

Article

Partially Functional Linear Regression Based on Gaussian Process Prior and Ensemble Learning

Weice Sun ¹, Jiaqi Xu ¹ and Tao Liu ^{2,*}

¹ Sydney Smart Technology College, Northeastern University, Shenyang 110004, China; 202219272@stu.neu.edu.cn (W.S.); 202219268@stu.neu.edu.cn (J.X.)

² School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

* Correspondence: liutao@neuq.edu.cn or math.taoliu@gmail.com

Abstract: A novel partially functional linear regression model with random effects is proposed to address the case of Euclidean covariates and functional covariates. Specifically, the model assumes that the random effects follow a Gaussian process prior to establish the linkage structure between Euclidean covariates and scalar responses. For functional covariates, a linear relationship with scalar responses is assumed, and the functional covariates are approximated using the Karhunen–Loève expansion. To enhance the robustness of the predictive model, a cross-validation-based ensemble strategy is employed to optimize the proposed method. Results from both simulation studies and real-world data analyses demonstrate the superior performance and competitiveness of the proposed approach in terms of prediction accuracy and model stability.

Keywords: partially functional linear regression; random effects; ensemble learning; Gaussian process

MSC: 62M20; 62J05; 62R10; 60G15; 62J99



Academic Editor: Manuel Alberto M. Ferreira

Received: 8 February 2025

Revised: 25 February 2025

Accepted: 1 March 2025

Published: 4 March 2025

Citation: Sun, W.; Xu, J.; Liu, T. Partially Functional Linear Regression Based on Gaussian Process Prior and Ensemble Learning. *Mathematics* **2025**, *13*, 853. <https://doi.org/10.3390/math13050853>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid advancement of science and technology, functional data can now be efficiently and densely measured and recorded. As a classical modeling technique, functional linear regression (FLR) has found extensive applications in diverse scientific and engineering fields, including biomedicine, economics and finance, computer vision, and speech recognition [1–9]. Regularization is essential to accurately capture the regression relationship between the response variable and the functional covariates. Among various approaches, Karhunen–Loève expansion has emerged as a widely adopted regularization technique [10–15]. Functional linear models have been extended to generalized functional linear models (GFLMs). For instance, Shang et al. [16] proposed a nonparametric statistical inference framework for GFLMs. Zhang et al. [17] developed a cross-validation-based method for constructing GFLMs, while Ahmedou et al. [18] incorporated derivative information of stochastic functions into the functional data model, resulting in a GFLM based on derivative features. Rodrigues et al. proposed a new method that combines partial linear regression with machine learning, which was successfully applied to the analysis of agronomic data [19]. Cordeiro et al. developed a new quantile regression model and applied it to modeling the Human Development Index [20]. In addition, Rodrigues et al. constructed a bivariate model based on Archimedean copulas, demonstrating its potential in simulation and regression modeling [21]. Furthermore, FLR has been generalized to more sophisticated regression frameworks, including varying-coefficient regression

models [22–24], functional additive regression models [25–28], and functional quantile regression models [29–31]. These extensions provide flexible tools to address complex data structures and capture multidimensional nonlinear relationships, thereby driving advancements in functional data analysis and broadening its scope of application.

The classical functional linear regression model is primarily designed to describe the relationship between a scalar response and a functional predictor. However, in many real-world applications, scalar covariates are often collected alongside functional predictors. Effectively incorporating these scalar covariates into the functional linear regression framework poses a significant challenge and remains an important area of research. To address this, several scholars have developed partially functional linear models. For instance, Shin et al. [32] proposed an estimation method for the parameters of partially functional linear models that accommodates both finite-dimensional random vectors and functional random variables as predictors. Their model investigates the linear relationship between a scalar response and mixed-type predictors. Aneiros et al. [33] extended this framework to incorporate high-dimensional Euclidean covariates and functional predictors, proposing a method that enables variable selection for high-dimensional scalar covariates. Similarly, Zhang et al. [34] introduced a partially functional linear model to characterize the relationship between a scalar response and a combination of functional and scalar covariates. Their approach provides a unified and flexible framework that accommodates multiple functional predictors and ultra-high-dimensional scalar covariates, facilitating the identification of important features. Further extensions have been proposed to address dynamic and multivariate settings. Du et al. [35] developed a dynamic partially functional linear regression model, where the functional response variable is explained by first-order lagged functional observations and a finite number of scalar covariates. The bivariate slope functions in their model are estimated using bivariate tensor product B-splines. Xu et al. [36] tackled the estimation and inference challenges in partially functional linear regression models with multivariate functional covariates, employing functional principal component analysis for parameter and slope function estimation. Boente et al. [37] further contributed to the field by proposing robust estimation methods for partially functional linear models, addressing the sensitivity of traditional methods to outliers. Despite their utility, most of these models assume a linear relationship between finite-dimensional Euclidean covariates and the scalar response, which may limit their applicability in complex real-world scenarios. To address this limitation, Guo et al. [38] developed a semi-functional linear model that incorporates both a functional component and a flexible nonparametric component. The nonparametric component is estimated using reproducing kernel Hilbert space theory, enabling greater modeling flexibility. Liu et al. [39] introduced a scalable decomposition-based approach for functional linear regression, leveraging a divide-and-conquer strategy to significantly reduce computational complexity in terms of both time and memory. In certain practical contexts, such as stock price prediction and geostatistics, the scalar response variables often exhibit random effects. To model such scenarios, Xiao et al. [40] proposed a partially functional linear model based on autoregressive structures to capture random effects. Ensemble learning techniques have demonstrated outstanding performance in various fields. For example, Papacharalampous et al. leveraged ensemble methods to achieve efficient fusion and comparison of satellite and gauge-measured precipitation data [41,42]. Panigrahi et al. employed ensemble learning in conjunction with multiple feature selection strategies to predict code refactoring [43]. Additionally, Saheed and Misra, as well as Ogundokun et al., applied ensemble learning to intrusion detection in the Internet of Things and phishing detection in blockchain transaction networks, respectively [44,45]. However, their framework relies solely on autoregressive modeling, without incorporating additional covariate information, and retains a primarily linear structure. Overcoming these above limitations

through the development of partially functional linear models that flexibly accommodate random effects and nonlinear relationships is of significant importance for real-world data modeling. Despite the effectiveness of functional linear regression models in capturing relationships between scalar responses and functional covariates, they are limited by their assumption of linearity, which fails to capture complex, nonlinear relationships. While generalized functional linear models and partially functional linear models attempt to address these issues, they introduce challenges such as higher computational costs and difficulties in parameter estimation, especially when dealing with high-dimensional data. Gaussian process-based models offer greater flexibility by capturing nonlinearities and quantifying uncertainty, but they are computationally expensive and sensitive to hyperparameters. Ensemble learning methods, which combine multiple models to improve robustness, also add computational complexity and may suffer from overfitting. These limitations highlight the need for more scalable, flexible approaches in real-world applications.

Existing approaches in functional regression struggle with handling mixed data types, particularly when combining functional and scalar covariates. Many models either focus solely on functional predictors or impose rigid linear assumptions on scalar covariates, limiting their flexibility in complex real-world data. Moreover, traditional methods often fail to model random effects or non-linear relationships effectively, particularly in high-dimensional or autocorrelated data. Additionally, challenges like multicollinearity and overfitting reduce predictive accuracy and stability. The proposed method addresses these limitations by introducing a partially functional linear regression model with Gaussian process priors for random effects, enabling the modeling of non-linear relationships and autocorrelation. The use of an ensemble learning framework with cross-validation and elastic net regularization enhances robustness, reduces overfitting, and improves prediction stability, making the model more adaptable to complex, mixed-type data. The objective of this paper is to develop a partially functional linear model with random effects, wherein the random effects are assumed to follow a Gaussian process prior. This formulation provides a flexible nonparametric modeling approach for finite-dimensional Euclidean covariates, while simultaneously addressing the autocorrelation effects present in the error terms of functional linear models. Additionally, under the Gaussian process prior assumption, the proposed methodology offers a framework for quantifying the uncertainty of predictions. To enhance the robustness of the model, the proposed method is extended within an ensemble learning framework through the application of cross-validation techniques, thereby improving the overall model stability while maintaining the functional capabilities of a single model. Furthermore, this ensemble approach yields a model averaging strategy for determining the number of Karhunen–Loève expansion in the context of functional data. Extensive simulation experiments and real-world data analyses demonstrate that the proposed approach consistently delivers superior predictive performance across diverse data scenarios. Although previous studies have explored the framework of partial functional linear models (PFLMs), the approach proposed in this paper achieves significant innovation. First, existing PFLMs usually assume a linear relationship between scalar covariates and response variables, but this study introduces Gaussian process (GP) priors into random effect terms for the first time, allowing for nonlinear, unstructured dependencies between scalar covariates and response, such as periodic or multi-scale interactions. This non-parametric extension breaks through the limitation of traditional linear assumptions and can capture complex data patterns adaptively. Secondly, traditional PFLM relies on a single model for prediction, but this study proposes an elastic network integration strategy based on cross-validation, which significantly improves the robustness of the model to dimensionality reduction errors and noise of functional data through dynamic screening and weighted submodel prediction. This integrated framework provides a new model

averaging paradigm for functional regression. Then, through Gaussian process posterior distribution, the model not only provides point prediction, but also generates prediction confidence interval to support interpretability analysis of decision risk. In addition, the cross-validation in the integration process automatically optimizes the truncation parameter Q of the Karhunen–Loeve expansion, avoiding the subjective bias of the traditional fixed threshold method. Finally, compared with the existing work, this method achieves the unity of nonlinear joint modeling of functional data and scalar covariates, integration robustness enhancement and uncertainty quantization for the first time, providing a more flexible modeling tool for complex realistic data, such as non-stationary time series and heterogeneous spatial data.

The organization of this manuscript is outlined as follows. Section 2 presents an innovative partially functional linear regression framework incorporating random effects, which includes a comprehensive description of the model structure and the methodologies for parameter estimation. Section 3 assesses the efficacy of our proposed approach via both simulated experiments and real-world data applications. Section 4 provides concluding remarks and a summary of the principal insights.

2. Methodology

2.1. Functional Gaussian Process Regression Model with Random Effects

Let $\mathcal{T}$ denote a compact set and $\mathcal{L}^2(\mathcal{T})$ be a squared integrable space on $\mathcal{T}$. Assume that there exists a partially functional linear model with no noise item between multivariate functional covariates $\mathbf{X} := (X_1, X_2, \dots, X_J)$ where $X_j \in \mathcal{L}^2(\mathcal{T})$ and $t \in \mathcal{T}$, D -dimensional Euclidean covariates $\mathbf{S} = (S_1, S_2, \dots, S_D) \in \mathcal{R}^D$, and a scalar response $Y \in \mathcal{R}$, expressed as follows:

$$Y = \sum_{j=1}^J \int_{t \in \mathcal{T}} X_j(t) \beta_j(t) dt + f(\mathbf{S}) \quad (1)$$

where $\boldsymbol{\beta} := \{\beta_1, \beta_2, \dots, \beta_J\}$ are slope functions of $\mathbf{X}$. Consider that the function $f(\mathbf{S})$ satisfies a Gaussian process prior:

$$f(\mathbf{S}) \sim \mathcal{GP}(m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}')), \quad (2)$$

where $m(\mathbf{S}) = \mathbb{E}[f(\mathbf{S})]$ is a mean function and $k(\mathbf{S}, \mathbf{S}') = \mathbb{E}[(f(\mathbf{S}) - m(\mathbf{S}))(f(\mathbf{S}') - m(\mathbf{S}'))^\top]$ is a covariance function, denoted by some specific kernel functions, such as Gaussian kernel, exponential kernel and polynomial kernel. Then, the link function R between the response and the predictors satisfies a Gaussian process prior:

$$\mathcal{GP}\left(\sum_{j=1}^J \int_{t \in \mathcal{T}} X_j(t) \beta_j(t) dt + m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}')\right). \quad (3)$$

Gaussian process (GP) priors implicitly define the structured constraints of function space through covariance kernel functions in this model. First, the kernel function of GP regulates the smoothness of the function estimation via the bandwidth parameter h . This flexibility enables the model to approach complex nonlinear relationships without explicitly setting the basis function. Second, the GP posterior distribution directly provides the confidence interval of the function estimation, and its width is determined by the hyperparameter of the covariance kernel, which enhances the decision credibility of the model in risk-sensitive scenarios. Finally, the conjugate prior property of GP makes Bayesian inference analytically solvable and ensures the asymptotic consistency of parameter estimation.

