

Bayesian learning with Gaussian processes for low-dimensional representations of time-dependent nonlinear systems

Shane A. McQuarrie¹, Anirban Chaudhuri², Karen E. Willcox², and Mengwu Guo^{*3}

¹*Scientific Machine Learning, Sandia National Laboratories*

²*Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin*

³*Centre for Mathematical Sciences, Lund University*

Abstract

This work presents a data-driven method for learning low-dimensional time-dependent physics-based surrogate models whose predictions are endowed with uncertainty estimates. We use the operator inference approach to model reduction that poses the problem of learning low-dimensional model terms as a regression of state space data and corresponding time derivatives by minimizing the residual of reduced system equations. Standard operator inference models perform well with accurate training data that are dense in time, but producing stable and accurate models when the state data are noisy and/or sparse in time remains a challenge. Another challenge is the lack of uncertainty estimation for the predictions from the operator inference models. Our approach addresses these challenges by incorporating Gaussian process surrogates into the operator inference framework to (1) probabilistically describe uncertainties in the state predictions and (2) procure analytical time derivative estimates with quantified uncertainties. The formulation leads to a generalized least-squares regression and, ultimately, reduced-order models that are described probabilistically with a closed-form expression for the posterior distribution of the operators. The resulting probabilistic surrogate model propagates uncertainties from the observed state data to reduced-order predictions. We demonstrate the method is effective for constructing low-dimensional models of two nonlinear partial differential equations representing a compressible flow and a nonlinear diffusion-reaction process, as well as for estimating the parameters of a low-dimensional system of nonlinear ordinary differential equations representing compartmental models in epidemiology.

Keywords: data-driven model reduction; uncertainty quantification; operator inference; Gaussian process; prediction uncertainty; parameter estimation; scientific machine learning

1 Introduction

Engineering design and analysis often involves many-query tasks, such as sensitivity analysis, uncertainty quantification, parameter estimation, and optimization, that require simulating a complex, expensive-to-simulate nonlinear system. To enable computationally efficient many-query simulations, we need data-driven surrogate models that can embed known physics to enable predictions outside of the training regime with respect to time and/or model inputs and build trust in such approximations by understanding the uncertainty in the predictions. This paper focuses on physics-based model reduction, the process of constructing low-dimensional systems as approximation models for high-dimensional processes, in the context of complex time-dependent dynamical systems. However, data-driven model reduction does not typically provide an assessment of uncertainty in the

^{*}Corresponding author. E-mail: mengwu.guo@math.lu.se.

predictions. We develop an interpretable, physics-aware probabilistic surrogate that addresses the challenge of characterizing prediction uncertainty in data-driven reduced-order modeling for nonlinear dynamical systems with sparse and/or noisy data.

Model reduction strategies confine system dynamics to a low-dimensional linear subspace or nonlinear manifold and define evolution equations for the coordinates of the original state in this reduced space [6, 54]. Classical reduced-order models (ROMs) are *intrusive* in the sense that a numerical solver for the system of interest, called the full-order model (FOM), is used explicitly in the construction of the ROM dynamics. In terms of implementation, intrusive ROMs are constructed by acting directly on code objects from the FOM. By contrast, *non-intrusive* or *data-driven* ROMs are constructed from system data alone, without needing access to any codes for solving the system of interest. This is a major advantage in scenarios without a FOM (e.g., experimental data), or in which the FOM can only be queried as a black box (e.g., proprietary, classified, or legacy codes). Furthermore, non-intrusive ROMs can learn from data that has been preprocessed with variable transformations or other advantageous alterations, whereas such preprocessing may severely complicate or even preclude intrusive ROM construction [52, 66]. Although non-intrusive ROMs can be constructed in many scenarios where intrusive ROMs are unavailable, they usually lack the accuracy guarantees and/or uncertainty estimates that many intrusive ROMs enjoy, such as in [32]. Data-driven ROMs, especially those of low intrinsic dimensions, do not usually come with rigorous accuracy guarantees and must therefore be equipped with probabilistic estimates to characterize uncertainty in the predictions, especially in the presence of sparse and/or noisy data. We focus on operator inference (OpInf) as the underlying ROM methodology in our work and build a probabilistic ROM.

Operator inference is a regression-based method for non-intrusively constructing interpretable ROMs from state data [25, 38, 48]. A polynomial model form is chosen based on the governing dynamics (or a variable transformation of the dynamics [52]), thereby embedding the physics structure of the problem into the ROM. The regression requires state data and their time derivatives. These time derivatives may be provided by a FOM, but more often they must be estimated from the state data, typically via finite differences. In the presence of sparse and/or noisy state data, the time derivative estimation and the OpInf regression problem in general suffer from accuracy issues. Uncertainty due to noise in the states, inaccuracies in the time derivative estimates, or model form error can be partially accounted for by introducing a regularization to the regression [42, 43, 51, 57] or by enforcing physics-motivated constraints [24, 27, 36, 61, 62]. Dealing with data noise remains a challenge, but some progress has been made in recent years. In [70], the re-projection scheme of [47] is coupled with an active-learning paradigm for strategically querying a FOM with noisy states. The procedure partially filters the training data, produces a data-driven ROM that can be related to an intrusive ROM, and decreases the estimated bias in the predicted states by at least one order of magnitude. In [69], a nonconvex output-based optimization is used instead of a residual-based regression to learn a ROM with polynomial structure. The method is computationally expensive compared with traditional OpInf but was shown to perform well in several scenarios with sparse data and significant noise pollution. However, OpInf ROMs do not come with rigorous characterization of uncertainty in the predictions. Our previous work [29] interprets OpInf as a Bayesian linear regression and develops a framework for producing probabilistic OpInf ROMs. The method poses a Gaussian noise model for the derivative of the state but still relies on high-quality time derivative data. In this paper, we utilize Gaussian processes to pose a noise model for the state, not its time derivative, and simultaneously obtain time derivative data in an analytical fashion along with a Bayesian version of OpInf.

Gaussian process (GP) regression [56] is a powerful tool for function approximation from data. The nonparametric Bayesian nature of GPs facilitates uncertainty quantification in the predictions, while its kernel-method essence ensures robustness in dealing with noisy and/or scarce data. Notwithstanding, GPs suffer from poor performance in temporal predictions beyond training coverage due to a general lack of physical consistency, as GP surrogate modeling is conducted in a black-box manner and no information about the governing equations is incorporated. Several recent strategies aim to overcome this challenge by synthesizing GP regression and

physics-based models [14, 31, 35, 55, 74]. In data-driven model reduction specifically, GP regression has been used to approximate the reduced trajectory over the time-parameter domain of interest [4, 15, 28, 30]. In this work, we combine GPs with OpInf to develop physics-aware probabilistic surrogates.

We develop a GP-based Bayesian OpInf method (GP-BayesOpInf) for predictive modeling of nonlinear dynamical systems that produces probabilistic predictions. Specifically, GPs are used for the likelihood definition in a Bayesian scheme of OpInf, combining the advantage of GP emulation in smoothing and interpolating imperfect data, and the predictive power of OpInf guaranteed by the preservation of formulation structure of the governing equations in projection-based model reduction. Based on the fact that Gaussianity is preserved through linear operations, differential-equation constraints of reduced states are embedded in the likelihood function. This does not require direct time-derivative approximations as in a Bayesian linear regression for equation fitting [29] and thus avoids generating low-quality derivative data when training trajectories are noisy and/or scarce. Thereafter, the resulting posterior estimate of the reduced operators can be formulated as a generalized least squares, in which an improved time derivative approximation is enabled by kernel regression and endowed with quantifiable uncertainty. We derive a closed form for the posterior distribution of the operators in OpInf, thus providing an interpretable, low-dimensional probabilistic surrogate model. The resulting physics-aware probabilistic surrogate can then sample realizations of operators from this posterior and integrate the low-dimensional dynamical system for each realization to provide predictions of the states of the dynamical system along with uncertainty in the predictions stemming from training data and modeling process.

