Analysis of Active/Inactive Patterns in the NHANES Data using Generalized Multilevel Functional Principal Component Analysis

Xinkai Zhou*

Department of Biostatistics, Johns Hopkins University

Julia Wrobel

Department of Biostatistics and Bionformatics, Emory University

Ciprian M. Crainiceanu

Department of Biostatistics, Johns Hopkins University

Andrew Leroux

Department of Biostatistics and Informatics, Colorado School of Public Health

Abstract

Between 2011 and 2014 NHANES collected objectively measured physical activity data using wrist-worn accelerometers for tens of thousands of individuals for up to seven days. Here we analyze the minute-level indicators of being active, which can be viewed as binary (because there is an active indicator at every minute), multilevel (because there are multiple days of data for each study participant), functional (because within-day data can be viewed as a function of time) data. To extract within- and between-participant directions of variation in the data, we introduce Generalized Multilevel Functional Principal Component Analysis (GM-FPCA), an approach based on the dimension reduction of the linear predictor. Scores associated with specific patterns of activity are shown to be strongly associated with time to death. Extensive simulation studies indicate that GM-FPCA provides accurate estimation of model parameters, is computationally stable, and is scalable in the number of study participants, visits, and observations within visits. R code for implementing the method is provided.

Keywords: accelerometry, functional data, GLMM, STAN

1 Introduction

Body-worn accelerometers allow high-resolution objective quantification of human physical activity. For example, both the US National Health and Nutrition Examination Survey (NHANES) and the UK Biobank study collected accelerometry data for tens of thousands of study participants. This paper focuses on the NHANES 2011-2014 accelerometry data, which was publicly released in December 2020. These data contain minute level summaries of physical activity for $13,603$ participants over seven days collected from wrist-worn accelerometers. Physical activity at each minute is summarized using a continuous measure called Monitor Independent Movement Summary units (MIMS) (John et al., 2019). In their original form, MIMS are difficult to interpret and translate into actionable information and physical activity recommendations. For this reason, it makes sense to first transform MIMS into measures that are easier to interpret. A few such transformations considered in the literature are: (1) binary data corresponding to active/inactive, walking/non-walking, or sleep/wake periods; (2) multinomial data corresponding to sedentary, low intensity physical activity (LIPA), and moderate to vigorous physical activity (MVPA); and (3) count data corresponding to the number of active seconds or steps in every minute. This paper focuses on the NHANES 2011-2014 binary active/inactive profile at the minute level over multiple days obtained by thresholding the continuous MIMS measures. However, the methods considered can be applied to any type of non-Gaussian data, including discrete data (e.g., binary, multinomial, counts) (Bothwell et al., 2022; Cui et al., 2022a; Gaston et al., 2008; Kass and Ventura, 2001; Kelly and Kass, 2012; Sebastián-González et al., 2010; Sentürk et al., 2014; Swihart et al., 2015) or continuous data with strong departures from normality (Gaynanova et al., 2020; Staicu et al., 2018).

To provide intuition about the data structure, Figure 1 displays the binary physical activity data for the $13,603$ subjects who participated in the NHANES 2011-2014 study. The binary active/inactive data is obtained by thresholding the MIMS data as $Z_{ijk} = \mathbf{1}_{\{W_{ijk} \geq 10.558\}}$, where W_{ijk} corresponds to the ith individual's MIMS measurement on day $j = 1, \ldots, J_i$ at minute k. For most study participants, $J_i = 7$, though some have a smaller number of days because some days were excluded for data quality purposes by NHANES. The threshold of 10.558 for physical inactivity was suggested by Karas et al. (2022). The top panel shows the binary (black dots correspond to active minutes while white dots correspond to inactive minutes) activity profiles for all study participants. Data are sorted by age on the $y-$axis and are shown from midnight to midnight on the x-axis for Wednesday (top left) and Saturday (top right). Figure 1 reveals activity patterns of a representative sample of the US population. A quick inspection of the plots indicates that (1) people generally wake up later and go to bed later on Saturday than on Wednesday; (2) people in their 20s tend to stay up past midnight and wake up later; and (3) children tend to be more inactive during the night and more active during the day than adults.

To visualize the day-to-day activity patterns, the lower panels show data for two randomly selected study participants, a 17 year-old (left) and a 62 year-old (right). Each sub-panel corresponds to one day and displays both the binary data (grey dots) and smooth estimates of the probability of being active (blue curves) (Wood, 2017). The plots indicate that there is substantial within- and between-day variability. Our goal is to model such generalized (e.g., binary) functional (e.g., minutes within a day) data observed at multiple time points
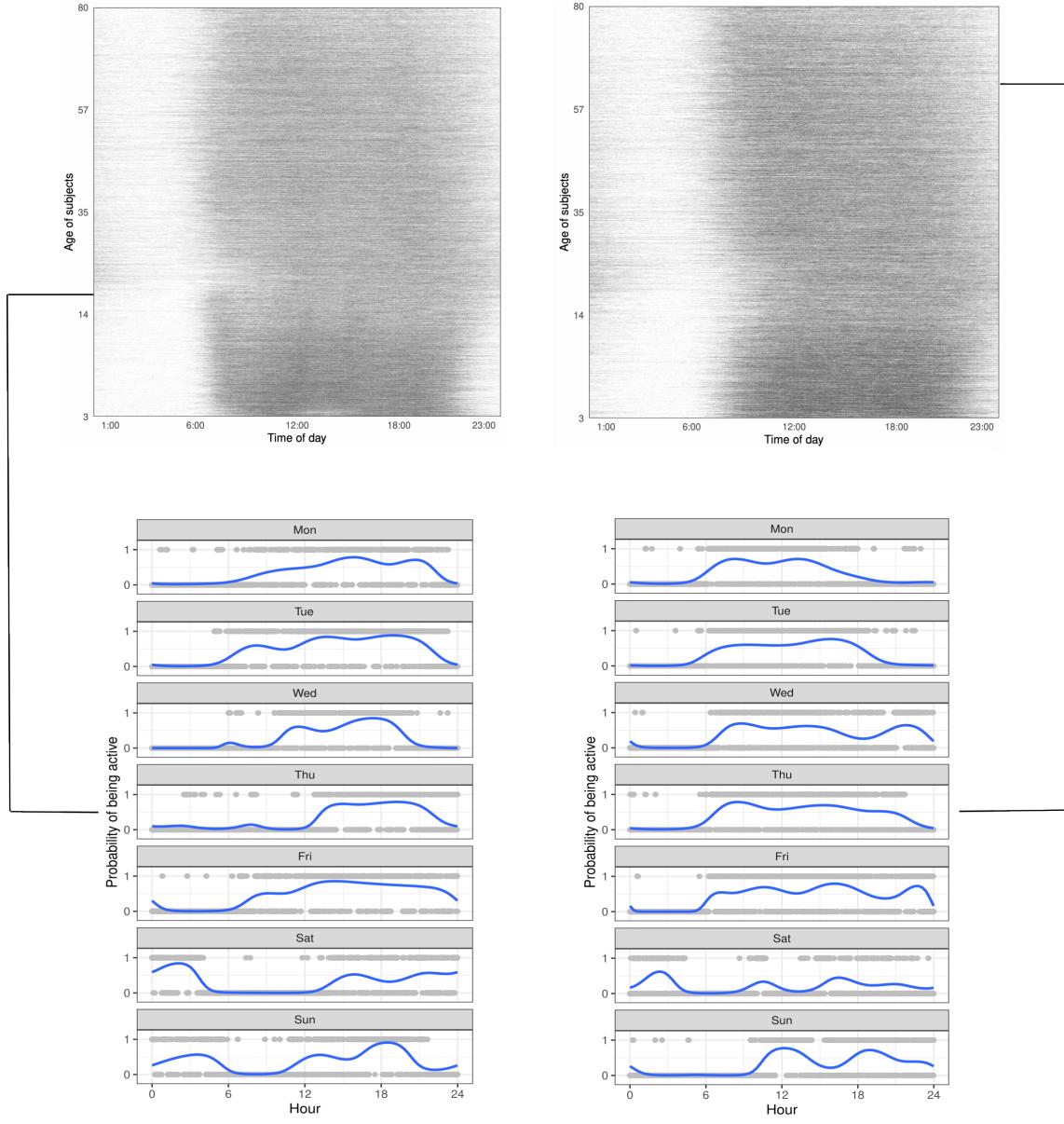
Figure 1: Top panels: binary active (black dots) / inactive (white dots) profiles for 13, 603 subjects in the NHANES 2011-2014 data for Wednesday (top left) and Saturday (top right). Bottom panels: zoom in on two subjects, a 17 year-old on the left and a 62 year-old on the right. Both the binary data (grey dots) and smooth estimates of the probability of being active (blue curves) are displayed.

(e.g., days).

We are not aware of any existing functional data analysis method that can handle data of the size and structure illustrated in Figure 1. Indeed, the binary active/inactive profiles in the

NHANES data exhibit the following characteristics: (1) a large number of study participants (13, 603) and observations per day (1, 440) (2) multiple days of observation (from 1 to 7); (3) substantial within- and between-day variability; (4) high within-participant correlation over the course of the day; and (5) binary observations. To address the combination of these challenges we introduce generalized multilevel functional principal component analysis (GM-FPCA). Here "generalized" refers to the non-Gaussianity of the measurements, "multilevel" refers to the multiple days within each individual, and "functional" refers to the functional representation of data, where the basic unit of observation is a curve measured over some domain (in our example, time). In our data set, each functional observation is a 24-hour physical activity trajectory for a given participant on a specific day.

The rest of the paper is organized as follows. Section 2 reviews the related functional data literature. Section 3 details our GM-FPCA method. Section 4 demonstrates the statistical and computational performance of our method through simulation for both binary and count data. Section 5 applies GM-FPCA to the multilevel active/inactive profiles obtained from the NHANES 2011-2014 accelerometer data. We conclude with a discussion in Section 6.

2   Literature review

Though we are not aware of any existing method that can handle generalized multilevel functional data of the size and complexity of the NHANES, there are several closely related ideas. GM-FPCA builds upon three methods in FDA: (1) functional principal components analysis (FPCA); (2) generalized functional principal component analysis (GFPCA) for generalized single-level functional data; and (3) multilevel functional principal component analysis (MF-PCA) for functional data observed at multiple visits. While there is a vast literature and reliable software for FPCA (Crainiceanu et al., 2023; Goldsmith et al., 2020; Jones and Rice, 1992; Ramsay and Silverman, 2005; Staniswalis and Lee, 1998; Xiao et al., 2016a), GFPCA and MFPCA require a closer look to understand the current state of the literature.

Generalized functional principal components analysis (GFPCA) extends FPCA to exponential-family outcomes such as binary or count data. More precisely, let Z_{ik} denote the non-Gaussian functional observation for participant $i = 1, \ldots, I$ at a point $s_k \in S$. The GFPCA model is

$$g\Big(\mathbb{E}\big[Z_{ik} \mid \{\xi_{il}\}_{l=1}^{\infty}\big]\Big) = \beta_0(s_k) + b_i(s_k) = \beta_0(s_k) + \sum_{l=1}^{\infty} \xi_{il}\phi_l(s_k) \,, \qquad (1)$$

where $g(\cdot)$ is a link function, $\beta_0(s)$ is the population-mean function on the linear predictor scale, $b_i(s)$ is a zero-mean Gaussian process which, by the Kosambi–Karhunen–Loève theorem, admits the expansion $b_i(s) = \sum_{l=1}^{\infty} \xi_{il}\phi_l(s)$, where $\{\phi_l(s)\}_l$ are population-level orthonormal eigenfunctions such that $\int_S \phi_l^2(s)ds = 1$, and $\xi_{il} \sim N(0, \lambda_l)$ are subject-specific uncorrelated scores.

Several methods exist for estimating Model (1), starting with Hall et al. (2008), who modeled Z_{ik} using a latent Gaussian process with mean and covariance estimators constructed using the Taylor expansion of $g(\cdot)$. Bayesian methods for non-Gaussian functional data were proposed by Gertheiss et al. (2017a) and van der Linde (2009) for a variety of sampling scenarios. An Exponential Family Functional Principal Component Analysis (EFPCA) was

proposed by Li et al. (2018), who used quadratic penalties on the scores and eigenfunctions of a low rank decomposition of the linear predictor. However, existing methods are either prohibitively slow for large datasets or are not accompanied by software. Notable exceptions are: (1) the variational Bayes approach for binary functional data (Wrobel et al., 2019) implemented in R via the `registr::bfpca()` function (Wrobel, 2018; Wrobel and Bauer, 2021); (2) the two-step GFPCA described in (Gertheiss et al., 2017b), which is implemented in the `registr::gfpca_twoStep()` function; and (3) a fast GFPCA method (Leroux et al., 2023) developed for any type of non-Gaussian data and deployed in the `fastGFPCA` package (Wrobel, 2023). Briefly, fast GFPCA reduces the computational complexity by estimating the latent processes in small bins along the functional domain, recovering eigenfunctions $\{\phi_l(s)\}$ using the fast covariance estimation (FACE) algorithm (Xiao et al., 2016b), and estimating the scores $\{\xi_{il}\}$ using a generalized linear mixed-model framework conditional on the eigenfunctions $\{\phi_l(s)\}$.

Multilevel functional principal components analysis (MFPCA) extends FPCA to the case when functional data are observed at multiple visits (days), but data are Gaussian. MFPCA was first proposed by Di et al. (2009), and is deployed in the `refund::mfpca.sc()` function (Goldsmith et al., 2020). A much improved version of the software was deployed recently in the `refund::mfpca.face()` function based on the paper by Cui et al. (2022b). Other developments include extensions to multivariate and multilevel functional data (Zhang et al., 2023).

