Daniel Ries[1]
Sandia National Laboratories,
Albuquerque, NM 87185
e-mail: dries@sandia.gov

Adah Zhang
Sandia National Laboratories,
Albuquerque, NM 87185
e-mail: azhang@sandia.gov

J. Derek Tucker
Sandia National Laboratories,
Albuquerque, NM 87185
e-mail: jdtuck@sandia.gov

Kurtis Shuler
Sandia National Laboratories,
Albuquerque, NM 87185
e-mail: kwshule@sandia.gov

Madeline Ausdemore
Los Alamos National Laboratory,
Los Alamos, NM 87545
e-mail: mausdemore@lanl.gov

A Framework for Inverse Prediction Using Functional Response Data

Inverse prediction models have commonly been developed to handle scalar data from physical experiments. However, it is not uncommon for data to be collected in functional form. When data are collected in functional form, it must be aggregated to fit the form of traditional methods, which often results in a loss of information. For expensive experiments, this loss of information can be costly. This paper introduces the functional inverse prediction (FIP) framework, a general approach which uses the full information in functional response data to provide inverse predictions with probabilistic prediction uncertainties obtained with the bootstrap. The FIP framework is a general methodology that can be modified by practitioners to accommodate many different applications and types of data. We demonstrate the framework, highlighting points of flexibility, with a simulation example and applications to weather data and to nuclear forensics. Results show how functional models can improve the accuracy and precision of predictions.
[DOI: 10.1115/1.4053752]

Keywords: artificial intelligence, inverse methods for engineering applications, statistical methods for engineering applications, machine learning for engineering applications

1 Introduction

Inverse prediction (IP) modeling is concerned with predicting the unknown values of independent variables that produced an observed, dependent response. In classical regression modeling, the values of the independent variables, often called "covariates" or "predictors," are known. The linear regression problem is to estimate the coefficients of a posited linear combination that captures the relationship between the independent variables and the dependent variable—the observed response. Inverse prediction modeling reverses the problem by starting with given coefficients (or other model parameters, usually estimated from training data), then uses the observed values of the dependent variable to predict the values of the independent variables that generated the observed response. IP is used in a variety of fields, including, but not limited to, forensic science [1], nuclear forensics [2,3], computer modeling [4,5], chemometrics [6], nutrition tracking [7], and geosciences [8,9].

We consider the problem of IP from the perspectives of a forward and an inverse model. The general mathematical form of a forward model associates a q-dimensional response $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_q)'$ with a p-dimensional vector of input variables $\mathbf{x} = (x_1, x_2, \ldots, x_p)'$. The relationship between the responses and inputs can be expressed as $Y = g(\mathbf{x}; \boldsymbol{\theta}) + \varepsilon$, where $g(\cdot)$ is the true underlying relationship (often expressed as a low-order polynomial for physical experiments, particularly when no theoretical causal relationship is known), $\boldsymbol{\theta}$ is a vector of unknown model parameters, and ϵ represents a random vector that captures noise in the observed data. These response surface models [10] are common in many applications and can be effective ways of summarizing relationships between inputs and responses.

The goal of the inverse model is to estimate the value of the unknown variable $\mathbf{X}^* = (X_1^*, X_2^*, \ldots, X_p^*)'$ that most likely produced a new observation $\mathbf{Y}^* = (Y_1^*, Y_2^*, \ldots, Y_q^*)'$. The relationship between $\mathbf{X}^*$ and $\mathbf{Y}^*$ is captured in a model whose parameters are estimated using a set of training data and whose performance is assessed using techniques described later. Typically, it is more convenient to directly construct an inverse model that predicts $\mathbf{X}^*$, rather than inverting a forward model. This approach may violate regression assumptions, however, as a standard linear regression model assumes that the independent variable is measured with negligible error, and it is the response that is associated with some error. This approach is also not well suited for traditional optimal design of experiments (DoEx). Direct inverse models can be difficult to develop from optimal DoEx since most literature does not consider the inverse problem as the goal to use $\mathbf{Y}^*$ to predict $\mathbf{X}^*$ [11,12].

In model calibration [5,13–15], the goal is to estimate, or calibrate, computer model parameters based on a small amount of observational data and limited, imperfect, computer model runs. This process is effectively performing IP to determine the values of computer model parameters which will align closest the computer model runs to the observed data while accounting for the computer model biases (model discrepancy). Verification and Validation (V&V) [16,17] is closely related to model calibration and considers assessing accuracy and reliability of computational models. According to Oberkampf and Trucano [16], verification is the identification and quantification of errors in computer models and validation is comparing computer model results to experimental data, while quantifying errors and estimate uncertainties. The main difference between the IP problem addressed in this paper and model calibration and V&V is the absence of a computer model approximating the underlying scientific mechanism. Because there is no computer model, the model discrepancy term is not needed since we model the response directly. For IP in this paper, the "causal" model $g(\cdot)$ is either assumed known through theoretical consideration, or approximated through a Taylor series expansion by a low-order polynomial. This approximation is reasonable for many physical experiments, which often only considers a 2–4 levels for experimental factors. Therefore, the IP framework proposed in this paper is a purely data-driven approach.

Inverse prediction modeling has traditionally been presented using scalar or vector values for both the dependent and independent variables. Many applications, however, handle data in functional form, where the observed values are naturally a function of time, frequency, location, or some other index [18]. The

[1]Corresponding author.
Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received November 15, 2021; final manuscript received January 31, 2022; published online May 20, 2022. Assoc. Editor: Christian Soize.

information of interest to the researcher is the function itself, and so aggregation or statistical summary reduces the usefulness of the data. In such cases, a functional data approach to inverse prediction is required. Functional data analysis (FDA) is the branch of statistics concerned with continuous data such as curves, surfaces, and functions [18,19]. Rather than considering a curve as a set of discrete data points over time, the entire curve is considered one functional measurement. Information is contained in the curve rather than at individual points along its length or in some aggregation of the function such as the mean. Handling data in this form allow for better treatment of its true structure and characteristics, such as the smoothness of the function which implicitly contains correlations that would need to be explicitly modeled otherwise.

In this paper, we present a functional inverse prediction (FIP) framework for IP problems where the response variable is functional. This framework is general enough to be widely applicable across research programs and problem domains, allowing the user to customize details of the forward and inverse models. Even though the framework is flexible, its structure captures the information in functional responses for IP and quantifies uncertainty for the inverse predictions produced.

This paper is structured as follows: in Sec. 2, we present the FIP framework at a high level, explaining briefly how the different parts of FIP fit together. In Sec. 3, details of the FIP framework are fleshed out, including a description of some possible basis representations of functional data, forward model forms, and inverse prediction loss functions. Uncertainty quantification (UQ) using the bootstrap method is also explained in Sec. 3. Section 4 presents a simulation study demonstrating how the FIP framework works in practice. Section 5 expands on the simulation study by applying the framework to a weather and a nuclear forensics problem. Section 6 discusses the results from the examples, reviews how the framework can be generalized to different scenarios, and provides suggestions for future improvements of the framework. Section 7 summarizes the FIP framework and results.

2 Functional Inverse Prediction Framework

The FIP framework is a general process using functional response data to perform IP on scalar independent variables. To ensure maximum flexibility, the FIP framework outlined here is developed with a minimal set of assumptions and does not rely on specific conditions or models. Figure 1 displays a diagram of the FIP framework. Section 3 details the specifics of each portion of the framework and gives examples and suggestions for modeling. The framework is divided into two phases: (i) the training phase and (ii) the inference phase. In the training phase, the forward model is fit using training data such as data already collected for the purpose of understanding the relationship between the responses and the independent variables. During this phase, uncertainty in the model is also quantified to characterize the variability associated with the modeled relationship. In the inference phase, the model is used for inverse prediction. When new response data are observed, they are run backwards through the model to predict the values of the independent variables that most likely generated the observed response. As in the training phase, uncertainty in the inverse predictions is also quantified.

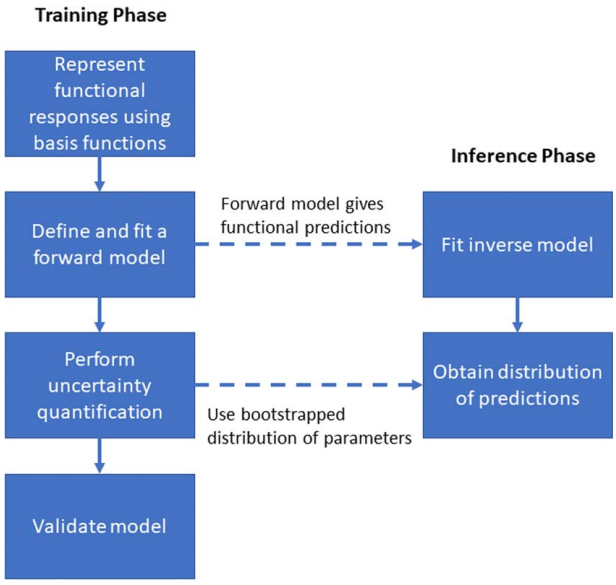

Fig. 1 FIP Framework. This is the general process to ensure for doing IP with functional response data.