The remainder of the paper is structured as follows. Section 2 provides a technical overview of the OpInf procedure for learning ROMs from state data. Section 3 develops a procedure that combines GP regression, Bayesian inference, and OpInf to learn probabilistic ROMs. Implementation details and a complete algorithm are presented in Section 4. GP-BayesOpInf is demonstrated for two nonlinear partial differential equations (PDEs) in Section 5. Section 6 shows that an adjustment to GP-BayesOpInf results in a parameter estimation strategy for a large class of systems of ordinary differential equations (ODEs). Finally, Section 7 concludes the paper. See Appendix C for a list of the major mathematical notation used throughout.

2 Reduced-order models via operator inference

Consider a high-dimensional system of ODEs,

$$\frac{d}{dt}\mathbf{q}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)), \quad \mathbf{q}(t_0) = \mathbf{q}_0, \quad t \in [t_0, t_f], \quad (2.1)$$

where $\mathbf{q}(t) \in \mathbb{R}^N$ is the state of the system at time t , $\mathbf{u}(t) \in \mathbb{R}^p$ represents inputs at time t (e.g., time-dependent source terms and/or boundary conditions), $\mathbf{f} : \mathbb{R}^N \times \mathbb{R}^p \rightarrow \mathbb{R}^N$ specifies the evolutionary system dynamics, $t_0 \in \mathbb{R}$ is the initial time, $\mathbf{q}_0 \in \mathbb{R}^N$ is the initial condition, and $t_f > t_0$ is the final time for which we want to know the state $\mathbf{q}(t)$. High-dimensional ODE systems with large N arise often from the spatial discretization of PDEs. A ROM of (2.1) is a system of ODEs with substantially fewer degrees of freedom whose solution—computed much more efficiently than directly solving (2.1)—can be used to approximate the solution to (2.1). The cost reduction resulting from using a ROM is crucial for applications requiring multiple solves of (2.1), e.g., with different input functions or initial conditions.

Projection-based ROMs introduce an affine approximation for the high-dimensional state,

$$\mathbf{q}(t) \approx \mathbf{V}\hat{\mathbf{q}}(t) + \bar{\mathbf{q}}, \quad (2.2)$$

where $\mathbf{V} \in \mathbb{R}^{N \times r}$ is the basis matrix, $\hat{\mathbf{q}}(t) \in \mathbb{R}^r$ is the reduced state at time t , $\bar{\mathbf{q}} \in \mathbb{R}^N$ is a fixed reference vector, and $r \ll N$ is the dimension of the reduced state. Both \mathbf{V} and $\bar{\mathbf{q}}$, and hence the reduced state dimension r , are either inferred from training observations of the state $\mathbf{q}(t)$ or chosen based on problem-specific knowledge

(e.g., $\bar{\mathbf{q}}$ may encode constant boundary conditions or represent a time-averaged solution). We typically require \mathbf{V} to have orthonormal columns, in which case dynamics for $\hat{\mathbf{q}}(t)$ are derived by substituting (2.2) into (2.1) and multiplying both sides by \mathbf{V}^\top , resulting in the system

$$\frac{d}{dt}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) := \mathbf{V}^\top \mathbf{f}(\mathbf{V}\hat{\mathbf{q}}(t) + \bar{\mathbf{q}}, \mathbf{u}(t)), \quad \hat{\mathbf{q}}(t_0) = \mathbf{V}^\top(\mathbf{q}_0 - \bar{\mathbf{q}}), \quad t \in [t_0, t_f], \quad (2.3)$$

where $\hat{\mathbf{f}}: \mathbb{R}^r \times \mathbb{R}^p \rightarrow \mathbb{R}^r$. For a wide class of problems, this approach preserves the dynamical structure of (2.1) in the ROM (2.3) [6]. In particular, if \mathbf{f} has a polynomial form, then so does $\hat{\mathbf{f}}$: for example, defining $\mathbf{c} \in \mathbb{R}^N$, $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{H} \in \mathbb{R}^{N \times N^2}$, and $\mathbf{B} \in \mathbb{R}^{N \times p}$, writing

$$\frac{d}{dt}\mathbf{q}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) = \mathbf{c} + \mathbf{A}\mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)] + \mathbf{B}\mathbf{u}(t), \quad (2.4a)$$

there exist corresponding $\hat{\mathbf{c}} \in \mathbb{R}^r$, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}$, and $\hat{\mathbf{B}} \in \mathbb{R}^{r \times p}$ such that

$$\frac{d}{dt}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) = \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] + \hat{\mathbf{B}}\mathbf{u}(t). \quad (2.4b)$$

In this work, we consider the scenario in which reduced-order operators such as $\hat{\mathbf{c}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{H}}$, and $\hat{\mathbf{B}}$ cannot be computed explicitly, as occurs when working with large, proprietary, or classified code bases, or in systems that must be transformed through a change of variables to first expose a polynomial structure similar to (2.4a) [52]. We will see examples of the latter in Section 5. In these cases, the polynomial structure of \mathbf{f} is known, but \mathbf{f} itself is computationally inaccessible and cannot be used directly to form $\hat{\mathbf{f}}$.

Operator inference (OpInf) is a non-intrusive projection-based method for learning polynomial ROMs by minimizing the model residual with respect to a collection of training data [38, 48]. In general, OpInf constructs ROMs that can be written as

$$\frac{d}{dt}\hat{\mathbf{q}}(t) = \hat{\mathbf{O}}\mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t)), \quad \hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad t \in [t_0, t_f], \quad (2.5)$$

where $\hat{\mathbf{O}} \in \mathbb{R}^{r \times d(r,p)}$ is the operator matrix, $\mathbf{d}: \mathbb{R}^r \times \mathbb{R}^p \rightarrow \mathbb{R}^{d(r,p)}$ is a known (typically nonlinear) function of the reduced state and input, and $d(r,p) \in \mathbb{N}$ is the operator dimension. The system (2.5) is linear with respect to the operator matrix $\hat{\mathbf{O}}$ but is usually nonlinear in the reduced state $\hat{\mathbf{q}}(t)$ and/or the input $\mathbf{u}(t)$ due to $\mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t))$. For example, the quadratic system (2.4b) can be written as (2.5) with $d(r,p) = 1 + r + r^2 + p$ and

$$\hat{\mathbf{O}} = [\hat{\mathbf{c}} \quad \hat{\mathbf{A}} \quad \hat{\mathbf{H}} \quad \hat{\mathbf{B}}] \in \mathbb{R}^{r \times d(r,p)}, \quad \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) = \begin{bmatrix} 1 \\ \hat{\mathbf{q}}(t) \\ \hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t) \\ \mathbf{u}(t) \end{bmatrix} \in \mathbb{R}^{d(r,p)}. \quad (2.6)$$

There are several ways to approach problems whose nonpolynomial nonlinear structure does not directly motivate the reduced structure (2.5). Many problems can be posed in polynomial form after a change of variables [52], a strategy that has found application in several model reduction contexts (see, e.g., [39] for an example of variable transformation enabling balanced truncation ROMs). OpInf models have also been demonstrated to be effective in some scenarios where the true dynamics can only be partially represented through a polynomial form [21, 42, 51, 66]. Additionally, a large class of nonpolynomial nonlinear terms may be incorporated via discrete empirical interpolation [5].