Extending these models to the case when both the outcomes are non-Gaussian and the functional data has a multilevel structure is not straightforward. However, several approaches have been proposed. In particular, Chen et al. (2013) introduced a reduced-rank method for the analysis of generalized functional data; however, they use a marginal approach, which does not estimate the random effects components that are central to our analysis. Scheipl et al. (2016) provides a framework for generalized function-on-scalar regression that can incorporate multilevel structure through subject-specific random effects; however, this approach is memory intensive and does not scale up well to the data application considered in this paper. For example, for a subset of 500 study participants from NHANES, the model did not run using 64GB of memory. With 128GB of memory, the method did run, but took 10 days to finish. To run the analysis on the entire data set may be possible by splitting larger data sets into smaller subsets and run analyses on each subset. However, exactly how these subset analyses would be combined across all study participants remains an open methods question that we cannot pursue in this paper. Goldsmith et al. (2015) proposed a generalized multilevel function-on-scalar regression based on FPCA of the underlying latent space. They used a Bayesian modeling approach based on posterior simulations. However, this approach is also not computationally feasible for the scale and complexity of the data considered here. As a point of reference, Goldsmith et al. (2015) analyzed an accelerometer dataset of 583 subjects with 5 days per subject. After downsampling the data to 144 points per day, the method still took 10 days to run. This approach is not feasible for data sets of the size of NHANES, which contains tens of thousands of subjects, multiple days per subject, and 1440 observations per day. Therefore, new computational approaches are necessary. This paper provides a new strategy for inference, describes its statistical and computational performance through extensive simulation studies, and applies it to the NHANES 2011-2014 data.

3 Methods

The observed data is of the form $\{s_k, Z_{ijk}\}$, where $s_k \in S$ $(k = 1, ..., K)$ is a point in the domain S and Z_{ijk} is the non-Gaussian observation for subject i $(i = 1, ..., I)$, visit j $(j = 1, ..., J)$, and at the point s_k. For example, in the dichotomized NHANES 2011-2014 data, Z_{111} is a binary variable indicating whether subject 1 was active at midnight on day 1. While data is only observed on a discrete grid $\{s_k,\ k = 1, ..., K\}$, we assume that they are generated by a latent continuous process. Specifically, we assume that

$$g[\mathbb{E}\{Z_{ijk}\}] = \beta_0(s_k) + a_i(s_k) + b_{ij}(s_k) \ , \tag{2}$$

where $g(\cdot)$ is a link function, $\beta_0(\cdot)$ is the population-mean function on the linear predictor scale, and $a_i(\cdot)$ and $b_{ij}(\cdot)$ are subject- and subject-visit level deviations from the population-mean function. We further assume that $a_i(s) \sim GP(0, K_a)$ and $b_{ij}(s) \sim GP(0, K_b)$ are mutually independent zero-mean Gaussian processes with covariance operators K_a and K_b, respectively.

The main goal of GM-FPCA is to decompose the variability in $a_i(s)$ and $b_{ij}(s)$ along their main directions of variation. By the Kosambi–Karhunen–Loève theorem,

$$a_i(s) = \sum_{l=1}^{\infty} \xi_{il}\phi_l^{(1)}(s) \ , \ b_{ij}(s) = \sum_{m=1}^{\infty} \zeta_{ijm}\phi_m^{(2)}(s) \ ,$$

where $\{\phi_l^{(1)}(\cdot)\}$ and $\xi_{il} \sim N(0, \lambda_l^{(1)})$ $(l = 1, 2, ..., \infty)$ are subject-level orthonormal eigenfunctions and scores, $\{\phi_m^{(2)}(\cdot)\}$ and $\zeta_{ijm} \sim N(0, \lambda_m^{(2)})$ $(m = 1, 2, ..., \infty)$ are subject-visit level orthonormal eigenfunctions and scores, and the scores ξ_{il} and ζ_{ijm} are assumed to be mutually independent. This leads to the GM-FPCA model

$$g\left(\mathbb{E}\left[Z_{ijk} \mid \{\xi_{il}\}_{l=1}^{\infty}, \{\zeta_{ijm}\}_{m=1}^{\infty}\right]\right) = \beta_0(s_k) + \sum_{l=1}^{\infty} \xi_{il}\phi_l^{(1)}(s_k) + \sum_{m=1}^{\infty} \zeta_{ijm}\phi_m^{(2)}(s_k) \ . \tag{3}$$

Fitting this model requires a "divide and conquer" approach, which partitions the problem into a sequence of smaller problems that are each tractable. First, we divide the data domain into K overlapping bins centered at the grid points $s_k \in S$, $k = 1, \ldots, K$. Second, we fit GLMMs in each data bin with both subject- and subject-visit random effects, and extract subject-visit estimates on the linear predictor scale. This step is crucial because it transforms non-Gaussian data into continuous predictors, upon which MFPCA developed for Gaussian data can be applied. Third, we conduct MFPCA on the estimates from the second step to obtain subject- and subject-visit level eigenfunction estimates. Fourth, a joint GLMM is fit conditional on the eigenfunctions estimated in the previous step. We now expound on each step and provide the necessary details.

Step 1: This step prepares the data for fitting local GLMMs in step (2) by organizing them into bins. Binning allows us to borrow strength from neighboring observations, improve stability of parameter estimates, and reduce the probability of singularity problems that arise when observations are zeros or ones. For simplicity we assume that the centers of the bins $\{s_k : k = 1, \ldots, K\}$ are equally spaced, but the methodology can be applied to unequally

spaced grids. Given a bin width w, the bin S_k centered at s_k contains the observations at $\{s_{k-\lceil \frac{w}{2} \rceil}, ..., s_k, ..., s_{k+\lceil \frac{w}{2} \rceil}\}$. On the boundary of the domain, if data is cyclic (e.g., 24-hour activity data that starts and ends at midnight), then bins can be formed by using data from the other end of the domain; otherwise, one can simply use asymmetric bins.

Step 2: This step takes the observed non-Gaussian data and converts them into the linear predictor scale. This is achieved by fitting a sequence of GLMMs of the form

$$g[\mathbb{E}\{Z_{ijk} \mid k : s_k \in S_k\}] = \beta_{0k}^* + a_{ik}^* + b_{ijk}^* \tag{4}$$

for each of the K bins, $S_1, ..., S_K$. The notation $\{Z_{ijk} \mid k : s_k \in S_k\}$ emphasizes that GLMMs are fit separately for each bin. The terms β_{0k}^*, a_{ik}^*, and b_{ijk}^* are the fixed effect intercept, subject random intercept, and subject-visit random intercept, respectively, for the bin S_k. The superscript * distinguishes these "local" quantities from their global counterparts defined in Equation 2. This step only requires fitting a generalized linear mixed effects model with a fixed intercept and nested two-level random intercepts, and is agnostic to the software or the fitting method (e.g., Bayesian, frequentist). The approach naturally deals with missing data at the subject or visit level and borrows strengths across individuals with sparser sampling points or less information (high likelihood of zeros or one). The estimated linear predictors $\widehat{\eta}_{ijk}^* = \widehat{\beta}_{0k}^* + \widehat{a}_{ik}^* + \widehat{b}_{ijk}^*$ are then used in Step 3.

Step 3: Decompose the $\widehat{\eta}_{ijk}^*$ estimates for each bin using MFPCA (Cui et al., 2022b; Di et al., 2009). We use the fast multilevel FPCA method (Cui et al., 2022b) because it works seamlessly with high dimensional data. MFPCA produces smooth estimates of eigenfunctions at subject- and subject-visit levels, denoted by $\widehat{\phi}_l^{(1)}(s)$ ($l = 1, ..., L$) and $\widehat{\phi}_m^{(2)}(s)$ ($m = 1, ..., M$), respectively. The choice of L and M will be discussed in supplementary materials Section S.2. These eigenfunctions will be used in Step 4.

Step 4: Given the estimated eigenfunctions $\{\widehat{\phi}_l^{(1)}(s)\}_{l=1}^L$ and $\{\widehat{\phi}_m^{(2)}(s)\}_{m=1}^M$, we need to estimate the population mean function and scores $\{\xi_{il}\}_{l=1}^L$ and $\{\zeta_{ijm}\}_{m=1}^M$. Conceptually, this can be achieved by fitting a global GLMM that includes a random slope for each eigenfunction. This model uses observations from the entire domain S, hence the word "global" to differentiate it from the local models in Step 2. The global model is

$$g[\mathbb{E}\{Z_{ijk} \mid k : s_k \in S\}] = \beta_0(s_k) + \sum_{l=1}^L \xi_{il} \widehat{\phi}_l^{(1)}(s_k) + \sum_{m=1}^M \zeta_{ijm} \widehat{\phi}_m^{(2)}(s_k). \tag{5}$$

Computationally, however, fitting model 5 using GLMM software packages such as `lme4` (Bates et al., 2015) is infeasible even for moderate data sizes (e.g., 1000 subjects, 10 visits per subject, and 100 sampling points).

To address this problem, we use a Bayesian hierarchical approach, which takes advantage of the simple structure of the full conditional distributions implied by model (5). For example, the full conditional for the random effect $[\zeta_{ijm} \mid \text{others}]$ is proportional to

$$[Z_{ijk} \mid \zeta_{ijm}, \text{others}][\zeta_{ijm} \mid \text{others}] ,$$

which does not have a closed-form, but are relatively easy to sample from using modern Bayesian software such as `JAGS` (Hornik et al., 2003) or `Stan` (Carpenter et al., 2017).

Step 4 is necessary because Step 3 can provide biased estimates of the scores and, consequently, of the subject- and subject/visit-specific latent trajectories. In the simulation section we will provide more insights into this issue.

The R code for implementing the algorithm and a discussion of practical issues such as choosing the bin width and the number of eigenfunctions can be found in supplementary materials.

4 Simulation study

In this section we simulate binary and Poisson multilevel functional data and evaluate how well the proposed method can recover the eigenfunctions, scores, and latent functional trajectories. For both binary and Poisson data, the latent continuous response η_{ijk} is generated from the following model:

$$\eta_{ijk} = \beta_0(s_k) + \sum_{l=1}^{4} \xi_{il}\phi_l^{(1)}(s_k) + \sum_{m=1}^{4} \zeta_{ijm}\phi_m^{(2)}(s_k),$$

where $\beta_0(s_k)$ denotes the intercept at time s_k, $\xi_{il} \sim N(0, \lambda_l^{(1)})$, $\zeta_{ijm} \sim N(0, \lambda_m^{(2)})$, $\{s_k = \frac{k}{K} : k = 0, 1, ..., K\}$, and K is the number of sampling points. We set the true eigenvalues $\lambda_l^{(1)} = 0.5^{l-1}, l = 1, 2, 3, 4$ and $\lambda_m^{(2)} = 0.5^{m-1}, m = 1, 2, 3, 4$. For simulating eigenfunctions, we follow (Di et al., 2009) and consider two cases:

Case 1. Mutually orthogonal bases.

Level 1: $\phi_l^{(1)}(s) = \{\sqrt{2}\sin(2\pi s), \sqrt{2}\cos(2\pi s), \sqrt{2}\sin(4\pi s), \sqrt{2}\cos(4\pi s)\}$.

Level 2: $\phi_m^{(2)}(s) = \{\sqrt{2}\sin(6\pi s), \sqrt{2}\cos(6\pi s), \sqrt{2}\sin(8\pi s), \sqrt{2}\cos(8\pi s)\}$.

Case 2. Mutually nonorthogonal bases between the two levels.

Level 1: same as Case 1.

Level 2: $\phi_1^{(2)}(s) = 1, \phi_2^{(2)}(s) = \sqrt{3}(2s - 1), \phi_3^{(2)}(s) = \sqrt{5}(6s^2 - 6s + 1), \phi_4^{(2)}(s) = \sqrt{7}(20s^3 - 30s^2 + 12s - 1)$.

Multilevel binary functional data are generated as $Z_{ijk} \sim \text{Bernoulli}\{\mu_{ijk}\}$, where $\text{logit}\{\mu_{ijk}\} = \eta_{ijk}$. Multilevel Poisson data are generated as $Z_{ijk} \sim \text{Poisson}\{\mu_{ijk}\}$, where $\log\{\mu_{ijk}\} = \eta_{ijk}$.

For Bernoulli we considered 32 simulation scenarios and for Poisson we considered 26 simulation scenarios. For each scenario we ran 20 simulations. More precisely, in addition to varying the eigenfunctions (2 scenarios), we also examined the effect of sample size $I = 50, 100, 200, 500, 1000$, number of visits $J = 2, 5, 10$, and number of sampling points $K = 100, 200, 500$. In the binary case the proportion of ones was controlled by the choice of the intercept function. We considered a constant intercept function $\beta_0(s) = b_0$ and varied $b_0 = 0, -1.5, -2.5, -3.5$, which corresponds to roughly 50%, 30%, 17%, and 9% of ones. When fitting models in Step 2 we have considered bins of a length corresponding to $w = 2\%, 5\%, 10\%, 15\%, 20\%$ of the number of sampling points. In step 4, we set the number of post warm-up iterations to $1,000$. Running simulations for all combinations of parameter

settings would have resulted in $1,800$ scenarios for Bernoulli and 450 for Poisson regressions, an impossible task given our computational resources. Instead, we varied one parameter at a time while keeping other parameters fixed.

To evaluate the proposed method, we calculate the mean squared error (MSE) of linear predictors across all subject-visit pairs

$$\frac{1}{I \times J \times K} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} [\widehat{\eta}_{ijk} - \eta_{ijk}]^2,$$

and the integrated squared error (ISE) of each eigenfunction:

$$\int_0^1 \{\widehat{\phi}_m^{(l)}(s) - \phi_m^{(l)}(s)\}^2 ds,$$

for $l = 1, 2$ (two levels) and $m = 1, 2, 3, 4$.

4.1   Simulation results

To illustrate how the proposed method performs for recovering eigenfunctions, Figure 2 displays the estimated eigenfunctions from an experimental setting with case 1 basis functions, sample size $I = 1000$, number of visits $J = 10$, number of sampling points $K = 100$, percentage of data used for constructing local bins $w = 5\%$, and fixed effect intercept $b_0 = 0$. Results indicate that the proposed method successfully decomposes variation into two levels and captures the shape of individual eigenfunctions.