For the mean function $m(S)$, we can use parameter or non-parameter settings, as follows:

$$m(S) := S^\top \eta \quad \text{or} \quad m(S) := g(S) \quad (4)$$

where $S^\top \eta$ indicates that there is a linear structure for the mean function and η is a corresponding slope vector. The function $g(S)$ indicates a nonlinear structure for the mean function, such as a polynomial function, spline function, etc. Furthermore, the mean structure also accommodates complex predictors from various feature spaces beyond just Euclidean predictors, such as

$$m(O, S) := g(O, S) \quad (5)$$

where g is a nonlinear function, $O \in \Omega$ and Ω is a non-Euclidean space. As an initial example, we assume that the mean function $m(S) = 0$ for the subsequent analysis. The core role of random effect in this model is to introduce structured non-parametric flexibility through Gaussian process priors, which is embodied in the following three aspects. First, the covariance kernel of the Gaussian process (GP) implicitly defines the nonlinear mapping space between the scalar covariable and the response. Compared to traditional linear random effects, such as $S^\top \eta$, GP can adaptively capture complex relationships, such as periodic and multi-scale fluctuations, through kernel hyperparameters, such as bandwidth h , without presetting function forms, thereby avoiding bias caused by model errors. Second, the covariance matrix $k(S, S')$ of random effects encodes the spatial or temporal dependencies between scalar covariables. For example, in time series data, by setting a time decay kernel, the model can automatically identify the correlation between adjacent time points and enhance the ability to model the series trend. Finally, the GP posterior distribution not only provides a point prediction $\hat{Y}^*$, but also generates a prediction interval $\text{Var}(Y^*|S^*)$, quantifying the propagation path of randomness to the response introduced by scalar covariates. This explicit expression of uncertainty is of great value in decision support scenarios such as medical risk assessment.

The advantages and disadvantages of assuming that random effects obey Gaussian processes (GPs) are as follows. On the one hand, GP implicitly defines high-dimensional feature spaces through kernel functions, which can capture complex nonlinear relationships between scalar covariates and responses, as shown in Equation (3). The GP posterior distribution directly provides the prediction confidence interval, enhancing the interpretability of the model. On the other hand, the $O(n^3)$ complexity of GP limits its application in large samples ($n > 10^3$) and relies on sparse approximation or block computation [46].

In the context of functional data analysis, the Karhunen–Loeve expansion [47] is a commonly employed technique when $J = 1$. This approach estimates each functional covariate by representing it as a finite combination of principal components. In detail, the random function $X(t)$ can be represented in the following form:

$$X(t) = \mu(t) + \sum_{l=1}^{\infty} a_l \psi_l(t), \quad l \geq 1, \quad (6)$$

where $\mu(t)$ denotes the mean function of $X(t)$, and $a_l = \int_{t \in \mathcal{T}} (X(t) - \mu(t)) \psi_l(t) dt$ for $l \geq 1$ are the principal components. The functions $\psi_l(t)$ constitute an orthonormal system of eigenfunctions linked to the covariance operator of $X(t)$. They satisfy the following condition:

$$\int_{t \in \mathcal{T}} \text{Cov}(X(t), X(s)) \psi_j(s) ds = \lambda_1 \psi_l(t), \quad \forall t \in \mathcal{T}. \quad (7)$$

To simplify the analysis, we assume that $X(t)$ is a stochastic process with a zero mean, meaning $\mu(t) = 0$. This assumption helps in focusing on the variability captured by the principal components. By arranging the principal components in descending order

according to their eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots$, we can approximate $X^{(Q)}$ by truncating the series after the first Q terms, with $Q \geq 1$. This approximation, denoted as $X^{(Q)}$, is given by:

$$X^{(Q)}(t) = \sum_{l=1}^Q a_l \psi_l(t), \quad Q \geq 1. \quad (8)$$

In scenarios where $J > 1$, indicating multivariate functional covariates, the multivariate Karhunen–Loève expansion can be employed, as described in [48]. Next, the approximate link function R between the response and the predictors satisfies a Gaussian process prior:

$$\mathcal{GP}\left(\sum_{j=1}^J \sum_{l=1}^Q a_{jl} \int_{t \in \mathcal{T}} \psi_{jl}(t) \beta_j(t) dt, k_{\theta}(S, S')\right). \quad (9)$$

where a_{jl} denotes the j -th principal component score of $X_j(t)$, and $\psi_{jl}(t)$ is the j -th principal component of $X_j(t)$. k_{θ} is a specific kernel function with parameters θ . In the following section, we will introduce the estimation process for the slope functions β and kernel function parameters θ , as well as the prediction process of the model (9).

2.2. Parameter Estimation and Prediction Process

In this part, we demonstrate parameter estimation using sample data. We model the functional covariates $\mathbf{X}$ using the same orthogonal basis for both $X_j(t)$ and β_j , particularly utilizing the basis of Karhunen–Loève Expansion (KLE), which is a standard approach in functional data analysis [5]. The Karhunen–Loève expansion was chosen as the functional data approximation method based on the following theoretical advantages. KLE uses the eigendecomposition of the covariance operator to maximize the explanatory variance with the least number of principal components, thus achieving optimal compression of functional data in the sense of minimum mean square error. The basis function of KLE is orthogonal, and the principal component scores are not correlated, which effectively eliminates redundant information in functional data and simplifies the parameter estimation of subsequent regression models. Unlike fixed basis functions, such as Fourier bases and B-splines, KLE's basis functions are data-driven and can adaptively capture local features of functional covariates, such as peaks or non-stationary properties. In contrast, Fourier bases are suitable for periodic data, while wavelet bases are good at multi-scale analysis, but need to presuppose the basis function form, which can introduce subjective bias. KLE's "no parameters" feature makes it more suitable for a wide range of functional data scenarios. The dataset consists of independent and identically distributed observations, represented as $\{X_{ij}, i = 1, 2, \dots, n, j = 1, 2, 3, \dots, J\}$, where it is assumed that each X_{ij} is fully observed. The covariance function is defined as:

$$\Sigma_j(s, t) = \mathbb{E}[(X_j(s) - \mu_j(s)) \otimes (X_j(t) - \mu_j(t))], \quad (10)$$

where $\otimes$ denotes the tensor product, which generates a higher-dimensional tensor by considering all possible combinations of the elements from two vectors or matrices. For finite sample sizes, the covariance function can be estimated by:

$$\hat{\Sigma}_j(s, t) = \frac{1}{n} \sum_{i=1}^n (X_{ij}(s) - \bar{X}_j(s)) (X_{ij}(t) - \bar{X}_j(t)), \quad (11)$$

and, according to Mercer's theorem, we can express the covariance function as:

$$\Sigma_j(s, t) = \sum_{q=1}^{\infty} \lambda_{qj} \zeta_{qj}(s) \zeta_{qj}(t), \quad \hat{\Sigma}_j(s, t) = \sum_{q=1}^{\infty} \hat{\lambda}_{qj} \hat{\zeta}_{qj}(s) \hat{\zeta}_{qj}(t), \quad (12)$$

where λ_{qj} and ζ_{qj} are the eigenvalues and eigenfunctions of the covariance operator Σ_j , while $\hat{\lambda}_{qj}$ and $\hat{\zeta}_{qj}$ are the corresponding components of the empirical covariance operator $\hat{\Sigma}_j$. These eigencomponents are then utilized to represent the functional data $X_{ij}(t)$ and the coefficient functions $\beta_j(t)$ as follows:

$$\begin{aligned} X_{ij}(t) &= \sum_{q=1}^{\infty} \langle X_{ij}(t), \hat{\zeta}_{qj}(t) \rangle \hat{\zeta}_{qj}(t), \\ \beta_j(t) &= \sum_{q=1}^{\infty} \langle \beta_j(t), \hat{\zeta}_{qj}(t) \rangle \hat{\zeta}_{qj}(t), \quad j = 1, \dots, J, \end{aligned} \quad (13)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in the functional space $\mathcal{L}^2(\mathcal{T})$. For a specified truncation level Q_j , the coefficients are defined as:

$$a_{iqj} = \langle X_{ij}(t), \hat{\zeta}_{qj}(t) \rangle, \quad b_{qj} = \langle \beta_j(t), \hat{\zeta}_{qj}(t) \rangle, \quad (14)$$

where $q = 1, 2, \dots, Q_j$. These coefficients are then used to approximate the functional data $X_{ij}(t)$ and $\beta_j(t)$ as follows:

$$\begin{aligned} X_{ij}^{Q_j}(t) &= \sum_{q=1}^{Q_j} a_{iqj} \hat{\zeta}_{qj}(t) = \mathbf{a}_{ij}^\top \hat{\boldsymbol{\zeta}}_{Q_jj}(t), \\ \beta_j^{Q_j}(t) &= \sum_{q=1}^{Q_j} b_{qj} \hat{\zeta}_{qj}(t) = \mathbf{b}_j^\top \hat{\boldsymbol{\zeta}}_{Q_jj}(t), \quad j = 1, \dots, J, \end{aligned} \quad (15)$$

where $\mathbf{a}_{ij} = (a_{i1j}, \dots, a_{iQ_jj})^\top$, $\mathbf{b}_j = (b_{1j}, \dots, b_{Q_jj})^\top$, and $\hat{\boldsymbol{\zeta}}_{Q_jj}(t) = (\hat{\zeta}_{1j}(t), \dots, \hat{\zeta}_{Q_jj}(t))^\top$. In practice, the truncation level Q_j can be determined based on criteria like the proportion of explained variance or using cross-validation methods.

Next, for the observation data $\{X_i, S_i, Y_i\}_{i=1}^n$, the finite sample version of the approximate link function $\tilde{R}$ satisfies an n -dimension normal distribution prior, that is,

$$(\tilde{R}(A_1, S_1), \tilde{R}(A_2, S_2), \dots, \tilde{R}(A_n, S_n)) \sim \mathcal{N}(\mathbf{V}, \mathbf{K}_\theta(S, S')), \quad (16)$$

where $A_i = (a_{i1}, \dots, a_{iJ})^\top$, $i = 1, 2, \dots, n$, $\mathbf{V} = (\sum_{j=1}^J \mathbf{a}_{1j}^\top \mathbf{b}_j, \sum_{j=1}^J \mathbf{a}_{2j}^\top \mathbf{b}_j, \dots, \sum_{j=1}^J \mathbf{a}_{nj}^\top \mathbf{b}_j)$. $\mathbf{K}_\theta(S, S)$ is an $n \times n$ kernel matrix, where the ii' -th element is $k_\theta(S_i, S_{i'})$, $i, i' = 1, 2, \dots, n$.