Given state-input pairs $\{(\mathbf{q}_j, \mathbf{u}_j)\}_{j=1}^{m-1}$ describing the solution of (2.1) at m discrete time instances t_0, \dots, t_{m-1} , a reference vector $\bar{\mathbf{q}}$, and a basis matrix \mathbf{V} , OpInf learns an operator matrix $\hat{\mathbf{O}}$ by compressing the observed

states as $\hat{\mathbf{q}}_j = \mathbf{V}^\top(\mathbf{q}_j - \hat{\mathbf{q}}) \in \mathbb{R}^r$, then solving the regression

$$\min_{\hat{\mathbf{O}}} \sum_{j=0}^{m-1} \left\| \hat{\mathbf{O}} \mathbf{d}(\hat{\mathbf{q}}_j, \mathbf{u}_j) - \dot{\hat{\mathbf{q}}}_j \right\|_2^2 = \min_{\hat{\mathbf{O}}} \left\| \hat{\mathbf{O}} \mathbf{D}^\top - \dot{\hat{\mathbf{Q}}} \right\|_F^2 = \min_{\hat{\mathbf{O}}} \left\| \mathbf{D} \hat{\mathbf{O}}^\top - \dot{\hat{\mathbf{Q}}}^\top \right\|_F^2, \quad (2.7a)$$

$$\dot{\hat{\mathbf{Q}}} = \begin{bmatrix} \dot{\hat{\mathbf{q}}}_0 & \dot{\hat{\mathbf{q}}}_1 & \cdots & \dot{\hat{\mathbf{q}}}_{m-1} \end{bmatrix} \in \mathbb{R}^{r \times m}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{d}(\hat{\mathbf{q}}_0, \mathbf{u}_0)^\top \\ \mathbf{d}(\hat{\mathbf{q}}_1, \mathbf{u}_1)^\top \\ \vdots \\ \mathbf{d}(\hat{\mathbf{q}}_{m-1}, \mathbf{u}_{m-1})^\top \end{bmatrix} \in \mathbb{R}^{m \times d(r,p)}. \quad (2.7b)$$

Here, $\dot{\hat{\mathbf{q}}}_j \in \mathbb{R}^r$ is an estimate of the time derivative $\frac{d}{dt} \hat{\mathbf{q}}(t) = \mathbf{V}^\top \frac{d}{dt} \mathbf{q}(t)$ at time t_j , usually computed from finite differences of the reduced snapshots $\hat{\mathbf{q}}_0, \dots, \hat{\mathbf{q}}_{m-1}$, and the matrix \mathbf{D} is called the data matrix. Because \mathbf{V} , $\hat{\mathbf{q}}$, and the observations $\{(\mathbf{q}_j, \mathbf{u}_j)\}$ are known, the only unknowns in (2.7) are the entries of the operator matrix $\hat{\mathbf{O}}$. Unless additional constraints on the unknown $\hat{\mathbf{O}}$ are imposed, (2.7) is a linear least-squares problem since (2.5) is linear with respect to $\hat{\mathbf{O}}$.

As a regression for $\hat{\mathbf{O}}$, (2.7a) decouples into r independent regressions,

$$\min_{\hat{\mathbf{o}}_i} \left\| \mathbf{D} \hat{\mathbf{o}}_i - \mathbf{z}_i \right\|_2^2, \quad (2.8)$$

where $\hat{\mathbf{o}}_i \in \mathbb{R}^{d(r,p)}$ is the i -th row of $\hat{\mathbf{O}}$ and $\mathbf{z}_i \in \mathbb{R}^k$ is the i -th row of $\dot{\hat{\mathbf{Q}}}$. It is often important to regularize the regression (2.8) in order for the resulting $\hat{\mathbf{O}}$ to produce a stable ROM [42, 57]. The most common way to do this is with a Tikhonov regularizer, i.e.,

$$\min_{\hat{\mathbf{o}}_i} \left\{ \left\| \mathbf{D} \hat{\mathbf{o}}_i - \mathbf{z}_i \right\|_2^2 + \left\| \mathbf{\Gamma}_i \hat{\mathbf{o}}_i \right\|_2^2 \right\}, \quad (2.9)$$

for some $\mathbf{\Gamma}_i \in \mathbb{R}^{d(r,p) \times d(r,p)}$. Alternatively, some scenarios call for constraints to be added to the regression, for example when preserving Hamiltonian or Lagrangian dynamics [27, 61, 62].

3 Bayesian operator inference with Gaussian processes

In deterministic OpInf (2.8), the training data are constructed from whatever (reduced) snapshot data are available, be they uniform or sparse in time, smooth or corrupted by observational noise. Here, we perform GP regression using the snapshot data and use the updated process to produce time derivative estimates and, ultimately, a probabilistic description for the reduced-order operator matrix $\hat{\mathbf{O}}$. The overall strategy is abbreviated as GP-BayesOpInf. See Appendix A for a brief overview of GPs.

3.1 Bayesian inference

In this section, we formulate a linear Bayesian inference for the operator matrix $\hat{\mathbf{O}}$ by separately considering each of its rows $\hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_r$, following the OpInf paradigm of learning each $\hat{\mathbf{o}}_i$ individually.

3.1.1 Likelihood and prior

We begin by modeling the reduced states $\hat{q}_1(t), \dots, \hat{q}_r(t)$ with independent GPs:

$$\hat{q}_i \sim \mathcal{GP}(0, \kappa_i(\cdot, \cdot)), \quad i = 1, \dots, r, \quad (3.1)$$

in which the mean function is assumed to be zero and $\kappa_i : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a kernel function specifying the covariance over the GP. If the kernel κ_i is at least once differentiable in each argument, then the time-derivative of each \hat{q}_i also follows a GP, and we have the joint processes [55, 56]

$$\begin{bmatrix} \hat{q}_i \\ \dot{\hat{q}}_i \end{bmatrix} \sim \text{vec-GP} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_i(\cdot, \cdot) & \partial_2 \kappa_i(\cdot, \cdot) \\ \partial_1 \kappa_i(\cdot, \cdot) & \partial_1 \partial_2 \kappa_i(\cdot, \cdot) \end{bmatrix} \right), \quad (3.2)$$

in which ∂_1 and ∂_2 respectively denote the partial derivatives with respect to the first and the second arguments.

Let $\mathbf{t} = (t_0, \dots, t_{m-1})^\top \in \mathbb{R}^m$ denote the $m \in \mathbb{N}$ time instances for which we have (reduced) snapshot data $\hat{\mathbf{q}}_0, \dots, \hat{\mathbf{q}}_{m-1} \in \mathbb{R}^r$. Define $\mathbf{Y}_i = \hat{q}_i(\mathbf{t})$ to be the random m -vector whose entries are the Gaussian random variables produced by evaluating the GP \hat{q}_i at each time instance in \mathbf{t} , and let $\mathbf{y}_i \in \mathbb{R}^m$ denote the snapshot data for the i -th reduced state \hat{q}_i at times \mathbf{t} , that is,

$$\begin{bmatrix} \hat{\mathbf{q}}_0 & \hat{\mathbf{q}}_1 & \cdots & \hat{\mathbf{q}}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_r^\top \end{bmatrix} \in \mathbb{R}^{r \times m}. \quad (3.3)$$

Note that \mathbf{y}_i is an observation of \mathbf{Y}_i . Likewise, let $\mathbf{t}' = (t'_0, \dots, t'_{m'-1})^\top \in \mathbb{R}^{m'}$ be the $m' \in \mathbb{N}$ time instances (typically with $m' \geq m$ and $[t'_0, t'_{m'-1}] \subset [t_0, t_{m-1}]$) where the GP is to produce estimates, and define $\mathbf{Z}_i = \dot{\hat{q}}_i(\mathbf{t}')$ to be the random m' -vector with entries given by the time derivative GP $\dot{\hat{q}}_i$ at times \mathbf{t}' .