We now turn to the estimation of eigenvalues. Figure 3 displays the boxplots of the estimated level one and level two eigenvalues for three different numbers of sampling points, K, while keeping the other experimental settings the same as in Figure 2. The solid black lines indicate the true eigenvalues. Results indicate that eigenvalues are well estimated with little or no bias at level one. At level two, there is a small amount of bias, but it decreases quickly as the number of sampling points K increases.

Table 1 summarises the MSE of linear predictors for binary and Poisson data using case 1 and case 2 basis functions and different number of sampling points. For binary data, results show that as the number of sampling points increases, the proposed method achieves higher accuracy in recovering the latent linear predictors. For Poisson data, MSE decreases as K increases from 100 to 200. At $K = 500$, however, we found that the MCMC algorithm in Step 4 converged successfully for only half of the 20 simulated data sets. For this reason, we do not report the MSE at $K = 500$ for Poisson data. Among the data sets where MCMC converged, the mean MSE at $K = 500$ is less than half of the MSE at $K = 200$. The non-convergence of MCMC may be caused by poor initialization or the sample size being too small relative to the number of sampling points. A more careful look into this issue is left to future work.

Simulation results for when the number of sampling points is fixed and the number of study participants, the number of visits, the bin width, or the fixed effect intercept increases are displayed in Table S.2 to S.8 in the supplementary materials. Indeed, as the sample size (number of subjects) increases, Table S.1 and S.2 show that both the ISE of eigenfunctions
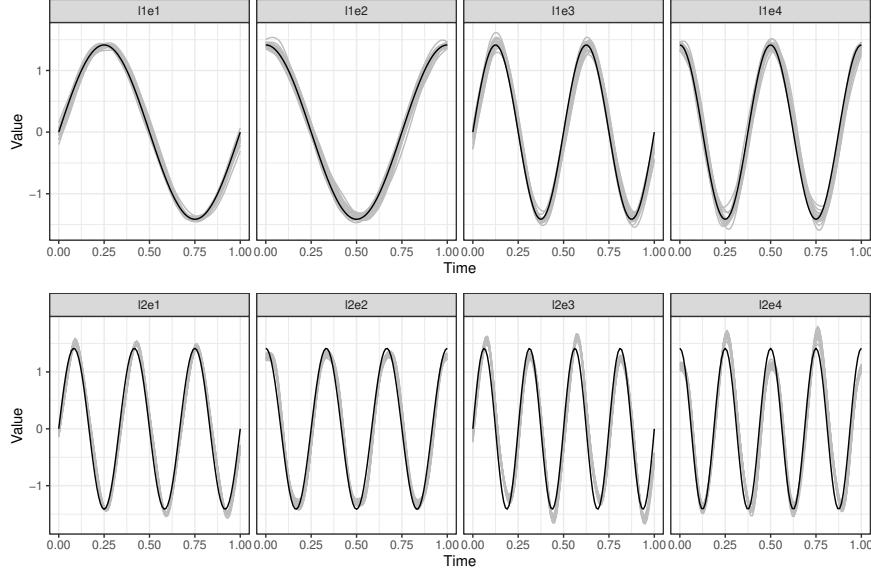
Figure 2: Estimated eigenfunctions for binary data from 20 simulations are plotted in grey. The true functions are shown in black. Data was generated using case 1 basis functions, sample size $I = 1000$, number of visits $J = 10$, number of sampling points $K = 100$, percentage of data used for constructing local bins $w = 5\%$, and fixed effect intercept $b_0 = 0$. The panel title "l1e1" refers to level one eigenfunction 1. Similarly for other eigenfunctions.

	Binary			Poisson		
Basis function	K = 100	K = 200	K = 500	K = 100	K = 200	K = 500
Case 1	0.267	0.145	0.062	0.028	0.014	—
Case 2	0.291	0.169	0.078	0.098	0.057	—

Table 1: MSE of linear predictors averaged over 20 simulations for different numbers of sampling points.

and the MSE of linear predictors generally go down. Table S.3 and S.4 show that a similar decrease is observed when the number of visits increases. As the bin width increases, however, Table S.5 shows that the ISE of eigenfunctions first goes down but then goes up. As explained in detail in supplementary materials Section S.2, this pattern is expected: the decrease is driven by having more data included in each bin, and the increase is driven by using local models, which assumes a constant mean (with fixed and random components) in each bin, to approximate functions that can exhibit substantial curvature in a large bin. Finally, for binary data, Table S.7 and S.8 show that both ISE and MSE go down when the percentages of zeros and ones become more balanced (closer to 50%). Results are similar for case 1 and case 2 basis functions.
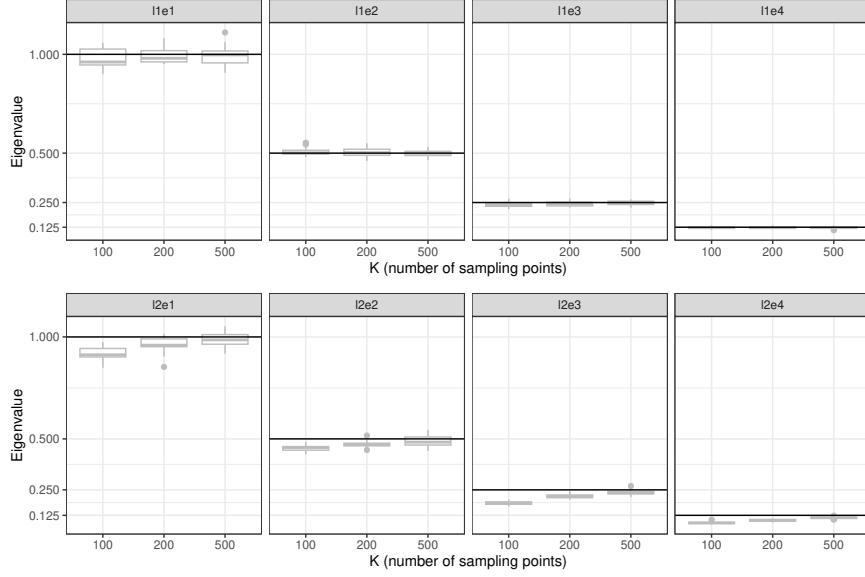
10

Figure 3: Boxplots of estimated eigenvalues for different number of sampling points K for binary data. The solid black lines are true eigenvalues. Data was generated using case 1 basis functions, sample size $I = 1000$, number of visits $J = 10$, number of sampling points $K = 100$, percentage of data used for constructing local bins $w = 5\%$, and fixed effect intercept $b_0 = 0$. The panel title "l1e1" refers to level one eigenfunction 1. Similarly for other eigenfunctions.

4.2   Why Step 4 is necessary

The scores for each subject-visit pair estimated in Step 3 are biased; therefore, in step 4 we re-estimate these scores. Figure 4 illustrates the difference between the score estimates in Steps 3 and 4. The experiment is based on a simulated data set with 1000 subjects, 5 visits per subject, and 500 sampling points along the domain. The top and middle panels plot the true scores for level one and level two eigenfunctions versus the estimated scores from Step 3 and Step 4. For level two scores, we only plot those at visit 1, but the same pattern can be observed for all visits. Results indicate that Step 4 corrects for the bias in the scores estimated from Step 3. The bottom panel displays the true latent continuous signal along with the estimated ones from Step 3 and Step 4 for four randomly chosen subject-visit pairs. The estimates from Step 4 are generally closer to the true trajectories.

4.3   Computation time

Table 2 displays the median computation time of 20 simulation runs for each step across different simulation scenarios. Each number is the median time to finish one simulation run using one computer node on our cluster. We used the cluster because we wanted simulations to finish within reasonable time. There is, however, substantial variability in computational performance within and between nodes, as each assigned node may be of different quality and may be more or less occupied at a particular time. This is why, for example, the median running time of Step 4 at $K = 200$ (1166 minutes) was shorter than that of $K = 100$ (1572
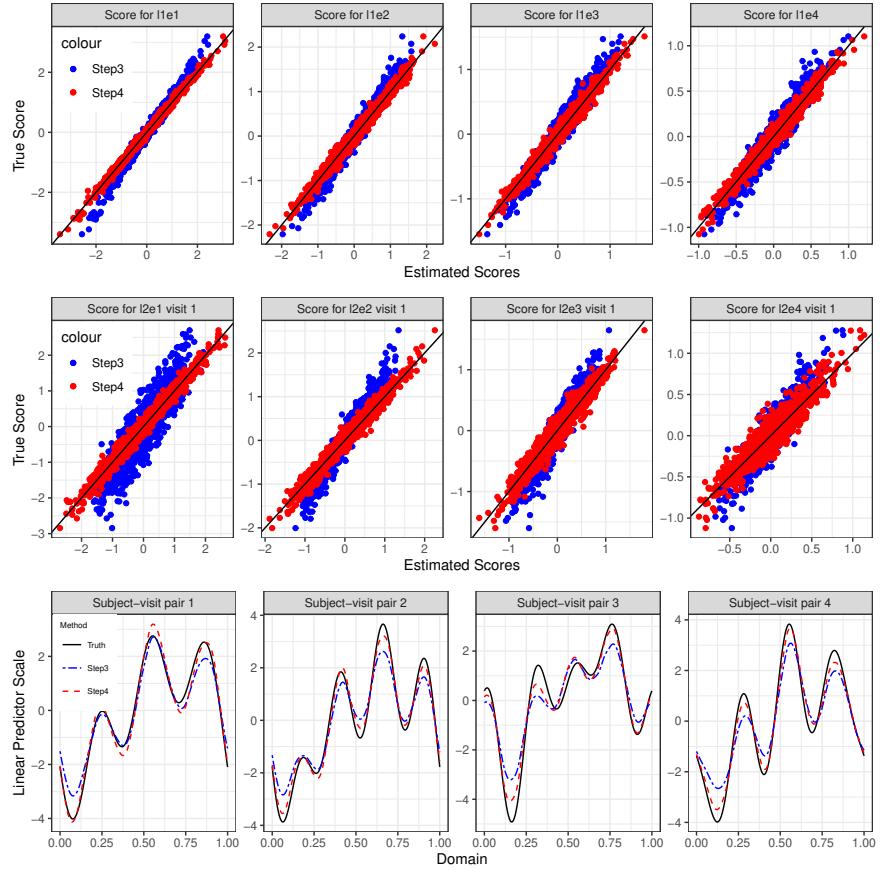
Figure 4: Top and middle: true scores versus those estimated from Step 3 and Step 4 of the GM-FPCA algorithm. For level two, we plot the scores from visit 1 for illustration. Bottom: recovery of the latent continuous signal from Step 3 and Step 4. The plots show four randomly selected subject-visit pairs. Note that the scale of the y-axis varies between subplots.

minutes) for Case 1 simulations with binary data. In fact, we observed that some older nodes can be two to three times slower than the newer ones. Despite these drawbacks, using a computing cluster provides important information about the computational performance of our algorithm. For binary data, Step 1 and Step 2 usually take less than 40 minutes to finish. For Poisson data, fitting local GLMMs in Steps 1 and Step 2 can take longer, but three hours is usually sufficient. These computation time can be further reduced by parallel computing; see Section 5 for details. Step 3 is fast regardless of the simulation setting and its run time can be safely ignored.

The computational bottleneck of GM-FPCA is in step 4, where subject-visit scores are estimated using the full data set. With $1,000$ post warm-up iterations, the median runtime ranges from half a day to 12 days depending on the scenario. The runtime of Step 4 for Poisson data at $K = 500$ is excluded because of the non-convergence issue described in the previous section. See Table S.10 and S.11 from the supplementary materials for runtime results at different sample sizes and number of visits.

Basis Function	K	Binary			Poisson		
		Step 1 and 2	Step 3	Step 4	Step 1 and 2	Step 3	Step 4
Case 1	100	12.16	0.008	1572	9.91	0.459	3231
	200	14.98	0.010	1166	35.7	0.599	11700
	500	36.33	0.024	3557	186.59	1.147	—
Case 2	100	9.01	0.007	678	10.68	0.410	7850
	200	15.66	0.010	1651	35.9	0.574	17836
	500	31.31	0.022	5881	192.98	1.130	—

Table 2: Median computation time (minutes) across 20 simulations. K denotes the number of sampling points. In this experiment, we fixed sample size $I = 1000$, number of visits $J = 10$, and percentage of data used for constructing local bins $w = 5\%$. For binary data, we used fixed effect intercept $b_0 = 0$.

5   Application to NHANES accelerometry data

5.1   Data description

The National Health and Nutrition Examination Survey (NHANES) is a large and ongoing study conducted by the United States Centers for Disease Control and Prevention (CDC) that provides a nationally representative sample of the non-institutionalized US population. NHANES collects data in two-year waves and wearable accelerometers were deployed in the 2003-2004, 2005-2006, 2011-2012, and 2013-2014 waves. Participants of the NHANES 2011-2014 waves were asked to wear a wrist-worn accelerometer, the ActiGraph model GT3X+ device (ActiGraph, Pensacola, FL), consecutively over 9 calendar days. The first and last days are partial days according to the study protocol, resulting in up to 7 full calendar days of data.

NHANES provides physical activity data expressed in Monitor Independent Movement Summary (MIMS) units (John et al., 2019), a summary statistic of the sub-second tri-axial acceleration signal from accelerometers, at three temporal resolutions: day, hour, and minute levels. Briefly, MIMS units are obtained by interpolating the raw acceleration signal to 100Hz, extrapolating the signal if it reaches the device's dynamic range, applying a bandpass filter (0.2–5.0 Hz), calculating the area under the curve (AUC) for each axis, and adding AUC across axes to obtain one value per epoch.

This paper focuses on the minute level data. For data quality control, we define a "good day of data" as the device being worn (the "wear" flag `PAXPREDM` $\in \{1, 2, 4\}$) for at least 95% of the time and not having data quality flags (`PAXFLGSM = ""`), and exclude participants who have no "good day of data". Out of the $14,693$ study participants who accepted to wear an accelerometer, $1,090$ were excluded for poor data quality, leaving $13,603$ for further analysis.

