Supplementary Material to “General Functional Concurrent Model”

Janet S. Kim[*] Arnab Maity[†] Ana-Maria Staicu[‡]

June 17, 2016

This Supplementary Material contains six sections. Appendix A discusses modifications required by the proposed estimation procedure when responses and/or covariates are observed on sparse and irregular grid of points. Appendix B describes transformation of the functional covariates required by our estimation procedure. Appendix C details the evaluation criteria used in our simulation experiment. Additional simulation results are included in Appendix D, and additional investigation for data applications are provided in Appendix E. Appendix F presents the implementation details.

A Model Estimation for Irregular and Sparse Design

The proposed method easily accommodates more realistic situations where the covariates and/or the responses are observed on sparse sampling design. Here we discuss the modifications required by the proposed estimation procedure to accommodate such sparseness. This approach was used in the simulation study as well as in the analysis of the dietary calcium absorption data.

Suppose for $i = 1, \dots, n$ we observe the functional covariates $X_{1,i}(t_{1ij})$ for $j = 1, \dots, m_{1i}$ and $X_{2,i}(t_{2ij})$ for $j = 1, \dots, m_{2i}$. Also, for $i = 1, \dots, n$ we observe the functional response $Y_i(t_{ik})$ for $k = 1, \dots, m_{Y,i}$. Our goal is to estimate the unknown model

[*] PhD Student, Department of Statistics, North Carolina State University, Raleigh, North Carolina 27695 (Email: jskim3@ncsu.edu)

[†] Associate Professor, Department of Statistics, North Carolina State University, Raleigh, North Carolina 27695 (Email: amaity@ncsu.edu)

[‡] Associate Professor, Department of Statistics, North Carolina State University, Raleigh, North Carolina 27695 (Email: ana-maria_staicu@ncsu.edu)

components, $\mu_Y(\cdot)$, F_1 and F_2, in our general functional concurrent model (GFCM), $Y_i(t) = \mu_Y(t) + F_1\{X_{1,i}(t), t\} + F_2\{X_{2,i}(t), t\} + \epsilon_i(t)$. Using basis expansions, an equivalent form of the model can be found as $Y_i(t) = \mathbb{B}_\mu(t)\Theta_\mu + \mathbb{Z}_{1,i}(t)\Theta_1 + \mathbb{Z}_{2,i}(t)\Theta_2 + \epsilon_i(t)$; here, $\mathbb{B}_\mu(t)$ and $\mathbb{Z}_{q,i}(t)$ are defined using pre-specified B-spline basis functions, as detailed in Section 2.2 of the main paper.

When the data has different observation points, we first smooth the covariates $X_{q,i}(\cdot)$ to obtain the estimated smooth curve $\widehat{X}_{q,i}(\cdot)$ of $X_{q,i}(\cdot)$, and we evaluate the smooth curve $\widehat{X}_{q,i}(\cdot)$ at the points t_{ik} - the points at which the response is observed. Then we approximate the penalized sum of squares by

$$\sum_{i=1}^{n}\left[\sum_{k=1}^{m_{Y,i}}\{Y_i(t_{ik}) - \mathbb{B}_\mu(t_{ik})\Theta_\mu - \mathbb{Z}_{1,i}(t_{ik})\Theta_1 - \mathbb{Z}_{2,i}(t_{ik})\Theta_2\}^2/m_{Y,i}\right] + \\ \Theta_\mu^T P_\mu\Theta_\mu + \Theta_1^T P_1\Theta_1 + \Theta_2^T P_2\Theta_2,$$

where P_μ, P_1 and P_2 are the penalty matrices defined in Section 2.2 of the main paper. To estimate the unknown parameter Θ_μ, Θ_1 and Θ_2, we minimize the above penalized criterion. For a simple illustration, let $\mathbb{Y}_i = [Y_i(t_{i1}), \ldots, Y_i(t_{im_{Y,i}})]^T$ and $\mathbb{E}_i = [\epsilon_i(t_{i1}), \ldots, \epsilon_i(t_{im_{Y,i}})]^T$ be the $m_{Y,i}$-dimensional vector of response and random errors for subject i. Also, define $\widetilde{\mathbb{B}}_\mu$ as $m_{Y,i} \times K_\mu$-dimensional matrix with the j-th row given by $\mathbb{B}_\mu(t_{ik})$ and $\mathcal{Z}_{q,i}$ as $m_{Y,i} \times K_{xq}K_{tq}$-dimensional matrix with the j-th row given by $\mathbb{Z}_{q,i}(t_{ik})$ $(q = 1, 2)$. We further denote $\mathcal{Z}_i = [\widetilde{\mathbb{B}}_\mu|\mathcal{Z}_{1,i}|\mathcal{Z}_{2,i}]$, $\Theta^T = [\Theta_\mu, \Theta_1, \Theta_2]^T$ and $\mathbb{P} = \mathrm{diag}(P_\mu, P_1, P_2)$. Then, the solution Θ has a closed form expression:

$$\widehat{\Theta} = H\{\textstyle\sum_{i=1}^{n}\mathcal{Z}_i^T\mathbb{Y}_i\}$$

with $H = \{\sum_{i=1}^{n}\mathcal{Z}_i^T\mathcal{Z}_i + \mathbb{P}\}^{-1}$. Furthermore, the variance of the parameter estimator can be calculated, following the same procedure described in Section 2.3 of the main paper:

$$\mathrm{var}(\widehat{\Theta}) = H\{\textstyle\sum_{i=1}^{n}\mathcal{Z}_i^T\mathbb{G}_i\mathcal{Z}_i\}H^T,$$

where $\mathbb{G}_i = \mathrm{cov}(\mathbb{E}_i)$ with dimension $m_{Y,i} \times m_{Y,i}$ for each i.