The entries of the observational data vector \mathbf{y}_i are noisy, which we model by an additive white noise term with Gaussian distribution $\mathcal{N}(0, \chi_i)$. With this setting, the joint distribution of \mathbf{Y}_i and \mathbf{Z}_i is deduced from (3.2) to be

$$\begin{bmatrix} \mathbf{Y}_i \\ \mathbf{Z}_i \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_i^{yy} & \mathbf{K}_i^{yz} \\ \mathbf{K}_i^{zy} & \mathbf{K}_i^{zz} \end{bmatrix} \right), \quad \text{with} \quad \begin{aligned} \mathbf{K}_i^{yy} &= \kappa_i(\mathbf{t}, \mathbf{t}) + \chi_i \mathbf{I}_m \in \mathbb{R}^{m \times m}, \\ \mathbf{K}_i^{zy} &= \partial_1 \kappa_i(\mathbf{t}', \mathbf{t}) = (\mathbf{K}_i^{yz})^\top \in \mathbb{R}^{m' \times m}, \\ \mathbf{K}_i^{zz} &= \partial_1 \partial_2 \kappa_i(\mathbf{t}', \mathbf{t}') \in \mathbb{R}^{m' \times m'}, \end{aligned} \quad (3.4)$$

where \mathbf{I}_m is the $m \times m$ identity matrix and, for two vectors $\mathbf{a} = (a_1, \dots, a_{n_a})^\top$ and $\mathbf{b} = (b_1, \dots, b_{n_b})^\top$, the shorthand $\kappa(\mathbf{a}, \mathbf{b})$ denotes the $n_a \times n_b$ matrix whose (i, j) -th entry is $\kappa(a_i, b_j)$.

Remark 3.1 (Choosing m'). The integer m' must be selected *a priori* by the practitioner. If m' is too large, fitting and using each GP becomes unnecessarily expensive; if m' is too small, the subsequent Bayesian inference may have insufficient data and produce unsatisfactory results. For the numerical examples in Section 5 and Section 6, we base m' on the interval length $t_{m-1} - t_0$ and uniformly space the entries of \mathbf{t}' in the interval $[t_0, t_{m-1}]$, i.e., $t_0 = t'_0 < t'_1 < \dots < t'_{m'-1} = t_{m-1}$ with $t'_{j+1} = t'_j + (t_{m-1} - t_0)/(m' - 1)$.

For a given reduced-order operator vector $\hat{\mathbf{o}}_i \in \mathbb{R}^{d(r,p)}$ and a data matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{m' \times d(r,p)}$ corresponding to the estimation times \mathbf{t}' , let $\mathbf{R}_i := \mathbf{Z}_i - \tilde{\mathbf{D}}\hat{\mathbf{o}}_i$ be the random m' -vector describing the residual for the i -th reduced-order system equation (2.5) at times \mathbf{t}' . We may now define a likelihood function, not only on the direct state observations $\mathbf{Y}_i = \mathbf{y}_i$, but also on the alignment of the time derivatives \mathbf{Z}_i with $\tilde{\mathbf{D}}\hat{\mathbf{o}}_i$, i.e., the satisfaction of residuals $\mathbf{R}_i = \mathbf{0}$:

$$\pi_{\text{like}}(\hat{\mathbf{o}}_i) := p(\mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0} \mid \hat{\mathbf{o}}_i, \tilde{\mathbf{D}}, \boldsymbol{\theta}_i) \propto \exp \left(-\frac{1}{2} \begin{bmatrix} \mathbf{y}_i \\ \tilde{\mathbf{D}}\hat{\mathbf{o}}_i \end{bmatrix}^\top \begin{bmatrix} \mathbf{K}_i^{yy} & \mathbf{K}_i^{yz} \\ \mathbf{K}_i^{zy} & \mathbf{K}_i^{zz} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_i \\ \tilde{\mathbf{D}}\hat{\mathbf{o}}_i \end{bmatrix} \right), \quad (3.5)$$

which is derived by substituting $\mathbf{Z}_i = \tilde{\mathbf{D}}\hat{\mathbf{o}}_i$ (i.e., $\mathbf{R}_i = \mathbf{0}$) into (3.4). Here, $\boldsymbol{\theta}_i$ consists of the noise variance χ_i and the hyperparameters of the kernel κ_i . The likelihood depends on $\boldsymbol{\theta}_i$ through the definition of the matrices \mathbf{K}_i^{yy} , \mathbf{K}_i^{yz} , \mathbf{K}_i^{zy} , and \mathbf{K}_i^{zz} in (3.4).

Remark 3.2 (Interpretation of the likelihood). In the likelihood definition (3.5), the direct observations $\mathbf{Y}_i = \mathbf{y}_i$ introduce the factor of GP regression where noisy and scarce data can be smoothed and interpolated, while $\mathbf{R}_i = \mathbf{0}$ incorporates the constraints of differential equations into the inference process, explicitly ensuring physical consistency in the resulting ROMs.

Remark 3.3 (Gaussianity preserved by affine reconstruction). The independent reduced state GP representations $\hat{q}_1, \dots, \hat{q}_r$ of (3.1) can be expressed together as a single vector-valued GP,

$$\hat{\mathbf{q}}(t) \sim \text{vec-}\mathcal{GP}(\mathbf{0}, \text{diag}(\kappa_1(t, t'), \dots, \kappa_r(t, t'))). \quad (3.6a)$$

If these reduced states are used in the approximation (2.2), the reconstruction of full-order states is given by

$$\mathbf{q}(t) \approx \mathbf{V}\hat{\mathbf{q}}(t) + \bar{\mathbf{q}} \sim \text{vec-}\mathcal{GP}(\bar{\mathbf{q}}, \mathbf{V}\text{diag}(\kappa_1(t, t'), \dots, \kappa_r(t, t'))\mathbf{V}^\top), \quad (3.6b)$$

in which the covariance $\mathbf{V}\text{diag}(\kappa_1(t, t'), \dots, \kappa_r(t, t'))\mathbf{V}^\top$ is a low-rank approximation of $\text{Cov}[\mathbf{q}(t), \mathbf{q}(t')]$. Hence, the GP-based modeling for the reduced states provides a quantification of uncertainties in the states on the full-order level as well.

We also define the prior on the reduced-order operators $\hat{\mathbf{o}}_i$ to be independent normal distributions, i.e.,

$$\pi_{\text{prior}}(\hat{\mathbf{o}}_i) := p(\hat{\mathbf{o}}_i \mid \boldsymbol{\gamma}_i) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{\Gamma}_i \hat{\mathbf{o}}_i\|_2^2\right), \quad \text{with } \boldsymbol{\Gamma}_i := \text{diag}(\boldsymbol{\gamma}_i), \quad (3.7)$$

where the entries of $\boldsymbol{\gamma}_i \in \mathbb{R}_+^{d(r,p)}$ are nonnegative and are considered hyperparameters.

3.1.2 Posterior

Combining (3.5)–(3.7), Bayes' rule gives the posterior distribution

$$\pi_{\text{post}}(\hat{\mathbf{o}}_i) := p(\hat{\mathbf{o}}_i \mid \mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0}, \tilde{\mathbf{D}}, \boldsymbol{\theta}_i, \boldsymbol{\gamma}_i) \propto \pi_{\text{like}}(\hat{\mathbf{o}}_i) \pi_{\text{prior}}(\hat{\mathbf{o}}_i). \quad (3.8)$$

From Schur's formula for block matrix inversion [1], one has

$$\begin{aligned} \begin{bmatrix} \mathbf{K}_i^{yy} & \mathbf{K}_i^{yz} \\ \mathbf{K}_i^{zy} & \mathbf{K}_i^{zz} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{W}_i^{yy} & \mathbf{W}_i^{yz} \\ \mathbf{W}_i^{zy} & \mathbf{W}_i^{zz} \end{bmatrix}, & \mathbf{W}_i^{yy} &= (\mathbf{K}_i^{yy} - \mathbf{K}_i^{yz}(\mathbf{K}_i^{zz})^{-1}\mathbf{K}_i^{zy})^{-1}, \\ & & \mathbf{W}_i^{yz} &= -\mathbf{K}_i^{zz}\mathbf{K}_i^{zy}(\mathbf{K}_i^{yy})^{-1} = (\mathbf{W}_i^{yz})^\top, \\ & & \mathbf{W}_i^{zz} &= (\mathbf{K}_i^{zz} - \mathbf{K}_i^{zy}(\mathbf{K}_i^{yy})^{-1}\mathbf{K}_i^{yz})^{-1}. \end{aligned} \quad (3.9)$$