Mortality data is obtained by linking the NHANES data to death certificate records from the National Center for Health Statistics (NCHS) (NCHS, 2022). At the time of this writing, the linked mortality files were available for public use up to December 31, 2019, which is used as the administrative censoring date.

Since the goal of our analysis is to study mortality and its association with physical activity patterns, we excluded $9,149$ subjects under the age of 50, where age was ascertained at the time of their interview with NHANES. We further excluded 9 subjects whose mortality status is missing. The analytic sample for the GM-FPCA analysis thus contains $4,445$ subjects. With up to 7 days of accelerometry data per subject, we have $28,023$ days of wear and $1,440$ (the number of minutes in a day) observations per day. Table S.9 from supplementary materials summarizes the demographics and traditional mortality risk factors of the analytic sample. As mentioned before, we thresholded MIMS at 10.558 (active if MIMS > 10.558 and inactive otherwise) (Karas et al., 2022) to obtain binary active/inactive profiles.

5.2   Analysis results

We applied the GM-FPCA algorithm to the multilevel binary activity data. To evaluate the algorithm's sensitivity to the bin width for fitting local GLMMs, we tried values of 10 minutes, 30 minutes, and 60 minutes. Using a bin width of 10 minutes led to singular fits in 15 bins and harder to interpret level one eigenfunctions (Figure S.3 in the supplementary materials). Increasing the bin width to 30 minutes or 60 minutes eliminated the singularity issue and resulted in more interpretable eigenfunctions. For this reason, we present results based on the bin width of 30 minutes (roughly 2% of the 24 hour interval).

Given the size of the NHANES data, implementing the algorithm efficiently can substantially reduce the computation time. Since there is no dependency between fitting local GLMMs in different bins, we can speed up computation by parallelizing Step 2. Implementing parallel fitting using shared memory space (e.g., on a laptop) is straightforward. However, to implement it using separate memory spaces (e.g., multiple compute nodes on a high performance cluster), we recommend first storing the data in a database. This allows each node to only load data from one bin at a time, which substantially reduces per-node memory allocation. In our analysis, each bin uses only 2% of the full data set because we chose a bin width of 30 minutes. For the NHANES data, parallelizing Step 2 saved 5 days of computation.

The estimated top six eigenfunctions from each level from Step 3 are plotted in Figure 5 (subject-level variability) and Figure 7 (subject-visit-level variability). Together, subject-level (level one) variability accounts for 47% of the total variability in the latent space, while the subject-visit-level (level two) accounts for the other 53%.
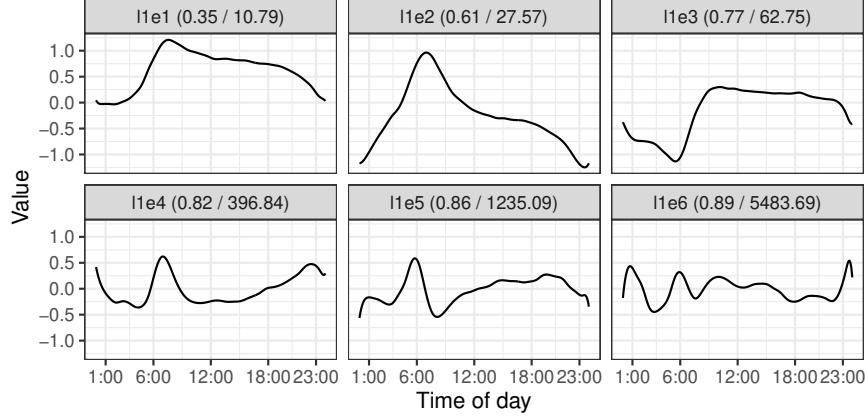

Figure 5: Top 6 level one eigenfunctions from the dichotomized NHANES data. Subplot titles indicate the eigenfunction, cumulative fraction of variance explained within level one, and the sum of squared second-order differences (SSSOD) statistic.

The title for each eigenfunction subplot indicates the level, eigenfunction number, cumulative fraction of variance explained within the level, and sum of square second-order differences (SSSOD, see supplementary materials Section S.2 for the definition) for each function. For example, the panel in the first row and second column of Figure 5 is labeled "l1e2 (0.61 / 27.57)". This indicates that this is the second eigenfunction (e2) at level one (l1), that the first two level one eigenfunctions ("l1e1" and "l1e2") explain 61% of the variability in level one, and that the SSSOD = 27.57. We do not propose a rigorous analysis of the SSSOD measure at this time, but point out that it increases substantially with the observed complexity of the functions. For example, the SSSOD of eigenfunctions one and two are one or two orders of magnitude smaller compared to that of eigenfunctions five and six, and these differences track with the visual inspection of the plots. Indeed, the first three eigenfunctions are relatively easy to interpret because they have smaller variation and together they explain 77% of the variation in level one. In contrast, the other eigenfunctions become increasingly difficult to interpret and together explain less than 23% of the variability at this level.

Recall that subject-level (level one) eigenfunctions capture the main directions of variation of the average within-subject physical activity trajectory. A closer inspection of the panel in the first row and column of Figure 5 (l1e1) indicates that the first eigenfunction is: (1) close to 0 from midnight till 4AM and from 11PM till midnight; and (2) positive from 4AM to 11PM, with a peak around 8AM. Therefore, a study participant with a positive score on this component would, on average, move more during the day and less during the night. To see this more intuitively, we first take a subset of subjects whose "l1e1" scores are in the highest and lowest 5% of the data set. Then for each subject in the subset, we smooth their binary activity data day by day using generalized additive models with cyclic cubic splines

and average the smooths across days, so that each subject has a smoothed and averaged activity profile. We further randomly sample five subjects from each of the two groups, the highest and lowest 5% "l1e1" score groups, and plot their smoothed activity profiles using shallow lines in the left panel of Figure 6. The solid lines in the plot are the average of all subjects from each of the two groups. The plot shows that subjects with high "l1e1" scores (red lines) have higher activity level during the day than those with low "l1e1" scores (blue lines). Similar plots for "l1e2" and "l1e3" are displayed in the middle and right panels of Figure 6 and can be interpreted similarly.
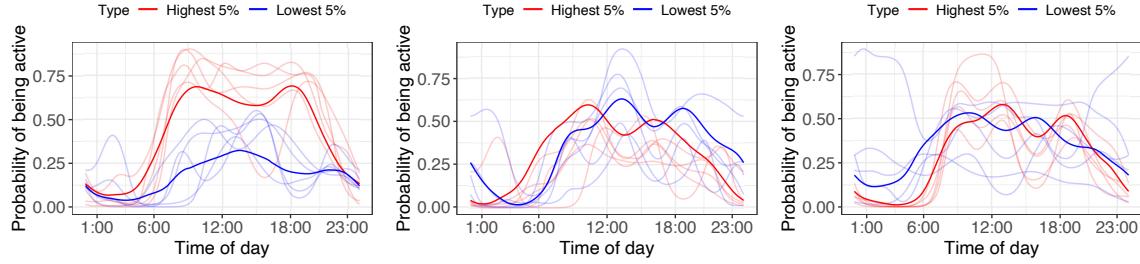

Figure 6: From left to right are plots for level one eigenfunction 1, 2, and 3. The thick lines are the average (across days and subjects) of the smoothed binary activity profile for those with the highest (red) and lowest (blue) 5% percent of FPC scores. The shallow lines are the average (across days) smoothed binary activity profile from 5 randomly selected subjects from each of the two groups.

Figure 7 displays information structured similarly as Figure 5, though it focuses on the day-to-day variation of physical activity patterns within subjects. Therefore, the interpretation is quite different from level one eigenfunctions, but the structure is consistent with our intuition: quantify the natural day-to-day variability of physical activity patterns. While level two eigenfunctions explain a similar proportion of total variability as level one eigenfunctions, the variability within level two is much more spread out across eigenfunctions. For example, the first six eigenfunctions at level two explain only 41% of the variability at this level, whereas the first six eigenfunctions at level one explain 89% of the variability at that level. This indicates that the patterns of day-to-day variability are more heterogeneous than those of subject-to-subject variability. While these findings are intuitive, it is important to see agreement between data-driven results and what is known/expected about human activity. Moreover, this approach provides quantification of variability at each level, as well as visualizations of the most important directions of variation.

We conduct Step 4 of the GM-FPCA algorithm by fitting the global model (5) conditional on the top five eigenfunctions from both level one and level two using the Bayesian approach. To keep the method simple while allowing the fixed-effect intercept to depend on time, we use the fixed effects estimates $\widehat{\beta_0}(s)$ obtained in Step 3 and plug them into the Step 4 model, where they are treated as fixed. This is not strictly necessary and the model could be refit jointly. We leave this problem for future analyses.

In spite of all dimension reduction efforts, the global model in Step 4 still raises extraordinary computational challenges. Indeed, recall that there are $4,445$ study participants, $28,023$ total days of wear, and $1,440$ obsevations per day. Therefore, the model has a total
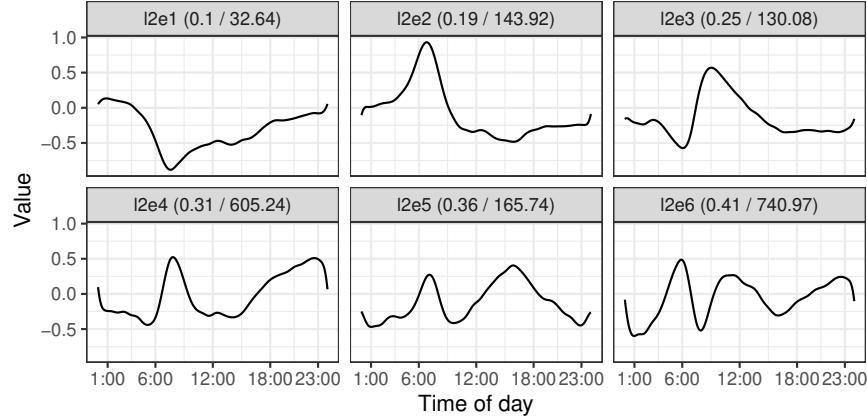
Figure 7: Top 6 level two eigenfunctions from the dichotomized NHANES data. "l2e1" refers to level two eigenfunction 1. Similar naming convention applies to other eigenfunctions.

of $162,340$ ($4445 \times 5 + 28023 \times 5$) parameters that need to be updated at every MCMC iteration. Here we use the same technique described by Leroux et al. (2023) and divide the data set into 7 subsets containing 635 subjects each. In the case of single level GFPCA, Leroux et al. (2023) showed that this approach provides close to identical results compared to joint fitting on the entire data set. Of course, this could be done jointly, but we suggest balancing what is reasonable with what is truly necessary.

After fitting the global model on each of the subsets, we obtain score estimates. We now investigate whether the subject-specific scores (level one) are associated with time to death after accounting for known predictors of mortality. After further excluding 127 subjects who have missing values in at least one of the covariates (see supplementary materials Table S.9 for the exact number of missingness for each covariate), we fit Cox proportional hazards models for the time to death outcome using two sets of covariates. Model 1 only includes traditional risk factors for mortality such as age, gender, and BMI; Model 2 further adds level one scores. Table 3 displays the results from the two models. In addition to hazard ratios and 95% confidence intervals, we also show the concordance and the integrated Brier scores for both models.

From model 2, we see that the scores of the first three level one eigenfunctions are strongly predictive of mortality even after accounting for traditional risk factors. For example, the hazard ratio for "l1e1" is 0.85 (95% CI [0.79, 0.92]), suggesting that subjects with higher scores on the level one eigenfunction 1 have lower risk of death when keeping other covariates in the model fixed. We have shown in the left panel of Figure 6 that subjects with higher "l1e1" scores tend to be more active during the day and less active during the night compared to those with lower "l1e1" scores, so the result agrees with our knowledge of what a healthy lifestyle should look like.

Similar to the simulation study, we recorded the computation time of GM-FPCA on the NHANES data. The median time it takes to fit a local GLMM is approximately 5 minutes, so a sequential analysis would have taken more than 5 days, but we avoided that by using the parallel fitting approach described earlier. Step 3 took only 8 seconds. As mentioned above, to run Step 4 we split the full data set into 7 subsets of 635 subjects each, and estimated

17

scores separately. With 1200 total iterations (200 warm-up, 1000 sampling), the median computation time of Step 4 across 7 subsets was 130 hours.

Table 3: Analysis of time to mortality in NHANES using Cox proportional hazards models with two sets of covariates.

