100 replications. Computation time more than 24 hours is denoted as ∞.

Design	J	Method	Balanced				Unbalanced			
			Time(s)	MISE(Y)	MISE(ϕ)	MISE(ψ)	Time(s)	MISE(Y)	MISE(ϕ)	MISE(ψ)
Complete	2	fast MFPCA	**1.81**	0.9450	0.0781	0.0319	**1.74**	0.9450	0.1203	0.0416
		MFPCA	**14.82**	0.9451	0.0797	0.0315	**21.51**	0.9571	0.1930	0.1286
	4	fast MFPCA	**1.98**	0.9532	0.0547	0.0126	**1.80**	0.9528	0.0634	0.0171
		MFPCA	**46.93**	0.9537	0.0483	0.0099	**75.87**	0.9597	0.0800	0.0463
	20	fast MFPCA	**2.33**	0.9614	0.0364	0.0056	**2.66**	0.9618	0.0314	0.0054
		MFPCA	**10146.82**	0.9617	0.0317	0.0019	**10334.10**	0.9626	0.0346	0.0088
	100	fast MFPCA	**6.02**	0.9626	0.0335	0.0042	**7.15**	0.9632	0.0332	0.0043
		MFPCA	∞	-	-	-	∞	-	-	-
Incomplete	2	fast MFPCA	**2.16**	0.9003	0.0942	0.0348	**2.03**	0.9010	0.1570	0.0461
		MFPCA	**3.58**	0.8921	0.0999	0.0614	**5.00**	0.9089	0.2496	0.1644
	4	fast MFPCA	**2.48**	0.9094	0.0616	0.0142	**2.48**	0.9100	0.0784	0.0200
		MFPCA	**11.07**	0.9147	0.0615	0.0161	**17.14**	0.9173	0.0842	0.0544
	20	fast MFPCA	**5.88**	0.9158	0.0432	0.0060	**6.08**	0.9162	0.0393	0.0067
		MFPCA	**733.01**	0.9254	0.0325	0.0037	**903.30**	0.9262	0.0399	0.0099
	100	fast MFPCA	**21.44**	0.9159	0.0384	0.0044	**23.72**	0.9160	0.0375	0.0046
		MFPCA	∞	-	-	-	∞	-	-	-

Table 3:

Simulation results for different L when $I = 100$ and $J = 2$. The computation time ("Time(s)"), MISE of $\mathbf{Y}$ ("MISE($\mathbf{Y}$)") and eigenfunctions ("MISE($\boldsymbol{\phi}$)", "MISE($\boldsymbol{\psi}$)") reported in the table are median values across 100 replications. Computation time more than 24 hours is denoted as ∞.

Design	L	Method	Balanced				Unbalanced			
			Time(s)	MISE(Y)	MISE($\boldsymbol{\phi}$)	MISE($\boldsymbol{\psi}$)	Time(s)	MISE(Y)	MISE($\boldsymbol{\phi}$)	MISE($\boldsymbol{\psi}$)
Complete	100	fast MFPCA	**1.81**	0.9450	0.0781	0.0319	**1.74**	0.9450	0.1203	0.0416
		MFPCA	**14.82**	0.9451	0.0797	0.0315	**21.51**	0.9571	0.1930	0.1286
	200	fast MFPCA	**1.67**	0.9722	0.0804	0.0277	**1.78**	0.9746	0.1193	0.0395
		MFPCA	**119.92**	0.9733	0.0785	0.0272	**334.66**	0.9830	0.1933	0.1398
	1000	fast MFPCA	**2.51**	0.9953	0.0758	0.0244	**3.06**	0.9976	0.1145	0.0368
		MFPCA	**16883.52**	0.9963	0.0784	0.0237	**24395.62**	1.0055	0.1794	0.1058
	5000	fast MFPCA	**7.79**	0.9990	0.0756	0.0246	**12.40**	1.0023	0.1103	0.0369
		MFPCA	∞	-	-	-	∞	-	-	-
	50000	fast MFPCA	**29.05**	1.0000	0.0767	0.0243	**68.71**	1.0028	0.1081	0.0362
		MFPCA	∞	-	-	-	∞	-	-	-
Incomplete	100	fast MFPCA	**2.16**	0.9003	0.0942	0.0348	**2.03**	0.9010	0.1570	0.0461
		MFPCA	**3.58**	0.8921	0.0999	0.0614	**5.00**	0.9089	0.2496	0.1644
	200	fast MFPCA	**2.52**	0.9515	0.0828	0.0291	**2.41**	0.9498	0.1308	0.0408
		MFPCA	**17.01**	0.9484	0.0914	0.0368	**26.18**	0.9579	0.2138	0.1426
	1000	fast MFPCA	**3.35**	0.9923	0.0811	0.0263	**3.86**	0.9949	0.1160	0.0363
		MFPCA	**1883.75**	0.9903	0.0788	0.0274	**2697.17**	1.0023	0.1889	0.1425
	5000	fast MFPCA	**10.09**	0.9985	0.0773	0.0247	**14.56**	1.0022	0.1147	0.0374
		MFPCA	∞	-	-	-	∞	-	-	-
	50000	fast MFPCA	**60.59**	1.0006	0.0761	0.0250	**91.23**	1.0034	0.1091	0.0368
		MFPCA	∞	-	-	-	∞	-	-	-