Therefore,

$$\begin{aligned} \pi_{\text{post}}(\hat{\mathbf{o}}_i) &\propto \pi_{\text{like}}(\hat{\mathbf{o}}_i) \pi_{\text{prior}}(\hat{\mathbf{o}}_i) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{y}_i^\top \mathbf{W}_i^{yy} \mathbf{y}_i + 2\hat{\mathbf{o}}_i^\top \tilde{\mathbf{D}}^\top \mathbf{W}_i^{zy} \mathbf{y}_i + \hat{\mathbf{o}}_i^\top \tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{D}} \hat{\mathbf{o}}_i + \hat{\mathbf{o}}_i^\top \boldsymbol{\Gamma}_i^\top \boldsymbol{\Gamma}_i \hat{\mathbf{o}}_i\right)\right) \\ &\propto \exp\left(-\frac{1}{2}(\hat{\mathbf{o}}_i - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\hat{\mathbf{o}}_i - \boldsymbol{\mu}_i)\right), \end{aligned} \quad (3.10)$$

$$\text{i.e., } \pi_{\text{post}}(\hat{\mathbf{o}}_i) = \mathcal{N}(\hat{\mathbf{o}}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i).$$

Here, the posterior mean of $\hat{\mathbf{o}}_i$ coincides with a *maximum a posteriori* (MAP) estimator, written as

$$\begin{aligned} \boldsymbol{\mu}_i &= (\hat{\mathbf{o}}_i)_{\text{MAP}} = -(\tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{D}} + \boldsymbol{\Gamma}_i^\top \boldsymbol{\Gamma}_i)^{-1} \tilde{\mathbf{D}}^\top \mathbf{W}_i^{zy} \mathbf{y}_i \\ &= (\tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{D}} + \boldsymbol{\Gamma}_i^\top \boldsymbol{\Gamma}_i)^{-1} \tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \mathbf{K}_i^{zy} (\mathbf{K}_i^{yy})^{-1} \mathbf{y}_i, \end{aligned} \quad (3.11)$$

and the posterior covariance matrix of $\hat{\boldsymbol{\theta}}_i$ is

$$\boldsymbol{\Sigma}_i = \left(\tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{D}} + \boldsymbol{\Gamma}_i^\top \boldsymbol{\Gamma}_i \right)^{-1}. \quad (3.12)$$

Note that

$$\tilde{\mathbf{z}}_i := \mathbf{K}_i^{zy} (\mathbf{K}_i^{yy})^{-1} \mathbf{y}_i \in \mathbb{R}^{m'} \quad (3.13)$$

describes the GP estimate of the time derivative $\frac{d}{dt} \hat{q}_i$ evaluated over \mathbf{t}' (see (4.4)). Thus, (3.11) can be abbreviated as $\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i \tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{z}}_i$.

3.1.3 Least squares formulation

The posterior mean $\boldsymbol{\mu}_i$ can be interpreted as a Tikhonov-regularized, generalized least-squares estimator of $\hat{\boldsymbol{\theta}}_i$ with the weight matrix \mathbf{W}_i^{zz} , whose inverse is the posterior covariance matrix of $\tilde{\mathbf{z}}_i$ through the GP regression of $(\mathbf{t}, \mathbf{y}_i)$. Specifically, $\boldsymbol{\mu}_i$ solves the problem

$$\min_{\boldsymbol{\eta}} \left\{ \left\| \tilde{\mathbf{D}} \boldsymbol{\eta} - \tilde{\mathbf{z}}_i \right\|_{\mathbf{W}_i^{zz}}^2 + \left\| \boldsymbol{\Gamma}_i \boldsymbol{\eta} \right\|_2^2 \right\}, \quad (3.14)$$

where $\|\bullet\|_{\mathbf{W}_i^{zz}} = (\bullet^\top \mathbf{W}_i^{zz} \bullet)^{1/2}$. Furthermore, an equivalent 2-norm least-squares problem is given by

$$\min_{\boldsymbol{\eta}} \left\| \begin{bmatrix} (\mathbf{W}_i^{zz})^{1/2} \tilde{\mathbf{D}} \\ \boldsymbol{\Gamma}_i \end{bmatrix} \boldsymbol{\eta} - \begin{bmatrix} (\mathbf{W}_i^{zz})^{1/2} \tilde{\mathbf{z}}_i \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \quad (3.15)$$

We use this least-squares formulation in the computation of $\boldsymbol{\mu}_i$ as it is more stable than solving the generalized normal equations in (3.11). This requires computing $(\mathbf{W}_i^{zz})^{1/2}$, which can be obtained by taking an eigendecomposition

$$(\mathbf{W}_i^{zz})^{-1} = \mathbf{K}_i^{zz} - \mathbf{K}_i^{zy} (\mathbf{K}_i^{yy})^{-1} \mathbf{K}_i^{yz} = \boldsymbol{\Psi}_i \boldsymbol{\Xi}_i \boldsymbol{\Psi}_i^\top, \quad (3.16)$$

then setting $(\mathbf{W}_i^{zz})^{1/2} = \boldsymbol{\Psi}_i \boldsymbol{\Xi}_i^{-1/2} \boldsymbol{\Psi}_i^\top$.

Remark 3.4 (Regularization of the covariance computation). In practice, computing (3.16) directly may be unstable due to the round-off errors introduced from solving for $\mathbf{C}_i := \mathbf{K}_i^{zy} (\mathbf{K}_i^{yy})^{-1} \mathbf{K}_i^{yz}$. As a remedy, we symmetrize \mathbf{C}_i and add a small amount of numerical regularization, replacing (3.16) with

$$(\mathbf{W}_i^{zz})^{-1} = \mathbf{K}_i^{zz} - \frac{1}{2} (\mathbf{C}_i + \mathbf{C}_i^\top) + \tau \mathbf{I}_{m'} = \boldsymbol{\Psi}_i \boldsymbol{\Xi}_i \boldsymbol{\Psi}_i^\top, \quad (3.17)$$

where $0 < \tau \ll 1$. The constant τ should be just large enough that the eigendecomposition of the above matrix yields all positive eigenvalues. The numerical examples in Section 5 and Section 6 use $\tau = 10^{-8}$.

3.2 Probabilistic reduced-order model predictions

Now that the reduced operator vectors $\hat{\boldsymbol{\theta}}_i$ are estimated with probability distributions $\pi_{\text{post}}(\hat{\boldsymbol{\theta}}_i)$, $1 \leq i \leq r$, they may be used to produce predictions over time through the ROM (2.5). In fact, the posterior distribution for the entire reduced operator matrix $\hat{\mathbf{O}}$ can be written as

$$\begin{aligned} \pi_{\text{post}}(\hat{\mathbf{O}}) &= p\left(\hat{\mathbf{O}} \mid \{\mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0}\}_{i=1}^r, \tilde{\mathbf{D}}, \{\boldsymbol{\theta}_i, \boldsymbol{\gamma}_i\}_{i=1}^r\right) \\ &\propto p\left(\{\mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0}\}_{i=1}^r \mid \hat{\mathbf{O}}, \tilde{\mathbf{D}}, \{\boldsymbol{\theta}_i\}_{i=1}^r\right) p\left(\hat{\mathbf{O}} \mid \{\boldsymbol{\gamma}_i\}_{i=1}^r\right) \\ &= \prod_{i=1}^r p\left(\mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0} \mid \hat{\boldsymbol{\theta}}_i, \tilde{\mathbf{D}}, \boldsymbol{\theta}_i\right) \prod_{i=1}^r p(\hat{\boldsymbol{\theta}}_i \mid \boldsymbol{\gamma}_i) = \prod_{i=1}^r \pi_{\text{like}}(\hat{\boldsymbol{\theta}}_i) \pi_{\text{prior}}(\hat{\boldsymbol{\theta}}_i). \end{aligned} \quad (3.18a)$$

