Annales Univ. Sci. Budapest., Sect. Comp. 58 (2025) 191–208

A REPRODUCING KERNEL FRAMEWORK FOR INTERPRETABLE PARAMETER ESTIMATION AND LEARNING IN DYNAMICAL SYSTEMS

Sándor Kovács and Gergő Tóth (Budapest, Hungary)

Communicated by László Szili

(Received 29 May 2025; accepted 3 August 2025)

Abstract. In this work, we investigate a suite of explainable machine learning models – including Kernel Ridge Regression, Gaussian Process Regression, Convolutional Neural Networks, and Multi-Layer Perceptrons – for parameter inference in systems of differential equations. By combining these models with explainability techniques such as SHapley Additive exPlanations, we extract explicit feature-to-parameter mappings, offering deeper insight into the inference process. Building on these insights, we propose lightweight, hand-engineered estimators that approximate parameter estimation tasks without requiring complex optimization. Additionally, we introduce a systematic methodology for dataset generation, incorporating time-series simulation and diverse feature extraction. Our results demonstrate that explainability-driven modeling can achieve accurate, interpretable, and computationally efficient parameter estimation, offering a new perspective on the integration of machine learning with domain-specific modeling.

1. Introduction

System identification, the process of building mathematical models from observed data, plays a fundamental role in understanding and controlling dynamical systems across disciplines such as engineering, biology, and epidemiology (cf. [4, 24]). Within this field, *parameter estimation* – the task of inferring the underlying parameters of differential equation systems (DE) – is critical for

Key words and phrases: RKHS methods, parameter estimation, ordinary differential equations.

2010 Mathematics Subject Classification: 46E22, 47E99.

https://doi.org/10.71352/ac.58.030825

First published online: 8 August 2025

ensuring that models accurately capture the essential dynamics of real-world phenomena.

Classical parameter estimation in DE models typically involves solving inverse problems, where observed trajectories are matched to model predictions through numerical optimization, sensitivity analysis, or Bayesian inference frameworks (cf. [27, 30, 7]). While these methods have achieved notable success, they often suffer from high computational cost, sensitivity to noise, and limited interpretability, particularly in complex or high-dimensional settings. Moreover, traditional techniques primarily focus on recovering best-fitting parameters, offering limited insight into the mechanisms by which features of the observed data inform parameter selection.

In contrast, this work aims to shift the focus from parameter recovery alone to *understanding and utilizing the parameter selection process.* Rather than solely fitting models to data, we seek to *extract explicit knowledge* from machine learning models about how different features of dynamic trajectories influence parameter inference.

To this end, we develop a framework that integrates explainable machine learning techniques – including *Kernel Ridge Regression* (KRR), *Gaussian Process Regression* (GPR) and comparative baselines such as *Convolutional Neural Networks* (CNNs) and *Multi-Layer Perceptrons* (MLPs) – with post-hoc explainability methods such as SHapley Additive exPlanations (SHAP).

In addition, we introduce a systematic methodology for dataset generation tailored to this dual objective. By simulating time-series data from systems of differential equations under controlled parameter variations and extracting diverse feature representations – including classical statistical descriptors and frequency-domain summaries such as spectrograms – we create rich experimental environments suited for both model training and interpretability analysis.

Through this combined focus on *explanation, knowledge extraction, and model compression*, we propose a framework that not only achieves accurate parameter estimation, but also promotes transparency, interpretability, and efficiency. This perspective opens new possibilities for integrating machine learning with domain science, where understanding underlying decision processes is valued alongside predictive performance.

2. Background

2.1. Reproducing kernel Hilbert spaces

Definition 2.1. Let $\mathcal{H}$ be a real Hilbert space of functions $f : \Omega \to \mathbb{R}$, where $\Omega \subseteq \mathbb{R}^d$ is non-empty. For each $y \in \Omega$, define the evaluation functional

$$L_y : \mathcal{H} \to \mathbb{R} \qquad \text{by} \qquad L_y(f) := f(y) \qquad \text{for all} \qquad f \in \mathcal{H}.$$

Then, $\mathcal{H}$ is called *reproducing kernel Hilbert space* (RKHS) if all evaluation functionals L_y are bounded, that is, $L_y \in \mathcal{H}^*$ for every $y \in \Omega$.

Unless stated otherwise, $\|\cdot\|$ denotes the standard Euclidean norm on $\mathbb{R}^d$. For the Hilbert space $\mathcal{H}$, we write $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ for the inner product, and $\|f\|_{\mathcal{H}} = = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ for the induced norm. It is well known that there exists a unique function $k : \Omega \times \Omega \to \mathbb{R}$ (cf. [3]), called the *reproducing kernel* of $\mathcal{H}$, satisfying

$$k(\cdot, y) \in \mathcal{H}, \quad \text{and} \quad f(y) = \langle f, k(\cdot, y)\rangle_{\mathcal{H}} \quad \text{for all } f \in \mathcal{H},\ y \in \Omega.$$

The kernel k is symmetric, $k(x, y) = k(y, x)$, and satisfies

$$k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle_{\mathcal{H}} \quad \text{for all } x, y \in \Omega.$$

Let $(f_n)_{n\in\mathbb{N}}$ be a sequence in $\mathcal{H}$. If (f_n) converges to $f \in \mathcal{H}$ in the norm of $\mathcal{H}$, meaning $\|f_n - f\|_{\mathcal{H}} \to 0$, then (f_n) converges pointwise to f on Ω; in other words, for every $y \in \Omega$, we have $f_n(y) \to f(y)$.