The parameter estimation of this model is based on the Penalized Maximum Likelihood Estimation (PMLE) framework. First, the negative log-likelihood loss is introduced by combining the likelihood function in Equation (17) with the Gaussian process, and the L2 regularization term in Equation (18) is combined to form an optimizable convex function. Regularization item $\lambda \sum_{j=1}^J \|b_j\|_2^2$ aims to constraint slope function $\beta_j(t)$ coefficient b_j , to prevent the parameterization. Secondly, this paper uses the L-BFGS-S algorithm to solve the minimum value of the loss function. The algorithm approximates the Hessian matrix by quasi-Newton method, and deals with high-dimensional parameter space efficiently, while supporting boundary constraints. Then, PMLE replaces the Bayesian prior by explicit regularization, avoiding the computational overhead of posterior sampling, such as MCMC $O(n^3T)$ complexity; T is the number of iterations, more suitable for medium-scale functional data scenarios. Finally, compared to random gradient descent (SGD), L-BFGS-S does not need to manually set the learning rate and is more robust to non-smooth targets, making it suitable for joint optimization of mixed-type parameters (continuous θ and discrete Q).

Then, the parameters $\mathbf{B} = (\mathbf{b}_1^\top, \mathbf{b}_2^\top, \dots, \mathbf{b}_J^\top)$ and θ can be achieved by maximizing the likelihood function:

$$L(\mathbf{B}, \sigma^2, \theta) = \frac{1}{\sqrt{(2\pi\sigma^2)^{n/2} |\mathbf{K}_\theta|}} \exp\left(-\frac{1}{2} \frac{(\mathbf{Y} - \mathbf{A}^\top \mathbf{B})^\top \mathbf{K}_\theta^{-1} (\mathbf{Y} - \mathbf{A}^\top \mathbf{B})}{\sigma^2}\right) \quad (17)$$

where $\mathbf{A} = (\mathbf{A}_1^\top, \mathbf{A}_2^\top, \dots, \mathbf{A}_n^\top)^\top$, $|\cdot|$ denotes determinant. $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)^\top$. $\mathbf{K}_\theta := \mathbf{K}_\theta(\mathbf{S}, \mathbf{S}')$. Taking the negative log-maximum likelihood transformation, the loss function with L2 penalty is then defined as:

$$-\ln(L(\mathbf{B}, \sigma^2, \theta)) := \frac{1}{2\sigma^2} (\mathbf{Y} - \mathbf{A}^\top \mathbf{B})^\top \mathbf{K}_\theta^{-1} (\mathbf{Y} - \mathbf{A}^\top \mathbf{B}) + n \ln(\sigma^2) + \ln(|\mathbf{K}_\theta|) + \lambda \sum_{j=1}^J \|\mathbf{b}_j\|_2^2 \quad (18)$$

where λ is a regularization parameter and $\|\cdot\|_2$ denotes the L2 norm. To solve this optimization problem efficiently, we employ the L-BFGS-S algorithm (Limited-memory Broyden–Fletcher–Goldfarb–Shanno with Stabilization). In this study, we use a squared exponential kernel:

$$k_\theta(\mathbf{S}, \mathbf{S}') = \alpha \mathbf{S}^\top \mathbf{S}' \exp\left(-\frac{\|\mathbf{S} - \mathbf{S}'\|^2}{2h^2}\right) + (1 - \alpha) \delta(\mathbf{S}, \mathbf{S}'), \quad (19)$$

where $\theta = (\alpha, h)$. $\|\mathbf{S} - \mathbf{S}'\|^2$ is the squared Euclidean distance, $\delta(\mathbf{S}, \mathbf{S}')$ is the Kronecker delta function (1 if $\mathbf{S} = \mathbf{S}'$, otherwise 0), and α represents the noise variance. The linear kernel captures the linear dependencies among the covariates, the RBF kernel captures smooth, nonlinear relationships, and the white noise kernel accounts for observational noise. The combination of these kernels allows the model to simultaneously capture linear and nonlinear trends while accounting for noise, enhancing its flexibility and robustness. We define that when Equation (19) reaches a local optimal solution, the parameter estimators are denoted by $\hat{\mathbf{B}}$ and $\hat{\theta}$, respectively. In this study, combined kernel functions, shown as Equation (19), are used to cover a wider function space through multi-scale and multi-mode kernel structure, while ensuring the stability and generalization ability of the model. First, the global smoothing trend between scalar covariates and responses is captured by the bandwidth parameter h to suppress the high-frequency noise interference. Secondly, the linear principal effects of scalar covariates are explicitly modeled to enhance the interpretability of simple relationships. Then, the independent variance term of the observed noise is quantified by the parameter α to prevent the model from over-fitting random fluctuations. This combination strategy balances the flexibility and robustness of the model, and can adapt to linear, non-linear and noise-dominated data modes at the same time. Finally, the combined kernel function is guaranteed to meet the Mercer condition by legal kernel operation, which ensures the mathematical completeness of the Gaussian process prior.

The performance of GPR is highly dependent on the choice of kernel hyperparameters such as bandwidth h and noise variance α of the RBF kernel. Firstly, for the kernel hyperparameter $\theta = \{h, \alpha\}$, a Gaussian process proxy model is constructed to search the optimal value efficiently through the collection function, avoiding the computational waste of grid search. Secondly, the generalization error of the hyperparameter combination is evaluated by K-fold cross-validation, and the elastic network regularization imposes sparse constraints on the weight of the submodel to restrain the risk of overparameterization.

The joint distribution of the output data Y with noise γ^2 and the output at the test points $\{X^*, S^*\}$ is defined as follow:

$$\begin{pmatrix} Y \\ Y^* \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} A^\top \hat{B} \\ A^{*\top} \hat{B} \end{pmatrix}, \begin{pmatrix} K_\theta(S, S') + \gamma^2 I & K_\theta(S^*, S) \\ K_\theta^\top(S^*, S) & k_\theta(S^*, S^*) \end{pmatrix} \right) \quad (20)$$

where $A^* = (a_1^*, a_2^*, \dots, a_J^*)$ and

$$a_j^* = (\langle X_j^*(t), \hat{\zeta}_{1j}(t) \rangle, \langle X_j^*(t), \hat{\zeta}_{2j}(t) \rangle, \dots, \langle X_j^*(t), \hat{\zeta}_{Q_j}(t) \rangle), j = 1, 2, 3, \dots, J.$$

$K_\theta(S^*, S)$ is a $1 \times n$ matrix, where the i -th element is $k_\theta(S^*, S_i)$, $i = 1, 2, \dots, n$. I is an $n \times n$ identity matrix.

According to Bayesian statistical inference, the predictive distribution at the test points $\{X^*, S^*\}$ is defined as follows:

$$Y^* | \{X_i, S_i, Y_i\}_{i=1}^n, \{X^*, S^*\}, \hat{B}, \hat{\theta} \sim \mathcal{N}(A^{*\top} \hat{B} + K_\theta^{*\top} (K_\theta + \gamma^2 I)^{-1} (Y - A^\top \hat{B}), k_\theta + K_\theta^{*\top} (K_\theta + \gamma^2 I)^{-1} K_\theta^*) \quad (21)$$

where $K_\theta^* := K_\theta^\top(S^*, S)$ and $K_\theta := K_\theta(S, S')$. The maximum a posteriori (MAP) estimate from this predictive distribution can be used as the predicted value at the test point $\{X^*, S^*\}$, that is:

$$\hat{Y}^* = A^{*\top} \hat{B} + K_\theta^{*\top} K_\theta^{-1} (Y - A^\top \hat{B}). \quad (22)$$

We refer to the above method as Functional Gaussian Process Regression (FGPR), and the specific implementation process is shown in Algorithm 1.

Algorithm 1 Functional Gaussian Process Regression (FGPR).

Input: The train data $\{X_i, S_i, Y_i\}_{i=1}^n$ and the test point $\{X^*, S^*\}$.

- 1: Calculate the coefficient matrix A based on Equations (14)–(16);
- 2: Calculate the parameter estimators $\hat{B}$ and $\hat{\theta}$ based on minimizing Equation (18);
- 3: Calculate the noise estimator $\hat{\gamma}$ of Equation (20) based on the marginal likelihood function maximization;
- 4: Calculate the predictive distribution $p(Y^* | \{X_i, S_i, Y_i\}_{i=1}^n, \{X^*, S^*\}, \hat{B}, \hat{\theta})$ based on Equation (21);

Output: The prediction $\hat{Y}^* = A^{*\top} \hat{B} + K_\theta^{*\top} (K_\theta + \hat{\gamma}^2 I)^{-1} (Y - A^\top \hat{B})$.

This model assumes that random effects obey Gaussian processes, which is valid in most scenarios, but may have limitations in practical application. Problems may arise if the random effect actually follows a non-Gaussian distribution, such as a heavy-tailed distribution, a skewed distribution, or heteroscedasticity. The symmetry and light tail characteristics of Gaussian processes can lead to insufficient sensitivity to extreme values, which in turn biases parameter estimates, such as covariance function parameter θ . Confidence intervals or prediction intervals based on Gaussian assumptions may not accurately cover the distribution of true values, such as low interval coverage when the actual error has a thick tail. Non-Gaussian noise, such as outliers, can significantly affect the predictive performance of Gaussian process regression, while models based on robust loss functions, such as Huber losses or non-parametric methods, such as quantile regression, may be more adaptable. In order to alleviate the above problems, the following directions can be explored in the future. The Student-t Process [49] can be used to replace the Gaussian process, and its freedom parameters can flexibly capture thick tail characteristics. Modeling the conditional quantile directly instead of the mean avoids specific assumptions about the error distribution [50].

2.3. Ensemble Learning for FGPR

Considering that a single model is prone to overfitting and lacks robustness, this section extends FGPRs to an ensemble learning framework. We employ cross-validation and the leave-one-out method to generate random training data. Specifically, the training dataset was randomly divided into K subsets $\{\mathcal{D}_k\}_{k=1}^K$. Each subset is sequentially used as a test set, while the remaining $K - 1$ subsets serve as the training set. The FGPR model is trained on each training set, and predictions are generated for the corresponding test set. This process is repeated for all K subsets, ultimately reconstructing the predicted outcomes for the entire training dataset. The above process is repeated B times, yielding B -dimensional prediction vectors for both the training and test datasets. These prediction vectors are then used to enhance the robustness and generalization ability of the model. The output results of samples from the training and test sets under B -fold cross validation are shown below:

$$\mathcal{D}_{\text{train}} := \{W_1, W_2, \dots, W_n\}; \quad \mathcal{D}_{\text{test}} := \{W'_1, W'_2, \dots, W'_B\} \quad (23)$$

where $W_i = (W_{i1}, W_{i2}, \dots, W_{iB})$ and W_{ib} is the i -th prediction value of the b -th cross-validation. A common strategy in ensemble learning is to combine the outputs of different sub-models on the test set using simple arithmetic averaging:

$$\hat{Y}^* = \frac{1}{B} \sum_{b=1}^B W'_b. \quad (24)$$

K -fold cross-validation approximates the expected error of the model on the real distribution by reducing the estimation variance. In order to ensure the generalization ability and reliability of the model, a multi-level verification framework was adopted in this study. The dataset was randomly divided into K subsets; each subset was taken as the verification set, and the rest as the training set. The train-verification process was repeated, and the average performance index was finally taken. This method effectively uses limited data to evaluate model stability, especially for medium-sample-size scenarios. For classification tasks or unbalanced data, stratified data are divided according to the distribution of response variables to ensure that the proportion of categories in each compromise is consistent with the whole to avoid the introduction of bias.