B Preprocessing of the Functional Covariates

In this section, we discuss the preprocessing of the functional covariates. One challenge of our estimation approach is that some B-splines might not have observed data on its support. This problem is more likely to arise when the covariate $X(\cdot)$ is observed on sparse and irregular gird of points, and realizations of the function, $X(t_j)$, are not dense over $\mathbb{R}$. To bypass this limitation, we propose to apply point-wise center/scaling transformation of the covariates; it is worthwhile to note that this problem is addressed by McLean et al. (2014) with a different approach.

We define point-wise center/scaling transformation of $X(t)$ by

$$X^*(t) = \{X(t) - \mu_X(t)\}/\sigma_X(t)$$

where $\mu_X(t)$ and $\sigma_X(t)$ are mean and standard deviation of $X(t)$. One can interpret the transformed covariate $X^*(t)$ as the amount of standard deviation $X(t)$ is away from the mean at time t. In practice, we estimate the mean and the standard deviation by the sample mean $\widehat{\mu}_X(t)$ and the sample standard deviation $\widehat{\sigma}_X(t)$, respectively, of the covariates. Thus for a fixed point t_j we will obtain realizations of the transformed covariates $\{X_i^*(t_j)\}_{i=1}^n$ based on the sample mean $\widehat{\mu}_X(t_j)$ and the sample standard deviation $\widehat{\sigma}_X(t_j)$ at the same point. The prediction procedure described in Section 2.4 of the main paper proceeds as before with the understanding that one now uses the transformed version of the new covariates, $X_0^*(t)$

$$X_0^*(t) = \{X_0(t) - \mu_X(t)\}/\sigma_X(t),$$

where we estimate the mean $\mu_X(t)$ and the standard deviation $\sigma_X(t)$ using the sample mean and the sample standard deviation obtained from the training data. Our empirical study has shown that the above transformation technique effectively controls the numerical stability issues, while still preserving predictive accuracy.

C Evaluation Criteria

In Section 5.2.1 of the main paper, we studied prediction performance of the proposed method through 1000 Monte Carlo simulations. Estimation and prediction accuracy were assessed using in-sample and out-of-sample root mean squared prediction error (RMSPE). The performance of variance estimation was measured through integrated coverage probability (ICP) and integrated length (IL) of the point-wise prediction intervals. We now describe how we define the above measures.

We define the in-sample RMSPE by

$$\text{RMSPE}^{\text{in}} = \frac{1}{1000 \cdot n} \sum\nolimits_{r=1}^{1000} \left[\sum\nolimits_{i=1}^{n} \frac{1}{m_i} \sum\nolimits_{k=1}^{m_i} \{ Y_i^{(r)}(t_{ik}) - \widehat{Y}_i^{(r)}(t_{ik}) \}^2 \right]^{\frac{1}{2}},$$

where $Y_i^{(r)}(t_{ik})$ and its estimate $\widehat{Y}_i^{(r)}(t_{ik})$ are from the r-th Monte Carlo simulation. The out-of-sample RMSPE, denoted by $\text{RMSPE}^{\text{out}}$, is defined similarly.

We approximate $(1-\alpha)$ level point-wise prediction intervals to observe coverage probabilities at the nominal level. The ICP at the $(1-\alpha)$ level is given by

$$\text{ICP}(1-\alpha) = \frac{1}{1000 \cdot n} \sum\nolimits_{r=1}^{1000} \sum\nolimits_{i=1}^{n} \int_0^1 I\{ Y_{0,i}^{(r)}(t) \in C_{1-\alpha,i}^{(r)}(t) \} dt,$$

where $C_{1-\alpha,i}^{(r)}(t)$ is the point-wise prediction interval from the r-th Monte Carlo simulation and $I(\cdot)$ is the indicator function. The prediction interval $C_{1-\alpha,i}^{(r)}(t)$ is as previously defined in Section 2.3 of the main paper.

We observe particular features of the prediction intervals by measuring their length. The IL of the $(1-\alpha)$ level prediction intervals is defined by

$$\text{IL}(1-\alpha) = \frac{1}{1000 \cdot n} \sum\nolimits_{r=1}^{1000} \sum\nolimits_{i=1}^{n} \int_0^1 2\text{MOE}_{1-\alpha,i}^{(r)}(t) dt$$

where $\text{MOE}_{1-\alpha,i}^{(r)}(t) = \Phi^{-1}(1-\alpha/2) \times \widehat{\text{var}}\{ Y_{0,i}^{(r)}(t) - \widehat{Y}_{0,i}^{(r)}(t) \}^{-\frac{1}{2}}$, and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function.

Although the IL of the prediction intervals remains small, the band might fluctuate dramatically at some points, and this will demonstrate a poor performance of variance estimation. Therefore, we examine the range of the estimated standard errors (SE). For the prediction intervals, we define the minimum SE by

$$\min(\mathrm{SE}) = \frac{1}{1000 \cdot n} \textstyle\sum_{r=1}^{1000} \sum_{i=1}^{n} \min_{t} \{2\mathrm{MOE}_{1-\alpha,i}^{(r)}(t)\}.$$

We define the maximum SE, denoted by max(SE), similarly. Then, R(SE)= $[\min(\mathrm{SE}), \max(\mathrm{SE})\}]$ provides the range of SE at the $(1 - \alpha)$ level.

D Additional Simulation Results

This section provides additional simulation results. In Appendix D.1, we show further investigation of the prediction error. In Appendix D.2 we compare the prediction results obtained by using different covariance estimation methods. In Appendix D.3, we present power curves corresponding to the case of densely sampled data and further discuss power performance of the proposed test.