Definition 2.2. (Positive semi-definite and positive definite kernels.) A symmetric function $\varphi : \Omega \times \Omega \to \mathbb{R}$ is called a *positive semi-definite kernel* if for any $n \in \mathbb{N}$, any $x_1, \dots, x_n \in \Omega$, and any $c_1, \dots, c_n \in \mathbb{R}$, we have:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \varphi(x_i, x_j) \geq 0.$$

It is called *positive definite* if the inequality is strict whenever the c_i are not all zero.

A fundamental result, the *Moore–Aronszajn theorem* (cf. [3]), states that for every symmetric, positive semi-definite function $\varphi : \Omega \times \Omega \to \mathbb{R}$, there exists a unique RKHS $\mathcal{H}_\varphi$ of functions $f : \Omega \to \mathbb{R}$ in which φ serves as the reproducing kernel.

In particular, commonly used kernels in statistical learning are positive semi-definite and hence correspond to RKHSs via the Moore–Aronszajn theorem. Two widely used examples are the Gaussian radial basis function (RBF) kernel k_σ (cf. [8]) and the Matérn kernel $k_{\nu,h}$ (cf. [26, 33]).

Example 2.1. (Gaussian RBF Kernel.) Let $\sigma > 0$. The Gaussian RBF kernel is defined by

$$k_\sigma(x, x') := \exp\left(-\frac{\|x - x'\|^2}{\sigma^2}\right) \qquad (x, x' \in \Omega).$$

Example 2.2. (Matérn Kernel.) Let $\nu > 0$ and $h > 0$. The Matérn kernel is defined by

$$k_{\nu,h}(x, x') := \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\|x - x'\|}{h}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{\|x - x'\|}{h}\right) \qquad (x, x' \in \Omega),$$

where K_ν denotes the modified Bessel function of the second kind and order ν (cf. [21]).

These kernels not only satisfy the conditions of the Moore–Aronszajn theorem, but also encode smoothness and scale information of the functions in the corresponding RKHS, making them essential tools in Gaussian process modeling and kernel methods.

For a comprehensive treatment of RKHS theory and its connections to learning and approximation, we refer the reader to [3, 6, 38].

3. Connections between Gaussian process regression and Kernel Ridge Regression

We assume a complete probability space $(\Omega, \mathcal{A}, \mathbb{P})$, where $\mathcal{A}$ is a σ-algebra on Ω, and $\mathbb{P}$ is a probability measure. A *random function* (or *stochastic process*) indexed by $\mathcal{I} \subset \mathbb{R}^d$ is a collection of real-valued measurable functions

$$f := \{f(x) : x \in \mathcal{I}\}, \quad \text{with} \quad f(x) : \Omega \to \mathbb{R}.$$

For each fixed $x \in \mathcal{I}$, the map $f(x)$ is a real-valued random variable, and we view f as a family of such variables indexed by the index set $\mathcal{I}$.

A vector-valued random variable $(Z_1, \dots, Z_n)^\top \in \mathbb{R}^n$ is said to follow a *multivariate normal distribution* if every linear combination $\sum_{i=1}^n a_i Z_i$, with $a_i \in \mathbb{R}$, is normally distributed. In this case, we write

$$(Z_1, \dots, Z_n)^\top \sim \mathcal{N}(\mu, \Sigma),$$

where $\mu \in \mathbb{R}^n$ is the mean vector with entries $\mu_i := \mathbb{E}[Z_i]$, and $\Sigma \in \mathbb{R}^{n\times n}$ is the covariance matrix with entries

$$\Sigma_{ij} := \operatorname{Cov}(Z_i, Z_j) = \mathbb{E}[(Z_i - \mu_i)(Z_j - \mu_j)].$$

A stochastic process $f = \{f(x) : x \in \mathcal{I}\}$ is called a *Gaussian process* if for any finite set $\{x_1, \dots, x_n\} \subset \mathcal{I}$, the random vector

$$(f(x_1), \dots, f(x_n))^\top \in \mathbb{R}^n$$

is multivariate normally distributed. That is,

$$(f(x_1), \dots, f(x_n))^\top \sim \mathcal{N}(m_X, \Phi_{XX}),$$

where

$$m_X := (m(x_1), \dots, m(x_n))^\top, \qquad \Phi_{XX} := (\Phi(x_i, x_j))_{i,j=1}^n,$$

and the *mean function* $m : \mathcal{I} \to \mathbb{R}$ and *covariance function* $\Phi : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ are defined as

$$\begin{aligned} m(x) &:= \mathbb{E}[f(x)], \\ \Phi(x, x') &:= \operatorname{Cov}(f(x), f(x')) = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \end{aligned}$$

In this case, we write

$$f \sim \mathcal{GP}(m, \Phi),$$

to denote that f is a Gaussian process with mean function m and covariance function Φ.

Conversely, any pair (m, Φ), where Φ is a symmetric positive definite kernel, uniquely determines a Gaussian process $f \sim \mathcal{GP}(m, \Phi)$ (cf. [14]), establishing a one-to-one correspondence between Gaussian processes and such pairs. In this context, the function Φ is a reproducing kernel, and by the Moore–Aronszajn theorem, it defines a unique RKHS associated with the process.