The training phase consists of four parts:

(1) Represent observed functional data as functional objects using basis functions.
(2) Choose and estimate the forward model.
(3) Bootstrap the forward model to quantify uncertainty in the forward model parameters.
(4) Evaluate the model using cross validation.

The training phase offers many opportunities for tailoring the framework to a specific problem. Discrete, observed measurements must be represented in some function space using basis functions that capture the smoothness of the response. The form and number of bases as well as the smoothing parameters can be chosen according to the goals of the specific application. The form of the forward model is also the modeler's choice: both linear and non-linear options will be introduced in Sec. 3.2. The last two steps of the training phase—uncertainty quantification and model validation—contribute to model interpretation. Uncertainty quantification of the forward model generates a distribution of values for the forward model parameters, which will then be used in the inference phase to generate inverse prediction intervals. Model fit is assessed through cross validation techniques, and common measures are defined in Sec. 3.4.

The inference phase consists of two parts:

(1) Estimate the inverse model.
(2) Assess uncertainty in inverse predictions using a modified bootstrap procedure.

In the inference phase, values of the independent variables are estimated from new observed data by minimizing the difference between the new dependent variable values and a functional prediction from the forward model, using model parameters estimated in the training phase. The minimization problem is solved using optimization techniques, which can be tailored to the specific problem, along with the associated loss function. Several options for loss functions are described in Sec. 3. To assess the uncertainty in the inverse predictions, we employ a modified bootstrap procedure using Monte Carlo simulations from the distributions of the forward model parameters generated in the training phase.

3 Components of Functional Inverse Prediction Framework

This section provides details for each of the steps in the FIP framework described in the previous section.

3.1 Basis Functions. Data can be functional in a theoretical sense, but observed data is always a collection of discrete, finite points. An essential assumption of FDA is that this collection of discrete points is representative of the underlying function of interest. The framework described here involves response data that is functional in a theoretical sense, and so the first step is to estimate a function from the observed discrete response data. This can be accomplished using basis functions to create a representation of the observed response data in some functional space.

Let $\{V_{ijk}\}_{k=1}^{n_i}$ be a vector containing n_i measurements on observation i of response variable j. We would like to represent $\{V_{ijk}\}_{k=1}^{n_i}$ on a functional space, as $V_{ij}(t)$, for some continuous domain t:

$$V_{ijk} = \mathbf{B}_t\boldsymbol{\zeta} + \delta_{ijk} \tag{1}$$

where for m basis functions, $\mathbf{B}_t$ is an n_i by m matrix of fully specified basis functions (note the change in notation from k to t: the rows of this matrix are the m values of the basis functions evaluated at observation k), $\boldsymbol{\zeta}$ is the m-element vector of coefficients on the basis functions, and δ_{ijk} is a mean-zero error term. Using this approach, the discretely collected data $\{V_{ijk}\}_{k=1}^{n_i}$ can be represented in functional space with argument t:

$$V_{ij}(t) = \mathbf{b}(t)'\hat{\boldsymbol{\zeta}} \tag{2}$$

where the m-element vector of basis functions is changed from discrete representation $\mathbf{B}_t$ to functional representation $\mathbf{b}(t)'$ and $\hat{\boldsymbol{\zeta}}$ are the coefficients on the basis functions estimated (for example) using ordinary least squares. Notice that the error term δ_{ijk} is dropped: it is common to assume that all noise is measurement noise and thus not part of the true response function.

Many options for the basis functions have been described in the literature, including Fourier basis [18], power series basis [18], B-splines [20], functional principal component analysis (fPCA) [18,19], and wavelets [21]. The choice of basis is left to the modeler, and this choice should consider the application. For example, applications whose underlying mechanism is expected to be relatively smooth might use B-splines, whereas applications containing data with extensive random microstructure might use wavelets.

3.2 Forward Model. With the response data represented in functional space alongside a vector of p independent variables $\mathbf{x}$, the relationship between the two can be modeled using a suitable forward model. The forward model represents the "cause and effect" part of the framework, where the p-vector of independent variables $\boldsymbol{x}_i$, $i = 1, \ldots, n$, are the causes, the functional dependent variables Y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, q$, are the effects, and the relationship between the two can be treated as either known or unknown. Although forward models are often characterized as "cause and effect", the relationship between the response and the independent variables need not be causal nor is a designed experiment needed to use the FIP framework.

3.2.1 Forward Model for Scalar Response. Because functional models add a layer of complexity, we first introduce forward modeling for the case of scalar responses and vectors of independent variables. Forward models for a scalar response can be expressed as follows:

$$Y_{ij} = g_j(\mathbf{x}_i; \boldsymbol{\theta}_j) + \varepsilon_{ij}, \quad i = 1, \ldots, n; \quad j = 1, \ldots, q \tag{3}$$

where $g_j(\cdot)$ represents a functional relationship, $\boldsymbol{\theta}$ is a vector of unknown model parameters, and ε_{ij}, $i = 1, \ldots, n$; $j = 1, \ldots, q$ are independent and identically distributed (iid) random variables with mean zero, and a finite, constant variance. The standard deviation for the jth component of ε_i is σ_j and the correlation between each ε_{ij} is assumed to be zero. The subscript on $g_j(\cdot)$ allows for the functional form of $g_j(\cdot)$ to change based on response variable j.

The p elements in the vector $\mathbf{x}_i$ are the independent variables for which inverse prediction is desired. The parameters, $\boldsymbol{\theta}_j$ are fixed, unknown values and are commonly estimated with least squares or maximum likelihood techniques. In practice, sample sizes are finite and the assumed likelihood is only an approximation of the true data model. The likelihood function and the form of $g(\cdot)$ are determined by both the data and subject matter knowledge to ensure interpretability and a good model fit. As with all modeling exercises, the model should be validated by assessing the fit of the model, the residual distributions, and the quality of the predictions. Several candidate forms of $g_j(\cdot)$ may be explored with variable selection methods such as criterion-based methods (Akaike information criterion or Bayesian information criterion) or step-wise procedures: independent variables that may not be related to the response may be considered for removal from the model. For the scalar case, in order for the inverse model to generate unique predictions, it must be the case that $q \geq p$ [2].

(a) Multivariate Adaptive Regression Splines. Multivariate adaptive regression splines (MARS) is a flexible non-linear modeling approach for scalar or vector responses, in which the function $g_j(\cdot)$ in Eq. (3) takes the form of an expansion of basis functions that are hinge functions of the independent variables [22]. The MARS forward model for a scalar response with m basis functions is

$$Y_{ij} = \sum_{l=1}^{m} \alpha_l B_l(\mathbf{x}_i) + \varepsilon_{ij}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, q \tag{4}$$

where $B_l(\mathbf{x}_i)$ is a basis function on the independent variables $\mathbf{x}_i$, α_l is the coefficient for the l^{th} basis function, and ε_{ij} is defined the same as for Eq. (3). The number of basis functions used, m, is a preset tuning parameter. When the responses are vectors, Eq. (4) reduces to a multivariate model. Because of their flexibility, MARS models can capture highly nonlinear relationships—an advantage when the underlying relationships between independent variables and responses is complex or involves non-smooth functions.

3.2.2 Forward Model for Functional Response. Forward models for functional responses are more complex than the scalar case, because the effect of scalar independent variables on entire functions must be determined. The model parameters are often themselves functions describing changes over the functional domain. As discussed in Sec. 3.2.1, forward models with scalar responses require that the number of response variables is larger than the number of independent variables. However, because of the richness of the responses and the flexibility of functional data, FDA allows for potentially relaxing this assumption such that a smaller number of responses may be sufficient to uniquely determine the values of the independent variable [3].