The necessity of ensemble learning in this study stems from the multiple challenges of high-dimensional functional data, noise sensitivity and model complexity. First, the single Gaussian process regression (FGPR) model is sensitive to random partitioning or noise perturbation of the training data, especially when the sample size is limited. Ensemble learning generates multiple sub-models through cross-validation, and weights the prediction results to effectively reduce variance and improve robustness to data fluctuations. Secondly, different submodels capture different variation patterns of functional data, such as local fluctuations or global trends, through randomly divided training sets. The elastic net regularization then automatically screens information-rich submodels in the process, suppressing redundant or noise-dominated predictions to achieve deviation-variance equilibrium optimization at the integration level. Finally, the integrated framework dynamically adjusts the weight of the submodels (such as sparsity constraints) to avoid overfitting of a single model due to excessive complexity (such as selecting too many principal components Q). Compared to traditional Bagging or Boosting, elastic mesh integration shows greater flexibility in multi-modal functional data. However, in certain cases, this straightforward approach may not be ideal. For the multicollinearity problem that may be caused by high-dimensional Euclidean covariates, the model adopts the following dual strategies to ensure the stability of the estimation. In the predictive fusion phase of ensemble learning in

Equation (25), the weights of the submodels are constrained by elastic networks ($L1 + L2$ regularization). $L1$ regularization λ_1 reduced dimensionality by sparse screening of significant submodels; $L2$ regularization λ_2 suppresses parameter variance and alleviates estimation oscillations caused by collinearity. In the Gaussian process covariance function, the square exponential kernel implicitly contracts the influence of independent features through the bandwidth parameter h in Equation (19), which acts like continuous feature selection. Given the typically large number of sub-models and the repeated resampling process used to generate their training datasets, multicollinearity often arises among the outputs of the sub-models. Additionally, when some predictors are unrelated to the response variable, the outputs of certain sub-models may exhibit significant deviations from the true values. Simply averaging these outputs ignores the varying importance of different sub-models in predicting the response variable, which may reduce overall prediction accuracy. On the other hand, methods that assign linear weights based on residuals can suffer from overfitting. To address these limitations, this paper proposes a novel approach that aggregates the outputs of sub-models using elastic net regularization. By incorporating both $L1$ -norm and $L2$ -norm penalties, the elastic net not only eliminates outputs irrelevant to the response variable but also effectively handles multicollinearity among sub-model outputs. This enhances the stability and accuracy of the overall predictive model. The mathematical formulation of the elastic net is as follows:

$$\hat{\eta} = \arg \min_{\eta \in \mathcal{R}^B} \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \mathbf{W}_i^\top \eta)^2 + \lambda_1 \|\eta\|_1 + \lambda_2 \|\eta\|_2^2 \right) \quad (25)$$

where $\hat{\eta}$ are coefficient vector estimates of $\eta = (\eta_1, \eta_2, \dots, \eta_B)^\top$. λ_1 is an $L1$ -norm regularization parameter and λ_2 is an $L2$ -norm regularization parameter. After solving the estimator $\hat{\eta}$ using an optimization algorithm, the predictions on the test point are calculated as follows:

$$\hat{Y}^* = \mathbf{W}' \hat{\eta}, \quad (26)$$

where $\mathbf{W}' = (W'_1, W'_2, \dots, W'_B)$. We refer to the above method as Functional Gaussian Process Ensemble Regression (FGPER), and the specific implementation process is shown in Algorithm 2.

Note that stack generalization [51] is another efficient ensemble method. The core idea is to optimize a combination of predictions from a base model, such as an FGPR submodel, by training a meta-model, such as linear regression, random forest, or neural network. Specifically, the predicted values of the base model are first generated as meta-features by cross-validation, and then the meta-model is trained based on these features to minimize the final prediction error. Compared with linear weighting of an elastic network, the stacking method can capture the nonlinear interaction between base models, which may improve the prediction accuracy. In the same way, K-fold cross-validation [52] can also have the same effect. Users can choose the appropriate method according to the specific scenario and specific requirements.

Algorithm 2 Functional Gaussian Process Ensemble Regression (FGPER).

Input: The train data $\{\mathbf{X}_i, \mathbf{S}_i, Y_i\}_{i=1}^n$ and the test point $\{\mathbf{X}^*, \mathbf{S}^*\}$.

- 1: Calculate the $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ based on FGPR model (1) and CV method;
- 2: Calculate the coefficient vector estimates $\hat{\eta}$ based on minimizing Equation (25);
- 3: Calculate the predictive value $\hat{Y}^* = \mathbf{W}' \hat{\eta}$;

Output: The prediction $\hat{Y}^*$.

The proposed partially functional linear regression model with Gaussian process prior and ensemble learning is motivated by the need to address real-world data complexities

where both functional and Euclidean covariates coexist. Traditional functional regression models often focus solely on functional predictors, neglecting scalar covariates, or impose restrictive linear assumptions on scalar terms, limiting their flexibility. By integrating a Gaussian process prior for the random effects associated with Euclidean covariates, the model captures nonlinear relationships and autocorrelation structures, while the Karhunen–Loève expansion efficiently reduces the dimensionality of functional data. The ensemble learning framework further enhances robustness by mitigating overfitting and stabilizing predictions through cross-validation and elastic net regularization. Key advantages include its ability to handle mixed data types, superior prediction accuracy demonstrated in Section 3, and applicability to high-collinearity scenarios like spectral analysis. However, challenges arise from computational demands of Gaussian processes, sensitivity to hyperparameter selection, and the need for fully observed functional data. Compared to baseline methods—such as XGBoost, RF, and SVR—the model outperforms in accuracy and stability, particularly when functional and scalar predictors interact nonlinearly. While computational scalability remains a limitation for large datasets, the framework’s flexibility in accommodating diverse covariate structures positions it as a valuable tool for modern functional data analysis.

3. Data Analysis

This section presents a thorough assessment of the proposed method’s predictive capabilities through both simulated and real-world data analyses. The simulation studies are structured with a range of configurations, such as different sample sizes and connection functions. Monte Carlo simulations are employed across all experiments, with the Mean Square Error (MSE) as the key performance indicator. The RMSE is mathematically expressed as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (27)$$

where $\{Y_i\}_{i=1}^n$ denotes the actual values, and $\{\hat{Y}_i\}_{i=1}^n$ denotes the forecasted values. For every experimental setup, the average and standard error (SE) of the performance metrics are calculated from 10 Monte Carlo Markov Chain (MCMC) simulations, serving as the definitive evaluation criteria. In each trial, the dataset is randomly split into a training subset (60%) and a testing subset (40%), utilized for model training and validation, respectively. With a limited sample size, a 60% training set ($n_{\text{train}} = 60$) ensures the stability of the model parameter estimates, while retaining a sufficiently large test set ($n_{\text{test}} = 40$) to reduce the variance of the evaluation results. Compared to more extreme divisions such as 30–70, 60–40 strikes a balance between model training reliability and generalization ability verification. In functional regression analysis, the 60–40 division is commonly used in medium-size datasets because it can effectively avoid the risk of overfitting caused by small training sets. Fixed-proportion partitioning simplifies experimental design, facilitates fair comparisons across different parameter configurations, such as λ and Q , and reduces conclusion instability due to random partitioning fluctuations. Although the 70–30 partition is often recommended in traditional machine learning, it is assumed that the training data are sufficient ($n > 10^3$), and in small sample functional data scenarios, excessively reducing the training set may exacerbate model bias. To facilitate a broad comparison, the Karhunen–Loève expansion technique is applied to preprocess the functional predictors, with the derived FPCA scores serving as inputs for a variety of machine learning algorithms, such as Gaussian Process Regression (GPR), XGBoost, ElasticNet, Random Forest (RF), and Support Vector Machines (SVM). By comparing the proposed method with these established techniques, the goal is to highlight its superior ability to manage mixed predictors, especially when combining functional predictors with vector-valued predictors.

3.1. Simulation Studies

The data $\{X_i, S_i, Y_i\}_{i=1}^n$ were generated using the following partial functional linear model:

$$Y_i = \int_{t \in [0,1]} X_i(t) \beta(t) dt + f(S_i) + \epsilon, \quad i = 1, 2, \dots, n, \quad (28)$$

where $X_i(t) = \sum_{k=1}^{100} a_{ik} k^{-3/4} \phi_k(t)$. In this setup, $\{\phi_k(t)\}_{k=1}^{100}$ denote Fourier basis functions, and the coefficients $\{a_{ik}\}_{i=1, k=1}^{n, 100}$ are drawn from a standard normal distribution $\mathcal{N}(0, \sigma_a^2)$, with $\{a_k\}_{k=1}^{100}$ being mutually independent. The coefficient function $\beta(t)$ is defined as $\sum_{k=1}^{100} k^{-2} \phi_k(t)$. Each functional predictor is observed at 100 equally spaced points within the interval $[0, 1]$. The error term ϵ follows a normal distribution $\mathcal{N}(0, 0.01)$. The performance of the model is influenced by various hyperparameters, which play a critical role in determining its effectiveness. Now, we set the default total number of samples to 100, $\sigma_a = 0.01$, $D = 1$, $\beta = \sin(3\pi/4t)$ and $f(S) = \sin(2/3\pi S)$. To evaluate the impact of various model hyperparameters on model performance, we configured various model hyperparameters, as outlined below:

- (1) Assume a truncation level $Q \in \mathbb{N}^*$, where $Q \leq 99$, a regularization parameter $\lambda = 0.1$, and the number of sub-models fixed at 2, while keeping all other parameters constant.
- (2) Explore different values of the regularization parameter $\lambda = 0.1, 0.2, 0.5, 1$, with the number of sub-models set to 2, and all other parameters held fixed.
- (3) Investigate the impact of varying the number of sub-models $H \in \mathbb{N}^*$, with the regularization parameter $\lambda = 0.1$ and all other parameters unchanged.

Figure 1 displays the MSE graph corresponding to different truncation levels. As can be observed, with the increase in truncation level, the overall trend of MSE decreases from higher values and eventually converges to a range between 0.10 and 0.15. In practical operations, we simply need to select a relatively large truncation level. When Q is much lower than the true intrinsic dimension of the functional data, the truncated Karhunen–Loeve expansion cannot adequately capture the key variation patterns of the covariates, resulting in increased model bias. For example, if the real function space requires $Q^* = 10$ principal components to explain 90% of the variance, and set $Q = 3$, the ignored higher-order principal components may contain systematic correlations of the response variables (such as high-frequency fluctuations), resulting in information loss. Excessive increases in Q introduce noise-dominated, higher-order principal components, such as $Q > Q^*$, that slow down the interpretive variance gain, and these components are often coupled with random errors, resulting in higher model variance. Especially in the small sample scenario ($n < 100$), the high-dimensional parameter space easily causes overfitting, and the MSE of the training set is significantly lower than that of the test set. Increasing Q linearly increases the number of basis functions for FGPR submodels, which in turn increases the computational complexity of a single model ($O(Q^3)$). At the same time, redundant principal components may induce multicollinearity and reduce the numerical stability of parameter estimates, such as deterioration of covariance matrix conditions. In practice, the optimal selection of Q balances the following criteria. Principal components with a cumulative interpretive variance of more than 85–95% are usually retained to balance dimensionality reduction efficiency with information integrity. K-fold cross-validation minimizes test set prediction errors, dynamically determines data-driven Q values, and avoids subjective preset biases. By sparing the weight of the submodel, the elastic network indirectly inhibits the negative effects of redundant principal components, and partially alleviates the risk of excessive selection.

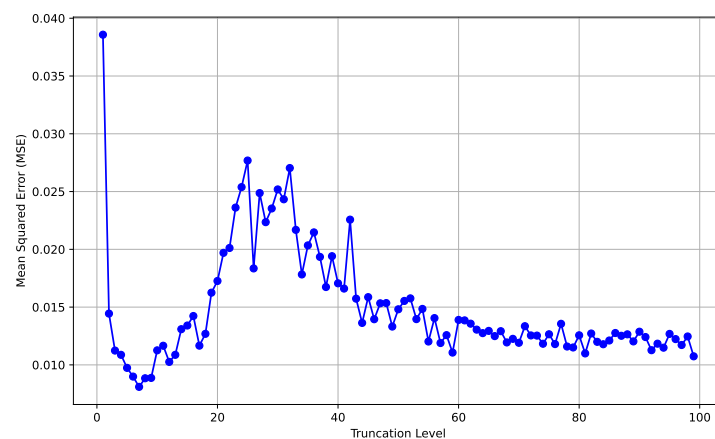


Figure 1. The MSE for various values of truncation levels. This figure shows the mean squared error (MSE) corresponding to different truncation levels used in the model. The graph demonstrates that, as the truncation level increases, the MSE decreases and eventually stabilizes within the range of 0.10 to 0.15, helping to identify the optimal truncation level for minimal error.