In this setting, given a dataset $X = (x_1, \ldots, x_n) \subset \mathcal{I}$ with observed outputs $Y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$, the posterior distribution of the process $f \sim \mathcal{GP}(m, \Phi)$ conditioned on the data remains a Gaussian process,

$$f \mid Y \sim \mathcal{GP}(\bar{m}, \bar{\Phi}),$$

with posterior mean and covariance functions

$$\begin{aligned}\bar{m}(x) &:= m(x) + \Phi_{xX}(\Phi_{XX} + \sigma^2 I_n)^{-1}(Y - m_X) \quad (x \in \mathcal{I}),\\ \bar{\Phi}(x, x') &:= \Phi(x, x') - \Phi_{xX}(\Phi_{XX} + \sigma^2 I_n)^{-1}\Phi_{Xx'} \quad (x, x' \in \mathcal{I}),\end{aligned}$$

where $\Phi_{xX} := (\Phi(x, x_1), \ldots, \Phi(x, x_n)) \in \mathbb{R}^n$, $\Phi_{Xx'} := \Phi_{xX}^\top$, and $\Phi_{XX} \in \mathbb{R}^{n\times n}$ is the Gram matrix. The vector $m_X \in \mathbb{R}^n$ contains the prior mean values at the training inputs. The posterior mean $\bar{m}$ gives the predicted average output at a new point x, while $\bar{\Phi}$ quantifies the updated uncertainty (cf. [39]).

3.1. Kernel Ridge Regression

Let $\psi : \Omega \times \Omega \to \mathbb{R}$ be a symmetric, positive definite kernel, and let $\mathcal{H}_\psi$ denote the corresponding RKHS. Using the same dataset (X, Y), kernel ridge regression seeks a function $\hat{f} \in \mathcal{H}_\psi$ minimizing the regularized empirical risk

$$\hat{f} := \operatorname*{arg\,min}_{f\in\mathcal{H}_\psi} \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}_\psi}^2,$$

where $\lambda > 0$ is a regularization parameter (cf. [35, 37, 32]).

By the representer theorem (cf. [31]), the solution $\hat{f}$ admits the explicit form

$$\hat{f}(x) = \psi_{xX}(\psi_{XX} + n\lambda I_n)^{-1}Y,$$

where $\psi_{XX} \in \mathbb{R}^{n\times n}$ is the kernel matrix with entries $\psi(x_i, x_j)$ and $\psi_{xX} \in \mathbb{R}^{1\times n}$ is the vector $[\psi(x, x_1), \ldots, \psi(x, x_n)]$.

There is a deep connection between GPR and KRR: the posterior mean function $\bar{m}$ of GPR coincides with the KRR solution $\hat{f}$ when the regularization parameter λ and the noise variance σ^2 are related by $\sigma^2 = n\lambda$ (cf. [21]).

4. Related work

Parameter estimation for DE models has long been a fundamental topic in the modeling of dynamic systems. Traditional techniques, such as least-squares fitting, adjoint sensitivity analysis, and maximum likelihood estimation (cf. [5, 9, 28, 29]), have been widely used to infer system parameters by minimizing the discrepancy between observed trajectories and model simulations. However, these classical approaches often encounter difficulties when applied to systems with nonlinear dynamics, noisy measurements, or incomplete observations. Issues such as computational cost, sensitivity to initial conditions, and the risk of convergence to local minima become particularly pronounced in complex, high-dimensional settings.

The integration of machine learning into the study of dynamical systems has opened new avenues for addressing some of these challenges. Neural networks, including recurrent architectures and physics-informed neural networks (PINNs), have been proposed for directly learning system behavior from time-series data (cf. [27]). Kernel-based methods offer flexible and probabilistic modeling frameworks that are well-suited for small-sample, high-noise environments (cf. [16]). Despite the success of these methods in improving predictive accuracy, most works remain focused on optimizing performance, with relatively little attention given to understanding the decision-making process behind parameter selection.

Among the theoretical frameworks supporting these modern approaches, the theory of RKHS provides a principled foundation for function approximation and regularization in supervised learning problems (cf. [16]). RKHS-based techniques offer strong theoretical guarantees on generalization and stability while enabling flexible nonparametric representations of complex mappings. These properties make kernel methods particularly attractive for parameter inference tasks, especially when system knowledge is limited or data quality is low.

Alongside improvements in modeling techniques, the need for interpretability in machine learning has become increasingly critical. Especially in scientific domains, understanding *why* a model makes certain predictions is as important as the predictions themselves. Methods such as SHAP (cf. [25, 10]) provide a unified, theoretically grounded approach to feature attribution, offering insights into the internal reasoning of complex models.

5. Methodology

We develop a general framework for interpretable parameter estimation from dynamical system trajectories, applicable across a variety of DE models. Our approach combines synthetic dataset generation, feature extraction, predictive model training, and explainability analysis, and is designed to be model-agnostic and adaptable to various systems.

We consider ordinary differential equations (ODEs) of the form

$$\dot{S}(t) = F(S(t), \theta), \qquad S(0) = S_0 \qquad (t \in \mathbb{R}_+, S_0 \in \mathbb{R}^d), \tag{5.1}$$

where $S(t) \in \mathbb{R}^d$ represents the system state at time t, $\theta \in \Theta \subset \mathbb{R}^p$ is a vector of parameters, and $F : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ defines the system dynamics.

To ensure that the initial value problem (5.1) admits a unique solution, we assume that the function F is continuous and satisfies a Lipschitz condition with respect to the state variable S, uniformly in θ. That is, there exists a constant $L > 0$ such that for all $S_1, S_2 \in \mathbb{R}^d$ and all $\theta \in \Theta$,

$$\|F(S_1, \theta) - F(S_2, \theta)\| \leq L\|S_1 - S_2\|,$$

where $\|\cdot\|$ denotes the standard Euclidean norm on $\mathbb{R}^d$. Under these assumptions, the Picard-–Lindelöf theorem [11] guarantees the existence of a unique local solution $S : [0, T] \to \mathbb{R}^d$ for some $T > 0$.

Although the system may involve multiple parameters, we focus on the estimation of a single parameter of interest $\theta \in \mathbb{R}$. The remaining parameters are either fixed or jointly sampled but not individually inferred.