A functional forward model can be written as

$$Y_{ij}(t) = g_j(\mathbf{x}_i; \boldsymbol{\theta}_j(t)) + \varepsilon_{ij}(t); \quad i = 1, \ldots, n; \quad j = 1, \ldots, q \tag{5}$$

where the regression coefficients $\boldsymbol{\theta}_j(t)$ are functions of the domain t and $\varepsilon_{ij}(t)$, $i = 1, \ldots, n$; $j = 1, \ldots, q$ are iid, mean-zero, second-order stationary stochastic processes. Equation (5) looks almost the same as the scalar version Eq. (3), with the only differences being the response and model parameters are functions. Many linear and nonlinear models can be cast in the general model framework of Eq. (5), including functional linear regression, described later.

a Functional Linear Regression. In functional linear regression, the function $g(\cdot)$ in Eq. (5) takes the form of a linear combination of regression coefficients and independent variables. Functional linear regression is conceptually similar to standard linear regression; however, in functional linear regression the response variables are functions of the domain t rather than scalars. As a result, the regression parameters are also functions of t. The functional regression model is

$$Y_{ij}(t) = \mathbf{x}_i'\boldsymbol{\beta}_j(t) + \varepsilon_{ij}(t); \quad i = 1, \ldots, n; \quad j = 1, \ldots, q \tag{6}$$

where the p regression coefficients in the vector $\boldsymbol{\beta}_j(t)$ are functions of the domain t and $\varepsilon_{ij}(t)$ is an iid, mean-zero, second-order stationary stochastic process, the same as for Eq. (5). Computationally, to estimate the regression coefficient functions $\boldsymbol{\beta}_j(t)$, the response, $Y_{ij}(t)$, is evaluated at a large number of values of t and ordinary least squares is performed for each t to obtain estimates of the regression coefficients at each t. Functional linear models are simple yet powerful statistical models whose parameters are easily interpreted. Additionally, the small number of parameters reduces the danger that the model will be overfit.

3.3 Inverse Model. The inverse model generates predictions for the p independent variables in $\mathbf{x}$ for newly observed response data Y^* by minimizing with respect to $\mathbf{x}$ the distance between the fitted forward model's estimated response $\hat{Y}$, and Y^*.

3.3.1 Inverse Model for Scalar Response. Given the estimated parameters $\hat{\boldsymbol{\theta}}_j$ and new responses $\{Y_j^*\}_{j=1}^q$, we estimate $\mathbf{X}^*$ with $\hat{\mathbf{x}}^*$. Note the capitalization of unobserved $\mathbf{X}^*$ indicates it is now treated as a random variable. The best estimate is the value that minimizes a specified loss function and describes the penalty for estimating Y^* with $\hat{y}^*$:

$$\hat{\mathbf{x}}^* = \underset{\mathbf{X}^*}{\operatorname{argmin}} \sum_{j=1}^{q} L(\hat{y}_j^*, Y_j^*) \tag{7}$$

where $L(\cdot)$ is the loss function and $\hat{y}_j^* = g_j(\mathbf{X}^*; \hat{\boldsymbol{\theta}}_j)$. The optimization in Eq. (7) is said to be the "inverse model." The most commonly used loss function is the squared error loss, where the minimization criterion is

$$\hat{\mathbf{x}}^* = \underset{\mathbf{X}^*}{\operatorname{argmin}} \sum_{j=1}^{q} (\hat{y}_j^* - Y_j^*)^2 \tag{8}$$

To keep large-magnitude responses from dominating this sum, all responses are usually normalized to a common scale.

3.3.2 Inverse Model for Functional Response. For inverse prediction, we assume that the newly observed response functions were generated from a single p-element vector of independent variable values (e.g., a single set of processing conditions) $\mathbf{X}^*$. To predict the values of the independent variables $\mathbf{X}^*$ from the observation's q functional responses, given $\hat{\boldsymbol{\theta}}_j(t), j = 1, \ldots, q$ and newly observed functions Y_j^*, the functional inverse model is

$$\hat{\mathbf{x}}^* = \underset{\mathbf{X}^*}{\operatorname{argmin}} \sum_{j=1}^{q} \int L(\hat{y}_j(t), Y_j^*(t))dt \tag{9}$$

where $\hat{y}_j(t) = g_j(\mathbf{X}^*, \hat{\boldsymbol{\theta}}_j(t))$ with loss function $L(\cdot)$. For example, the squared-error loss for functional data is

$$\hat{\mathbf{x}}^* = \underset{\mathbf{X}^*}{\operatorname{argmin}} \sum_{j=1}^{q} \int \left(\hat{y}_j(t) - Y_j^*(t)\right)^2 dt \tag{10}$$

In practice, the integral in Eq. (10) is approximated with a sum over a large number of discrete values of t.

3.4 Uncertainty Quantification of Inverse Predictions. The inverse prediction model in Eq. (9) will always generate a unique predicted value for the independent variables. However, it does not provide an assessment of uncertainty on that prediction. Given the complexity and diversity of inverse prediction models compatible with this framework, bootstrap procedures are a logical choice for providing uncertainty estimates [23]. The modified bootstrap procedure to provide uncertainty quantification for inverse prediction is as follows: let $(\mathbf{x}, \mathbf{Y})$ be training data consisting of n observations, and $\mathbf{Y}^*$ be newly observed functional response data with which to predict unobserved independent variables $\mathbf{X}^*$. Then, for $b = 1, \ldots, B$:

(1) Create the bth bootstrap sample $(\mathbf{x}^b, \mathbf{Y}^b)$ by sampling n times with replacement from $(\mathbf{x}, \mathbf{Y})$,
(2) Train forward model in Eq. (5) with $(\mathbf{x}^b, \mathbf{Y}^b)$, to generate $\hat{\boldsymbol{\theta}}^b$
(3) Perform inverse prediction in Eq. (9) with $\mathbf{Y}^*$ and $\hat{\boldsymbol{\theta}}^b$ and record prediction $\hat{\mathbf{x}}^{*b}$.

For a vector of p independent variables, prediction standard errors for the sth, $s = 1, \ldots, p$ variable are given by computing the standard deviation of $\{\hat{x}_s^{*b}\}_{b=1}^B$. Prediction intervals can be constructed using the same procedure, such that the true value for x_s will lie between the $\alpha/2$ and $(1-\alpha/2)$ quantiles of $\{\hat{x}_s^{*b}\}_{b=1}^B$ approximately $(1-\alpha)\%$ of the time. The value of B should be large (10,000 is generally sufficient in our experience.)

3.5 Model Performance and Assessment. The predictive capabilities of the model can be assessed using cross validation (CV) procedures. Using k-fold CV, the training data is grouped into k sets of equal size, then the model is trained using $k-1$ sets and evaluated for performance on the set that has been withheld. This process is repeated for each of the k sets, prediction performance is quantified at each step with a loss function $L(\cdot)$, and overall performance is measured by the average loss over the k sets. Leave-one-out CV (LOOCV) is a special case of k-fold CV where $k = n$, the sample size. In the presence of large quantities of data, k-fold CV is preferred over LOOCV in order to reduce computational costs.

Different applications can call for different loss functions depending on the researcher's goals. The root mean-squared error (RMSE) loss function is a common choice for evaluating predictive ability:

$$RMSE(\hat{x}_s) = \sqrt{\frac{\sum_{i=1}^{n} (\hat{x}_{ij} - x_{ij})^2}{n-1}} \tag{11}$$

Different candidates for the forward model can be compared using CV RMSE: the model with lowest RMSE has the best predictive performance in terms of squared-error loss. When using RMSE to evaluate a single model, the scale of the data values must be considered. If the RMSE is higher than the standard deviation of the data, a simple intercept-only model performs better than the model under consideration. Because raw values of RMSE are hard to interpret, centering and scaling independent variables $\mathbf{x}$ before modeling is helpful, since a value of 1 describes the variability associated with using the sample mean as the inverse prediction.

4 Simulation Study

For functional data analysis, we assume the data are generated by an underlying continuous process that can be expressed in functional form. Nevertheless, in practice we observe and record discrete response values, and the data used in model computations are also necessarily discrete. In standard multivariate analysis, the data values are used directly for model computations; the response values fed into the model are just the observed values. In functional data analysis, the discrete response data are first represented in functional form using basis functions, and the discrete values of those basis functions are used in the model computations. In this section, we illustrate the FIP framework using two forward model forms—functional linear regression and MARS—and compare the results of the functional models to a standard regression approach in which the response data are not represented in functional form.