D.1 Further Investigation of Prediction Error

D.1.1 Additional Simulations for Irregular and Sparse Design

We provide additional simulation results corresponding to another level of sparseness. This simulation study is based on the sparsely sampled data, but the data is now sampled at relatively increased number of points t; for convenience, we call this setting moderately sparse design. The sparse design scenario considered in the main paper is based on the smaller number of time points t, and we continue to call this setting sparse design. For the moderately sparse design, we used $m_i \overset{iid}{\sim} Uniform(29, 41)$ points to evaluate the response and covariate functions, and then we assessed the predictive accuracy of our estimation procedure using the evaluation criteria defined in Appendix C. Table 1 shows prediction results from the sparse design as well as the moderately sparse design. As expected, the moderately sparse setting improves both the in-sample and out-of-sample prediction accuracy on an average.

Table 1: Summary of RMSPE$^{\text{in}}$, RMSPE$^{\text{out}}$, ICP, IL, and R(SE) based on 1000 simulated data sets. The models fitted by our method and the linear FCM are indicated by `GFCM` and `FCM`, respectively.

						$1-\alpha=0.95$						$1-\alpha=0.90$				$1-\alpha=0.85$			
		RMSPE$^{\text{in}}$		RMSPE$^{\text{out}}$		ICP		IL		R(SE)		ICP		IL		ICP		IL	
n	$\mathbb{E}_i$	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`	`GFCM`	`FCM`
Scenario B (true relationship is non-linear), $m_i \overset{iid}{\sim} Unif(20,31)$ **(sparse design)**																			
	$\mathbb{E}_i^1$	1.16	3.55	1.18	3.76	0.964	0.928	4.52	12.60	[3.71, 7.16]	[3.32, 23.46]	0.929	0.903	3.79	10.57	0.891	0.880	3.32	9.25
100	$\mathbb{E}_i^2$	1.46	3.66	1.47	3.87	0.959	0.932	5.75	13.17	[5.09, 8.04]	[4.69, 23.72]	0.918	0.904	4.82	11.06	0.875	0.877	4.22	9.68
	$\mathbb{E}_i^3$	1.94	3.89	2.06	4.10	0.949	0.942	7.69	14.51	[5.82, 11.09]	[8.13, 24.67]	0.900	0.911	6.46	12.17	0.852	0.879	5.65	10.65
	$\mathbb{E}_i^1$	1.15	3.65	1.00	3.68	0.973	0.935	4.42	12.90	[3.67, 6.90]	[3.23, 23.25]	0.941	0.912	3.71	10.83	0.904	0.889	3.25	9.48
300	$\mathbb{E}_i^2$	1.46	3.76	1.33	3.79	0.966	0.941	5.66	13.50	[5.06, 7.79]	[4.70, 23.50]	0.927	0.915	4.75	11.33	0.885	0.889	4.16	9.91
	$\mathbb{E}_i^3$	1.96	3.99	1.91	4.03	0.958	0.948	7.64	14.80	[5.87, 10.64]	[8.28, 24.56]	0.913	0.919	6.41	12.42	0.866	0.890	5.61	10.87
Scenario B (true relationship is non-linear), $m_i \overset{iid}{\sim} Unif(29,41)$ **(moderately sparse design)**																			
	$\mathbb{E}_i^1$	1.09	3.55	1.10	3.76	0.962	0.932	4.26	12.62	[3.62, 6.53]	[3.42, 23.06]	0.924	0.908	3.58	10.59	0.883	0.885	3.13	9.27
100	$\mathbb{E}_i^2$	1.41	3.66	1.41	3.87	0.958	0.938	5.54	13.24	[5.02, 7.47]	[4.87, 23.34]	0.915	0.911	4.65	11.11	0.870	0.883	4.07	9.72
	$\mathbb{E}_i^3$	1.90	3.90	2.02	4.10	0.948	0.943	7.55	14.52	[5.69, 10.67]	[8.18, 24.34]	0.899	0.912	6.34	12.19	0.849	0.881	5.55	10.67
	$\mathbb{E}_i^1$	1.08	3.64	0.96	3.68	0.969	0.942	4.20	12.91	[3.61, 6.36]	[3.46, 22.84]	0.934	0.919	3.52	10.84	0.894	0.896	3.08	9.49
300	$\mathbb{E}_i^2$	1.41	3.75	1.31	3.79	0.963	0.946	5.49	13.53	[5.01, 7.33]	[4.93, 23.12]	0.921	0.920	4.61	11.36	0.877	0.895	4.03	9.94
	$\mathbb{E}_i^3$	1.92	3.98	1.89	4.03	0.955	0.948	7.49	14.77	[5.75, 10.23]	[8.38, 24.14]	0.909	0.920	6.29	12.40	0.861	0.891	5.50	10.85
Scenario A (true relationship is linear), $m_i \overset{iid}{\sim} Unif(20,31)$ **(sparse design)**																			
	$\mathbb{E}_i^1$	0.92	0.92	0.90	0.90	0.955	0.955	3.61	3.61	[3.55, 3.77]	[3.55, 3.77]	0.907	0.908	3.03	3.03	0.859	0.859	2.65	2.65
100	$\mathbb{E}_i^2$	1.28	1.28	1.27	1.27	0.952	0.952	5.04	5.04	[4.96, 5.26]	[4.96, 5.26]	0.904	0.904	4.23	4.23	0.854	0.855	3.70	3.70
	$\mathbb{E}_i^3$	1.81	1.84	1.89	1.86	0.943	0.947	7.14	7.17	[5.38, 9.56]	[5.37, 9.54]	0.891	0.896	6.00	6.01	0.840	0.846	5.25	5.26
	$\mathbb{E}_i^1$	0.92	0.92	0.90	0.90	0.955	0.955	3.60	3.60	[3.55, 3.72]	[3.55, 3.73]	0.908	0.908	3.02	3.02	0.859	0.859	2.64	2.64
300	$\mathbb{E}_i^2$	1.28	1.28	1.27	1.27	0.953	0.953	5.03	5.03	[4.97, 5.19]	[4.97, 5.19]	0.904	0.904	4.22	4.22	0.855	0.855	3.69	3.69
	$\mathbb{E}_i^3$	1.84	1.85	1.85	1.84	0.949	0.950	7.16	7.17	[5.45, 9.19]	[5.45, 9.19]	0.899	0.900	6.01	6.02	0.848	0.850	5.26	5.27
Scenario A (true relationship is linear), $m_i \overset{iid}{\sim} Unif(29,41)$ **(moderately sparse design)**																			
	$\mathbb{E}_i^1$	0.91	0.91	0.90	0.90	0.953	0.953	3.58	3.58	[3.54, 3.70]	[3.54, 3.71]	0.905	0.905	3.00	3.00	0.856	0.857	2.63	2.63
100	$\mathbb{E}_i^2$	1.28	1.28	1.27	1.27	0.952	0.952	5.02	5.02	[4.95, 5.21]	[4.95, 5.21]	0.903	0.903	4.21	4.21	0.854	0.854	3.69	3.69
	$\mathbb{E}_i^3$	1.81	1.83	1.90	1.85	0.943	0.947	7.14	7.15	[5.37, 9.36]	[5.37, 9.27]	0.891	0.896	5.99	6.00	0.840	0.846	5.24	5.25
	$\mathbb{E}_i^1$	0.91	0.91	0.90	0.90	0.954	0.954	3.57	3.57	[3.54, 3.67]	[3.54, 3.67]	0.906	0.906	3.00	3.00	0.857	0.857	2.62	2.62
300	$\mathbb{E}_i^2$	1.28	1.28	1.27	1.27	0.952	0.952	5.01	5.01	[4.96, 5.17]	[4.96, 5.17]	0.903	0.903	4.20	4.20	0.854	0.854	3.68	3.68
	$\mathbb{E}_i^3$	1.83	1.84	1.85	1.84	0.948	0.949	7.14	7.15	[5.44, 9.04]	[5.44, 9.03]	0.897	0.899	5.99	6.00	0.847	0.849	5.24	5.25