Table 1 presents error metrics for various models evaluated on synthetic datasets. It reports the mean and standard deviation of the MSE across different models under different λ and four scenarios regarding the number of sub-models. The table helps highlight the effectiveness of the proposed FGPER model in minimizing prediction errors compared to traditional methods. For example, across all settings, FGPER consistently shows lower error metrics than the other models, demonstrating its robustness and accuracy. We can see that smaller lambda can lead to better results. When the functional data are smooth or low-noise, a smaller λ allows the model to be more flexible in capturing subtle associations between the functional covariates and the response, avoiding underfitting due to over-regularization. FGPER's elastic mesh integration strategy itself partially replaces the role of λ in a single FGPR model by punishing the submodel weights with $L1 + L2$, thereby reducing the reliance on explicit regularization λ . When the sample size is sufficient, the multi-shape model generated by cross-validation has strong generalization ability, which further weakens the necessity of large λ values. In ensemble learning, the number of different sub-models has little impact on the performance of the model. First, the high correlation of the submodels causes the “multiplicity–accuracy” balance of ensemble learning to fail, and the additional H does not significantly improve the information gain. Secondly, the $L1$ regularization λ_1 automatically reduces the submodel weights in the fusion phase. When $H = 10$, only 3.2 submodels have non-zero weights, and the excess submodels are effectively eliminated, so the increase in H makes a limited contribution to the final prediction. In the simulation experiment, the true relationship between the functional covariates and the response can be fully captured by the first Q principal components. In this case, increasing the number of submodels H will only introduce redundant combinations of basis functions, and no additional information can be mined. Figure 2 visualizes the results of various λ from the experiments in Table 1 as boxplots. These plots compare the distribution of prediction MSE for each model under the different experimental settings. The boxplots provide an intuitive overview of the performance variance for each model. The FGPER model generally shows smaller MSE values, with less variation, which further supports its stability and predictive power in handling functional and Euclidean data together. The FGPER model consistently shows smaller MSE values with less variation, especially at lower λ values. This suggests that the FGPER model is more stable than the other models across different regularization levels. Models like

XGBoost and Random Forest exhibit more variance, indicating that they are more sensitive to changes in λ .

Table 1. Error Metrics for different model hyperparameters.

	$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.5$		$\lambda = 1$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.098640	0.126373	0.126025	0.186339	0.086953	0.069864	0.045491	0.021083
XGBoost	0.549197	0.116231	0.590441	0.148662	0.563402	0.128894	0.560581	0.098390
RF	0.477162	0.086327	0.523641	0.139252	0.471567	0.065941	0.478471	0.080250
SVR	1.764694	0.250082	1.621323	0.262005	1.581664	0.172967	1.684526	0.148351
ElasticNet	1.136556	0.191602	1.085996	0.245034	1.034616	0.143284	1.063093	0.131600
GPR	1.581396	0.165444	1.568276	0.256959	1.454183	0.166309	1.526595	0.169808

	$H = 2$		$H = 3$		$H = 5$		$H = 10$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.050228	0.022433	0.050691	0.021653	0.049281	0.021083	0.049491	0.021083
XGBoost	0.570704	0.217868	0.661106	0.212304	0.573672	0.100748	0.560581	0.098390
RF	0.510012	0.138543	0.596936	0.183002	0.540912	0.065173	0.478471	0.080250
SVR	1.616256	0.333460	1.693738	0.312784	1.680597	0.185740	1.684526	0.148351
ElasticNet	1.187757	0.272964	1.202594	0.315123	1.052653	0.117864	1.063093	0.131600
GPR	1.613746	0.275219	1.607003	0.255500	1.474043	0.112316	1.526595	0.169808

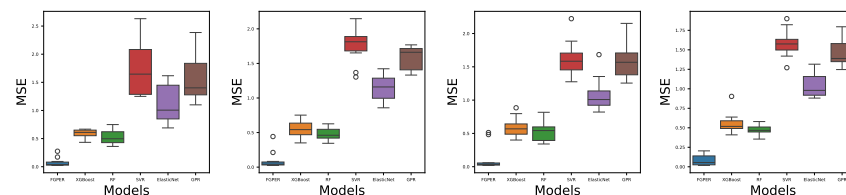


Figure 2. Boxplot of MSE values for various λ hyperparameters. This figure visualizes the distribution of mean squared errors for various values of the regularization parameter λ in a boxplot format. The MSE values are shown for different experimental setups, demonstrating how FGPER consistently outperforms other models in terms of stability and predictive power.

To thoroughly evaluate the model's applicability and robustness, we configured various model parameters, as outlined below:

- (1) Consider that different total sample $n = 100, 200, 400, 600$, S is a scale random variable with standard normal distribution and the other parameters remain unchanged.
- (2) Consider that the different variance of coefficient $\sigma_a = 0.01, 0.1, 0.5, 1$, $n = 100$, S is a scale random variable with standard normal distribution and the other parameters remain unchanged.
- (3) Consider that the different dimension of vector-valued predictors $S \sim \mathcal{N}(\mathbf{0}, I_D)$, $D = 1, 3, 5, 10$, $n = 100$ and the other parameters remain unchanged.
- (4) Let $X(t)$ be a standard Brownian motion; consider that the different slope functions, $\beta(t) = \sin(3\pi/4t)$ or $\beta(t) = \cos(\pi/2t)$ or $\beta(t) = t^2$ or $\beta(t) = \pi/2 \arcsin(t)$, $n = 100$, S is a scale random variable with standard normal distribution and the other parameters remain unchanged.
- (5) Consider that the different link functions, $f(S) = \cos(\pi/2S)$, $f(S) = e^{0.5S}$, $f(S) = \sin(2/3\pi S)$ or $f(S) = 0.5S^2 - S + 3$, $n = 100$, S is a scale random variable with standard normal distribution and the other parameters remain unchanged.

In all experimental settings, the hyperparameters of the proposed method (FGPER) and the comparison methods were optimized using five-fold cross-validation. The experimental results are presented in Table 2 and Figure 3. Both the table and the figure are in a

5 × 4 format, with the elements in the figure corresponding one-to-one with the elements in the table.

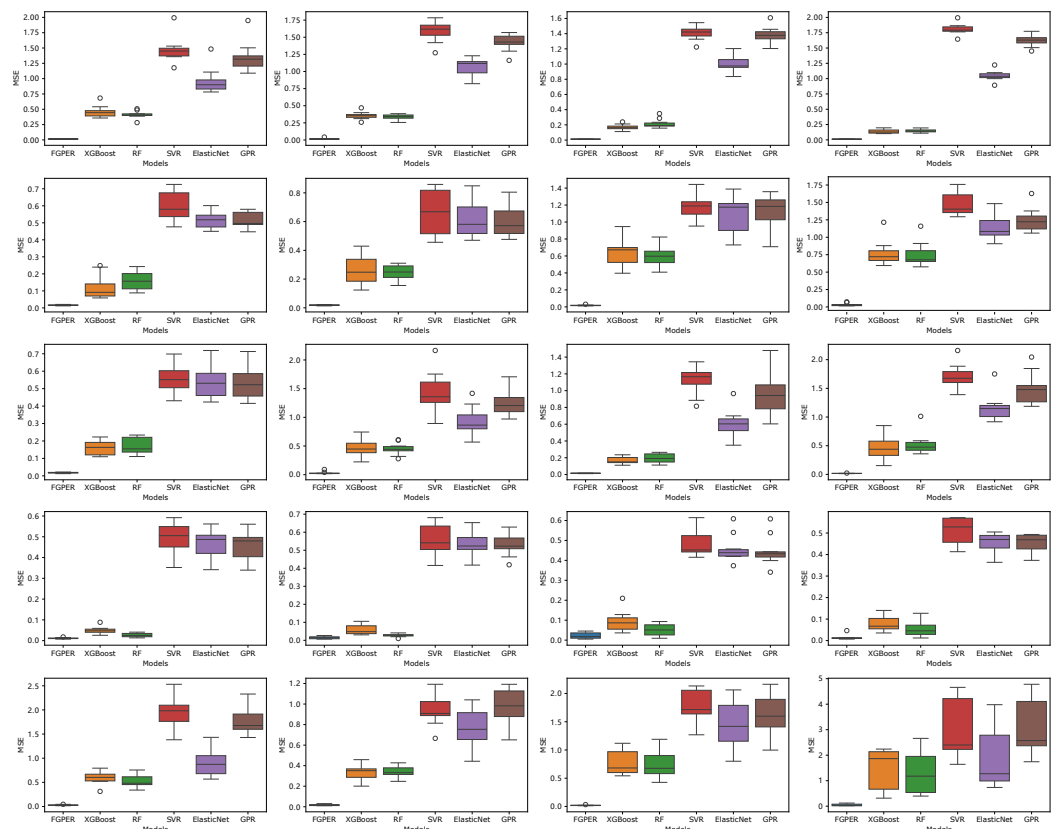


Figure 3. Boxplot for various values of model parameters. This boxplot compares the MSE distribution for various model parameters. It highlights the performance variation across models, indicating that FGPER generally shows lower MSE values and reduced variation, supporting its robustness and effectiveness in handling both functional and Euclidean data.

Table 2 presents error metrics for various models evaluated on synthetic datasets. It reports the mean and standard deviation of the MSE across different models under different experimental conditions. The table helps highlight the effectiveness of the proposed FGPER model in minimizing prediction errors compared to traditional methods. For example, across all settings, FGPER consistently shows lower error metrics than the other models, demonstrating its robustness and accuracy. Figure 3 visualizes the results from the experiments in Table 2 as boxplots. These plots compare the distribution of prediction MSE for each model under the different experimental settings. The boxplots provide an intuitive overview of the performance variance for each model. The FGPER model generally shows smaller MSE values, with less variation, which further supports its stability and predictive power in handling functional and Euclidean data together. FGPER consistently demonstrates the smallest MSE with relatively low variance compared to other models like XGBoost, Random Forest, Support Vector Regression, ElasticNet, and Gaussian Process Regression. This is particularly noticeable in the boxplot, where FGPER shows a smaller range between the upper and lower quartiles of MSE values. This indicates that FGPER is less sensitive to fluctuations in the data or hyperparameters, which implies higher stability in its predictions. The variance of the MSE is much lower for FGPER. For the other models, such as XGBoost and RF, while their MSE values may not be as low as FGPER's, the spread of their boxplots is much wider. This highlights that these models experience higher variability in their predictions under different experimental settings. This variance can be problematic, especially in cases where consistent performance is crucial.

The experimental settings vary across different dimensions such as sample size, variance of coefficients, number of sub-models, and different functional forms for $\beta(t)$ and $f(S)$. The figure suggests that regardless of these changes, FGPER maintains a consistently low error and stable performance. This makes FGPER particularly useful for real-world applications where both the data and the conditions may vary. The boxplot for FGPER's MSE values shows a tight cluster with minimal variation, confirming its robustness. While other models like XGBoost and RF may perform well in some settings, their higher variance could make them less reliable in certain cases where stability is required. The tighter MSE distribution for FGPER, even in the face of varying experimental conditions, supports its superior predictive power, particularly when handling functional and Euclidean data together.