Let $\mathbf{x}_1 = (x_{11}, x_{21}, \ldots, x_{n1})'$ be the first independent variable, where $\{x_{i1}\}_{i=1}^n$ take values on the sequence from 1.1 to 3.0, by 0.1. Similarly, let $\mathbf{x}_2 = (x_{12}, x_{22}, \ldots, x_{n2})'$ be the second independent variable, where $\{x_{i2}\}_{i=1}^n$ take values on the sequence from 0.5 to 2.4, by 0.1, in random order. Randomizing the order of the values in $\mathbf{x}_2$ ensures there is no correlation between $\mathbf{x}_1$ and $\mathbf{x}_2$. The functional observations, $\mathbf{Y}_1(t) = (Y_{11}(t), Y_{21}(t), \ldots, Y_{n1}(t))'$, $\mathbf{Y}_2(t) = (Y_{12}(t), Y_{22}(t)\ldots, Y_{n2}(t))'$ are generated by Eq. (12) and (13), respectively,

$$Y_{i1}(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}x_{i1}} e^{-\frac{1}{2x_{i1}^2}u^2} du \tag{12}$$

$$Y_{i2}(t) = x_{i1} \sin(x_{i2}t) \tag{13}$$

For computational purposes, we evaluate a large number of observations, n_t, over the domain t of each function, generating an $n \times n_t$ observation matrix for each response variable $\mathbf{Y}_1(t)$ and $\mathbf{Y}_2(t)$. Figure 2 plots $\mathbf{Y}_1(t)$ for $t \in (-4, 4)$, with each line colored by its respective $\mathbf{x}_1$ value. Figure 3 plots $\mathbf{Y}_2(t)$ for $t \in (-2, 2)$,

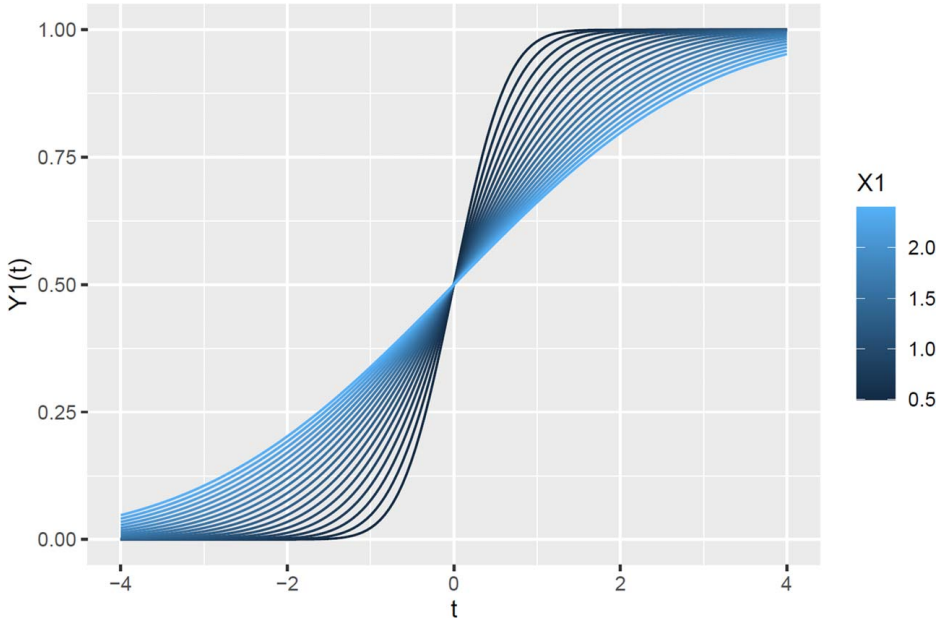

Fig. 2 Simulated functional responses $Y_1(t)$, each colored by its respective x_1 value. Generated according to Eq. (12).

with each line colored by its respective $\mathbf{x}_2$ value. Inspection of these plots shows clearly the relationship between the shape of the response function and the value of the independent variable.

We fit both a functional linear and a MARS forward model using the two independent variables and a basis representation of the two observation matrices. In this example, the $n \times 2$ matrix of response values used in the computation are given by the scores from the first axis of an fPCA decomposition of the two observation matrices (although other basis representations can be used). To compare the FIP model to a model that doesn't consider the functional nature of the data, we also used the simple row means of the two $n \times n_t$ observation matrix, concatenated into an $n \times 2$ matrix, as the response. The row mean is the discrete equivalent of the expected value of the function that generated the observations, $\int t\, Y(t) dt$. The mean-only model is equivalent to the usual multiple regression model for the linear model case, or a standard MARS model, neither of which account for correlation in the response data due to "nearness" on the functional domain t.

Table 1 shows the RMSE of the IP evaluated using LOOCV. The left side shows results for the functional model and the right side shows results for the mean-only model, with prediction standard errors in parentheses. The mean-only model fails to recover the relationship between $\mathbf{Y}(t)$ and $\mathbf{x}$, while the functional model performs well. Both the linear and MARS models show good prediction performance on $\mathbf{x}_1$, but $\mathbf{x}_2$ is more difficult. MARS outperforms the linear model in both cases, as expected since the generating functions are non-linear. The prediction standard errors for the FIP

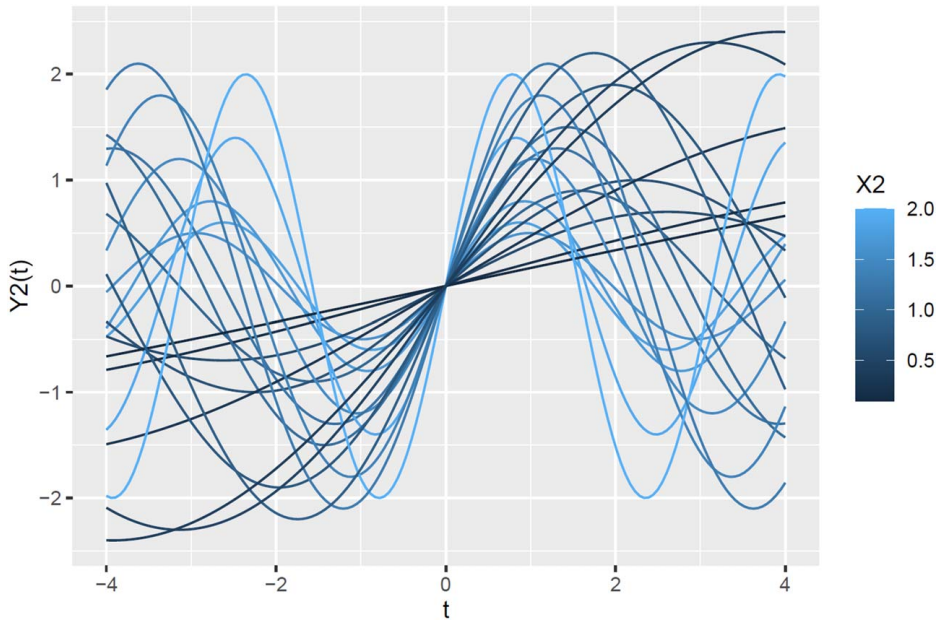

Fig. 3 Simulated functional responses $Y_2(t)$, each colored by its respective x_2 value. Generated according to Eq. (13).

Table 1 RMSE for IP of simulation study. Left side has results for functional model and right side for mean-only model, standard errors are in parentheses

	Functional		Mean only	
	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_1$	$\mathbf{x}_2$
Linear	0.06 (0.04)	0.57 (0.26)	1.80 (1.27)	1.56 (1.20)
MARS	0.02 (0.04)	0.51 (0.26)	0.97 (1.28)	0.97 (1.18)

Note: Mean-only model fails to see relationship between $\mathbf{Y}(t)$ and $\mathbf{x}$ while functional model performs well.

model are consistent across the linear and MARS forward models, with very little error in predicting $\mathbf{x}_1$.

In this simple example, taking the mean of the response and ignoring the functional nature of the data would lead to extremely poor inverse predictions. The next section applies the FIP framework to real data.

5 Applications

In this section, we apply the FIP framework to two empirical datasets: the Canadian weather data presented in Ref. [18], and original data collected for a nuclear forensics FIP project. In both applications, we compare FIP to the mean-only method described in Sec. 4.

5.1 Canadian Weather. Average daily temperature and precipitation values from 35 cities in Canada are included in the R package **fda** [24]. Figure 4 displays average monthly temperatures colored by the latitude of the respective city, while Fig. 5 displays monthly averages for precipitation. Using the FIP framework, we would like to determine through inverse prediction the latitudes and longitudes of the 35 cities.

Let $Y_{i1}(t)$ be the average monthly temperature for city i, $i = 1, \ldots, 35$, $Y_{i2}(t)$ is the average monthly precipitation for city i, x_{i1} is the latitude of city i, and x_{i2} is the longitude of city i. We first obtain fPCA representations of the temperature and precipitation functions and use the first four principal components to fit both a linear and MARS functional forward model. As in Sec. 4, we also fit a mean-only model for comparison. Latitudes and longitudes were standardized to make interpretation of the RMSE values easier.