D.1.2 Additional Simulations for Complicated Error Structure

Next, we further discuss prediction accuracy investigated in Section 5.2.1 of the main paper. In Table 1 of the main paper, the value of $\mathrm{RMSPE}^{\mathrm{in}}$ corresponding to non-stationary error covariance ($\mathbb{E}_i^3$) slightly increases with larger sample size. To investigate whether the results are valid, one needs to compare the values of $\mathrm{RMSPE}^{\mathrm{in}}$ with the true standard deviation of the error process. In the simulation, the true standard deviation (averaged over t) of the error process can be computed as:

- $\mathrm{sd}(\mathbb{E}_i^1) = \sqrt{0.8} \approx 0.89$.
- $\mathrm{sd}(\mathbb{E}_i^2) = \sqrt{0.8 + 0.8} \approx 1.26$.
- $\mathrm{sd}(\mathbb{E}_i^3) = \sqrt{2 + 0.75^2 + 0.8} \approx 1.83$.

Table 2 summarizes the $\mathrm{RMSPE}^{\mathrm{in}}$ with the estimated standard errors in the parentheses for different sampling scenarios and for the error process corresponding to $\mathbb{E}_i = \mathbb{E}_i^3$. From this table, one should expect to see that the true standard deviation is within 2 standard errors of the estimated $\mathrm{RMSPE}^{\mathrm{in}}$. In Table 2, as the number of observations per curve decreases (see the column for the sparse design), the values of $\mathrm{RMSPE}^{\mathrm{in}}$ are more different to the true standard deviation of the error process. As the complexity of the true model increases, the values of $\mathrm{RMSPE}^{\mathrm{in}}$ are slightly different from the true standard deviation; see the results corresponding to Scenario C, which incorporates two functional covariates and has the most complicated form of the true model. Nevertheless, the results indicate that the target values (true standard deviation of the error process) are within two standard deviations of the estimated values, and thus our estimation method still preserves prediction accuracy.

D.2 Further Investigation of Different Covariance Estimation Methods

In the literature, there are various approaches to estimate the covariance of the residual process: using local polynomial smoothing and using global smoothing via B-spline basis functions. So far the latter approach is implemented in `R`, whereas the former one is implemented in `Matlab`. The proposed method in Section 2.3 of the main paper is implemented by `fpca.sc` function in `refund` `R` package, which uses the tensor product

Table 2: Summary of RMSPE$^{\text{in}}$ and the estimated standard errors (in parentheses) obtained by fitting the GFCM based on 1000 simulations. The simulation settings correspond to the case where $\mathbb{E}_i = \mathbb{E}_i^3$.

		dense design	moderately sparse design	sparse design
Scenario	n	$m = 81$	$m_i \overset{iid}{\sim} Unif(29, 41)$	$m_i \overset{iid}{\sim} Unif(20, 31)$
A	100	1.80 (0.08)	1.81 (0.08)	1.81 (0.08)
	300	1.83 (0.05)	1.83 (0.05)	1.84 (0.05)
B	100	1.84 (0.08)	1.90 (0.09)	1.94 (0.09)
	300	1.87 (0.05)	1.92 (0.05)	1.96 (0.05)
C	100	1.80 (0.08)	1.86 (0.08)	1.91 (0.09)
	300	1.85 (0.05)	1.90 (0.05)	1.94 (0.05)

bi-variate P-splines. In this section, we further consider two alternative approaches: (i) Yao et al. (2005) which is implemented using the `Matlab` toolbox `PACE` and (ii) Xiao et al. (2016) implemented in `R` using `fpca.face` function of the `refund` package (Huang et al. 2015). We carried out additional simulation studies to compare the performance of variance estimation for two cases - when the functional covariates are observed densely or sparsely and with measurement error. In the interest of space, we only considered the situations where we have a single functional covariate (see Scenario A and B defined in Section 5.1 of the main paper). Table 3 presents the results for the $n = 100$ and $\mathbb{E}_i = \mathbb{E}_i^3$ obtained by fitting the GFCM. As in the table, the results are quite robust to the methods. However, one remark is that `face` is originally not developed for sparsely sampled data, and one might encounter a problem if sparseness of the sampling is significantly increased; for example, we only observe 5%$\sim$10% of the data for each curve.