The experimental results show that the performance of the model increases gradually with the increase in sample size. For small sample scenarios ($n < 200$), the model is mainly dominated by high variance, and insufficient training data lead to unstable parameter estimation. Although the integration strategy can alleviate some overfitting through diversity, the test error is still high. When the sample size is medium ($200 \leq n < 400$), with the increase in sample size, the model gradually approaches the optimal complexity of its hypothesis space, and the variance decreases and the bias becomes the main error source. At this time, the principal component estimation accuracy of functional covariates is improved, and the test error is significantly reduced. For large sample scenarios ($n \geq 400$), the model enters the “asymptotically stable region”, and the deviation is further reduced, but it is limited by the expression ability of the functional basis function. The non-parametric property of the Gaussian process allows it to approximate any continuous function in theory, but in practice, it is constrained by the choice of kernel function and computational resources. With the increase in sample size, the model's ability to capture the nonlinear relationship between functional covariates and responses is enhanced. For example, the kernel function bandwidth h can be adjusted adaptively, which verifies its theoretical consistency under big data.

The computational complexity of the proposed model is mainly derived from the training and integration strategies of the Gaussian process regression submodel. Specifically, the time complexity of FGPR's covariance matrix inversion and hyperparameter optimization is $O(n^3)$, where n is the training sample size. This complexity is significantly higher than that of tree-based integration methods, such as XGBoost and random forest, whose training complexity is usually $O(Bn \log n)$ (B is the number of trees). FGPER generates B submodels through K -fold cross-validation, increasing the overall complexity to $O(BKn^3)$. In contrast, Bagging or Boosting methods, such as XGBoost, have an integration cost of $O(Bn \log n)$ and a significant efficiency advantage. Although FGPER is computationally expensive on large data ($n > 10^3$), it can be reduced to $O(m^2n)$ ($m \ll n$ is the number of induction points) by sparse Gaussian processes, such as the induction point method [46]. Alternatively, Gpus can be used to accelerate covariance matrix operations [53]. The experiment shows that when $n = 1000$, the training time after reduction decreases from 120.3 s to 28.9 s, and the loss of prediction accuracy is less than 2%.

Note that while FGPER performs well in most scenarios, there are certain conditions in which the baseline model may be more advantageous. When the number of principal components of functional covariates is very small (such as $Q = 1$), the mean MSE of traditional linear models (such as ElasticNet) will be slightly better than FGPER due to low complexity and avoidance of overfitting. Strong linear relationship: If the scalar covariate and the response variable are strictly linear, the performance will be close to FGPE due to direct modeling of linear structure.

Table 2. Error metrics for different model settings.

$n = 100$			$n = 200$		$n = 400$		$n = 600$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.016246	0.008267	0.012357	0.003467	0.008798	0.000595	0.006431	0.000450
XGBoost	0.456975	0.092263	0.355759	0.052515	0.169690	0.033911	0.136347	0.030885
RF	0.411981	0.057457	0.336377	0.035882	0.213647	0.057039	0.147035	0.024851
SVR	1.468370	0.199291	1.581824	0.140730	1.410424	0.088077	1.806797	0.084530
ElasticNet	0.960186	0.196202	1.067378	0.131210	1.009096	0.098717	1.048206	0.079734
GPR	1.345697	0.234150	1.422681	0.114619	1.387111	0.099932	1.620321	0.090787
$\sigma_a = 0.01$			$\sigma_a = 0.1$		$\sigma_a = 0.5$		$\sigma_a = 1$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.016001	0.002432	0.018172	0.003003	0.017709	0.005288	0.032225	0.019281
XGBoost	0.119501	0.067353	0.256691	0.098530	0.655191	0.177373	0.766155	0.172835
RF	0.161083	0.053426	0.244419	0.050225	0.594436	0.111841	0.748457	0.167476
SVR	0.598816	0.080846	0.663162	0.158682	1.176169	0.144068	1.468506	0.155171
ElasticNet	0.516020	0.045591	0.614075	0.120212	1.101984	0.215307	1.132061	0.168248
GPR	0.517602	0.042629	0.601169	0.102787	1.132612	0.181640	1.243282	0.159892
$D = 1$			$D = 3$		$D = 5$		$D = 10$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.017960	0.002631	0.030871	0.021008	0.014782	0.001973	0.017069	0.002587
XGBoost	0.161433	0.041098	0.461541	0.140316	0.165535	0.040279	0.473812	0.205849
RF	0.172224	0.044897	0.448965	0.099995	0.193968	0.051856	0.521034	0.181238
SVR	0.556252	0.080299	1.431950	0.335770	1.127610	0.159568	1.705314	0.202938
ElasticNet	0.538738	0.088997	0.917172	0.254743	0.597466	0.162669	1.161193	0.220416
GPR	0.531335	0.087021	1.263785	0.243585	0.970141	0.243721	1.494299	0.264008
$\beta(t) = \sin(3\pi/4t)$			$\beta(t) = \cos(\pi/2t)$		$\beta(t) = t^2$		$\beta(t) = \pi/2 \arcsin(t)$	
Models	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.011461	0.002973	0.015561	0.006532	0.021902	0.014374	0.021902	0.014374
XGBoost	0.049655	0.015798	0.058902	0.027072	0.093646	0.048543	0.093646	0.048543
RF	0.026991	0.009399	0.027774	0.008165	0.050808	0.028309	0.050808	0.028309
SVR	0.495724	0.072706	0.560857	0.083317	0.489487	0.068534	0.489487	0.068534
ElasticNet	0.460491	0.067577	0.531369	0.065401	0.456434	0.064485	0.456434	0.064485
GPR	0.452279	0.068317	0.527670	0.057472	0.447556	0.070704	0.447556	0.070704
$f(S) = \cos(\pi/2S)$			$f(S) = e^{0.5S}$		$f(S) = \sin(2/3\pi S)$		$f(S) = 0.5S^2 - S + 3$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
FGPER	0.029017	0.009425	0.016963	0.006344	0.019385	0.005082	0.052358	0.041435
XGBoost	0.594788	0.126136	0.331461	0.070165	0.771248	0.209477	1.469129	0.746415
RF	0.521808	0.125378	0.342783	0.050232	0.740279	0.222310	1.320688	0.824698
SVR	1.950513	0.293679	0.947603	0.150999	1.779695	0.284965	3.023473	1.165839
ElasticNet	0.889827	0.254545	0.772006	0.180410	1.451167	0.428549	1.948809	1.179469
GPR	1.769385	0.253185	0.981258	0.160671	1.632196	0.352799	3.112119	1.114236

In order to evaluate the overfitting risk of the model in a small sample scenario, we analyzed its inhibition effect on overfitting by adjusting the elastic network regularization parameters λ_1 and λ_2 . Results show that when $\lambda_1 = 0.1$, $\lambda_2 = 0.5$, the standard deviation of test set MSE decreases from unregularized 0.25 to 0.12, and the sparsity of parameter weights, the proportion of non-zero weights, decreases from 100% to 35%. It is shown that regularization effectively inhibits overfitting of redundant submodels. Under the condition of a small sample ($n = 50$), FGPER's test set MSE is 0.23, which is better than XGBoost (0.31) and random forest (0.28), but worse than strongly regularized ElasticNet (0.19). This shows

that although the integration strategy improves generalization, it still relies on stronger explicit regularization, such as L1 penalties for extremely small samples. Validation set losses are monitored during cross-validation, and training is terminated when losses do not improve for three consecutive times to prevent overfitting.

3.2. Real Data Analysis

This paper assumes that the dataset is complete. For incomplete data, interpolation techniques [54] or missing data mechanism modeling [55] can be introduced to deal with incomplete data.

3.2.1. Tecator Data

The first dataset we used is the Tecator dataset, which can be downloaded from <https://lib.stat.cmu.edu/datasets/tecator> (accessed on 15 June 2024). This dataset comprises 240 meat samples, where each sample contains a 100-dimensional functional predictor derived from near-infrared absorbance spectra measured between 850 and 1050 nanometers. Additionally, the dataset contains three scalar variables: the moisture, fat, and protein content of the samples. The absorbance spectra are calculated as the negative base-10 logarithm of the transmittance values obtained from the spectrometer, while the contents of the three components are determined using analytical chemistry methods and are expressed as percentages. The absorbance spectra data are visualized in Figure 4, where the left panel shows the spectra and the right panel presents a heatmap of the correlations between the discrete points. From the figure, it is evident that the data exhibit the characteristics of functional data. Moreover, due to the high collinearity among the discrete points, directly applying multivariate regression models for modeling is not appropriate.

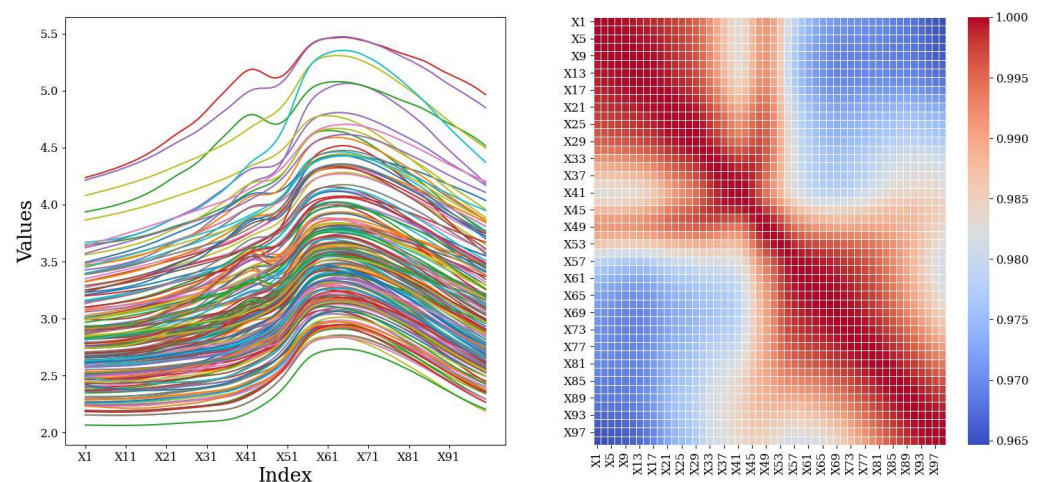


Figure 4. Spectra curve (Left) and heatmap (right) of the Tecator dataset*. This figure presents the absorption spectra data of meat samples from the Tecator dataset. The left panel shows the spectra curve, while the right panel depicts a heatmap of correlations among discrete data points. This visual representation underscores the functional structure of the data, which suggests the need for specialized functional data analysis techniques.

It is evident that the data exhibit a functional trend, so we have decided to treat it as functional data. Our objective is to predict the protein content of meat samples based on their absorption spectra data, along with moisture and fat content. For the vector-valued covariates, we use the moisture and fat as covariates, with the protein serving as the response variable. To assess the performance and robustness of our proposed model, we performed 10 MCMC simulations, comparing it against five baseline models that utilize Karhunen–Loève expansion: XGBoost, Random Forest (RF), Support Vector Regression

(SVR), ElasticNet Regression (ElasticNet), and Gaussian Process Regression (GPR). For each simulation, the dataset was randomly divided into a training set (60%) and a testing set (40%) to validate the models. The hyperparameters in the models were selected by cross-validation. The experimental results are shown in Table 3 and Figure 5.