Table 2 displays RMSE values and prediction standard errors for the inverse predictions using LOOCV and bootstrapping. Overall, the functional approach generates more accurate predictions and smaller standard errors than the mean-only model. Latitude predictions are better than longitude predictions for both modeling approaches and both forward model types. The linear model actually performs better than the flexible MARS model, indicating that the more complex model may overfit the data, even after penalization.

5.2 Nuclear Forensics. Nuclear forensics is a field of forensics which focuses on identifying and attributing to a set of processing conditions interdicted special nuclear materials such as Plutonium (Pu). Knowing the processing conditions used to produce, the interdicted material is helpful in determining where the material originated. We apply the FIP framework to several types of functional data generated by this nuclear forensics application, showcasing several ways to cast difficult data structures into functional space.

Experts at Pacific Northwest National Laboratory conducted an experiment which replicated historical and modern Pu processing methods and conditions. We analyze a subset of the data consisting of 24 runs, where each run employed the same set of three processing conditions (independent variables) whose values varied across runs. We denote the processing conditions Condition 1, Condition 2, and Condition 3. For each run, the resulting Pu material was imaged with a scanning electron microscope (SEM) to generate images of the Pu particles. Each run i produced n_i images, and each image contained n_p particles. The minimum number of particles imaged per run was 118 and the maximum was 831.

The images were segmented using Los Alamos National Laboratory's Morphological Analysis for Material Attribution (MAMA) software [25]. Segmentation post-processing extracts the unique particles and generates measurements for characteristics such as particle area and aspect ratio. Figure 6 shows an example SEM image from the PNNL experiment containing multiple particles.

5.2.1 Particle Characteristics. For each particle, MAMA measures particle area, aspect ratios, convexity/circularity, and gradient/shadings. As mentioned in the previous section, each

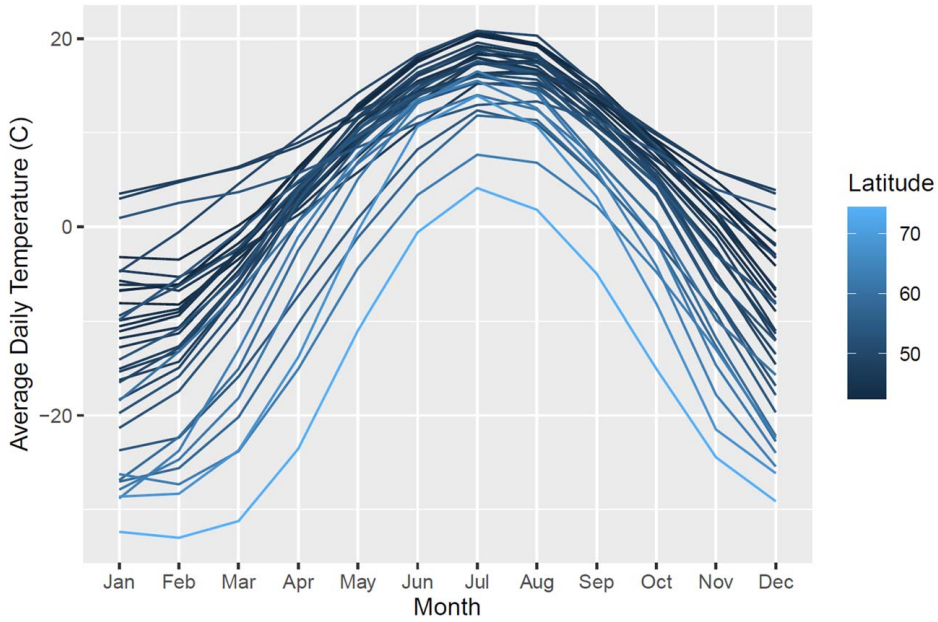

Fig. 4 Monthly average temperatures for 35 cities in Canada, colored by the latitude of the city

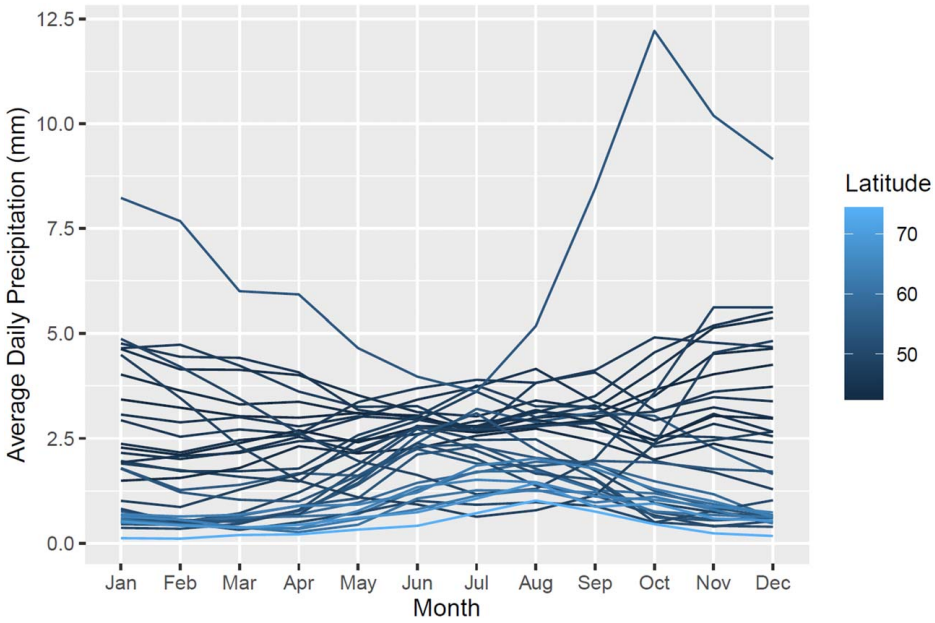

Fig. 5 Monthly average precipitation for 35 cities in Canada, colored by the latitude of the city

experimental run contains multiple images with multiple particles in each image, resulting in hundreds of MAMA measurements for a single experimental run and set of processing conditions. These hundreds of observations can be conceptualized as a probability distribution of responses, and the functional analysis relates the cumulative distribution function (CDF) of the observations Y, $F_y(t) = P(Y \leq t)$, to the set of processing conditions employed during each run. The center, spread, and shape of various MAMA measurements can then be related to the processing features instead of only the mean. An approach introduced by Ref. [3] uses the distribution data by developing a forward model that regresses the CDFs of MAMA measurements on their respective processing conditions through functional regression.

To convert distributional data to functional, let $\{Y_{ijk}\}_{k=1}^{n_i}$ be a vector of observations from experimental run $i = 1, \ldots, n$ and response variable $j = 1, \ldots, q$ of length n_i corresponding to a vector of processing conditions $\mathbf{x}_i$. To compare distributions across runs, the empirical CDF of $\{Y_{ijk}\}_{k=1}^{n_i}$ can be estimated using

$$Y_{ij}(t) = \frac{1}{n_i} \sum_{k=1}^{n_i} I(Y_{ijk} \leq t) \tag{14}$$

where Y_{ijk} represents the kth element in $\mathbf{Y}_{ij}$, t is the domain for the CDF of Y_{ij}, and $I(\cdot)$ is the indicator function. To visualize the relationship between distributions and processing conditions, Fig. 7 shows the CDFs of particle area convexity for each run, colored by the value of processing Condition 1. Certain relationships are visible; for example, higher values of Condition 1 lead to stronger convexity in the response curve.

Table 2 RMSE for IP Canadian weather application

	Functional		Mean only	
	Latitude	Longitude	Latitude	Longitude
Linear	0.47 (0.22)	0.70 (0.53)	0.52 (0.32)	0.99 (0.73)
MARS	0.50 (0.37)	1.07 (0.74)	0.84 (0.46)	1.83 (0.90)

Note: Left side has results for functional model and right side for mean-only model, standard errors are in parentheses. The mean-only model fails to see relationship between $\mathbf{Y}(t)$ and $\mathbf{x}$ while the functional model performs well.

5.2.2 Particle Shape. An alternative approach to relating distributions of particle characteristics to processing conditions is by using the particle's shape, since different values of the processing conditions can result in different particle shapes. Shape analysis is a relatively new field, and we direct interested readers to Ref. [19] for a detailed introduction.

The MAMA segmentations extract the boundaries (shape) of each particle, and these 2D boundaries are analyzed using the elastic shape framework described in detail in the Appendix. In summary, particle shapes within a run are aligned to their Karcher mean, a process which returns a set of warping functions describing the sampling variability along the curves and a set of rotation matrices representing rotational variability of the curves. This has the effect of stretching/compressing and bending parts of the curves to match each other in an optimal fashion.