Dataset generation

Synthetic datasets are constructed by sampling parameter vectors $\theta \in \Theta \subset \subset \mathbb{R}^p$ from a bounded domain that reflects prior knowledge or physical constraints. Each sampled θ defines an instance of the dynamical system (5.1), which is numerically integrated over a fixed time interval $[0, T]$. In our experiments, we employ an explicit Runge–Kutta method of order 5(4) with adaptive step size control, specifically the Dormand–Prince scheme (cf. [13, 1]).

Let $\hat{s}_\theta : [0, T] \to \mathbb{R}^d$ denote the numerically integrated trajectory corresponding to parameter θ. To account for measurement noise, we optionally add Gaussian perturbations:

$$\tilde{s}_\theta(t) := \hat{s}_\theta(t) + \varepsilon(t), \quad \varepsilon(t) \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_d),$$

where $\varepsilon(t) \in \mathbb{R}^d$ is multivariate Gaussian noise with zero mean and isotropic covariance $\sigma^2 I_d \in \mathbb{R}^{d \times d}$.

Each noisy trajectory $\tilde{s}_\theta(t)$ is transformed into a fixed-dimensional feature vector $x \in \mathbb{R}^q$, paired with its associated target parameter $y = \theta$. The final dataset consists of input-output pairs $(x_i, y_i) \in \mathbb{R}^q \times \mathbb{R}$.

Learning and explainability

The objective is to learn a predictive function

$$f^* : \mathbb{R}^q \to \mathbb{R}, \quad f^*(x) \approx \theta^*,$$

mapping extracted features to the corresponding target parameter value. We consider multiple machine learning models to approximate f^*, including kernel-based methods and neural networks.

KRR (cf. [31]) is employed with positive definite kernels to produce regularized function estimators within a RKHS. GPR (cf. [39]) offers a Bayesian nonparametric framework, yielding both point predictions and closed-form posterior uncertainty estimates. As neural network baselines, we include MLPs (cf. [20]) for general-purpose regression and CNNs (cf. [23]), especially for tasks involving spectrogram-based representations (cf. [40]). Specific architectures, kernel functions, and hyperparameter configurations are detailed in the experimental section.

To interpret the trained models and assess the role of different input features, we incorporate post-hoc explainability tools. For neural networks and other black-box regressors, we apply SHAP (cf. [25]), a game-theoretic method that attributes prediction differences to feature contributions. SHAP values are computed by approximating the Shapley value for each input feature, offering a principled measure of local importance. These per-instance explanations are further aggregated across the dataset to obtain global feature relevance.

In summary, our framework integrates simulation-based dataset generation, feature-informed learning, interpretable modeling, and post-hoc explanation to deliver both accurate and transparent parameter inference from observed system trajectories.

6. Experimental setup

6.1. Model description

The experiments are based on a two-dimensional nonlinear dynamical system, initially proposed for the study of bifurcation phenomena in chemical and biological processes (cf. [22]). The system dynamics are described by the coupled ordinary differential equations

$$\dot{S}_1 = \lambda - aS_1S_2 + \beta S_2 - \delta_1 S_1, \tag{6.1}$$

$$\dot{S}_2 = aS_1S_2 - \beta S_2 - \delta_2 S_2, \tag{6.2}$$

where $S_1(t)$ and $S_2(t)$ denote the system states at time t, and $\lambda, a, \beta, \delta_1, \delta_2$ are positive parameters controlling inflow, nonlinear interactions, recovery, and decay processes, respectively.

6.2. Data generation

A synthetic dataset was constructed by numerically solving the Scheurle–Seydel model for a range of parameter values. A total of $N = 500$ independent trajectories were simulated over the time interval $[0, 10]$, evaluated at $T = 501$ uniformly spaced time points. All simulations were initialized from the fixed initial condition:

$$[S_1(0), S_2(0)] := [1.0, 1.0].$$

The diffusion parameters were fixed as $\delta_1 := 0.01$and $\delta_2 := 0.01$, while the remaining model parameters were sampled independently from uniform distributions:

$$\lambda \sim \mathcal{U}(0.5, 100.0), \qquad a \sim \mathcal{U}(0.01, 0.05), \qquad \beta \sim \mathcal{U}(0.05, 0.15).$$

Let $\theta := (\lambda, a, \beta) \in \Theta \subset \mathbb{R}^3$ denote a sampled parameter vector. For each θ, the associated ODE system was numerically integrated using the Dormand–Prince method—an explicit Runge–Kutta scheme of order 5(4) with adaptive step size control (cf. [13, 1]).

Let $s_{\theta,1}(t)$ and $s_{\theta,2}(t)$ denote the two components of the numerically computed solution trajectory $s_\theta(t) \in \mathbb{R}^2$ corresponding to the parameter vector θ. These noiseless signals serve as the foundation for all subsequent feature extraction and spectral analysis.

To simulate observational noise, additive Gaussian noise with isotropic variance $\sigma^2 := 1$ was independently applied to each component:

$$\tilde{s}_{\theta,i}(t) := s_{\theta,i}(t) + \epsilon_i(t), \qquad \epsilon_i(t) \sim \mathcal{N}(0, \sigma^2), \quad i = 1, 2.$$

All simulations were implemented in `Python` using `NumPy` (cf. [18]) for numerical computation and `SciPy` (cf. [36]) for ODE integration. A fixed random seed ensured reproducibility.