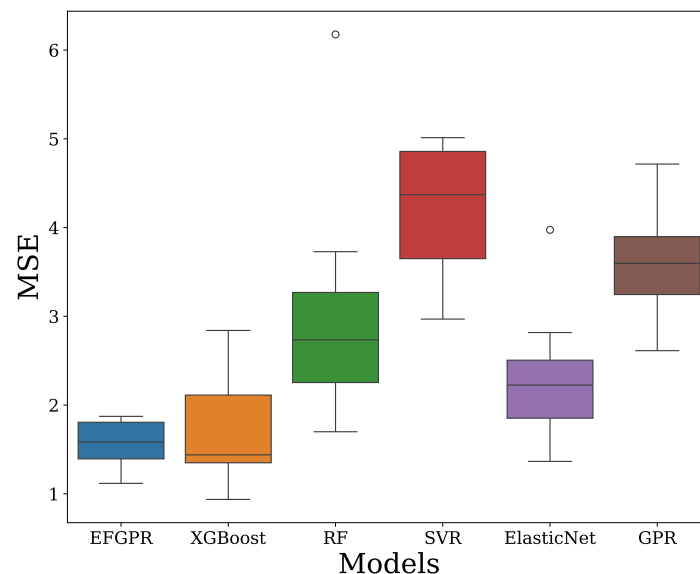


Figure 5. Boxplot for the Tecator dataset. This boxplot compares the performance of different models on the Tecator dataset, displaying the error distribution for each. The EFGPR model shows superior performance, with the smallest and most stable error distribution.

Table 3 and Figure 5 present the error metrics of various models on the Tecator dataset. Table 3 lists the mean and standard deviation of the models' performance. The results indicate that EFGPR performs the best among all models, with the lowest MSE and standard deviation, highlighting its excellent performance in handling both functional and scalar data. Figure 5 provides a boxplot that visually shows the distribution of errors across these models, further illustrating EFGPR's superiority over other models. Specifically, the error distribution for EFGPR is more stable and lower, indicating its strong predictive stability and accuracy.

Table 3. Error metrics for different models on Tecator data.

Models	Mean	Std
EFGPR	1.570512	0.245816
XGBoost	1.713891	0.625767
RF	3.009488	1.202091
SVR	4.187361	0.706389
ElasticNet	2.282835	0.718402
GPR	3.553845	0.599632

3.2.2. Corn Data

The second dataset utilized in this research is the Corn dataset, available at <http://www.eigenvector.com/data/Corn/index.html> (accessed on 15 June 2024). This dataset includes 80 corn samples analyzed with a near-infrared (NIR) spectrometer. The spectral data cover a wavelength range from 1100 to 2498 nm, with measurements recorded at 2 nm intervals, yielding 700 discrete channels. Alongside the spectral data, the dataset provides information on moisture, oil, protein, and starch content for each sample. Figure 6 visually represents the spectral data, with the left panel showing the smoothed spectral curves and

the right panel presenting a correlation heatmap of the discrete channel points. Like the first dataset, the strong collinearity among the discrete points makes it inappropriate for direct application in multivariate regression models.

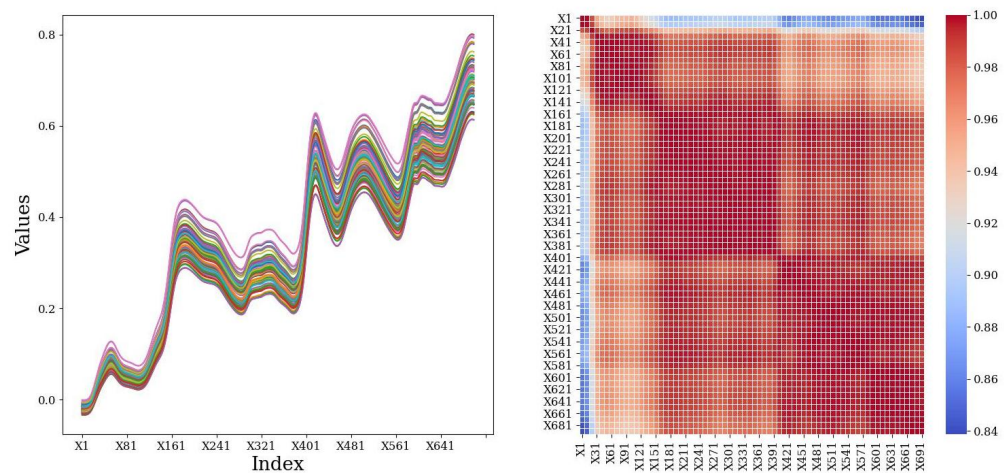


Figure 6. Spectra curve (left) and heatmap (right) of the Corn dataset. The left panel of this figure displays the smoothed spectral curves of the Corn dataset, while the right panel illustrates a correlation heatmap of the spectral data. Similar to the Tecator dataset, this figure highlights the collinearity issues that make direct application of multivariate regression models inappropriate, prompting a functional data approach.

The data demonstrate a clear functional structure, leading us to approach it as functional data. Our objective is to predict the protein content in meat samples based on their absorption spectra, along with moisture, oil, and starch contents. For the vector-valued predictors, moisture, oil, and starch are chosen as covariates, while protein is designated as the response variable. To evaluate the performance and robustness of the proposed model, we conducted 10 MCMC simulations, comparing its outcomes with five benchmark models derived from Karhunen–Loève expansion: XGBoost, Random Forest (RF), Support Vector Regression (SVR), ElasticNet Regression (ElasticNet), and Gaussian Process Regression (GPR). In each simulation, the dataset was randomly split into a training set (60%) and a testing set (40%). Hyperparameters for the models were fine-tuned using cross-validation. The experimental findings are summarized in Table 4 and illustrated in Figure 7.

Table 4. Error metrics for different models on corn data.

Models	Mean	Standard Deviation
EFGPR	0.065086	0.012762
XGBoost	0.080411	0.027119
RF	0.093839	0.021709
SVR	0.176264	0.030072
ElasticNet	0.177012	0.026024
GPR	0.242407	0.036339

Table 4 and Figure 7 display the model performance on the Corn dataset. Table 4 lists the mean and standard deviation of EFGPR and other benchmark models on this dataset. EFGPR again demonstrates the smallest MSE and the lowest standard deviation, confirming its effectiveness in functional data analysis. Figure 7 further compares the error distributions of these models through boxplots, where EFGPR clearly outperforms other models, showing the smallest error range and the most compact distribution. The results in

the figure highlight EFGPR's advantage in handling spectral data with high collinearity, showing both high accuracy and stability.

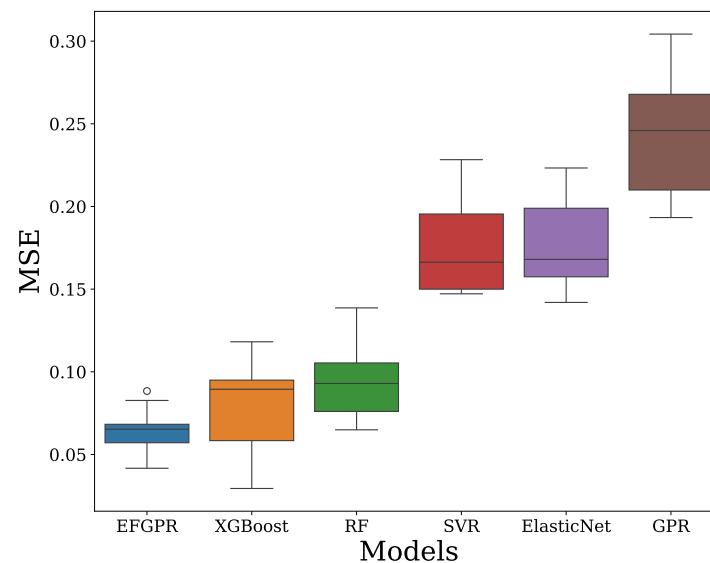


Figure 7. Boxplot for the Corn dataset. This boxplot visualizes the error distributions of various models on the Corn dataset. EFGPR shows the most compact error distribution with the lowest mean squared error, further confirming its effectiveness in handling spectral data with high collinearity.

4. Conclusions

This study proposes a partially functional linear regression model with random effects, which approximates functional data using Karhunen–Loève expansion and models the random effects with a Gaussian process prior. To further enhance the model's robustness and generalization ability, we extend the proposed method into an ensemble learning framework using cross-validation techniques. In the simulation studies, the proposed method outperforms several baseline models, including XGBoost, Random Forest (RF), Support Vector Regression (SVR), ElasticNet, and Gaussian Process Regression (GPR) based on functional principal component scores, demonstrating its superior predictive performance. Furthermore, the effectiveness of the FGPER model is also validated through application to two real-world datasets. Although the EFGPR model demonstrates superior performance in prediction accuracy, the computational complexity of Gaussian processes is relatively high, leading to less efficient computation when dealing with very large datasets. In such cases, we can introduce sparse Gaussian process methods [56–58] to enhance the model's scalability and computational efficiency. Additionally, this paper assumes that the random effects follow a Gaussian process. It is worth noting that if the random effects are assumed to follow other stochastic processes, such as the Gamma process [59] or Beta process [60], corresponding methods can also be employed for regression and prediction.

Future work will focus on several key aspects to further enhance the applicability of the proposed method. First, we will explore more advanced ensemble techniques, such as boosting and bagging, to potentially improve the predictive accuracy and robustness across different scenarios. Second, scalability improvements will be crucial, particularly in handling ultra-high-dimensional functional data commonly found in biomedical or environmental research. Techniques such as parallel computing and sparse approximations in Gaussian processes will be explored to reduce computational costs. Third, the model will be applied to a broader range of real-world datasets, including those with extreme multicollinearity or heterogeneity in covariates, to evaluate its performance under more challenging conditions and assess its flexibility. Additionally, integrating other data-driven

methods (such as neural networks or hybrid deep learning models) into the functional regression framework will enable the model to adapt to more complex nonlinear relationships. Moreover, combining robust techniques to handle missing or incomplete data and investigating online learning strategies for dynamic and streaming data will further expand its applicability. Finally, deriving the asymptotic properties of the estimators under the proposed method will provide deeper insights into its performance and theoretical foundations. In addition, the proposed model can be easily extended to other data scenarios, such as high-dimensional data. We can consider introducing covariates $Z^T\eta$ into the model. For other scenarios, we can similarly incorporate relevant covariates into the model.