Algorithm 3.1 Given data, compute the moments of the posterior operator distribution.

```

1: procedure OPPOST(
    Data matrix  $\tilde{\mathbf{D}} \in \mathbb{R}^{m' \times d}$ ,
    GP time derivative estimates  $\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_r \in \mathbb{R}^{m'}$ ,
    Weight matrices  $(\mathbf{W}_1^{zz})^{1/2}, \dots, (\mathbf{W}_r^{zz})^{1/2} \in \mathbb{R}^{m' \times m'}$ ,
    Prior variances  $\gamma_1, \dots, \gamma_r \in \mathbb{R}^d$ 
)
2:   for  $i = 1, \dots, r$  do
3:      $\mathbf{\Gamma}_i \leftarrow \text{diag}(\gamma_i)$ 
4:      $\boldsymbol{\mu}_i \leftarrow \text{argmin}_{\boldsymbol{\eta}} \{ \|\tilde{\mathbf{D}}\boldsymbol{\eta} - \tilde{\mathbf{z}}_i\|_{\mathbf{W}_i^{zz}}^2 + \|\mathbf{\Gamma}_i\boldsymbol{\eta}\|_2^2 \}$  # Solve numerically via (3.15).
5:      $\boldsymbol{\Sigma}_i \leftarrow (\tilde{\mathbf{D}}^\top \mathbf{W}_i^{zz} \tilde{\mathbf{D}} + \mathbf{\Gamma}_i^\top \mathbf{\Gamma}_i)^{-1}$ 
6:   return  $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \dots, (\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ 

```

The likelihood separates into a product because (1) the OpInf regression (2.7a)–(2.7b) decouples the inference of the operator matrix $\hat{\mathbf{O}}$ into its rows $\hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_r$, and (2) we model the reduced modes independently with individual one-dimensional GPs (3.1). The separation of the prior is a result of the independence in the prior definition (3.7). Hence the posterior of $\hat{\mathbf{O}}$ can be treated row-by-row independently, i.e.,

$$\pi_{\text{post}}(\hat{\mathbf{O}}) = \prod_{i=1}^r \pi_{\text{post}}(\hat{\mathbf{o}}_i), \quad (3.18b)$$

and $\hat{\mathbf{O}}$ is sampled from the product of the posterior distributions $\pi_{\text{post}}(\hat{\mathbf{o}}_1), \dots, \pi_{\text{post}}(\hat{\mathbf{o}}_r)$.

For a given sample of $\hat{\mathbf{O}}$, we evaluate the reduced state $\hat{\mathbf{q}}(t)$ by solving the corresponding ROM (2.5). The reduced state $\hat{\mathbf{q}}(t)$ therefore has the following distribution:

$$\begin{aligned} p(\hat{\mathbf{q}}(t) \mid \{\mathbf{Y}_i = \mathbf{y}_i, \mathbf{R}_i = \mathbf{0}\}_{i=1}^r, \tilde{\mathbf{D}}, \{\boldsymbol{\theta}_i, \gamma_i\}_{i=1}^r) &= \int p(\hat{\mathbf{q}}(t) \mid \hat{\mathbf{O}}) \pi_{\text{post}}(\hat{\mathbf{O}}) \, d\hat{\mathbf{O}} \\ &= \int p(\hat{\mathbf{q}}(t) \mid \hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_r) \prod_{i=1}^r \pi_{\text{post}}(\hat{\mathbf{o}}_i) \, d\hat{\mathbf{o}}_i, \end{aligned} \quad (3.19)$$

in which all the $\hat{\mathbf{o}}_i$'s are marginalized. Here, the notation $p(\hat{\mathbf{q}}(t) \mid \hat{\mathbf{O}})$ indicates a deterministic model evaluation for a given instance of $\hat{\mathbf{O}}$; indeed, one can write the reduced state $\hat{\mathbf{q}}(t)$ as a function of $\hat{\mathbf{O}}$, i.e., $\hat{\mathbf{q}}(t; \hat{\mathbf{O}})$. Note, however, that the distribution of $\hat{\mathbf{q}}$ is not necessarily Gaussian, even though the posterior of $\hat{\mathbf{O}}$ is, primarily because $\hat{\mathbf{q}}(t; \hat{\mathbf{O}})$ is nonlinear with respect to $\hat{\mathbf{O}}$. Having constructed a posterior distribution for $\hat{\mathbf{O}}$, we use Monte Carlo sampling over the operator posterior to estimate the moments of $\hat{\mathbf{q}}(t)$. Specifically, an individual sample $\hat{\mathbf{O}}_k$ defines an instance of the (deterministic) ROM (2.5),

$$\frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{O}}_k \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t)), \quad \hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad t \in [t_0, t_f],$$

which is solved numerically for $\hat{\mathbf{q}}(t)$ over $[t_0, t_f]$ using standard time-stepping integration methods. Each numerical solve is computationally inexpensive due to the low dimensionality of the system. After repeating this procedure for $k = 1, \dots, K$, the collection of K trajectories is used to estimate the mean and higher-order moments of the reduced state at time t , thereby establishing an uncertainty indicator for the model predictions.

3.3 Multiple trajectories

In many situations we have data for $\ell > 1$ trajectories of a single system due to variable initial conditions or input functions (but with the same set of system parameters, such as material properties). In this scenario,

observed states from each of the ℓ trajectories are used together to construct a global basis of rank r , then r GPs—one for each reduced mode, as before—are trained for each trajectory. The observation times \mathbf{t} may vary by trajectory, but we use the same estimation times \mathbf{t}' for each trajectory and mode.

The GP estimates for the i -th mode from each trajectory are used jointly to learn a posterior for the operator matrix row $\hat{\mathbf{o}}_i$. Using a superscript (l) to represent the l -th trajectory over $1 \leq l \leq \ell$, for each $i = 1, \dots, r$ we are concerned with the posterior

$$\begin{aligned} \pi_{\text{post}}(\hat{\mathbf{o}}_i) &= p\left(\hat{\mathbf{o}}_i \mid \left\{ \mathbf{Y}_i^{(l)} = \mathbf{y}_i^{(l)}, \mathbf{R}_i^{(l)} = \mathbf{0}, \tilde{\mathbf{D}}^{(l)}, \boldsymbol{\theta}_i^{(l)} \right\}_{l=1}^{\ell}, \gamma_i\right) \propto \pi_{\text{prior}}(\hat{\mathbf{o}}_i) \prod_{l=1}^{\ell} \pi_{\text{like}}^{(l)}(\hat{\mathbf{o}}_i), \\ &\text{with } \pi_{\text{like}}^{(l)}(\hat{\mathbf{o}}_i) = p\left(\mathbf{Y}_i^{(l)} = \mathbf{y}_i^{(l)}, \mathbf{R}_i^{(l)} = \mathbf{0} \mid \hat{\mathbf{o}}_i, \tilde{\mathbf{D}}^{(l)}, \boldsymbol{\theta}_i^{(l)}\right), \end{aligned} \quad (3.20)$$

where $\tilde{\mathbf{D}}^{(l)}$ is formed from the GP state estimates for the l -th trajectory, i.e., $\tilde{\mathbf{y}}_1^{(l)}, \dots, \tilde{\mathbf{y}}_r^{(l)}$. The posterior mean and covariance respectively become

$$\boldsymbol{\mu}_i = \left(\sum_{l=1}^{\ell} (\tilde{\mathbf{D}}^{(l)})^{\top} \mathbf{W}_i^{zz(l)} \tilde{\mathbf{D}}^{(l)} + \boldsymbol{\Gamma}_i^{\top} \boldsymbol{\Gamma}_i \right)^{-1} \sum_{l=1}^{\ell} (\tilde{\mathbf{D}}^{(l)})^{\top} \mathbf{W}_i^{zz(l)} \tilde{\mathbf{z}}_i^{(l)}, \quad (3.21a)$$

$$\boldsymbol{\Sigma}_i = \left(\sum_{l=1}^{\ell} (\tilde{\mathbf{D}}^{(l)})^{\top} \mathbf{W}_i^{zz(l)} \tilde{\mathbf{D}}^{(l)} + \boldsymbol{\Gamma}_i^{\top} \boldsymbol{\Gamma}_i \right)^{-1}, \quad (3.21b)$$

in which $\tilde{\mathbf{z}}_i^{(l)} := (\mathbf{K}_i^{zy(l)} (\mathbf{K}_i^{yy(l)})^{-1} \mathbf{y}_i^{(l)})$ and $\mathbf{W}_i^{zz(l)} := (\mathbf{K}_i^{zz(l)} - \mathbf{K}_i^{zy(l)} (\mathbf{K}_i^{yy(l)})^{-1} \mathbf{K}_i^{yz(l)})^{-1}$ is the corresponding weight matrix. In this case, the generalized least-squares problem (3.14) for $\boldsymbol{\mu}_i$ becomes

$$\min_{\boldsymbol{\eta}} \left\{ \sum_{l=1}^{\ell} \left\| \tilde{\mathbf{D}}^{(l)} \boldsymbol{\eta} - \tilde{\mathbf{z}}_i^{(l)} \right\|_{\mathbf{W}_i^{zz(l)}}^2 + \|\boldsymbol{\Gamma}_i \boldsymbol{\eta}\|_2^2 \right\}. \quad (3.22)$$