The optimally aligned and rotated curves are represented using the square-root velocity functions, $q_{ika}(t)$ for particle a in image k, $k = 1, \ldots, n_i$ from experimental run i, which removes the effect of

Fig. 6 Example of SEM image of Pu from PNNL experiment showing multiple particles

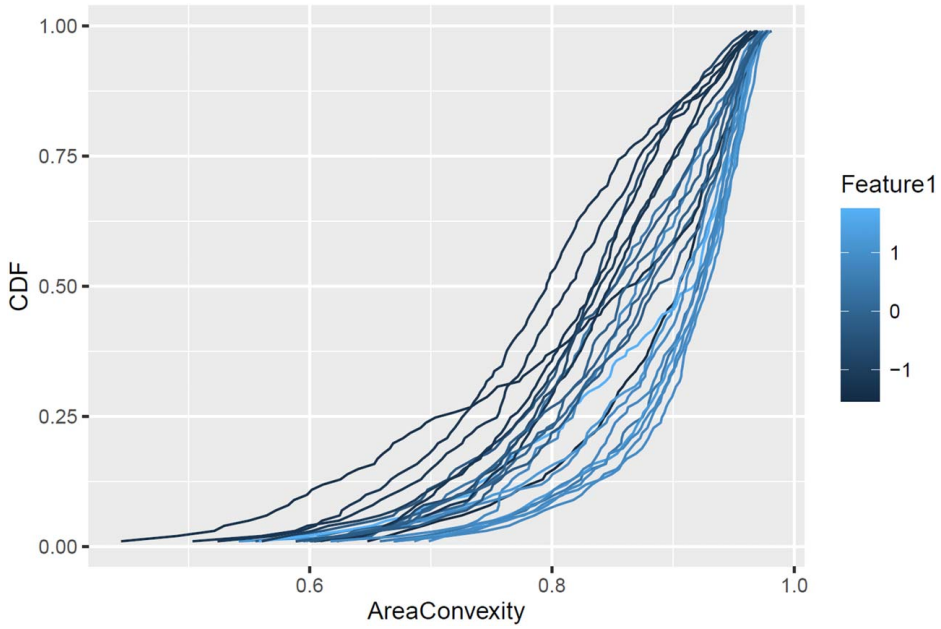

Fig. 7 Run level CDFs of MAMA measurements used in model colored by value of Condition 1

scale. For each shape, a shape-PCA is computed where the functional principal shape components are used as a basis for the forward model. Note that, in contrast to the particle characteristics described in the previous section, the analysis of particle shapes is conducted at the particle level rather than the run level. The response matrix therefore contains $\sum_{i=1}^{n} n_i$ rows, and in the matrix of independent variables (processing conditions), the values for each run i are repeated n_i times. This framework results in metrics, statistics, and models that are invariant to arbitrary rotation, scaling, translation, and re-parametrization of individual curves. For a more complete review, the reader is referred to Refs. [19,26]. This technique is an extension of the functional work in Refs. [27,28] and provides the same advantage of simultaneous registration and comparison of the shapes of the curves with respect to the elastic metric. Figure 8 shows the first four principal directions for one of the experimental runs. The modes of shape variation can be seen with respect to the black middle shape, the overall mean.

5.2.3 Particle Texture. Crystal structure and habit can be linked to varying processing conditions associated with final Pu material. Texture, as captured by SEM imagery, is characterized by local and non-local features that are a function of the intensity

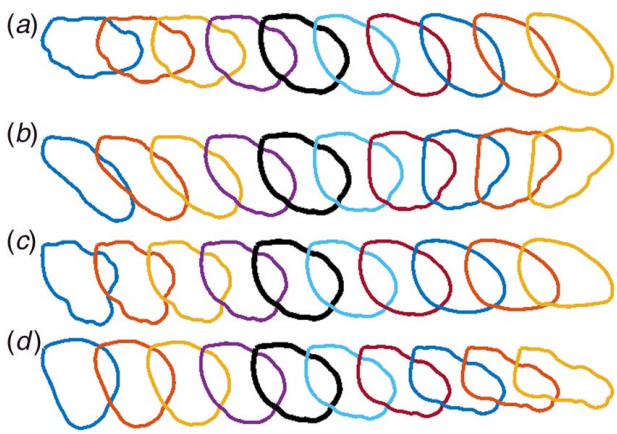

Fig. 8 First four principal directions of the shape PCA: (*a*) PC 1, (*b*) PC 2, (c) PC 3, and (*d*) PC 4

of the pixels in the images. Collectively, these features define a potential texture "signature" related to the processing conditions under which the material was produced. An example showing particle images from two different processing conditions is shown in Fig. 9. Informally, the two particle textures may be described as "rough" and "flat."

The texture model uses a dataset of segmented SEM images taken over the 24 runs in the experimental design. The particle images within each run capture the distribution of Pu texture characteristics produced under those conditions. The three particles in Fig. 6, for example, were produced from the same experimental run and therefore three particle-level texture samples for those processing conditions.

Image filters used as a preprocessing step can capture features of particle texture that may not be reflected in the raw pixel intensities of the SEM image. A standard deviation (SD) filter, which calculates the SD of nearby pixel values in a neighborhood around an input pixel, acts as an edge detector highlighting neighborhoods of high variance with the images. Formally, let Z_{hw} be the raw value of some input pixel z from an SEM image with dimension $H \times W$. The local standard deviation around this input pixel is calculated using the raw pixel values from a square neighborhood around z, that is, the set of pixels $z_{h'w'}$ where $h' = h-u, \ldots, h+u$ and $w' = w-u, \ldots, w+u$. We let $u = 1$, resulting in a 9×9 pixel neighborhood, with the input pixel for which the SD is calculated in the center. For pixels along the image's border, padding values are added using a mirror reflection. Using this procedure, each input pixel z from image k, $k = 1, \ldots, n_i$ from run i has a corresponding local SD value Y_{ikz}. The CDF of these SD values (Eq. (14)) for each particle can then be used as a response function.

Considering the images in Fig. 9, the "rough" particle has more high-variance neighborhoods (i.e., more "edges') than "flat particles," resulting in characteristically different CDFs of local SD values. As a result, empirical CDFs estimated from the local SD values can be used effectively as a response function which relates back to the material processing conditions. However, unlike the estimation procedure for empirical CDFs described in Sec. 5.2.1 in Eq. (14), the functional response from the local SD values are CDFs produced at the particle level rather than the run level, similar to the procedure for shape analysis.

5.2.4 FIP Framework Applied to Nuclear Forensics Responses. For this application, we use an fPCA basis, choosing

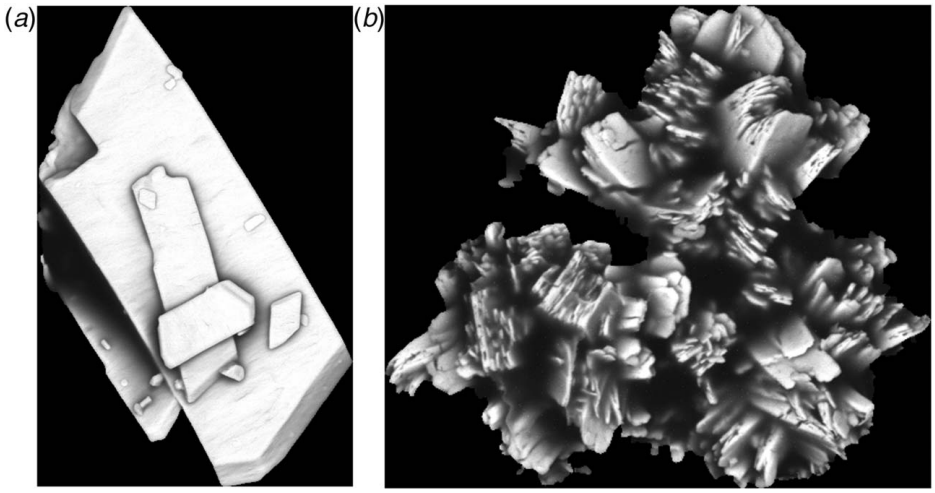

Fig. 9 Particle textures from two different processing: (*a*) flat particle and (*b*) rough particle conditions

the number of fPC basis functions to ensure >90% of the variability is explained by the fPCs. In the case of the particle characteristics analysis, the first three fPCs of the CDFs of the four response variables for each run are used as the model response. For the particle shape analysis, the first 74 fPCs from the shape of each particle are used as the model response. Finally, for the particle texture analysis, the first five fPCs from the CDF of texture of each particle are used as the model response. For the forward models, we implement both the functional and mean-only linear and MARS models described in Sec. 3.2.2. The general inverse prediction procedure is the same for both linear and MARS forward models, and both are optimized using squared error loss as in Eq. (10).