D.3 Power Performance of the Tests

In Section 5.2.2 of the main paper, we discussed power performance of the proposed tests for sparsely sampled data. When the sampling design is dense, power properties are very similar to power properties corresponding to the sparse design, and we provide the results in Figure 1.

Table 3: Summary of ICP, IL, and R(SE) for sample size 100 and $\mathbb{E}_i = \mathbb{E}_i^3$ based on 1000 simulations. Results are obtained by fitting the GFCM.

Scenario B (true relationship is non-linear)							
	$1-\alpha=0.95$			$1-\alpha=0.90$		$1-\alpha=0.85$	
Method	ICP	IL	R(SE)	ICP	IL	ICP	IL
$m = 81$ (dense design)							
`fpca.sc`	0.946	7.34	[5.53, 9.84]	0.895	6.16	0.844	5.39
`fpca.face`	0.940	7.22	[5.49, 9.82]	0.886	6.06	0.835	5.30
`PACE`	0.943	7.29	[5.43, 9.73]	0.889	6.12	0.839	5.35
$m_i \overset{iid}{\sim} Unif(29, 41)$ (moderately sparse design)							
`fpca.sc`	0.948	7.55	[5.69, 10.67]	0.899	6.34	0.849	5.55
`fpca.face`	0.943	7.37	[5.64, 10.24]	0.890	6.18	0.841	5.41
`PACE`	0.948	7.62	[5.65, 11.12]	0.897	6.40	0.849	5.60
$m_i \overset{iid}{\sim} Unif(20, 31)$ (sparse design)							
`fpca.sc`	0.949	7.69	[5.82, 11.09]	0.900	6.46	0.852	5.65
`fpca.face`	0.939	7.38	[5.59, 11.38]	0.885	6.19	0.834	5.42
`PACE`	0.950	7.85	[5.84, 11.78]	0.902	6.59	0.855	5.77

Scenario A (true relationship is linear)							
	$1-\alpha=0.95$			$1-\alpha=0.90$		$1-\alpha=0.85$	
Method	ICP	IL	R(SE)	ICP	IL	ICP	IL
$m = 81$ (dense design)							
`fpca.sc`	0.943	7.16	[5.41, 9.10]	0.891	6.01	0.839	5.26
`fpca.face`	0.936	7.06	[5.37, 9.37]	0.880	5.92	0.828	5.18
`PACE`	0.939	7.13	[5.31, 9.08]	0.884	5.98	0.833	5.24
$m_i \overset{iid}{\sim} Unif(29, 41)$ (moderately sparse design)							
`fpca.sc`	0.943	7.14	[5.37, 9.36]	0.891	5.99	0.840	5.24
`fpca.face`	0.935	7.03	[5.37, 9.24]	0.879	5.90	0.828	5.16
`PACE`	0.941	7.21	[5.31, 9.76]	0.887	6.05	0.837	5.29
$m_i \overset{iid}{\sim} Unif(20, 31)$ (sparse design)							
`fpca.sc`	0.943	7.14	[5.38, 9.56]	0.891	6.00	0.840	5.25
`fpca.face`	0.925	6.87	[4.94, 10.19]	0.866	5.76	0.812	5.04
`PACE`	0.943	7.27	[5.35, 10.20]	0.890	6.10	0.840	5.34