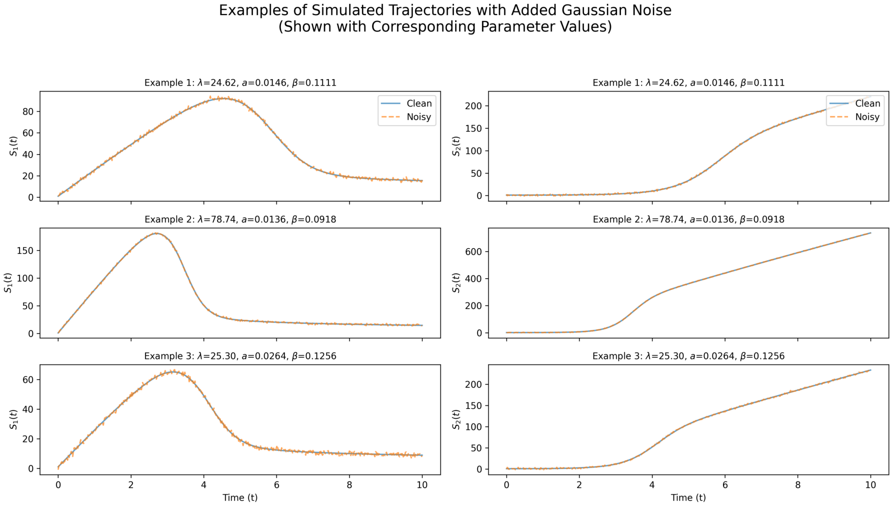

Figure 1. Simulated trajectories of $S_1(t)$ and $S_2(t)$ with Gaussian noise ($\sigma = 1$).

6.3. Feature extraction

To facilitate downstream learning tasks, we extracted a structured collection of statistical and spectral features from each simulated trajectory. These features were derived from the two components $s_{\theta,1}(t)$ and $s_{\theta,2}(t)$ of the numerically computed state trajectories.

Statistical features. For each trajectory, we computed descriptive statistics separately for $s_{\theta,1}$ and $s_{\theta,2}$, including the mean, standard deviation, minimum, and maximum values. These scalar features provide a compact summary of the distributional properties of the system's behavior over time. Any undefined or infinite values encountered during computation were replaced with zeros to maintain numerical stability.

Frequency-domain features. To characterize periodic behavior, we applied the Fast Fourier Transform (FFT) to both state variables. The DC component (zero-frequency term) was excluded, and the magnitudes of the first 20 non-DC Fourier coefficients were retained as features. The choice of 20 coefficients was treated as a tunable hyperparameter; increasing this number did not yield significant improvements in prediction performance based on empirical validation.

Spectrogram features. To capture joint time–frequency structure, we computed spectrograms for each trajectory using the short-time Fourier transform (STFT). Let $M_1 \in \mathbb{R}^{F\times T}$denote the spectrogram of $s_{\theta,1}$, where $F = 33$ is the number of frequency bins and $T = 15$ is the number of time windows. These values were determined by applying an FFT window size of 64 and a hop length of 32 to the 501-point time series.

To compress dynamic range and enhance low-magnitude patterns, a logarithmic transformation was applied:

$$\widetilde{M}_1(i,j) := \log(M_1(i,j) + \epsilon), \quad \epsilon > 0.$$

We then standardized the transformed matrix:

$$\begin{aligned}
M_1^{\mathrm{std}}(i,j) &:= \frac{\widetilde{M}_1(i,j) - \mu_1}{\sigma_1}, \\
\mu_1 &:= \frac{1}{FT}\sum_{i=1}^{F}\sum_{j=1}^{T}\widetilde{M}_1(i,j), \\
\sigma_1^2 &:= \frac{1}{FT}\sum_{i=1}^{F}\sum_{j=1}^{T}(\widetilde{M}_1(i,j) - \mu_1)^2.
\end{aligned}$$

The same process was applied to $s_{\theta,2}$ to obtain $M_2^{\mathrm{std}} \in \mathbb{R}^{F\times T}$. The two standardized spectrograms were then vertically concatenated to form a matrix $M \in \mathbb{R}^{2F\times T}$, which served as the input representation for CNN models. The normalization step ensures comparability across samples and promotes numerical stability during training.

All feature extraction steps were implemented in `Python` using `NumPy`, `SciPy`, and `Matplotlib`. Sampling frequencies were inferred from the discretized time grids used during numerical integration.

6.4. Model training and evaluation setup

We trained and evaluated four types of models for the parameter estimation task: KRR (cf. [31]), GPR (cf. [39]), MLPs (cf. [20]), and CNNs (cf. [23]). For KRR, we tested multiple kernels, including RBF, polynomial (degree 3), and Laplacian kernels. The regularization parameter λ was set to either 0.1 or 1.0, and the RBF/Laplacian kernel bandwidth γ was fixed at 0.1. For GPR, we used RBF and Matérn covariance kernels with smoothness parameter $\nu := 1.5$, and set the regularization parameter $\lambda := 10^{-5}$, with 10 optimizer restarts for marginal likelihood maximization.

The MLP models consisted of two fully connected hidden layers with 64 and 32 units respectively, ReLU (Rectified Linear Unit) activations, and were trained using a learning rate of 0.001, a batch size of 32, and for 100 epochs. CNNs were trained directly on spectrogram images generated from the trajectories, using three convolutional layers with 16, 32, and 64 filters (each with 3×3 kernels), followed by 2×2 max-pooling and a dropout layer with rate 0.5. The CNNs were trained with a learning rate of 0.001, batch size of 16, and for 50 epochs.

Each model was trained to predict a single target parameter (λ, a, or β) from the extracted features and evaluated on a held-out test set using three metrics: mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2) (cf. [19, 17]). Hyperparameters such as regularization strength (KRR, GPR), learning rates, and batch sizes (MLP, CNN) were selected based on standard heuristics and validated through preliminary experiments.

To assess interpretability, we applied SHAP (cf. [25]). For kernel-based models, we used RKHS-structured SHAP values when feasible (cf. [10]), while for neural models we employed model-agnostic SHAP approximations. Feature importance rankings were compared across models to assess the consistency and reliability of the extracted patterns.