5.2.5 Results. We evaluate model performance using metrics of predictive ability as in Sec. 3.5 and generate prediction uncertainties as described in Sec. 3.4. A slight modification to the bootstrap procedure was made due to the small sample size: the Fractional-Random-Weight Bootstrap [29] samples weights from a Dirichlet distribution that sum to the sample size, and these weights are assigned to each observation for the forward model estimation, for each bootstrap sample. This allows for the original experimental design and still measure uncertainty without extrapolation. Tables 3 and 4 show RMSE for the linear forward model and the MARS forward model, respectively, for each of the characteristic response functions. Standard errors for predictions are in parentheses. Overall, the functional response model that uses the cumulative distribution function performs best, with similar RMSE for linear and MARS forward models. In general, the more flexible MARS does not perform better than the linear model. The results from the FIP framework compared to the mean-only model show some evidence that the functional representations of the data at the run level are informative for IP. However, the relatively large prediction errors for all models indicate that this data is noisy. The poor performance of the particle-level approaches could be due to the considerable particle-to-particle variation, and a more complex model that includes a random effect for run may improve performance.

Table 3 RMSE of inverse predictions for linear models using CV with standard errors in parentheses

Measure	Condition 1	Condition 2	Condition 3
Distributional	0.84 (0.32)	0.73 (0.28)	0.96 (0.27)
Shape	1.34 (0.16)	1.19 (0.13)	1.19 (0.20)
Texture	1.18 (0.07)	1.45 (0.10)	1.17 (0.08)
Mean-only	1.40 (0.13)	0.84 (0.08)	1.16 (0.10)

Table 4 RMSE of inverse predictions for MARS models using CV with standard errors in parentheses

	Condition 1	Condition 2	Condition 3
Distributional	0.94 (0.17)	0.82 (0.26)	0.96 (0.09)
Shape	1.00 (0.69)	1.28 (0.25)	0.97 (0.60)
Texture	1.24 (0.39)	1.19 (0.11)	1.25 (0.30)
Mean-only	0.94 (0.07)	0.99 (0.23)	1.10 (0.30)

6 Discussion

The simulation study and applications demonstrated the FIP framework in use, and how a functional approach compares to a simple aggregation of the data through the mean summary statistic. The FDA approach to IP improved predictive performance, uncertainty quantification, or both. This framework is useful not only in situations where IP is needed but can also be used as a tool in scientific inference, as FIP models which perform well can suggest new scientific hypotheses about relationships seen empirically in strong IP results.

There are several avenues for future work on this framework. The basis representation of the functional response does not consider the error in the choice of basis functions or the basis approximation of the response function; this error could propagate forward to increase the prediction uncertainties. Future modifications of the framework will allow for multivariate response models that account for correlations among response functions, as addressed in Sec. 5.2.5, allowing the framework access to more data to improve predictions and uncertainty estimates. Extending FIP to a Bayesian framework would also be useful, as posterior distributions can be used to estimate precision without the need for bootstrapping, and subject matter knowledge can be incorporated through priors.

7 Conclusion

This paper presented a framework for inverse prediction in scenarios where response data are functional. The FDA approach maximizes the information contained in the data to provide the inverse model with greater prediction power. The framework was created to be flexible and customizable to many different problems and applications. Different methods for representing functional data through basis functions are introduced, and the FIP framework does not

force a particular specification. Examples of methods to represent the data as functional are given in both the simulation study and in applications to weather and nuclear forensics. The forward model, which is sometimes thought of as representing the "cause and effect" relationship, is also left to the analyst to tailor to the application; we present two options but many others are available. For the inverse model, we present a general optimization procedure, but again allow for customization through different loss functions. Finally, the FIP framework quantifies uncertainty using bootstrap procedures, providing prediction intervals for inverse predictions. The framework presented for inverse prediction modeling with functional data provides researchers with a straightforward, general approach that can be applied to a variety of specialized inverse prediction problems.

Acknowledgment

We acknowledge the Department of Homeland Security National Technical Forensics Center for funding this work at Sandia National Laboratories and Los Alamos National Laboratory. The authors thank Audrey L. McCombs for her thoughtful insights into revisions of the manuscript. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness. SAND2022-0974 J.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data and information that support the findings of this article are available upon request.

Nomenclature

- t = reserved as general argument for functional data
- $\hat{y}$ = a hat over a Latin letter indicates a predicted value
- Y^* = values with an asterisk indicate an unobserved or future value (not used to train model)
- $Y(t)$ = values that show as a function of t are functional data variables
- θ = greek letters are unknown model parameters to be estimated unless otherwise stated
- $\hat{\theta}$ = a hat over a Greek letter indicates a parameter estimate

Appendix: Elastic Shape Analysis of Open Curves

Consider an absolutely continuous, parametrized curve $\xi : [0, 1] \to \mathbb{R}^2$. Define the set of all re-parameterizations as the set of diffeomorphisms

$$\Gamma = \{\gamma : [0, 1] \to [0, 1] | \gamma(0) = 0, \gamma(1) = 1, \gamma \text{ increasing}\} \quad \text{(A1)}$$

By definition, γ and γ^{-1} are absolutely continuous. Define a re-parameterization of the curve ξ as the composition $\xi \circ \gamma$, where $\gamma \in \Gamma$. A major problem that arises under this framework is if the distance between two curves ξ_1 and ξ_2 are taken using standard metrics. Most papers use the standard $\mathbb{L}^2$ metric $\|\xi_1 - \xi_2\|$ to find the distance. However, the standard $\mathbb{L}^2$ metric is not isometric with respect to the group action of Γ. That is, for $\gamma \in \Gamma$, $\|\xi_1 - \xi_2\| \neq \|\xi_1 \circ \gamma - \xi_2 \circ \gamma\|$. Since there is the desire to be invariant to re-parameterization and the $\mathbb{L}^2$ metric is not isometric with respect to the group action, it is not possible to compute distances that are invariant to re-parameterization. In order to achieve this property, Srivastava et al. [26] introduced a novel representation of curves. They represent a shape by its square-root velocity function (SRVF)

$$q(t) = \frac{\dot{\xi}(t)}{\sqrt{|\dot{\xi}(t)|}} \quad \text{(A2)}$$

The warping of q is given by group action as $(q, \gamma) \equiv (q \circ \gamma)\sqrt{\dot{\gamma}}$. It is easy to show that for two SRVF's q_1 and q_2, $\|q_1 - q_2\| = \|(q_1, \gamma) - (q_2, \gamma)\|$. Therefore, the action of the group Γ is isometric on the space of SRVFs.

Another important motivation for the SRVF representation is that elastic Riemannian metric (a metric that measures a combination of stretching and bending to optimally deform on curve into another) is equal to the standard $\mathbb{L}^2$ metric, and therefore much simpler and easier to compute. Since $|q(t)|^2 = |\dot{\xi}(t)|$, the square of the $\mathbb{L}^2$-norm of any SRVF is equal to the length of the corresponding curve ξ. That is, $\|q\|^2 = \int_0^1 |q(t)|^2 \, dt = \int_0^1 |\dot{\xi}(t)| \, dt = L_\xi$. Therefore, the $\mathbb{L}^2$ norm of SRVF's of unit-length curves is one, and the space of such curves is a Hilbert Sphere (unit sphere in infinite dimensional function space)

$$\mathbb{S}^\infty = \{q \in \mathbb{L}^2([0, 1], \mathbb{R}^2) | \|q\| = 1\} \quad \text{(A3)}$$

The space $\mathbb{S}^\infty$ represents the SRVF's of all unit-length open curves, and once endowed with the $\mathbb{L}^2$ Riemannian metric, becomes a Riemannian manifold.