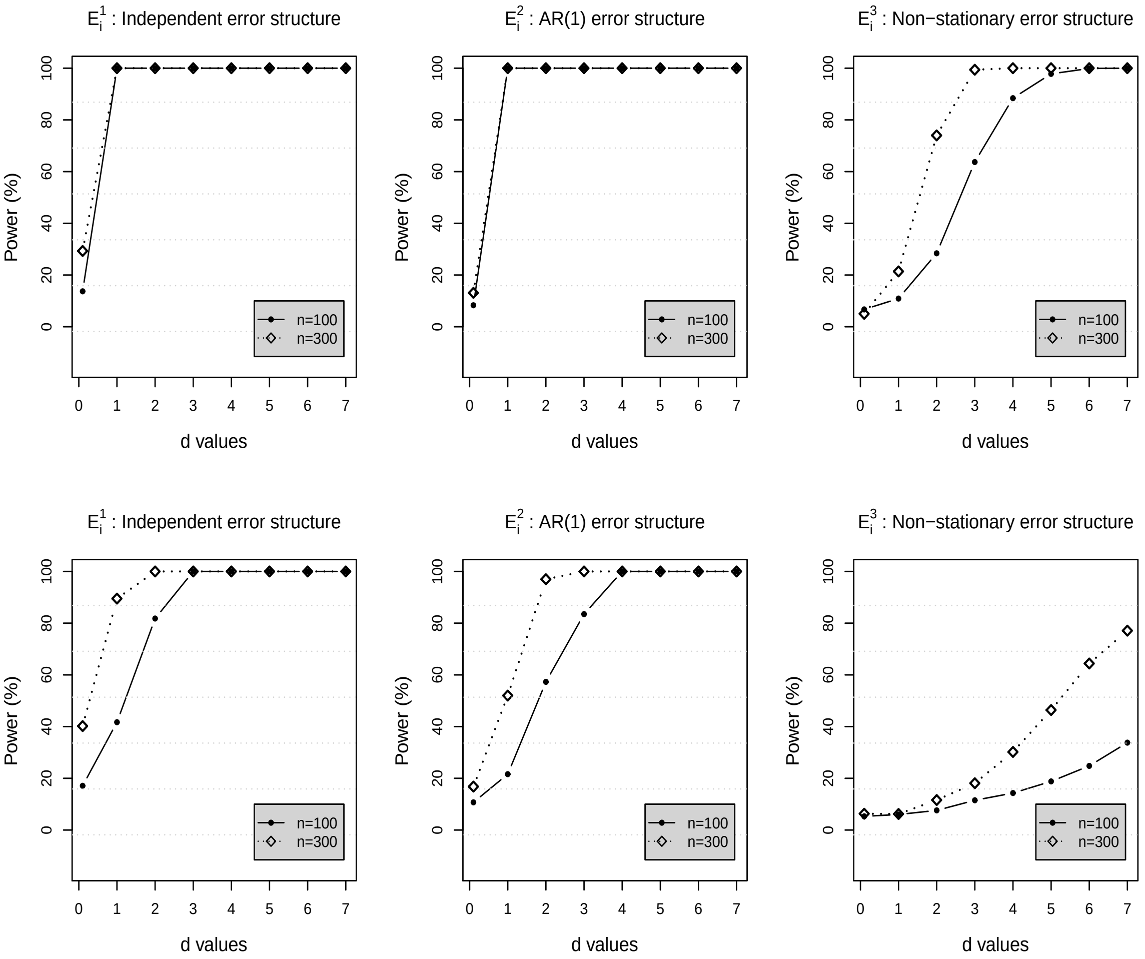

Figure 1: Powers (×100) of the tests at significance level $\alpha = 5\%$. The top (bottom) panel displays the results from scenario A (scenario B) for the setting where sampling design is dense. The error process in the left, middle and right panels is assumed to be $\mathbb{E}_i^1$, $\mathbb{E}_i^2$ and $\mathbb{E}_i^3$, respectively.

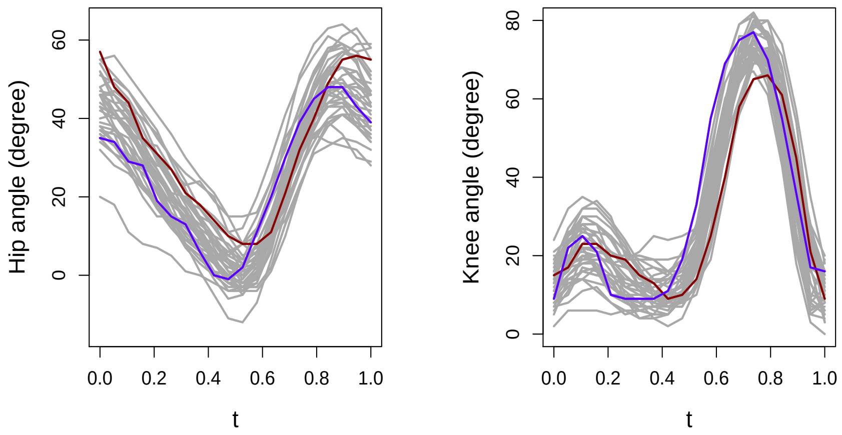

Figure 2: Longitudinal measurements of the hip angle (left) and the knee angle (right) is obtained from 39 children while they go through a single gait cycle.

E Further Investigation of Real Data Examples

E.1 Additional Figures

We present plots of gait data and dietary calcium absorption data illustrated in Section 5.3 of the main paper. Figure 2 displays individual trajectories of the hip angle and the knee angle along the gait cycles in [0, 1] interval. Figure 3 shows the observed individual trajectories of the calcium intake and absorption along the patient's age at the visit.

E.2 Further Investigation of Gait Data Example

The curves in gait data are quite different from the ones used in the simulation studies, and furthermore the sample size is smaller than the one studied in the simulation studies. In this section, we confirm the results of the gait data analysis by investigating the performance of proposed method using a generating model that mimic the feature of the gait data. The purpose of the numerical study is to assess predictive accuracy based on the simulated data sets and to ensure that our method is reliable also for smaller sample sizes; in the gait data example, the sample size is 39.

The new simulation study generates the covariates $X_i(t)$ from a process with the mean and covariance functions that equal their estimated counterparts from the data. For this purpose, we first apply the FPCA to the observed hip angles using the entire 39 subjects,

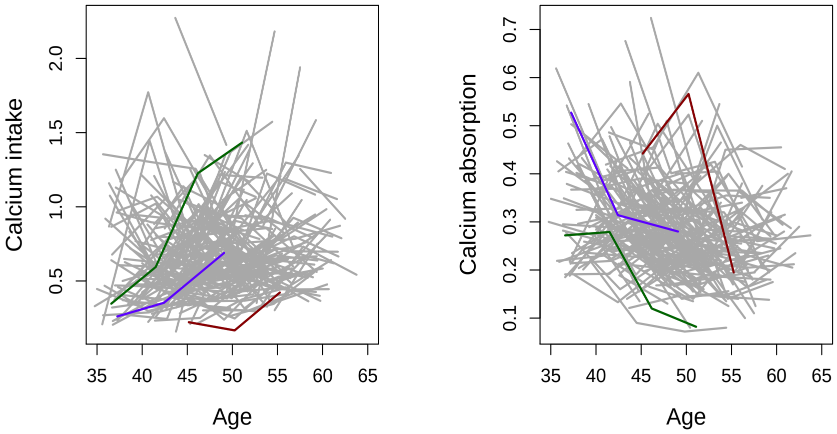

Figure 3: Longitudinal measurements of calcium intake (left) and calcium absorption (right) obtained from 188 patients.

and obtain a smoothed version of n curves by computing $\widehat{X}_i^{\text{sim}}(t_j) = \widehat{\mu}_X(t) + \sum_{j=1}^{K} \xi_{ik}\widehat{\phi}_k(t)$, where ξ_{ik} $(k = 1, \ldots, K)$ are normally distributed with zero-mean and variance $\widehat{\lambda}_k$ $(i = 1, \ldots, n)$. The estimates $\widehat{\mu}_X(t)$, $\widehat{\phi}_k(t)$, and $\widehat{\lambda}_k$ are obtained from the observed data, and the finite truncation K is chosen by setting the percent variance to 99%. It is assumed that hip angles are observed with some noise, and we generate noisy covariate trajectories from $W_{ij}^{\text{sim}} = \widehat{X}_i^{\text{sim}}(t_{ij}) + \delta_{ij}$. The noise δ_{ij} are normally distributed with zero-mean, and the noise variance is estimated from the original data. For the training data, we consider n =30,100, and 300 subjects. For the test data, we consider 9 subjects. For evaluation points t, we use the same time points used in the data analysis.