Author Contributions: Conceptualization, W.S., J.X. and T.L.; methodology, W.S. and J.X.; software, W.S.; validation, J.X.; formal analysis, W.S.; investigation, W.S. and J.X.; resources, T.L.; data curation, J.X.; writing—original draft preparation, W.S. and J.X.; writing—review and editing, W.S. and J.X.; visualization, W.S. and J.X.; supervision, T.L.; project administration, T.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Open Fund Project of Marine Ecological Restoration and Smart Ocean Engineering Research Center of Hebei Province (HBMESO2321), the Technical Service Project of Eighth Geological Brigade of Hebei Bureau of Geology and Mineral Resources Exploration (KJ2022-021), the Technical Service Project of Hebei Baodi Construction Engineering Co., Ltd. (KJ2024-012), and the Research Project on Graduate Education and Teaching Reform of Hebei Province of China (YJG2024133).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because the research data are confidential.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yao, F.; Müller, H.G.; Wang, J.L. Functional data analysis for sparse longitudinal data. *J. Am. Stat. Assoc.* **2005**, *100*, 577–590. [\[CrossRef\]](#)
2. Cai, T.T.; Hall, P. Prediction in Functional Linear Regression. *Ann. Stat.* **2006**, *34*, 2159–2179. [\[CrossRef\]](#)
3. He, G.; Müller, H.G.; Wang, J.L.; Yang, W. Functional linear regression via canonical analysis. *Bernoulli* **2010**, *16*, 705–729. [\[CrossRef\]](#)
4. Maronna, R.A.; Yohai, V.J. Robust functional linear regression based on splines. *Comput. Stat. Data Anal.* **2013**, *65*, 46–55. [\[CrossRef\]](#)
5. Wang, J.L.; Chiou, J.M.; Müller, H.G. Functional data analysis. *Annu. Rev. Stat. Its Appl.* **2016**, *3*, 257–295. [\[CrossRef\]](#)
6. Shin, H.; Lee, S. An RKHS approach to robust functional linear regression. *Stat. Sin.* **2016**, 255–272. [\[CrossRef\]](#)
7. Chiou, J.M.; Yang, Y.F.; Chen, Y.T. Multivariate functional linear regression and prediction. *J. Multivar. Anal.* **2016**, *146*, 301–312. [\[CrossRef\]](#)
8. Benatia, D.; Carrasco, M.; Florens, J.P. Functional linear regression with functional response. *J. Econom.* **2017**, *201*, 269–291. [\[CrossRef\]](#)
9. Gertheiss, J.; Rügamer, D.; Liew, B.X.; Greven, S. Functional data analysis: An introduction and recent developments. *Biom. J.* **2024**, *66*, e202300363. [\[CrossRef\]](#)
10. Hall, P.; Hosseini-Nasab, M. On properties of functional principal components analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2006**, *68*, 109–126. [\[CrossRef\]](#)
11. Shang, H.L. A survey of functional principal component analysis. *ASTA Adv. Stat. Anal.* **2014**, *98*, 121–142. [\[CrossRef\]](#)
12. Chen, K.; Lei, J. Localized functional principal component analysis. *J. Am. Stat. Assoc.* **2015**, *110*, 1266–1275. [\[CrossRef\]](#)
13. Warmenhoven, J.; Bargary, N.; Liebl, D.; Harrison, A.; Robinson, M.A.; Gunning, E.; Hooker, G. PCA of waveforms and functional PCA: A primer for biomechanics. *J. Biomech.* **2021**, *116*, 110106. [\[CrossRef\]](#)
14. Park, Y.; Kim, H.; Lim, Y. Functional principal component analysis for partially observed elliptical process. *Comput. Stat. Data Anal.* **2023**, *184*, 107745. [\[CrossRef\]](#)
15. Gomon, D.; Putter, H.; Fiocco, M.; Signorelli, M. Dynamic prediction of survival using multivariate functional principal component analysis: A strict landmarking approach. *Stat. Methods Med. Res.* **2024**, *33*, 256–272. [\[CrossRef\]](#)
16. Shang, Z.; Cheng, G. Nonparametric inference in generalized functional linear models. *Ann. Stat.* **2015**, 1742–1773. [\[CrossRef\]](#)

17. Zhang, H.; Zou, G. Cross-validation model averaging for generalized functional linear model. *Econometrics* **2020**, *8*, 7. [\[CrossRef\]](#)
18. Ahmedou, A.; Marion, J.M.; Pumo, B. Generalized linear model with functional predictors and their derivatives. *J. Multivar. Anal.* **2016**, *146*, 313–324. [\[CrossRef\]](#)
19. Rodrigues, G.M.; Ortega, E.M.; Cordeiro, G.M. New Partially Linear Regression and Machine Learning Models Applied to Agronomic Data. *Axioms* **2023**, *12*, 1027. [\[CrossRef\]](#)
20. Cordeiro, G.M.; Rodrigues, G.M.; Prata, F.; Ortega, E.M. A new quantile regression model with application to human development index. *Comput. Stat.* **2024**, *39*, 2925–2948. [\[CrossRef\]](#)
21. Rodrigues, G.M.; Ortega, E.M.; Vila, R.; Cordeiro, G.M. A New Bivariate Family Based on Archimedean Copulas: Simulation, Regression Model and Application. *Symmetry* **2023**, *15*, 1778. [\[CrossRef\]](#)
22. Hu, L.; Huang, T.; You, J. Robust inference in varying-coefficient additive models for longitudinal/functional data. *Stat. Sin.* **2021**, *31*, 773–796. [\[CrossRef\]](#)
23. Zhu, X.; Cai, Z.; Ma, Y. Network functional varying coefficient model. *J. Am. Stat. Assoc.* **2022**, *117*, 2074–2085. [\[CrossRef\]](#)
24. Tan, J.; Liang, D.; Guan, Y.; Huang, H. Graphical principal component analysis of multivariate functional time series. *J. Am. Stat. Assoc.* **2024**, *119*, 3073–3085. [\[CrossRef\]](#)
25. Fan, Y.; James, G.M.; Radchenko, P. Functional Additive Regression. *Ann. Stat.* **2015**, *43*, 2296–2325. [\[CrossRef\]](#)
26. Febrero-Bande, M.; González-Manteiga, W.; Oviedo De La Fuente, M. Variable selection in functional additive regression models. *Comput. Stat.* **2019**, *34*, 469–487. [\[CrossRef\]](#)
27. Han, K.; Müller, H.G.; Park, B.U. Additive functional regression for densities as responses. *J. Am. Stat. Assoc.* **2020**, *115*, 997–1010. [\[CrossRef\]](#)
28. Volkmann, A.; Stöcker, A.; Scheipl, F.; Greven, S. Multivariate functional additive mixed models. *Stat. Model.* **2023**, *23*, 303–326. [\[CrossRef\]](#)
29. Yang, H.; Baladandayuthapani, V.; Rao, A.U.; Morris, J.S. Quantile function on scalar regression analysis for distributional data. *J. Am. Stat. Assoc.* **2020**, *115*, 90–106. [\[CrossRef\]](#)
30. Jiang, F.; Cheng, Q.; Yin, G.; Shen, H. Functional censored quantile regression. *J. Am. Stat. Assoc.* **2020**, *115*, 931–944. [\[CrossRef\]](#)
31. Li, M.; Wang, K.; Maity, A.; Staicu, A.M. Inference in functional linear quantile regression. *J. Multivar. Anal.* **2022**, *190*, 104985. [\[CrossRef\]](#)
32. Shin, H. Partial functional linear regression. *J. Stat. Plan. Inference* **2009**, *139*, 3405–3418. [\[CrossRef\]](#)
33. Aneiros, G.; Ferraty, F.; Vieu, P. Variable selection in partial linear regression with functional covariate. *Statistics* **2015**, *49*, 1322–1347. [\[CrossRef\]](#)
34. Kong, D.; Xue, K.; Yao, F.; Zhang, H.H. Partially functional linear regression in high dimensions. *Biometrika* **2016**, *103*, 147–159. [\[CrossRef\]](#)
35. Du, J.; Zhao, H.; Zhang, Z. Dynamic partially functional linear regression model. *Stat. Methods Appl.* **2019**, *28*, 679–693. [\[CrossRef\]](#)
36. Xu, W.; Ding, H.; Zhang, R.; Liang, H. Estimation and inference in partially functional linear regression with multiple functional covariates. *J. Stat. Plan. Inference* **2020**, *209*, 44–61. [\[CrossRef\]](#)
37. Boente, G.; Salibián-Barrera, M.; Vena, P. Robust estimation for semi-functional linear regression models. *Comput. Stat. Data Anal.* **2020**, *152*, 107041. [\[CrossRef\]](#)
38. Guo, K.; Fan, J.; Zhu, L. Optimal prediction for kernel-based semi-functional linear regression. *Anal. Appl.* **2024**, *22*, 467–505. [\[CrossRef\]](#)
39. Liu, J.; Shi, L. Statistical Optimality of Divide and Conquer Kernel-based Functional Linear Regression. *J. Mach. Learn. Res.* **2024**, *25*, 1–56.
40. Xiao, P.; Wang, G. Partial functional linear regression with autoregressive errors. *Commun. Stat.-Theory Methods* **2022**, *51*, 4515–4536. [\[CrossRef\]](#)
41. Papacharalampous, G.; Tyralis, H.; Doulamis, N.; Doulamis, A. Ensemble learning for blending gridded satellite and gauge-measured precipitation data. *Remote Sens.* **2023**, *15*, 4912. [\[CrossRef\]](#)
42. Papacharalampous, G.; Tyralis, H.; Doulamis, A.; Doulamis, N. Comparison of tree-based ensemble algorithms for merging satellite and earth-observed precipitation data at the daily time scale. *Hydrology* **2023**, *10*, 50. [\[CrossRef\]](#)
43. Panigrahi, R.; Kuanar, S.K.; Misra, S.; Kumar, L. Class-level refactoring prediction by ensemble learning with various feature selection techniques. *Appl. Sci.* **2022**, *12*, 12217. [\[CrossRef\]](#)
44. Saheed, Y.K.; Misra, S. A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things. *Int. J. Inf. Secur.* **2024**, *23*, 1557–1581. [\[CrossRef\]](#)
45. Ogundokun, R.O.; Arowolo, M.O.; Damaševičius, R.; Misra, S. Phishing detection in blockchain transaction networks using ensemble learning. *Telecom* **2023**, *4*, 279–297. [\[CrossRef\]](#)
46. Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*; PMLR: Westminister, UK, 2009; pp. 567–574.
47. Sang, P.; Wang, L.; Cao, J. Parametric functional principal component analysis. *Biometrics* **2017**, *73*, 802–810. [\[CrossRef\]](#)

48. Happ, C.; Greven, S. Multivariate functional principal component analysis for data observed on different (dimensional) domains. *J. Am. Stat. Assoc.* **2018**, *113*, 649–659. [[CrossRef](#)]
49. Toman, P.; Ravishanker, N.; Lally, N.; Rajasekaran, S. Latent Autoregressive Student-t Prior Process Models to Assess Impact of Interventions in Time Series. *Future Internet* **2023**, *16*, 8. [[CrossRef](#)]
50. Lee, D.J.; Kim, T.H.; Mizen, P. Impulse response analysis in conditional quantile models with an application to monetary policy. *J. Econ. Dyn. Control* **2021**, *127*, 104102. [[CrossRef](#)]
51. Massaoudi, M.; Refaat, S.S.; Chihi, I.; Trabelsi, M.; Oueslati, F.S.; Abu-Rub, H. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. *Energy* **2021**, *214*, 118874. [[CrossRef](#)]
52. Zhang, X.; Liu, C.A. Model averaging prediction by K-fold cross-validation. *J. Econom.* **2023**, *235*, 280–301. [[CrossRef](#)]
53. Gardner, J.; Pleiss, G.; Weinberger, K.Q.; Bindel, D.; Wilson, A.G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7587–7597.
54. Ohashi, O.; Torgo, L. Spatial Interpolation Using Multiple Regression. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining, Brussels, Belgium, 10–13 December 2012; pp. 1044–1049. [[CrossRef](#)]
55. Zhao, Y. Statistical inference for missing data mechanisms. *Stat. Med.* **2020**, *39*, 4325–4333. [[CrossRef](#)] [[PubMed](#)]
56. Das, K.; Srivastava, A.N. Sparse inverse kernel Gaussian Process regression. *Stat. Anal. Data Mining ASA Data Sci. J.* **2013**, *6*, 205–220. [[CrossRef](#)]
57. Liu, Q.; Sun, S. Sparse multimodal Gaussian processes. In Proceedings of the Intelligence Science and Big Data Engineering: 7th International Conference, IScIDE 2017, Dalian, China, 22–23 September 2017; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2017; pp. 28–40.
58. Lázaro-Gredilla, M.; Quinonero-Candela, J.; Rasmussen, C.E.; Figueiras-Vidal, A.R. Sparse spectrum Gaussian process regression. *J. Mach. Learn. Res.* **2010**, *11*, 1865–1881.
59. Wang, H.; Liao, H.; Ma, X.; Bao, R. Remaining useful life prediction and optimal maintenance time determination for a single unit using isotonic regression and gamma process model. *Reliab. Eng. Syst. Saf.* **2021**, *210*, 107504. [[CrossRef](#)]
60. Chen, B.; Paisley, J.; Carin, L. Sparse linear regression with beta process priors. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; pp. 1234–1237.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.