We can write this in a form that is similar to (3.14) by noting that

$$\sum_{l=1}^{\ell} \left\| \tilde{\mathbf{D}}^{(l)} \boldsymbol{\eta} - \tilde{\mathbf{z}}_i^{(l)} \right\|_{\mathbf{W}_i^{zz(l)}}^2 = \left\| \begin{bmatrix} \tilde{\mathbf{D}}^{(1)} \\ \vdots \\ \tilde{\mathbf{D}}^{(\ell)} \end{bmatrix} \boldsymbol{\eta} - \begin{bmatrix} \tilde{\mathbf{z}}_i^{(1)} \\ \vdots \\ \tilde{\mathbf{z}}_i^{(\ell)} \end{bmatrix} \right\|_{\mathbf{W}_i}^2, \quad (3.23)$$

where \mathbf{W}_i is the block-diagonal matrix $\mathbf{W}_i = \text{diag}(\mathbf{W}_i^{zz(1)}, \dots, \mathbf{W}_i^{zz(\ell)})$. Hence, the multiple-trajectory case can be written as the original case with a concatenated data matrix, block diagonal weights, and concatenated time derivative estimates.

Remark 3.5 (Parametric problems). The strategy presented here assumes that the underlying reduced operators comprising $\hat{\mathbf{O}}$ are constant across all trajectories, which is the case when only initial conditions or input terms are altered. Parametric problems in which the reduced operators vary by trajectory require additional specialized treatment (see, e.g., [43]) and remain a subject for future work.

4 Algorithmic procedure

The inference to define the posterior parameters (3.11)–(3.12) depends on the choice of kernel hyperparameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_r$, the data matrix $\tilde{\mathbf{D}}$, and the prior variance parameters $\gamma_1, \dots, \gamma_r$. This section provides strategies for selecting each of these quantities and details the associated numerical algorithms. For simplicity we return to the single-trajectory PDE setting, but the approach readily generalizes to cases with data from multiple trajectories (see Appendix B).

Algorithm 4.1 Gaussian process regression with derivative estimation.

```

1: procedure GPFIT(
    Observation times  $\mathbf{t} \in \mathbb{R}^m$ ,
    Observed data  $\mathbf{y} \in \mathbb{R}^m$ ,
    Estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Numerical regularization  $0 < \tau \ll 1$ 
)
2:    $\boldsymbol{\theta}^* \leftarrow \operatorname{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{t}, \mathbf{y})$            # Maximize the marginal likelihood (4.1).
3:    $\mathbf{K}^{yy} \leftarrow \kappa(\mathbf{t}, \mathbf{t}; \boldsymbol{\theta}^*) + \chi \mathbf{I}_m$ 
4:    $\mathbf{K}^{zy} \leftarrow \partial_1 \kappa(\mathbf{t}', \mathbf{t}; \boldsymbol{\theta}^*)$ 
5:    $\mathbf{K}^{zz} \leftarrow \partial_1 \partial_2 \kappa(\mathbf{t}', \mathbf{t}'; \boldsymbol{\theta}^*)$ 
6:    $\tilde{\mathbf{y}} \leftarrow \kappa(\mathbf{t}', \mathbf{t}; \boldsymbol{\theta}^*) (\mathbf{K}^{yy})^{-1} \mathbf{y}$            # GP state estimate over  $\mathbf{t}'$ .
7:    $\tilde{\mathbf{z}} \leftarrow \mathbf{K}^{zy} (\mathbf{K}^{yy})^{-1} \mathbf{y}$            # GP time derivative estimate over  $\mathbf{t}'$ .
8:    $\mathbf{C} \leftarrow \mathbf{K}^{zy} (\mathbf{K}^{yy})^{-1} (\mathbf{K}^{zy})^\top$ 
9:    $\boldsymbol{\Psi} \boldsymbol{\Xi} \boldsymbol{\Psi}^\top \leftarrow \mathbf{K}^{zz} - \frac{1}{2} (\mathbf{C} + \mathbf{C}^\top) + \tau \mathbf{I}_{m'}$    # Eigendecomposition of  $(\mathbf{W}^{zz})^{-1}$  (see Remark 3.4).
10:   $(\mathbf{W}^{zz})^{1/2} \leftarrow \boldsymbol{\Psi} \boldsymbol{\Xi}^{-1/2} \boldsymbol{\Psi}^\top$ 
11:  return  $\tilde{\mathbf{y}}, \tilde{\mathbf{z}}, (\mathbf{W}^{zz})^{1/2}$            # State / time derivative estimates and weighting matrix.

```

4.1 Kernel hyperparameter selection

For each $i = 1, \dots, r$, we select $\boldsymbol{\theta}_i$ through the one-dimensional GP regression of the (noisy, reduced) snapshot data $(\mathbf{t}, \mathbf{y}_i)$. Specifically, we approximately solve the simplified maximum marginal log-likelihood problem

$$\begin{aligned} \boldsymbol{\theta}_i^* &= \operatorname{argmax}_{\boldsymbol{\theta}_i} \mathcal{L}(\boldsymbol{\theta}_i; \mathbf{t}, \mathbf{y}_i) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}_i} \left\{ -\frac{1}{2} \mathbf{y}_i^\top (\kappa(\mathbf{t}, \mathbf{t}; \boldsymbol{\theta}_i) + \chi_i \mathbf{I}_m)^{-1} \mathbf{y}_i - \frac{1}{2} \log |\kappa(\mathbf{t}, \mathbf{t}; \boldsymbol{\theta}_i) + \chi_i \mathbf{I}_m| - \frac{m}{2} \log(2\pi) \right\}, \end{aligned} \quad (4.1)$$

where χ_i is included in $\boldsymbol{\theta}_i$. Exact solutions of this nonconvex optimization are difficult to obtain, but there are several open-source resources for obtaining satisfactory approximate solutions. Our implementation utilizes the Python library `scikit-learn` [46], which applies the limited-memory bound-constrained BFGS algorithm [9] available from `scipy.optimize` [72].