The response $Y_i(\cdot)$ is generated using two choices for true function $F(\cdot, \cdot)$: a linear version $F^{\text{L,gait}}(x, t)$ and a non-linear version $F^{\text{NL,gait}}(x, t)$. We define the linear version by $F^{\text{L,gait}}(x, t) = \beta_0(t) + \beta_1(t)x$, where $\beta_0(t) = \widehat{\beta}_0^*(t) - \widehat{\beta}_1^*(t)\widehat{\mu}_X(t)/\widehat{\sigma}_X(t)$ and $\beta_1(t) = \widehat{\beta}_1^*(t)/\widehat{\sigma}_X(t)$. Here, $\widehat{\beta}_0^*(t)$ and $\widehat{\beta}_1^*(t)$ are the intercept and the slope estimated from the gait data. Such formulation allows $F^{\text{L,gait}}(x, t)$ to mimic the fitted curve of the gait data. We define the non-linear version by $F^{\text{NL,gait}}(x, t) = \exp(xt/12) - x$. To generate the random errors $\epsilon_i(t)$, we first obtain residuals from the fitted linear model of the gait data, and employ the FPCA methods to estimate the variance of the random errors. Figure 4 displays the simulated covariates $\widehat{X}_i^{\text{sim}}(t)$ $(i = 1, \ldots, 100)$ evaluated at the points t_j (leftmost panel) as well as the response curves $Y_i(t)$ obtained from the models

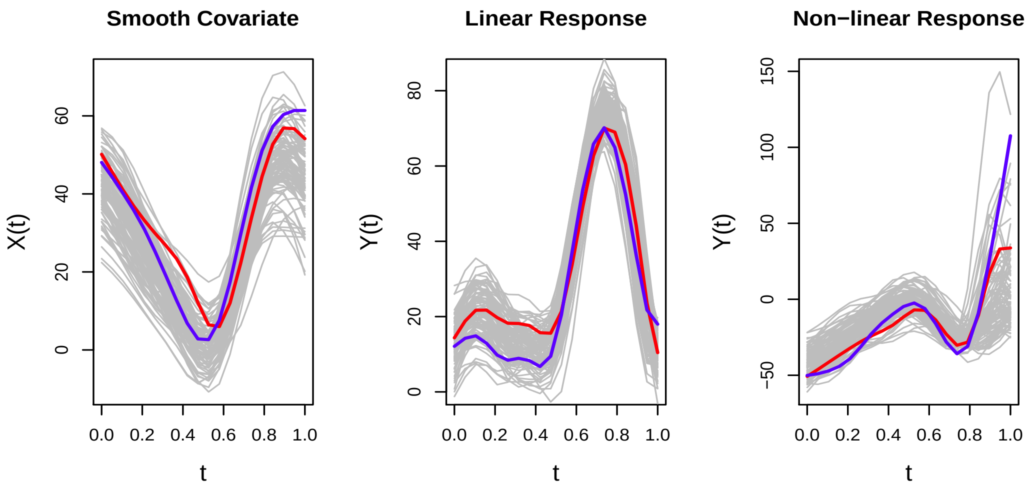

Figure 4: Displayed plots are curves for sample size 100 simulated based on the gait data. The smoothed version of covariate functions $\widehat{X}_i^{\text{sim}}$ are presented in the left. The middle and the rightmost panel present response curves $Y_i(t)$ generated based on $F^{\text{L,gait}}(x,t)$ and $F^{\text{NL,gait}}(x,t)$, respectively. The last two subjects are highlighted in different colors.

$F^{\text{L,gait}}(x,t)$ (middle panel) and $F^{\text{NL,gait}}(x,t)$ (rightmost panel). In the plot, when the true function is $F^{\text{L,gait}}(x,t)$, patterns in the response curves are very similar to the ones from the original gait data.

Finally, we assess the prediction performance of the proposed method using 1000 samples, and compare its performance with the linear FCM. We fit the GFCM using $K_x = K_t = 11$ cubic B-splines for x and t. When the true function is $F^{\text{L,gait}}(x,t)$ (the top three panels in Table 4), the overall predictive performance of the GFCM and the linear FCM is relatively similar. These results indicate that the underlying relationship between the covariate and the response is linear, as we investigated through the gait data analysis in the main paper. When the true function is $F^{\text{NL,gait}}(x,t)$ (the bottom three panels in Table 4), the GFCM better captures the complex non-linear relationships than the linear FCM in all scenarios. Other measures also indicate that the prediction variance obtained from the linear FCM is less accurate. Therefore, when the underlying relationship is complex, the GFCM outperforms the linear FCM.

Table 4: Summary of RMSPE$^{\text{in}}$, RMSPE$^{\text{out}}$, ICP, IL, and R(SE) obtained from the simulation studies of the gait data example. The models fitted by our method and the linear FCM are indicated by `GFCM` and `FCM`, respectively.