7. Results and discussion

Our work can be situated within the broader family of approaches leveraging RKHS theory for modeling and inference tasks involving dynamical systems.

Geng [15] demonstrated the effectiveness of RKHS constructions for directly solving complex nonlinear differential equations, while González et al. (cf. [16]) extended RKHS-based formulations to the problem of parameter estimation in noisy ODE systems. More broadly, Steinke et al. (cf. [34]) presented a unifying view of kernel methods, highlighting the deep connections between positive definite kernels, regularization operators, and differential equations, thus emphasizing the natural role of RKHS methods in structured dynamical modeling tasks.

In this context, we designed a comparative evaluation involving four modeling approaches: KRR, GPR, MLP, and CNN. Models were trained on features derived from statistical summaries, Fourier-transform representations, and spectrograms, aiming to predict key system parameters. The evaluation metrics included mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2).

The empirical results, summarized in Table 1, showed that kernel-based methods (KRR and GPR) consistently outperformed neural models (MLP and CNN) across all feature types and target parameters, even without extensive hyperparameter tuning. This observation suggests that the structure captured by kernels is well aligned with the underlying system dynamics, providing a strong inductive bias for the task at hand.

Target	**Model (Kernel/Arch) – Feat**	**MAE/RMSE/R^2**
λ	KRR (Laplacian) – Stat	0.0079 / 0.0105 / 0.8734
λ	**GPR (RBF) – Stat**	**0.0007 / 0.0012 / 1.0000**
λ	GPR (Matérn) – Stat	0.0073 / 0.0283 / 1.0000
λ	MLP (64–32) – Stat	1.3686 / 1.8003 / 0.9953
λ	MLP (64–32) – FFT	1.7632 / 2.4863 / 0.9911
λ	CNN (3C, DO 0.5) – Spec	20.2090 / 23.8824 / 0.1781
a	KRR (Laplacian) – Stat	0.0016 / 0.0026 / 0.9516
a	**GPR (Matérn) – Stat**	**0.0007 / 0.0015 / 0.9836**
a	GPR (RBF) – Stat	0.0008 / 0.0014 / 0.9860
a	MLP (64–32) – Stat	0.0016 / 0.0023 / 0.9624
a	MLP (64–32) – FFT	0.0020 / 0.0028 / 0.9433
a	CNN (3C, DO 0.5) – Spec	0.0067 / 0.0085 / 0.4817
β	KRR (Laplacian) – Stat	0.0200 / 0.0239 / 0.3477
β	**GPR (RBF) – Stat**	**0.0010 / 0.0025 / 0.9927**
β	GPR (Matérn) – Stat	0.0014 / 0.0022 / 0.9945
β	MLP (64–32) – Stat	0.0039 / 0.0053 / 0.9677
β	MLP (64–32) – FFT	0.0100 / 0.0131 / 0.8052
β	CNN (3C, DO 0.5) – Spec	0.0288 / 0.0336 / −0.2874

Table 1. Performance on parameter estimation. Format: ”Model (Kernel/Arch) – Feature type”.

Although additional tuning of MLP and CNN models could potentially narrow the performance gap, this work focused on a systematic comparison using standard configurations. Hence, the superior results of KRR and GPR models support the relevance of RKHS-inspired methodologies as a theoretically grounded and practically effective framework for parameter inference in nonlinear dynamical systems.

Best performing models

Across all three target parameters (λ, a, and β), the GPR model with an RBF kernel, trained on statistical features, achieved the best overall performance. For λ, the GPR model with an RBF kernel reached a MAE of 0.0007, an RMSE of 0.0012, and an R^2score of 1.0000. For a, the best result was obtained by the GPR model with a Matérn kernel ($\nu = 1.5$), yielding a MAE of 0.0007, RMSE of 0.0015, and $R^2$of 0.9836. For β, GPR with an RBF kernel again performed best, with a MAE of 0.0010, RMSE of 0.0025, and R^2 of 0.9927.

Among the evaluated models, KRR also showed strong performance-particularly the variant using the Laplacian kernel for λ-though it consistently trailed GPR in predictive accuracy. Neural network-based models, including MLPs and CNNs, were able to learn meaningful patterns, but their performance was significantly lower, especially when trained on frequency-domain features or spectrogram images.

Feature importance analysis for λ

To further understand the learned models, we performed SHAP-based feature attribution analysis. Figures 2, 3, and 4 illustrate the feature importance for KRR, GPR, and MLP models, respectively.

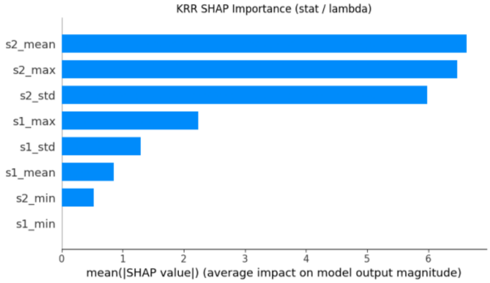

Figure 2. SHAP bar plot for KRR model predicting λ.

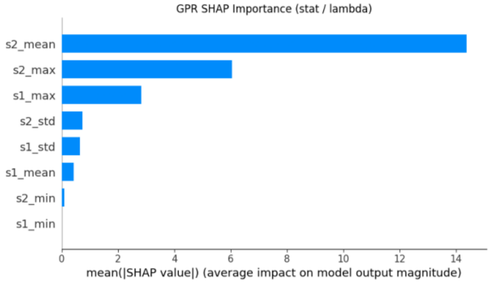

Figure 3. SHAP bar plot for GPR model predicting λ.