A geodesic on a Riemannian manifold is defined as shortest length path with respect to the chosen metric between two points on the manifold. Geodesic distance is the length of the path that is used as the distance between shapes. Since the geometric structure of the manifold is known, $\mathbb{S}^\infty$, the geodesic can be expressed analytically. With respect to the $\mathbb{L}^2$ metric, the geodesic between two points on the unit sphere is defined as the arch segment of the great circle that passes through those two points. Therefore, the geodesic distance is the arc length of this path and mathematically the geodesic path $\nu : [0, 1] \to \mathbb{S}^\infty$ with $\nu(0) = q_1$ and $\nu(1) = q_2$ is given by

$$\nu(\tau) = \frac{1}{\sin(\theta)} [\sin((1 - \tau)\theta) q_1 + \sin(\tau\theta) q_2] \quad \text{(A4)}$$

where $\theta = \cos^{-1}(\langle q_1, q_2 \rangle)$ is the arc length between q_1 and q_2 and $\langle \cdot, \cdot \rangle$ is the $\mathbb{L}^2$ inner product. Therefore, the geodesic distance is defined as $d_{\mathbb{S}^\infty} = \theta$. Figure 10 shows a diagram of a geodesic path and distance between two points on $\mathbb{S}^\infty$.

The space $\mathbb{S}^\infty$ is called the *pre-shape space* of elastic curves, as the invariants of translation and scaling have been removed.

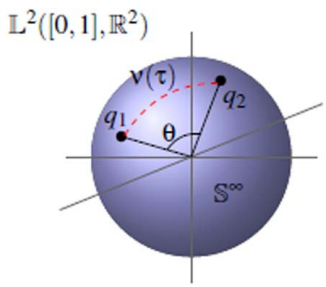

Fig. 10 A geodesic between two points on $\mathbb{S}^\infty$

However, the rotation and re-parameterization have yet to be removed. The rotation, or special orthogonal group, $SO(2)$ defines the usual two-dimensional rotations around a point and acts by isometries with respect to the $\mathbb{L}^2$ metric. Therefore, the quotient space $S_s^O = \mathbb{S}^\infty/(SO(2) \times \Gamma)$ is the similarity invariant, elastic shape space of open curves. Elements of the quotient space are equivalence classes $[q] = \text{closure}\{O(q, \gamma) | O \in SO(2), \gamma \in \Gamma\}$, where $q \in \mathbb{S}^\infty$. Therefore, the distance between orbits $[q_1]$ and $[q_2]$ in the shape space is

$$d_{S_s^O}([q_1], [q_2]) = \inf_{O \in SO(2), \gamma \in \Gamma} d_{\mathbb{S}^\infty}(q_1, O(q_2, \gamma)) \tag{A5}$$

and $d_{\mathbb{S}^\infty}$ is the geodesic distance. The optimization in Eq. A5 is performed via Procrustes rigid body alignment for $SO(2)$ and via dynamic programming in the case of Γ. Both of these methods are described in Ref. [26], and the reader is referred to that paper for more details of this framework.

Given that there is a distance between orbits, the Karcher Mean curve can be calculated given a set of curves $\{\xi_1, \ldots, \xi_N\}$ and their corresponding SRVFs, $\{q_1, \ldots, q_N\}$. This computation is found finding the orbit that minimizes the sum of squared distances, i.e.,

$$u = \underset{[q] \in \mathbb{S}^\infty}{\text{argmin}} \sum_{i=1}^{N} d_{S_s^O}([q], [q_i])^2 \tag{A6}$$

References

[1] Wells, J., and LaMotte, L. R., 1995, "Estimating Maggot Age From Weight Using Inverse Prediction," J. Forensic Sci., **40**(4), pp. 585–590.

[2] Lewis, J. R., Zhang, A., and Anderson-Cook, C. M., 2018, "Comparing Multiple Statistical Methods for Inverse Prediction in Nuclear Forensics Applications," Chemom. Intell. Lab. Syst., **175**, pp. 116–129.

[3] Ries, D., Lewis, J., Zhang, A., Anderson-Cook, C., Wilkerson, M., Wagner, G., and Gravelle, J., 2019, "Utilizing Distributional Measurements of Material Characteristics From SEM Images for Inverse Prediction.," J. Nucl. Mater. Manage., **XLVII**(1), pp. 37–46.

[4] Haaland, D. M., and Thomas, E. V., 1988, "Partial Least-Squares Methods for Spectral Analyses. 1. Relation to Other Quantitative Calibration Methods and the Extraction of Qualitative Information," Anal. Chem., **60**(11), pp. 1193–1202.

[5] Kennedy, M. C., and O'Hagan, A., 2001, "Bayesian Calibration of Computer Models," J. R. Stat. Soc.: Seri. B (Stat. Methodol.), **63**(3), pp. 425–464.

[6] Martens, H., and Naes, T., 1992, *Multivariate Calibration*, John Wiley & Sons, New York.

[7] Kaaks, R., Riboli, E., and Van Staveren, W., 1995, "Calibration of Dietary Intake Measurements in Prospective Cohort Studies," Am. J. Epidemiol., **142**(5), pp. 548–556.

[8] Krasnopolsky, V. M., and Schiller, H., 2003, "Some Neural Network Applications in Environmental Sciences. Part I: Forward and Inverse Problems in Geophysical Remote Measurements," Neural Netw., **16**(3-4), pp. 321–334.

[9] Sun, N.-Z., 2013, *Inverse Problems in Groundwater Modeling*, Vol. 6, Springer Science & Business Media, New York.

[10] Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M., 2016, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Hoboken, NJ.

[11] Anderson-Cook, C., Burr, T., Hamada, M. S., Ruggiero, C., and Thomas, E. V., 2015, "Design of Experiments and Data Analysis Challenges in Calibration for Forensics Applications," Chemom. Intell. Lab. Syst., **149**(B), pp. 107–117.

[12] Anderson-Cook, C. M., Hamada, M. S., and Burr, T., 2016, "The Impact of Response Measurement Error on the Analysis of Designed Experiments," Q. Reliab. Eng. Inter., **32**(7), pp. 2415–2433.

[13] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D., 2004, "Combining Field Data and Computer Simulations for Calibration and Prediction," SIAM J. Sci. Comput., **26**(2), pp. 448–466.

[14] Higdon, D., Gattiker, J., Williams, B., and Rightly, M., 2008, "Computer Model Calibration Using High-Dimensional Output," J. Am. Stat. Assoc., **103**(482), pp. 570–583.

[15] Lee, G., Kim, W., Oh, H., Youn, B., and Kim, N. H., 2019, "Review of Statistical Model Calibration and Validation–From the Perspective of Uncertainty Structures," Struct. Multidiscipl. Optim., **60**, pp. 1619–1644.

[16] Oberkampf, W. L., and Trucano, T. G., 2002, "Verification and Validation in Computational Fluid Dynamics," Prog. Aerosp. Sci., **38**, pp. 209–272.

[17] Roy, C. J., and Oberkampf, W. L., 2011, "A Comprehensive Framework for Verification, Validation, and Uncertainty Quantification in Scientific Computing," Comput. Methods. Appl. Mech. Eng., **200**, pp. 2131–2144.

[18] Ramsay, J. O., and Silverman, B. W., 2005, *Functional Data Analysis*, Springer, New York.

[19] Srivastava, A., and Klassen, E. P., 2016, *Functional and Shape Data Analysis*, Springer, New York.

[20] De Boor, C., and De Boor, C., 1978, *A Practical Guide to Splines*, Vol. 27, Springer-Verlag, New York.

[21] Daubechies, I., 1992, *10 Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

[22] Friedman, J. H., 1991, "Multivariate Adaptive Regression Splines," Ann. Stat., **19**(1), pp. 1–67.

[23] Efron, B., and Tibshirani, R., 1994, *An Introduction to the Bootstrap*, Chapman & Hall, New York.

[24] Ramsay, J. O., Graves, S., and Hooker, G., 2021, *fda: Functional Data Analysis*. R package version 5.5.0.

[25] Porter, R., Ruggiero, C., Harvey, N., Kelly, P., Tandon, L., Wilkerson, M., Kuhn, K., and Schwartz, D., 2016, MAMA User Guide v 1.2. Technical Report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

[26] Srivastava, A., Klassen, E., Joshi, S., and Jermyn, I., 2011, "Shape Analysis of Elastic Curves in Euclidean Spaces," IEEE Trans. Pattern Anal. Mach. Intell., **33**(7), pp. 1415–1428.

[27] Tucker, J. D., Wu, W., and Srivastava, A., 2013, "Generative Models for Functional Data Using Phase and Amplitude Separation," Comput. Stat. Data Anal., **61**(C), pp. 50–66.

[28] Marron, J. S., Ramsay, J., Sangalli, L., and Srivastava, A., 2015, "Functional Data Analysis of Amplitude and Phase Variation," Stat. Sci., **30**(4), pp. 468–484.

[29] Xu, L., Gotwalt, C., Hong, Y., King, C. B., and Meeker, W. Q., 2020, "Applications of the Fractional-Random-Weight Bootstrap," Am. Stat., **4**(74), pp. 345–358.