	Hazard Ratio (95% CI)	
	Model 1	Model 2
Age	1.09*** (1.08, 1.10)	1.09*** (1.08, 1.10)
Gender: Female	0.72*** (0.61, 0.84)	0.78** (0.66, 0.92)
BMI (Ref: Normal)		
Underweight	1.33 (0.88, 2.03)	1.36 (0.90, 2.07)
Overweight	0.63*** (0.51, 0.76)	0.62*** (0.51, 0.75)
Obese	0.63*** (0.51, 0.77)	0.59*** (0.48, 0.72)
Race (Ref: Non-Hispanic White)		
Non-Hispanic Black	0.80* (0.66, 0.97)	0.83 (0.68, 1.01)
Mexican American	0.63** (0.45, 0.88)	0.65* (0.47, 0.91)
Non-Hispanic Asian	0.50*** (0.34, 0.72)	0.51*** (0.36, 0.74)
Other Hispanic	0.62** (0.45, 0.86)	0.65* (0.47, 0.90)
Other	1.12 (0.66, 1.88)	1.13 (0.67, 1.91)
Overrall Health (Ref: Excellent)		
Very good	0.89 (0.64, 1.24)	0.92 (0.66, 1.28)
Good	0.75* (0.60, 0.95)	0.77* (0.61, 0.97)
Fair	1.26* (1.05, 1.52)	1.28* (1.06, 1.54)
Poor	1.85*** (1.41, 2.43)	1.92*** (1.46, 2.53)
Diabetes (Ref: No)		
Borderline	0.82 (0.53, 1.26)	0.83 (0.54, 1.28)
Yes	1.18 (0.99, 1.41)	1.18 (0.99, 1.40)
Arthritis: Yes	0.89 (0.75, 1.04)	0.88 (0.75, 1.04)
CHF: Yes	1.48*** (1.18, 1.86)	1.43** (1.14, 1.80)
CHD: Yes	1.15 (0.91, 1.46)	1.17 (0.92, 1.48)
Heart Attack: Yes	1.20 (0.93, 1.53)	1.21 (0.95, 1.56)
Stroke: Yes	1.16 (0.93, 1.44)	1.11 (0.89, 1.38)
Cancer: Yes	1.28** (1.08, 1.52)	1.27** (1.07, 1.51)
Mobility Problem: Any difficulty	2.11*** (1.76, 2.51)	2.06*** (1.72, 2.45)
Alcohol Consumption (Ref: Never drinker)		
Former drinker	1.26 (0.99, 1.61)	1.26 (0.98, 1.60)
Moderate drinker	0.83 (0.65, 1.07)	0.88 (0.69, 1.12)
Heavy drinker	1.25 (0.84, 1.85)	1.33 (0.90, 1.98)
Not available	1.10 (0.77, 1.57)	1.12 (0.78, 1.60)
Cigarette Smoking (Ref: Never)		
Former	1.06 (0.89, 1.27)	1.08 (0.90, 1.28)
Current	1.30* (1.02, 1.65)	1.29* (1.01, 1.64)
l1e1		0.85*** (0.79, 0.92)
l1e2		1.14* (1.03, 1.27)
l1e3		0.89** (0.81, 0.97)
l1e4		0.94 (0.81, 1.09)
l1e5		0.93 (0.78, 1.10)
Concordance	0.806	0.810
Integrated Brier score	0.068	0.066

Note:	*p<0.05; **p<0.01; ***p<0.001

6 Discussion

Motivated by the need to analyze large binary multilevel physical activity data, we have developed the GM-FPCA method. While the initial motivation comes from analyzing multilevel binary functional data, our method is general and can handle other types of non-Gaussian multilevel functional outcomes such as categorical or count data. As demonstrated through extensive simulation studies, GM-FPCA provides reasonable estimates of eigenfunctions and scores for a large number of scenarios with varying number of subjects, visits per subject, and number of sampling points. The scalability of the GM-FPCA method can be seen through both simulation studies and the NHANES 2011-2014 accelerometry data analysis, where the data set contains $4,445$ subjects, $28,023$ days of wear, and $1,440$ observations per day.

While GM-FPCA can scale up to the size of large accelerometry data such as NHANES, we acknowledge that Step 4 of the algorithm still takes a long time. A compromise between accuracy and efficiency could be achieved by downsampling the domain in Step 4. A preliminary attempt in this direction is described in the supplementary materials, while a more careful study is left to future work. The other direction of future research is to incorporate covariates in Step 2 when fitting local GLMMs. This would offer the advantage of regressing out the variability induced by known covariates such as age and gender.

To summarize, we developed a novel statistical method called GM-FPCA for analyzing generalized (non-Gaussian) multilevel functional data, and confirmed activity patterns that are associated with higher mortality through an analysis of the binary NHANES 2011-2014 accelerometry data. These findings are easy to interpret and has the potential for wide public health impact. We believe that GM-FPCA would prove indispensable for researchers working with large wearable device data. Therefore, we make the code of GM-FPCA available both in the paper and as part of the R package `fastGFPCA`, available at `https://github.com/julia-wrobel/fastGFPCA`.

Supplementary Material

S.1 Implementation

The code below (Listing. 1) provides the implementation of Steps 1-3 of the algorithm for binary data. The code is short and concise, and allows for easy implementation and modification.

Listing 1: R code for the GM-FPCA algorithm for binary data.

```
1 library(tidyverse)
2 library(lme4)
3 library(refund)
4 dat <- read.csv("...")
5 fit_ls  <- vector(mode = "list", length = S+1)
6 for(j in 0:S){
7    # Step 1: Create data bins.
8    sind_j <- (j - ceiling(bin_len/2)):(j + ceiling(bin_len/2)) %% (S+1)
9    df_j <- df %>% filter(time %in% sind_j)
10
11   # Step 2: Fit local GLMMs; extract linear predictor estimates.
12   fit_j <- glmer(
13     value ~ 1 + (1|id) + (1|id:visit),
14     data = df_j, family = binomial,
15   )
16   fit_ls[[j+1]] <- list(
17     "id_visit" = rownames(coef(fit_j)$"id:visit"),
18     "eta_i" = rep(coef(fit_j)$"id"[[1]], each = J) +
19       coef(fit_j)$"id:visit"[[1]],
20     "time" = j
21   )
22 }
23
24 # Step 3: Run MFPCA
25 fit_df <- bind_rows(fit_ls)
26 id_visit <- strsplit(fit_df$id_visit, ":")
27 fit_df$id <- sapply(id_visit, function(x) as.integer(x[1]))
28 fit_df$visit <- sapply(id_visit, function(x) as.integer(x[2]))
29 fit_df <- arrange(fit_df, id, visit, time)
30 fit_df_wide <- pivot_wider(
31     select(fit_df, c(id, visit, time, eta_i)),
32     id_cols = c(id, visit), names_from = time,
33     values_from = eta_i, names_prefix = "T"
34 )
35 mfpca.bin <- mfpca.face(
36     as.matrix(select(fit_df_wide, -c(id, visit))),
37     id = fit_df_wide$id,
38     visit = fit_df_wide$visit,
```

```
39    argvals = 0:S,
40    pve = 0.95
41  )
```

This code below (Listing. 2) provides the Stan code for implementing Step 4 of the algorithm. We only provide code for binary outcomes, but it can easily be modified for Poisson outcomes.

Listing 2: Stan code for Step 4 of the algorithm.

```
1  functions {
2    real partial_sum_lpmf(array[] int y, int start, int end, vector eta) {
3      return bernoulli_logit_lupmf(y | eta[start:end]);
4    }
5  }
6  data {
7    int<lower=0> N;    // number of individuals
8    int<lower=0> J;    // number of repeated functions within individuals
9    int<lower=0> S;    // number of observations per function
10   int<lower=0> K;    // number of level 1 eigenfunctions
11   int<lower=0> L;    // number of level 2 eigenfunctions
12   int<lower=0> Nobs; // number of total observations
13   array[Nobs] int<lower=0,upper=1> y; // outcome (binary functional data)
14   matrix[K,S+1] efuncs_l1; // level 1 eigen functions
15   matrix[L,S+1] efuncs_l2; // level 2 eigen functions
16   int<lower=1> grainsize;
17 }
18 parameters {
19   real beta_0;    // global avereage
20   matrix[N, K] xil1;    // level 1 random effects
21   matrix[N*J, L] xil2;    // level 2 random effects
22
23   vector<lower=0>[K] sigmal1sq; // var of the l1 random intercept
24   vector<lower=0>[L] sigmal2sq; // var of the l2 random intercept
25 }
26 transformed parameters {
27   matrix[N, K] xil1_sc;    // level 1 random effects
28   matrix[N*J, L] xil2_sc;    // level 2 random effects
29   for(k in 1:K){
30       xil1_sc[,k] = xil1[,k] * sqrt(sigmal1sq[k]);
31   }
32   for(k in 1:L){
33       xil2_sc[,k] = xil2[,k] * sqrt(sigmal2sq[k]);
34   }
35 }
36 model {
37   int inx;
38   int inx_ij;
39   vector[Nobs] eta_vec;
40   matrix[N, S+1] eta_mat_i;
41   matrix[N*J, S+1] eta_mat_ij;
42
```

3

```
43    // priors
44    beta_0 ~ normal(0, 10);
45    to_vector(xil1) ~ normal(0,1);
46    to_vector(xil2) ~ normal(0,1);
47
48    sigmal1sq ~ inv_gamma(1, 1);
49    sigmal2sq ~ inv_gamma(1, 1);
50
51    eta_mat_i = xil1_sc * efuncs_l1;
52    eta_mat_ij = xil2_sc * efuncs_l2;
53    inx = 1;
54    for(i in 1:N){
55            for(j in 1:J){
56                    eta_mat_ij[inx,] = eta_mat_ij[inx,] + eta_mat_i[i,];
57                    inx = inx + 1;
58            }
59    }
60    eta_vec = to_vector((eta_mat_ij)');
61
62   target += reduce_sum(partial_sum_lpmf, y, grainsize, eta_vec + beta_0);
63 }
```

S.2 Practical considerations

S.2.1 Choosing the bin width

In practice, if the bin width is too small, there may not be enough variation between subjects or between visits among the same subject to fit the local GLMM model in certain bins; convergence or singularity issues can arise. The problem is exacerbated when the number of visits per subject is small (e.g., 2-5). At the other extreme when bins are too large, the curvature of the latent process may not be well captured. This is because each local GLMM model essentially estimates a mean (with fixed and random components) in each bin and for each function. If the functions exhibit substantial curvature in a large bin, then the model would not provide a good fit to the data. From a practical standpoint, we have found that one does not need to find the perfect bin width to obtain reasonable results. Indeed, there are many choices that provide similar results and a good strategy is to try a few different bin widths.

To inform the bin selection process, we developed the R function `SummarizeVariability()`, which summarizes both subject- and subject-visit level variability across all bins for a given bin width. For binary data it calculates the within-bin percentages of subject- and subject-visit level units whose observations are all zeros or ones. In the NHANES data, choosing a bin width that ensured that less than 50% of the units at the subject/visit level are all zeros or ones was enough to avoid GLMM fitting problems (no convergence warnings). This criteria can be further relaxed as the model does not need to converge in every single bin. However, when the model fails in many adjacent bins, we recommend increasing the bin width and carefully analyzing the structure of the data in that region of the functional domain. Choosing the bin width can be assisted by careful analysis of what works and what fails as well as by conducting a reasonable sensitivity analysis.

S.2.2 Choosing the number of eigenfunctions

The most common approach for choosing the number of eigenfunctions is based on the cumulative percentage of variance explained. While this strategy is effective, it sometimes leads to selecting too many eigenfunctions. Eigenfunctions that rank lower do not contribute much to the variance explained, slow down Step 4, and are more challenging to interpret because they typically show more oscillation compared to higher-ranked eigenfunctions. To inform the choice of the eigenfunctions, we propose to use the sum of squared second-order differences (SSSOD), which is the discrete analog of the integrated squared second-order derivative (ISSOD). A higher SSSOD corresponds to more oscillations in an eigenfunction, which reduces its interpretability.

$$
\begin{aligned}
\text{ISSOD} &= \int_S [f''(s)]^2 ds \,, \\
\text{SSSOD} &= K^2 \sum_{k=1}^{K-2} [f(s_{k+2}) - 2f(s_{k+1}) + f(s_k)]^2 \,.
\end{aligned}
$$

S.3 More on Step 4

S.3.1 Prior distribution

We perform a sensitivity analysis to examine whether eigenvalues are sensitive to the choice of prior distributions for the variance parameters, σ_l^2 and σ_m^2, of the random effects. We focus on binary data and fix the sample size $I = 500$, number of visits $J = 10$, number of sampling points $K = 100$, percentage of data used for constructing local bins $w = 5\%$, and fixed effect intercept $b_0 = 0$. Besides Inv-Gamma$(1, 1)$, which was used in simulation and real data analysis, we also considered Inv-Gamma$(0.001, 0.001)$, Uniform$(0, 20)$, and Half-Cauchy$(0, 10)$. The last two prior distributions were suggested by Gelman (2006). Figure S.1 shows the boxplots of the posterior mean of eigenvalues over 20 repeats under different priors. Results show that the posterior means of the eigenvalues are not sensitive to the choice of priors. We repeated the experiment using $J = 2$ visits instead of 10 and observed similar results. An explanation for the "insensitivity" of the posterior to the parameters of the inverse-gamma distribution is provided in (Crainiceanu et al., 2005). Let B_b denote the scale parameter for the inverse-gamma prior and b denote the corresponding score vector. Crainiceanu et al. (2005) showed that as long as B_b is small compared to $\|b\|^2/2$, the posterior will not be sensitive to the choice to B_b.
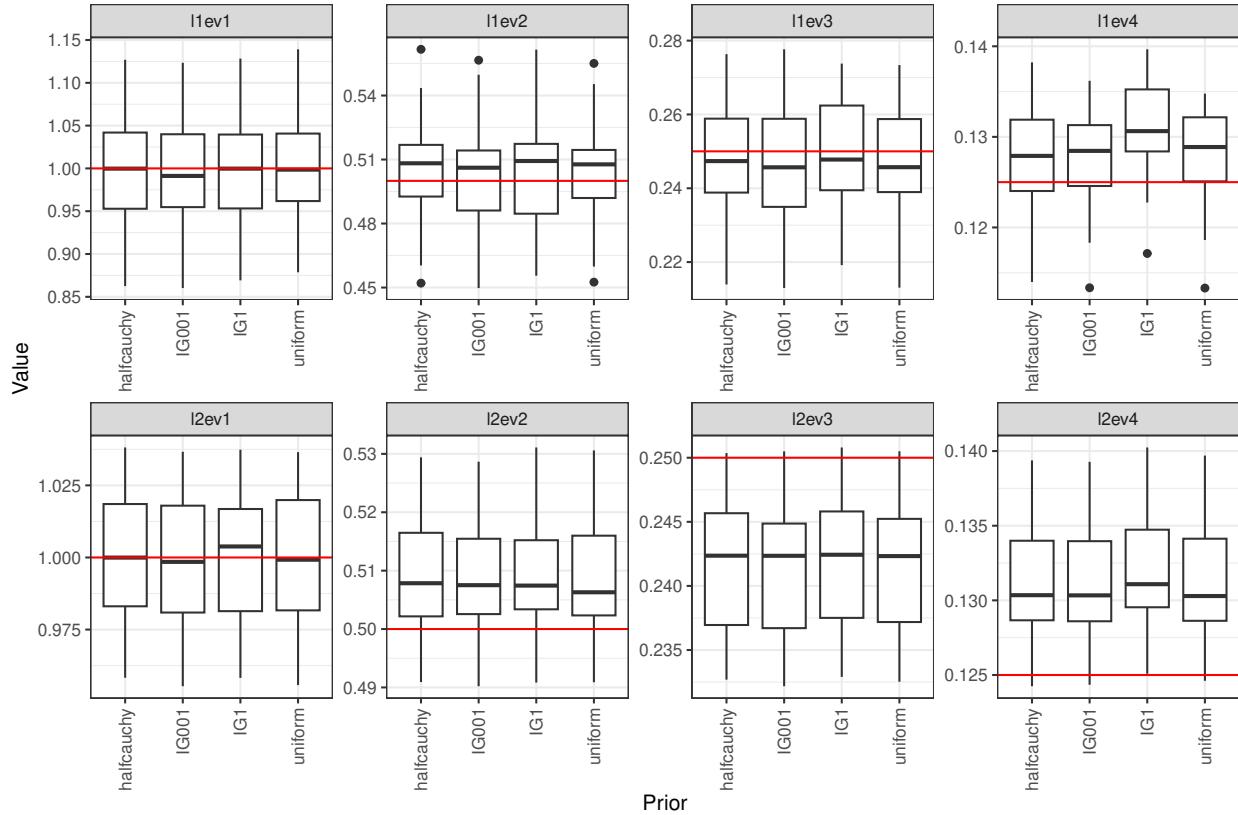
Figure S.1: Boxplots of the posterior mean of eigenvalues over 20 repeats under different priors. Red horizontal lines indicate the truth.