Across all three models, the two most influential features remained identical, with only the third-ranked feature differing slightly. Despite this minor variation, no substantial differences could be observed in the overall SHAP explanations, suggesting a strong consistency in the learned feature importance

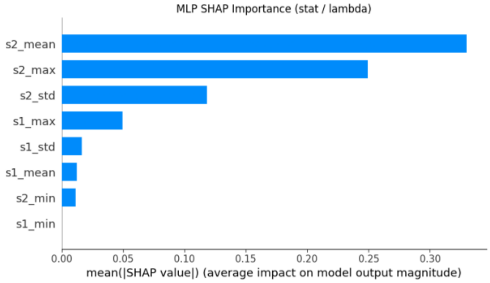

Figure 4. SHAP bar plot for MLP model predicting λ.

patterns. These results reinforce that the extracted statistical features robustly capture the essential dynamics governing the parameter λ across different modeling approaches.

Moreover, these results highlight that simple statistical summaries can be highly informative and sufficient for parameter inference tasks, outperforming models trained on more complex feature types such as frequency-domain or spectrogram representations.

Comparison of feature types

The empirical evaluation highlights several key insights. Statistical features consistently outperformed Fourier-transform and spectrogram-based representations across all model types and target parameters. This finding emphasizes the importance of carefully engineered features that summarize the system dynamics rather than relying solely on raw or transformed signal representations.

Overall, the results demonstrate that combining simulation-based data generation, feature extraction, kernel-based learning, and model interpretability yields a powerful and transparent framework for inferring dynamic system parameters.

8. Conclusion and future work

We proposed a feature-based learning framework for parameter estimation in differential equation systems, combining predictive modeling with post-hoc explainability. Experiments on synthetic ODE datasets evaluated multiple methods, including kernel-based approaches, neural networks, and Gaussian processes. Kernel Ridge Regression and Gaussian Process Regression consistently achieved higher predictive accuracy and robustness compared to neural network baselines.

Notably, SHAP-based feature attribution across KRR, GPR, and MLP models revealed highly consistent results, with the top three features remaining the same across methods, differing only in order. This robustness suggests that reliable insights into parameter–feature relationships can be extracted even when model architectures vary.

While the proposed framework demonstrated strong empirical performance, further theoretical analysis is needed to formalize the connection between system dynamics and feature relevance. Additionally, extending validation beyond synthetic datasets remains an important step toward establishing broader practical applicability.

Building on these findings, future work will extend the methodology to stable multi-parameter estimation, develop data-driven techniques for identifying informative parameter ranges, and design adaptive dataset generation strategies to improve training efficiency and interpretability.

To enable a fair comparison, we also should scale up the dataset and employ computationally optimized models and techniques, as current kernel-based methods do not scale efficiently, and evaluating neural architectures like MLPs and CNNs requires sufficient data volume to fully leverage their capacity.

References

[1] **Alexander, R.,** Solving ordinary differential equations I: Nonstiff problems (E. Hairer, S.P. Norsett, and G. Wanner), *Siam Review*, **32** (1990), 485.
https://doi.org/10.1137/1032091

[2] **Allen, J.B. and L.R. Rabiner,** Short-time spectral analysis, synthesis, and modification by discrete Fourier transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **25** (1977), 235–238.
https://doi.org/10.1109/TASSP.1977.1162950

[3] **Aronszajn, N.,** Theory of reproducing kernels, *Transactions of the American Mathematical Society*, **68** (1950), 337–404.
https://doi.org/10.1090/S0002-9947-1950-0051437-7

[4] **Åström, K.J. and P. Eykhoff,** System identification – a survey, *Automatica*, **7** (1971), 123–162.
https://doi.org/10.1016/0005-1098(71)90059-8

[5] **Bellman, R. and K.J. Åström,** On structural identifiability, *Mathematical biosciences*, **7** (1970), 329–339.
https://doi.org/10.1016/0025-5564(70)90132-X

[6] **Berlinet, A. and C. Thomas-Agnan,** *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer Science & Business Media, 2011.
https://doi.org/10.1007/978-1-4419-9096-9

[7] **Brunton, S.L, J.L. Proctor and J.N. Kutz,** Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. U.S.A.*, **113** (2016), 3932–3937.
https://doi.org/10.1073/pnas.1517384113

[8] **Buhmann, M.D.,** Radial basis functions, *Acta Numerica*, **9** (2000), 1–38.
https://doi.org/10.1017/S0962492900000015

[9] **Cao, Y., S. Li, L. Petzold and R. Serban,** Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution, *SIAM Journal on Scientific Computing*, **24(3)** (2003), 1076–1089.
https://doi.org/10.1137/S1064827501380630

[10] **Chau, S.L., R. Hu, J. Gonzalez and D. Sejdinovic,** RKHS-SHAP: Shapley values for kernel methods, *Advances in Neural Information Processing Systems*, **35** (2022), 13050–13063.

[11] **Coddington, E.A. and Levinson, N.,** *Theory of Ordinary Differential Equations*, McGraw-Hill, 1955.

[12] **Cooley, J.W. and J.W. Tukey,** An algorithm for the machine calculation of complex Fourier series, *Mathematics of Computation*, **19** (1965), 297–301.
https://doi.org/10.2307/2003354

[13] **Dormand, J.R. and P.J. Prince,** A family of embedded Runge–Kutta formulae, *Journal of Computational and Applied Mathematics*, **6(1)** (1980), 19–26.
https://doi.org/10.1016/0771-050X(80)90013-3

[14] **Dudley, R.M.,** *Real Analysis and Probability*, Chapman and Hall/CRC, 2018.
https://doi.org/10.1201/9781351076197

[15] **Geng, F.,** A new reproducing kernel Hilbert space method for solving nonlinear fourth-order boundary value problems, *Applied Mathematics and Computation*, **213** (2009), 163–169
https://doi.org/10.1016/j.amc.2009.02.053

[16] **González, J., I. Vujačić and E. Wit,** Reproducing kernel Hilbert space based estimation of systems of ordinary differential equations, *Pattern Recognition Letters*, **45** (2014), 26–32.
https://doi.org/10.1016/j.patrec.2014.02.019

[17] **Goodfellow, I., Y. Bengio and A. Courville,** *Deep Learning*, MIT Press, 2016.