(a) True model is $F(x,t) = F^{\mathbf{L,gait}}(x,t)$

							$1-\alpha=0.95$				$1-\alpha=0.90$				$1-\alpha=0.85$			
	RMSPE$^{\text{in}}$		RMSPE$^{\text{out}}$		ICP		IL		R(SE)		ICP		IL		ICP		IL	
n	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM
30	4.83	5.00	5.41	5.26	0.920	0.933	18.98	19.36	[12.96, 33.92]	[13.56, 34.61]	0.862	0.878	15.93	16.25	0.808	0.827	13.84	14.22
100	5.08	5.12	5.20	5.16	0.942	0.945	19.48	19.58	[14.47, 35.24]	[14.57, 35.46]	0.890	0.894	16.35	16.43	0.840	0.844	14.31	14.38
300	5.13	5.14	5.14	5.13	0.947	0.948	19.55	19.59	[14.78, 35.44]	[14.81, 35.50]	0.897	0.898	16.41	16.44	0.847	0.848	14.36	14.39

(b) True model is $F(x,t) = F^{\mathbf{NL,gait}}(x,t)$

							$1-\alpha=0.95$				$1-\alpha=0.90$				$1-\alpha=0.85$			
	RMSPE$^{\text{in}}$		RMSPE$^{\text{out}}$		ICP		IL		R(SE)		ICP		IL		ICP		IL	
n	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM	GFCM	FCM
30	4.89	7.04	6.24	7.74	0.907	0.930	19.69	24.33	[13.21, 35.13]	[13.79, 64.59]	0.846	0.878	16.52	20.42	0.791	0.829	14.46	17.87
100	5.24	7.47	5.67	7.36	0.936	0.945	20.18	25.11	[14.53, 36.26]	[14.90, 69.24]	0.883	0.898	16.94	21.07	0.833	0.852	14.82	18.44
300	5.33	7.59	5.47	7.24	0.945	0.950	20.24	25.33	[14.83, 36.33]	[15.35, 70.92]	0.893	0.904	16.99	21.26	0.844	0.858	14.87	18.60

F Implementation Details

We implemented our proposed estimation and testing methodology using `R` software. The model components of the GFCM, $Y_i(t) = \mu_Y(t) + F_1\{X_{1,i}(t), t\} + F_2\{X_{2,i}(t), t\} + \epsilon_i(t)$, can be estimated using the `gam`/`bam` functions of `mgcv` package (Wood 2015). The smoothing parameter choice is automatic in `gam`/`bam`; we use REML criteria to select the smoothing parameters. For the sparsely sampled design, we employ the `R` package `refund` to carry out FPCA. In the following, we illustrate how the `R` software codes can be used to implement our procedures.

We first pool all observed data. Let `trep` be the N-dimensional vector of evaluation points pooled from all subjects. Let `y.vec` be the N-dimensional vectors of response and transformed covariate, and let and `x1.vec` and `x2.vec` be the transformed covariates of dimension N, respectively, where the evaluation points of the functional correspond to the vector `trep`. The transformed covariate indicates that the point-wise center/scaling transformation is applied. Then a simple command

```
fit <-gam(y.vec ~ s(trep, bs=‘‘ps’’, k=Kmu) + te(x1.vec, trep, bs=‘‘ps’’,
k=c(Kx1, Kt1)) + te(x2.vec, trep, bs=‘‘ps’’, k=c(Kx2, Kt2)), method=‘‘REML’’)
```

performs our estimation procedure. The function `s()` estimates the marginal smooth mean of response, $\mu_Y(t)$, and the number of basis functions for this term is indicated by `Kmu`. The function `te()` specifies the tensor product of basis functions. The `bs` argument selects the type of penalized splines. In our case, we set `bs=‘‘ps’’` to incorporate B-splines with the second order difference penalties. The `k` argument specifies the number of basis functions; for example, when estimating the term $F_2\{X_{1,i}(t), t\}$, the number of basis functions is `Kx1` for `x1.vec` and `Kt1` for `trep`. Smoothing for the penalized splines is indicated by `method=‘‘REML’’`. The `gam()` function will automatically offer the parameter estimates $\widehat{\Theta}$ as well as the estimated response `y.vecEst` at the points `trep`. For the large data sets, one can use `bam()` in place of `gam()`.

To estimate $G(\cdot, \cdot)$ at specific time points, one may use `fpca.sc` function of `refund` package in `R`. The model residuals are computed from `res.vec=y.vec-y.vecEst`; FPCA is then applied to the residuals using the function `fpca.sc` of `R` package `refund` as

```
fpc <- fpca.sc(matrix(res.vec, nrow=n, ncol=m, byrow=TRUE), pve, var=TRUE).
```

Note that the residuals must be transformed into a matrix format in `fpca.sc`. The `pve` argument specifies the percent of variance explained by the first few eigencomponents such as `pve=0.9` or `pve=0.95`. We set `var=TRUE` to estimate the variance of measurement errors σ^2. This procedure offers the estimate of eigencomponents $\{\phi_k(\cdot), \lambda_k\}$ and the estimate of σ^2, which will be used to reconstruct $G = \mathrm{cov}(\mathbb{E}_i)$. For the case where the data are observed on a sparse grid of points, the $n \times m$-dimensional matrix of the residuals contains `NA`s as components, considered as missing values. Nevertheless, the `fpca.sc()` function can still estimate the underlying smooth curves and the eigencomponents.

References

Huang, L., F. Scheipl, J. Goldsmith, J. Gellar, J. Harezlak, M. W. McLean, B. Swihart, L. Xiao, C. Crainiceanu, and P. Reiss (2015). *refund: Regression with Functional Data.* R package version 0.1-13.

McLean, M. W., G. Hooker, A. M. Staicu, F. Scheipl, and D. Ruppert (2014). Functional generalized additive models. *Journal of Computational and Graphical Statistics 23*, 249–269.

Wood, S. N. (2015). *Mixed GAM computation vehicle with GCV/AIC/REML smoothness estimation.* R package version 1.8-7.

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

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