S.3.2 Downsampling in Step 4

This section explores downsampling along the functional domain in Step 4 to speed up the estimation of scores. We emphasize that downsampling is not needed in Step 2 because fitting GLMMs in local bins is fully parallelizable; it is also not needed in Step 3 because MFPCA (Cui et al., 2022b) is highly scalable.

We use a simulated data set so that we know the ground truth. The data set was simulated from $I = 1000$, $J = 5$, $K = 500$, $\beta_0 = 0$, and the response variable is binary. We used 2% of the data to construct local bins. After Step 3, we have the estimated eigenfunctions at all 500 sampling points. To downsample, we take 100 and 250 equally spaced sampling points from the 500 points, and estimate scores using only eigenfunctions and binary responses at the subset of points. Figure S.2 shows the estimated scores against the truth for l1e1 (level one eigenfunction 1), l2e1, l1e4, and l2e4. Scores from Step 3 are also included for comparison. We can see that for top eigenfunctions (l1e1 and l2e1), even downsampling at a rate of 20% (i.e., taking only 100 sampling points) does not drastically lower the quality of the scores and still provides clear debiasing benefit compared to scores obtained from Step 3. For bottom eigenfunctions (l1e4 and l2e4), however, a higher downsampling rate such as 50% is needed to secure reasonable score estimates. This is particularly true for level 2 eigenfunctions. In summary, if one is only interested in using the top one or two eigenfunctions for downstream analysis, we would recommend an aggressive downsampling rate in Step 4 to significantly cut computational cost.
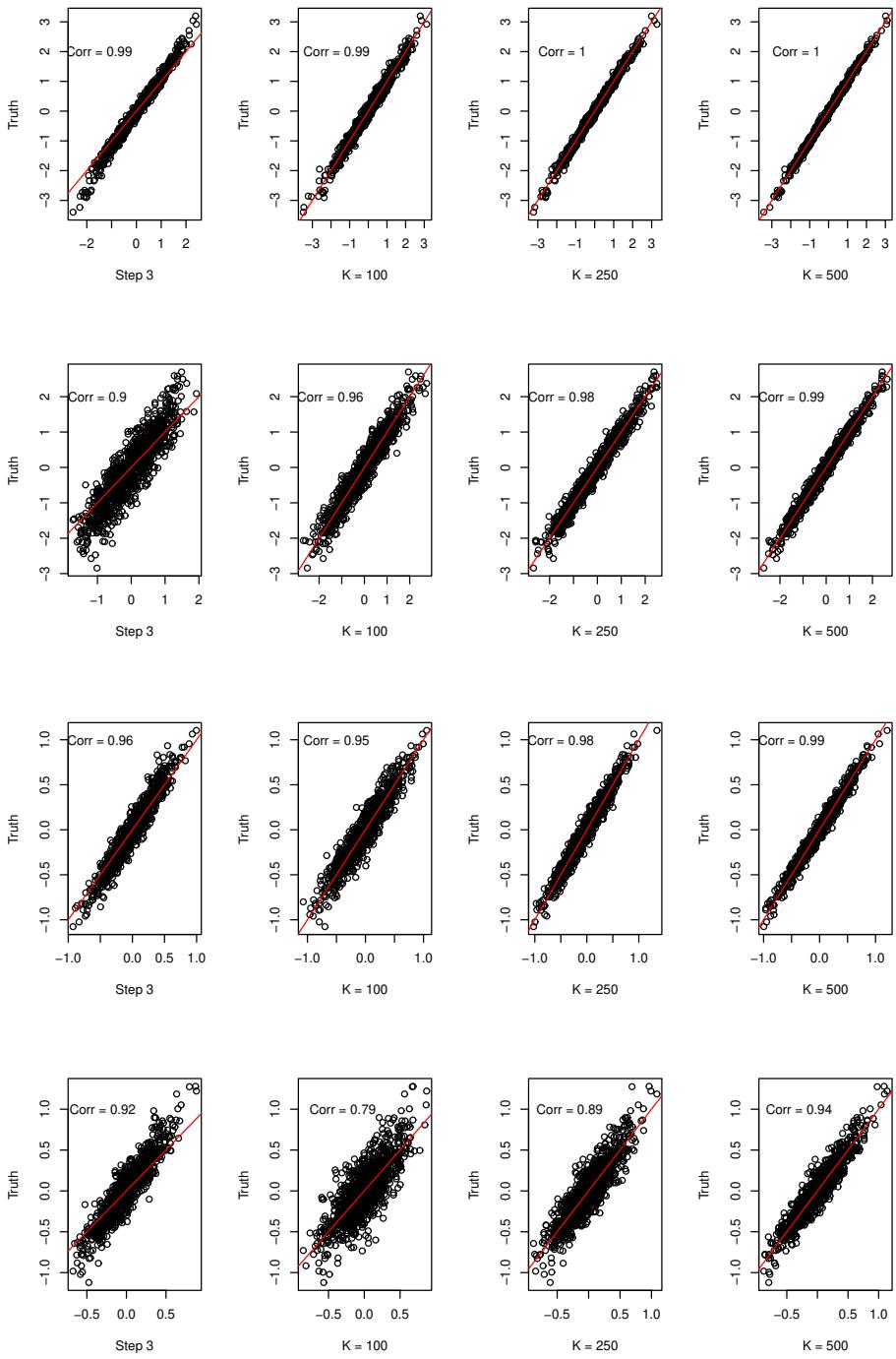
Figure S.2: True scores versus those estimated using all sampling points (K=500) and using a subset of points (K=100, 200). Scores obtained from Step 3 are added for comparison. Top to bottom rows: l1e1, l2e1, l1e4, l2e4.

S.4 More simulation results

S.4.1 Effect of Sample Size

In this experiment, we vary the sample size (number of study participants) while fixing other parameters at the following values: number of visits $J = 10$, number of sampling points

$K = 100$, and percentage of data used for constructing local bins $w = 5\%$. For binary data, we used fixed effect intercept $b_0 = 0$. Table S.1 and S.2 display the ISE of eigenfunctions and MSE of linear predictors for different data distributions and basis functions as a function of sample size $I = 50, 100, 200, 500, 1000$. Results indicate that ISE decreases quickly with increased sample size and that linear predictors are estimated increasingly well when the number of study participants increases.

Basis function	Eigenfunction	Binary					Poisson				
		N = 50	N = 100	N = 200	N = 500	N = 1000	N = 50	N = 100	N = 200	N = 500	N = 1000
Case 1	l1e1	0.072	0.042	0.017	0.009	0.006	0.083	0.027	0.024	0.012	0.007
	l1e2	0.127	0.063	0.022	0.017	0.008	0.115	0.056	0.035	0.015	0.009
	l1e3	0.173	0.072	0.046	0.031	0.022	0.190	0.083	0.032	0.027	0.021
	l1e4	0.215	0.084	0.055	0.030	0.024	0.180	0.062	0.028	0.025	0.022
	l2e1	0.052	0.042	0.044	0.034	0.041	0.035	0.039	0.041	0.037	0.036
	l2e2	0.080	0.054	0.053	0.039	0.044	0.037	0.042	0.042	0.036	0.035
	l2e3	0.110	0.092	0.082	0.084	0.080	0.077	0.073	0.068	0.068	0.066
	l2e4	0.133	0.111	0.097	0.092	0.088	0.081	0.077	0.071	0.071	0.069
Case 2	l1e1	0.051	0.043	0.012	0.005	0.004	0.066	0.051	0.013	0.005	0.003
	l1e2	0.114	0.066	0.025	0.017	0.009	0.123	0.067	0.022	0.008	0.005
	l1e3	0.160	0.067	0.028	0.021	0.010	0.160	0.062	0.027	0.014	0.011
	l1e4	0.189	0.101	0.050	0.029	0.015	0.130	0.051	0.018	0.009	0.007
	l2e1	0.035	0.023	0.019	0.015	0.014	0.021	0.005	0.003	0.002	0.002
	l2e2	0.045	0.023	0.023	0.016	0.015	0.022	0.012	0.004	0.003	0.002
	l2e3	0.058	0.042	0.034	0.030	0.027	0.019	0.019	0.010	0.009	0.007
	l2e4	0.073	0.049	0.043	0.036	0.034	0.029	0.022	0.018	0.018	0.016

Table S.1: ISE of eigenfunctions averaged over 20 repeats for different sample sizes. "l1e1" stands for level 1 eigenfunction 1. Similarly for other eigenfunctions.

Basis function	Binary					Poisson				
	N = 50	N = 100	N = 200	N = 500	N = 1000	N = 50	N = 100	N = 200	N = 500	N = 1000
Case 1	0.292	0.281	0.271	0.268	0.267	0.042	0.034	0.031	0.029	0.028
Case 2	0.313	0.297	0.294	0.291	0.291	0.102	0.103	0.095	0.096	0.098

Table S.2: MSE of linear predictors averaged over 20 repeats for different sample sizes.

S.4.2 Effect of the number of visits

In this experiment, we vary the number of visits while fixing other parameters at the following values: sample size $I = 1000$, number of sampling points $K = 100$, and percentage of data used for constructing local bins $w = 5\%$. For binary data, we used fixed effect intercept $b_0 = 0$. Table S.3 and S.4 display the ISE of eigenfunctions and MSE of linear predictors for different data distributions and basis functions as a function of the number of visits $J = 2, 5, 10$. Results indicate that both ISE and MSE are estimated increasingly well when the number of visits increases.

Basis function	Eigenfunction	Binary			Poisson		
		$J = 2$	$J = 5$	$J = 10$	$J = 2$	$J = 5$	$J = 10$
Case 1	l1e1	0.011	0.007	0.006	0.010	0.006	0.007
	l1e2	0.026	0.009	0.008	0.013	0.010	0.009
	l1e3	0.208	0.023	0.022	0.025	0.022	0.021
	l1e4	1.726	0.091	0.024	0.070	0.026	0.022
	l2e1	0.039	0.044	0.041	0.049	0.035	0.036
	l2e2	0.053	0.048	0.044	0.050	0.034	0.035
	l2e3	0.098	0.079	0.080	0.073	0.070	0.066
	l2e4	0.118	0.089	0.088	0.079	0.074	0.069
Case 2	l1e1	0.009	0.004	0.004	0.007	0.005	0.003
	l1e2	0.040	0.012	0.009	0.008	0.006	0.005
	l1e3	0.398	0.014	0.010	0.019	0.010	0.011
	l1e4	1.563	0.401	0.015	0.281	0.011	0.007
	l2e1	0.024	0.017	0.014	0.004	0.001	0.002
	l2e2	0.028	0.018	0.015	0.006	0.002	0.002
	l2e3	0.041	0.029	0.027	0.010	0.007	0.007
	l2e4	0.047	0.036	0.034	0.019	0.017	0.016

Table S.3: ISE of eigenfunctions averaged over 20 repeats for different numbers of visits.

Basis function	Binary			Poisson		
	$J = 2$	$J = 5$	$J = 10$	$J = 2$	$J = 5$	$J = 10$
Case 1	0.467	0.298	0.267	0.057	0.035	0.028
Case 2	0.398	0.312	0.291	0.120	0.097	0.098

Table S.4: MSE of linear predictors averaged over 20 repeats for different numbers of visits.

S.4.3   Effect of bin width

In this experiment, we vary the bin width while fixing other parameters at the following values: sample size $I = 1000$, number of visits $J = 10$, and number of sampling points $K = 100$. For binary data, we used fixed effect intercept $b_0 = 0$. Table S.5 and S.6 display the ISE of eigenfunctions and MSE of linear predictors for different data distributions and basis functions as a function of the bin width, expressed as a percentage of the number of total sampling points. We can see that with the exception of Poisson data using case 2 eigenfunctions, the MSE linear predictors first goes down and then goes up. Similar trend is observed for the ISE of many eigenfunctions too. To understand what happened to Poisson data using case 2 eigenfunctions, we further conducted experiments at $w = 3\%$ and 4%. The MSE of linear predictors are 0.092 and 0.093, respectively, which suggest that $w = 2\%$ is likely the best bin width for this scenario.