[18] **Harris, C.R., K.J. Millman, S.J. van Der Walt et al.,** Array programming with NumPy, *Nature*, **585** (2020), 357–362.
https://doi.org/10.1038/s41586-020-2649-2

[19] **Hastie, T., R. Tibshirani, J.H. Friedman,** *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
https://doi.org/10.1007/978-0-387-84858-7

[20] **Hornik, K., M. Stinchcombe and H. White,** Multilayer feedforward networks are universal approximators, *Neural networks*, **2** (1989), 359–366.
https://doi.org/10.1016/0893-6080(89)90020-8

[21] **Kanagawa, M., P. Hennig, D. Sejdinovic and B.K. Sriperumbudur,** Gaussian processes and kernel methods: A review on connections and equivalences, https://arxiv.org/abs/1807.02582v1
https://doi.org/10.48550/arXiv.1807.02582

[22] **Kovács, S.,** Delay in decision making causes oscillation, *Nonlinearity*, **17** (2004), 2267.
https://doi.org/10.1088/0951-7715/17/6/013

[23] **LeCun, Y., L. Bottou, Y. Bengio and P. Haffner,** Gradient-based learning applied to document recognition, in: *Proceedings of the IEEE*, **86** (1998), 2278–2324.
https://doi.org/10.1109/5.726791

[24] **Lennart, L.,** System Identification: Theory for the User, *PTR Prentice Hall, Upper Saddle River, NJ*, **28** (1999), 540.

[25] **Lundberg, S.M. and S.I. Lee,** A unified approach to interpreting model predictions, *Advances in neural information processing systems*, **30** (2017).

[26] **Matérn, B.,** Spatial variation, *Meddelanden från Statens Skogsforskningsinstitut*, **49**(5) (1960), 1–144.

[27] **Raissi, M., P. Perdikaris, G.E. Karniadakis and E. George,** Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics*, **378** (2019), 686–707.
https://doi.org/10.1016/j.jcp.2018.10.045

[28] **Raue, A., M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, et al.,** Lessons learned from quantitative dynamical modeling in systems biology, *PloS ONE*, **8(9)** (2013), e74335.
https://doi.org/10.1371/journal.pone.0074335

[29] **Rothenberg, T.J.,** Identification in parametric models, *Econometrica: Journal of the Econometric Society*, **39(3)** (1971), 577–591.
https://doi.org/10.2307/1913267

[30] **Rudy, S.H., S.L. Brunton, J.L. Proctor and J.N. Kutz,** Data-driven discovery of partial differential equations, *Science Advances*, **3(4)** (2017), e1602614.
https://doi.org/10.1126/sciadv.1602614

[31] **Schölkopf, B., R. Herbrich and A.J. Smola,** A generalized representer theorem, In: Helmbold, D., Williamson, B. (eds) *Computational Learning Theory, COLT*, Lecture Notes in Computer Science, vol. 2111, 2001.
https://doi.org/10.1007/3-540-44581-1_27

[32] **B. Schölkopf and A.J. Smola,** *Learning with kernels*, MIT Press, 2001.
https://doi.org/10.7551/mitpress/4175.001.0001

[33] **Stein, M.L.,** *Interpolation of Patial Data: Some Theory for Kriging*, Springer Series in Statistics, Springer-Verlag, New York, 1999.
https://doi.org/10.1007/978-1-4612-1494-6

[34] **Steinke, F, and B. Schölkopf,** Kernels, regularization and differential equations, *Pattern Recognition*, **41** (2008), 3271–3286.
https://doi.org/10.1016/j.patcog.2008.06.011

[35] **Vapnik, V.N.,** *Statistical Learning Theory*, Wiley-Interscience, 1998.

[36] **Virtanen, P., R. Gommers, T.E. Oliphant, et al.,** SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nature Methods*, **17** (2020), 261–272.
https://doi.org/10.1038/s41592-019-0686-2

[37] **Wahba, G.,** Spline models for observational data, CBMS-NSF Regional Conferences series in applied mathematics, SIAM, 1990.
https://doi.org/10.1137/1.9781611970128

[38] **Wendland, H.,** *Scattered Data Approximation*, Cambridge University Press, 2004.
https://doi.org/10.1017/CBO9780511617539

[39] **Williams, C.K. and C.E. Rasmussen,** *Gaussian Processes for Machine Learning*, MIT Press Cambridge, MA, 2006.
https://doi.org/10.7551/mitpress/3206.001.0001

[40] **Xie, L., C. Li, X. Zhang, S. Zhai, Y. Fang, Q. Shen and Z. Wu,** TRLS: A time series representation learning framework via spectrogram for medical signal processing,
https://arxiv.org/abs/2401.05431v1 (2024),
https://doi.org/10.48550/arXiv.2401.05431

Sándor Kovács
https://orcid.org/0000-0001-7051-5075
ELTE Eötvös Loránd University
Department of Numerical Analysis
Budapest
Hungary
alex@ludens.elte.hu

Gergő Tóth
https://orcid.org/0009-0000-2899-1975
ELTE Eötvös Loránd University
Doctoral School of Informatics
Budapest
Hungary
q253k0@inf.elte.hu