Basis function	Eigenfunction	Binary					Poisson				
		W = 2%	W = 5%	W = 10%	W = 15%	W = 20%	W = 2%	W = 5%	W = 10%	W = 15%	W = 20%
Case 1	l1e1	0.006	0.006	0.006	0.006	0.006	0.007	0.007	0.007	0.007	0.007
	l1e2	0.010	0.008	0.008	0.007	0.007	0.009	0.009	0.008	0.007	0.007
	l1e3	0.023	0.022	0.021	0.020	0.020	0.022	0.021	0.021	0.020	0.020
	l1e4	0.029	0.024	0.022	0.021	0.020	0.023	0.022	0.021	0.021	0.022
	l2e1	0.045	0.041	0.042	0.046	0.058	0.035	0.036	0.036	0.037	1.843
	l2e2	0.071	0.044	0.044	0.050	0.125	0.036	0.035	0.035	0.042	1.657
	l2e3	0.098	0.080	0.090	1.793	1.965	0.066	0.066	0.067	1.980	1.977
	l2e4	0.136	0.088	0.118	1.950	1.988	0.078	0.069	1.943	1.981	1.987
Case 2	l1e1	0.004	0.004	0.006	0.014	0.023	0.003	0.003	0.005	0.011	0.020
	l1e2	0.011	0.009	0.008	0.007	0.008	0.006	0.005	0.005	0.005	0.006
	l1e3	0.010	0.010	0.018	0.046	0.083	0.010	0.011	0.018	0.044	0.075
	l1e4	0.039	0.015	0.011	0.014	0.027	0.009	0.007	0.007	0.012	0.025
	l2e1	0.111	0.014	0.005	0.002	0.001	0.006	0.002	0.001	0.001	0.001
	l2e2	0.090	0.015	0.006	0.004	0.005	0.003	0.002	0.002	0.003	0.004
	l2e3	0.166	0.027	0.015	0.022	0.035	0.010	0.007	0.009	0.019	0.031
	l2e4	0.175	0.034	0.038	0.080	0.123	0.014	0.016	0.030	0.071	0.113

Table S.5: ISE of eigenfunctions averaged over 20 repeats for different bin widths.

Basis function	Binary					Poisson				
	W = 2%	W = 5%	W = 10%	W = 15%	W = 20%	W = 2%	W = 5%	W = 10%	W = 15%	W = 20%
Case 1	0.296	0.267	0.275	0.565	0.607	0.030	0.028	0.249	0.700	0.733
Case 2	0.325	0.291	0.293	0.308	0.327	0.085	0.098	0.109	0.141	0.169

Table S.6: MSE of linear predictors averaged over 20 repeats for different bin widths.

S.4.4 Effect of the fixed-effect intercept

In this experiment, we focus on binary data and vary the fixed-effect intercept while fixing other parameters at the following values: sample size $I = 1000$, number of visits $J = 10$, number of sampling points $K = 100$, and percentage of data used for constructing local bins $w = 5\%$. Table S.7 and S.8 display the ISE of eigenfunctions and MSE of linear predictors for different data distributions and basis functions as a function of the intercept. Results indicate that both ISE and MSE are estimated increasingly well when the percentage of zeros and ones become more balanced (closer to 50%).

Basis function	Eigenfunction	Binary			
		b0 = -3.5	b0 = -2.5	b0 = -1.5	b0 = 0
Case 1	l1e1	0.007	0.006	0.006	0.006
	l1e2	0.013	0.010	0.009	0.008
	l1e3	0.028	0.024	0.023	0.022
	l1e4	0.047	0.029	0.027	0.024
	l2e1	0.045	0.043	0.041	0.041
	l2e2	0.076	0.054	0.047	0.044
	l2e3	0.252	0.110	0.093	0.080
	l2e4	1.232	0.428	0.162	0.088
Case 2	l1e1	0.006	0.005	0.005	0.004
	l1e2	0.016	0.011	0.009	0.009
	l1e3	0.017	0.013	0.010	0.010
	l1e4	0.674	0.107	0.024	0.015
	l2e1	0.544	0.068	0.026	0.014
	l2e2	0.365	0.036	0.018	0.015
	l2e3	0.500	0.086	0.040	0.027
	l2e4	0.312	0.057	0.038	0.034

Table S.7: ISE of eigenfunctions averaged over 20 repeats for different intercepts.

Basis function	Binary			
	b0 = -3.5	b0 = -2.5	b0 = -1.5	b0 = 0
Case 1	0.724	0.463	0.324	0.267
Case 2	0.895	0.545	0.378	0.291

Table S.8: MSE of linear predictors averaged over 20 repeats for different intercepts.

S.5 More NHANES data results

S.5.1 Summary of demographic variables and traditional risk factors

	Mean (SD) or N (%)
Sample Size	4445
Age	64.73 (9.45)
Gender	
Female	2318 (52.1)
BMI	
Normal	1125 (25.3)
Underweight	75 (1.7)
Overweight	1470 (33.1)
Obese	1710 (38.5)
NA	65 (1.5)
Race	
Non-Hispanic White	1963 (44.2)
Non-Hispanic Black	1123 (25.3)
Mexican American	428 (9.6)
Non-Hispanic Asian	427 (9.6)
Other Hispanic	430 (9.7)
Other Race	74 (1.7)
Overall Health	
Excellent	1743 (39.2)
Very good	334 (7.5)
Good	1041 (23.4)
Fair	1069 (24.0)
Poor	258 (5.8)
Diabetes	
No	3279 (73.8)
Borderline	181 (4.1)
Yes	982 (22.1)
NA	3 (0.1)
Arthritis	
No	2459 (55.3)
Yes	1975 (44.4)
NA	11 (0.2)
CHF	
No	4162 (93.6)
Yes	268 (6.0)
NA	15 (0.3)
CHD	
No	4076 (91.7)
Yes	341 (7.7)
NA	28 (0.6)
Heart Attack	
No	4107 (92.4)
Yes	332 (7.5)
NA	6 (0.1)
Stroke	
No	4121 (92.7)
Yes	318 (7.2)
NA	6 (0.1)
Cancer	
No	3707 (83.4)

Yes	737 (16.6)
NA	1 (0.0)
Alcohol Consumption	
Never drinker	697 (15.7)
Former drinker	1081 (24.3)
Moderate drinker	2161 (48.6)
Heavy drinker	257 (5.8)
Not Available	249 (5.6)
Smoking	
Never	2224 (50.0)
Former	1477 (33.2)
Current	741 (16.7)
NA	3 (0.1)
Mobility Problem	
No difficulty	2929 (65.9)
Any difficulty	1511 (34.0)
NA	5 (0.1)

Table S.9: Summary of demographic variables and traditional risk factors for subjects over the age of 50 from the NHANES 2011-2014 accelerometry data set.

S.5.2 Choosing bin width

To choose a bin width for the NHANES data, we tried three values: 10 minutes, 30 minutes, and 60 minutes. We fit local GLMMs using all three values and compare the estimated eigenfunctions. Figure S.3 and S.4 show level 1 and level 2 eigenfunctions, respectively. We can see that results are very similar when using a bin width of 30 minutes or 60 minutes. With a bin width of 10 minutes, we can see that while some eigenfunctions (e.g., l1e3, l1e6, l2e2, l2e3, with possible difference in signs) are similar to 30 minutes and 60 minutes, other are not. In addition, we got singular fits in 31 bins when using a bin width of 10 minutes, which is not surprising given the number of subject-visit pairs that have all zeros (little movement when people are asleep) in bins during the night (Figure S.5 top right). As bin width increases to 30 minutes, we no longer have singular fits.

S.6 More computation time results

In Section 4.3 of the main text, we presented the computation time of GMFPCA as the number of sampling points K varies. In this section, we present additional computation time results when the sample size (Table S.10) or the number of visits (Table S.11) vary.
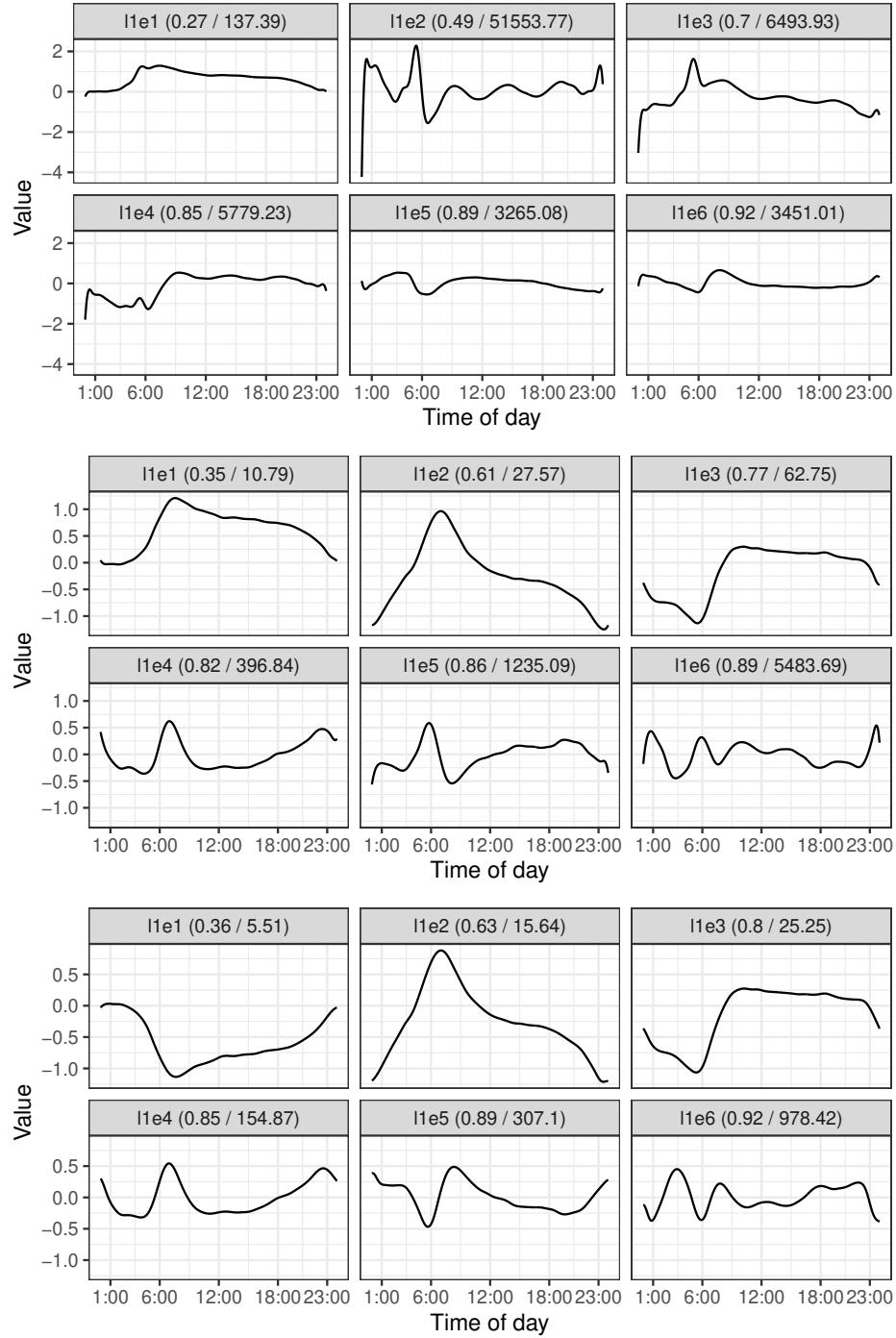
Figure S.3: Level 1 eigenfunctions obtained under bin widths 10 minutes (top), 30 minutes (middle), and 60 minutes (bottom) for the NHANES data.
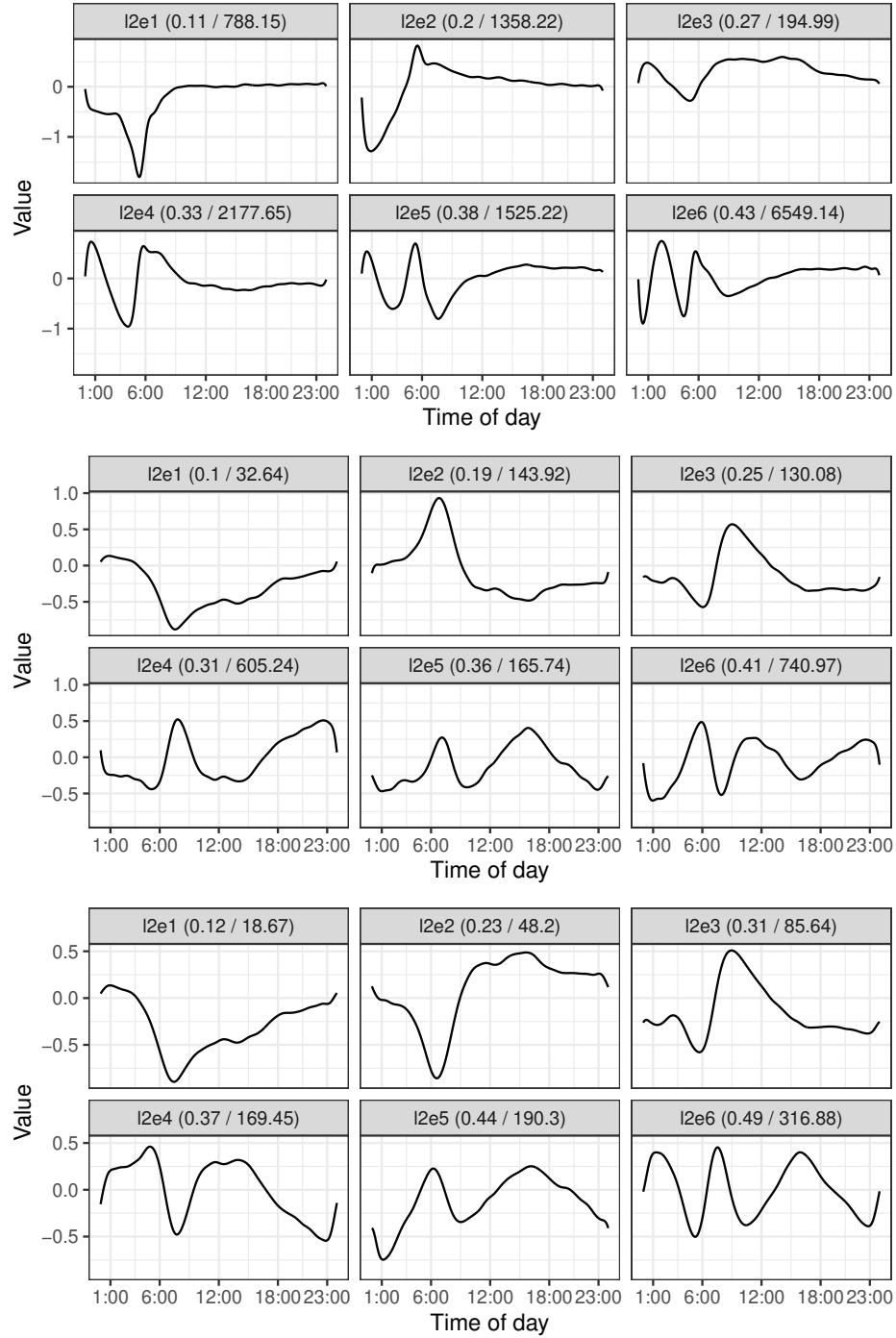
Figure S.4: Level 2 eigenfunctions obtained under bin widths 10 minutes (top), 30 minutes (middle), and 60 minutes (bottom) for the NHANES data.
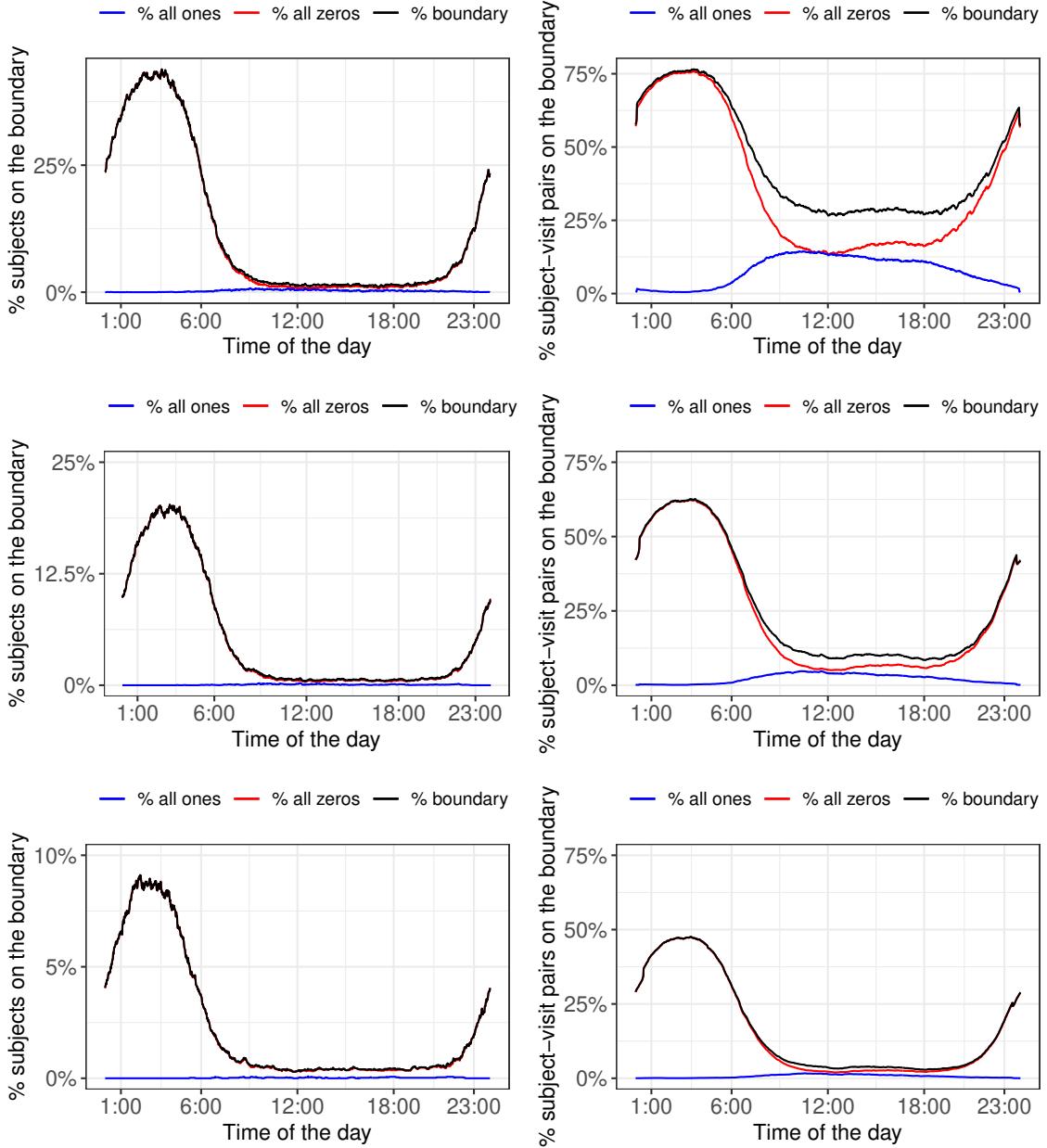
Figure S.5: Percentage of subjects (left) and subject-visit pairs (right) whose binary activity values are on the boundary in each bin for bin widths of 10 minutes (top), 30 minutes (middle), and 60 minutes (bottom).

Basis Functions	Sample Size	Binary			Poisson		
		Step 1 and 2	Step 3	Step 4	Step 1 and 2	Step 3	Step 4
Case 1	50	0	0.002	22	1	0.002	47
	100	1	0.003	60	1	0.002	265
	200	2	0.003	95	2	0.003	544
	500	4	0.005	543	5	0.005	1424
	1000	12	0.008	1572	10	0.008	3231
Case 2	50	0	0.002	11	1	0.002	296
	100	1	0.003	78	1	0.003	763
	200	2	0.003	97	2	0.003	1377
	500	4	0.005	551	5	0.004	3240
	1000	9	0.007	678	11	0.007	7850

Table S.10: Median computation time (minutes) across 20 repeats. In this experiment, we fixed number of visits $J = 10$, number of sampling points $K = 100$, and percentage of data used for constructing local bins $w = 5\%$. For binary data, we used fixed effect intercept $b_0 = 0$.

Basis Functions	# Visits	Binary			Poisson		
		Step 1 and 2	Step 3	Step 4	Step 1 and 2	Step 3	Step 4
Case 1	2	2	< 0.001	142	2	< 0.001	474
	5	4	< 0.001	296	5	< 0.001	1047
	10	12	0.010	1572	10	0.010	3231
Case 2	2	2	< 0.001	369	2	< 0.001	1715
	5	5	< 0.001	307	5	< 0.001	3605
	10	9	0.010	678	11	0.010	7850

Table S.11: Median computation time (minutes) across 20 repeats. In this experiment, we fixed sample size $I = 1000$, number of sampling points $K = 100$, and percentage of data used for constructing local bins $w = 5\%$. For binary data, we used fixed effect intercept $b_0 = 0$.

References

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Bothwell, S., Kaizer, A., Peterson, R., Ostendorf, D., Catenacci, V., and Wrobel, J. (2022). Pattern-based clustering of daily weigh-in trajectories using dynamic time warping. *Biometrics*.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1).

Chen, H., Wang, Y., Paik, M. C., and Choi, H. A. (2013). A marginal approach to reduced-rank penalized spline smoothing with application to multilevel functional data. *Journal of the American Statistical Association*, 108(504):1216–1229. PMID: 24497670.

Crainiceanu, C. M., Goldsmith, J., Leroux, A., and Cui, E. (2023). *Functional Data Analysis with R*. Springer New York, NY, USA.

Crainiceanu, C. M., Ruppert, D., and Wand, M. P. (2005). Bayesian analysis for penalized spline regression using winbugs. *Journal of Statistical Software*, 14(14):1–24.

Cui, E., Leroux, A., Smirnova, E., and Crainiceanu, C. (2022a). Fast univariate inference for longitudinal functional models. *Journal of Computational and Graphical Statistics*, 31(1):219–230.

Cui, E., Li, R., Crainiceanu, C. M., and Xiao, L. (2022b). Fast multilevel functional principal component analysis. *Journal of Computational and Graphical Statistics*, 0(ja):1–33.

Di, C.-Z., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. (2009). Multilevel functional principal component analysis. *The Annals of Applied Statistics*, 3(1):458.

Gaston, M., Leon, T., and Mallor, F. (2008). Functional data analysis for non homogeneous poisson processes. In *2008 Winter Simulation Conference*, pages 337–343.

Gaynanova, I., Punjabi, N., and Crainiceanu, C. (2020). Modeling continuous glucose monitoring (CGM) data during sleep. *Biostatistics*, 23(1):223–239.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534.

Gertheiss, J., Goldsmith, J., and Staicu, A.-M. (2017a). A note on modeling sparse exponential-family functional response curves. *Computational Statistics & Data Analysis*, 105:46–52.

Gertheiss, J., Goldsmith, J., and Staicu, A.-M. (2017b). A note on modeling sparse exponential-family functional response curves. *Computational Statistics & Data Analysis*, 105:46–52.

Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Di, C., Gellar, J., Harezlak, J., McLean, M., Swihart, B., Xiao, L., Crainiceanu, C., and Reiss, P. (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.

Hall, P., Müller, H.-G., and Yao, F. (2008). Modelling sparse generalized longitudinal observations with latent Gaussian processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):703–723.

Hornik, K., Leisch, F., Zeileis, A., and Plummer, M. (2003). JAGS: A program for analysis

of bayesian graphical models using gibbs sampling. In *Proceedings of DSC*, volume 2.

John, D., Tang, Q., Albinali, F., and Intille, S. (2019). An open-source monitor-independent movement summary for accelerometer data processing. *Journal for the Measurement of Physical Behaviour*, 2(4):268 – 281.

Jones, M. and Rice, J. A. (1992). Displaying the important features of large collections of similar curves. *The American Statistician*, 46(2):140–145.

Karas, M., Muschelli, J., Leroux, A., Urbanek, J. K., Wanigatunga, A. A., Bai, J., Crainiceanu, C. M., and Schrack, J. A. (2022). Comparison of accelerometry-based measures of physical activity: Retrospective observational data analysis study. *JMIR mHealth and uHealth*, 10(7):e38077.

Kass, R. and Ventura, V. (2001). A spike train probability model. *Neural Computation*, 13:1713–1720.

Kelly, R. and Kass, R. (2012). A framework for evaluating pairwise and multiway synchrony among stimulus-driven neurons. *Neural Computation*, 24:2007–2032.

Leroux, A., Crainiceanu, C., and Wrobel, J. (2023). Fast generalized functional principal components analysis.

Li, G., Huang, J. Z., and Shen, H. (2018). Exponential family functional data analysis via a low-rank model. *Biometrics*, 74(4):1301–1310.

NCHS (2022). Continuous NHANES public-use linked mortality files, 2019. `https://www.cdc.gov/nchs/data-linkage/mortality-public.htm`.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer New York, NY, USA.

Scheipl, F., Gertheiss, J., and Greven, S. (2016). Generalized functional additive mixed models. *Electronic Journal of Statistics*, 10(1):1455 – 1492.

Sebastián-González, E., Sánchez-Zapata, J., Botella, F., and Ovaskainen, O. (2010). Testing the heterospecific attraction hypothesis with time-series data on species co-occurrence. *Proceedings: Biological Sciences*, 277(1696):2983–2990.

Sentürk, D., Dalrymple, L., and Nguyen, D. (2014). Functional linear models for zero-inflated count data with application to modeling hospitalizations in patients on dialysis. *Statistics in Medicine*, 33(27):4825–4840.

Staicu, A., Crainiceanu, C., Reich, D., and Ruppert, D. (2018). Modeling functional data with spatially heterogeneous shape characteristics. *Biometrics*, 68(2):331–343.

Staniswalis, J. and Lee, J. (1998). Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418.

Swihart, B., Punjabi, N., and Crainiceanu, C. (2015). Modeling sleep fragmentation in sleep hypnograms: An instance of fast, scalable discrete-state, discrete-time analyses. *Computational Statistics and Data Analysis*, 89:1–11.

van der Linde, A. (2009). A Bayesian latent variable approach to functional principal components analysis with binary and count data. *AStA Advances in Statistical Analysis*, 93(3):307–333.

Wood, S. (2017). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2 edition.

Wrobel, J. (2018). register: Registration for exponential family functional data. *Journal of Open Source Software*, 3(22):557.

Wrobel, J. (2023). *fastGFPCA: Fast generalized principal components analysis.*

https://github.com/julia-wrobel/fastGFPCA.

Wrobel, J. and Bauer, A. (2021). registr 2.0: Incomplete curve registration for exponential family functional data. *Journal of Open Source Software*, 6(61):2964.

Wrobel, J., Zipunnikov, V., Schrack, J., and Goldsmith, J. (2019). Registration for exponential family functional data. *Biometrics*, 75(1):48–57.

Xiao, L., Zipunnikov, V., Ruppert, D., and Crainiceanu, C. (2016a). Fast covariance estimation for high-dimensional functional data. *Statistics and computing*, 26(1):409–421.

Xiao, L., Zipunnikov, V., Ruppert, D., and Crainiceanu, C. (2016b). Fast covariance estimation for high-dimensional functional data. *Statistics and computing*, 26(1):409–421.

Zhang, J., Siegle, G. J., Sun, T., D'andrea, W., and Krafty, R. T. (2023). Interpretable principal component analysis for multilevel multivariate functional data. *Biostatistics*, 24(2):227–243.