4.2 Data selection

The data matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{m' \times d(r,p)}$ driving the Bayesian inference contains data at the estimation points \mathbf{t}' , yet we only are provided with snapshot data corresponding to \mathbf{t} . To populate $\tilde{\mathbf{D}}$, we utilize the one-dimensional GP \hat{q}_i of (3.1) to estimate the state at times \mathbf{t}' ,

$$\tilde{\mathbf{y}}_i := \kappa_i(\mathbf{t}', \mathbf{t}) (\kappa_i(\mathbf{t}, \mathbf{t}) + \chi_i \mathbf{I}_m)^{-1} \mathbf{y}_i = \kappa_i(\mathbf{t}', \mathbf{t}) (\mathbf{K}_i^{yy})^{-1} \mathbf{y}_i, \quad (4.2)$$

which provides a smooth interpolation of the trajectory. This is especially beneficial for cases when the snapshot data $\mathbf{y}_1, \dots, \mathbf{y}_r$ are noisy. Collecting the filtered state estimates $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_r$, we construct $\tilde{\mathbf{D}}$ akin to (2.7b):

$$\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{d}(\tilde{\mathbf{q}}_0, \mathbf{u}(t'_0))^\top \\ \mathbf{d}(\tilde{\mathbf{q}}_1, \mathbf{u}(t'_1))^\top \\ \vdots \\ \mathbf{d}(\tilde{\mathbf{q}}_{m'-1}, \mathbf{u}(t'_{m'-1}))^\top \end{bmatrix} \in \mathbb{R}^{m' \times d(r,p)}, \quad \begin{bmatrix} \tilde{\mathbf{q}}_0 & \tilde{\mathbf{q}}_1 & \cdots & \tilde{\mathbf{q}}_{m'-1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{y}}_1^\top \\ \vdots \\ \tilde{\mathbf{y}}_r^\top \end{bmatrix} \in \mathbb{R}^{r \times m'}. \quad (4.3)$$

Algorithm 4.2 Estimate the mean error of the Bayesian OpInf model for a given prior variance.

```

1: procedure OPINFERROR(
    Prior variances  $\gamma_1, \dots, \gamma_r \in \mathbb{R}^d$ ;
    Regression arguments  $\alpha = \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_r, (\mathbf{W}_1^{zz})^{1/2}, \dots, (\mathbf{W}_r^{zz})^{1/2}\}$ ,
    GP state estimates  $\tilde{\mathbf{Q}} \in \mathbb{R}^{r \times m'}$ 
    Input function  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^p$ ,
    GP estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Stability margin parameter  $\varphi > 0$ ,
    Number of samples  $n_s \in \mathbb{N}$ ,
    Final prediction time  $t_f > t'_0$ 
)
2:    $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \dots, (\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) \leftarrow \text{OPPOST}(\alpha, \gamma_1, \dots, \gamma_r)$  # Algorithm 3.1.
3:    $\hat{\mathbf{q}}_0 \leftarrow \tilde{\mathbf{Q}}_{:,0}$  # Initial condition.
4:    $B \leftarrow \varphi \cdot \max_{ij} |\tilde{\mathbf{Q}}_{ij}|$  # Bound for stability check.
5:   for  $k = 1, \dots, n_s$  do
6:      $\hat{\mathbf{O}}_k \leftarrow \text{sample from } [\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \cdots \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)]^T$  # Sample operator matrix.
7:      $\hat{\mathbf{Q}} \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{O}}_k \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t)), \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0 \text{ over } t \in [t'_0, t_f]$ 
8:     if  $\max_{ij} |\hat{\mathbf{Q}}_{ij}| > B$  then # Detect and disqualify unstable ROMs.
9:       return  $\infty$ 
10:     $\hat{\mathbf{Q}}_k \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{O}}_k \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}(t)), \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0 \text{ over } t \in \mathbf{t}'$ 
11:     $\hat{\mathbf{Q}} = \frac{1}{n_s} \sum_{k=1}^{n_s} \hat{\mathbf{Q}}_k$  # Sample mean of the ROM solutions.
12:    return  $\|\hat{\mathbf{Q}} - \tilde{\mathbf{Q}}\|$  # Error compared to GP state estimates.

```

From (4.2), we explicitly see the derivative relationship between the state estimates $\tilde{\mathbf{y}}_i$ and the time derivative estimates $\tilde{\mathbf{z}}_i$:

$$\partial_1 \tilde{\mathbf{y}}_i = \partial_1 \kappa_i(\mathbf{t}', \mathbf{t})(\mathbf{K}_i^{yy})^{-1} \mathbf{y}_i = \mathbf{K}_i^{zy} (\mathbf{K}_i^{yy})^{-1} \mathbf{y}_i = \tilde{\mathbf{z}}_i. \quad (4.4)$$

Algorithm 4.1 enumerates the steps for the GP regression for a single reduced mode. Apart from solving the optimization problem (4.1), the computational cost is $\mathcal{O}(m^3 + (m')^3)$, dominated by the expense of inverting \mathbf{K}^{yy} and factorizing $(\mathbf{W}^{zz})^{-1}$. These matrices are symmetric, hence we use Cholesky factorization and back substitution to apply $(\mathbf{K}^{yy})^{-1}$ in steps 6–8 and use symmetric QR for the eigendecomposition of $(\mathbf{W}^{zz})^{-1}$ in step 9, both via standard tools available in `scipy.linalg` [72].

4.3 Prior variance selection

The vector $\gamma_i \in \mathbb{R}^d$ parameterizing the prior distribution $\pi_{\text{prior}}(\hat{\mathbf{O}}_i)$ plays the role of a Tikhonov regularizer in the generalized least-squares regression (3.14) and strongly affects the stability of the resulting ROM. To methodically select $\gamma_1, \dots, \gamma_r$, we extend the approach introduced in [42], which balances model stability with accuracy in reproducing training data. For a candidate set $\{\gamma_i\}_{i=1}^r$, we solve the corresponding Bayesian inference (3.11)–(3.12) for $\{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}_{i=1}^r$, sample $n_s \in \mathbb{N}$ times from the posterior operator distribution $\hat{\mathbf{O}} \sim [\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \cdots \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)]^T$, and solve the resulting reduced systems (2.5) over the full time domain of interest $[t_0, t_f]$. Recall that $t_f > t_{m-1}$, that is, the full simulation domain extends beyond where noisy data are observed. If the magnitude of any of the reduced model solutions exceeds a predetermined bound computed from the training data, the candidate set is disqualified for failing to produce a sufficiently stable model; otherwise, the sample mean of the model solutions is compared to the filtered state data $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_r$. A grid search, followed by a derivative-free optimization, is used to select optimal $\{\gamma_i^*\}_{i=1}^r$ that minimizes the difference between the sample mean and the training data while not violating the stability criterion. Details are

Algorithm 4.3 Bayesian Operator Inference with Gaussian processes (GP-BayesOpInf).

```

1: procedure GPBAYESOPINF(
    Observation times  $\mathbf{t} \in \mathbb{R}^m$ ,
    Snapshots  $\mathbf{Q} \in \mathbb{R}^{N \times m}$ ,
    Reduced dimension  $r \ll N$ ,
    Estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Numerical regularization  $0 < \tau \ll 1$ ,
    Input function  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^p$ 
)
2: # GP regression on reduced training data.
3:  $\mathbf{V} \leftarrow \mathbf{basis}(\mathbf{Q}; r)$  # Rank- $r$  basis (e.g., POD).
4:  $[\mathbf{y}_1 \ \cdots \ \mathbf{y}_r]^\top \leftarrow \mathbf{V}^\top \mathbf{Q}$  # Compress training data.
5: for  $i = 1, \dots, r$  do
6:      $\tilde{\mathbf{y}}_i, \tilde{\mathbf{z}}_i, (\mathbf{W}_i^{zz})^{1/2} \leftarrow \mathbf{GPFIT}(\mathbf{t}, \mathbf{y}_i, \mathbf{t}', \tau)$  # Algorithm 4.1.
7: # OpInf regression to construct a probabilistic ROM.
8:  $\tilde{\mathbf{Q}} \leftarrow [\tilde{\mathbf{y}}_1 \ \cdots \ \tilde{\mathbf{y}}_r]^\top$ 
9:  $\tilde{\mathbf{D}} \leftarrow \mathbf{data\_matrix}(\tilde{\mathbf{Q}}, \mathbf{u}(\mathbf{t}'))$  # Data matrix (4.3) from GP state estimates.
10:  $\boldsymbol{\alpha} \leftarrow \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_r, (\mathbf{W}_1^{zz})^{1/2}, \dots, (\mathbf{W}_r^{zz})^{1/2}\}$  # Regression arguments.
11:  $\boldsymbol{\gamma}_1^*, \dots, \boldsymbol{\gamma}_r^* \leftarrow \mathbf{argmin}_{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_r} \mathbf{OPINFERROR}(\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_r; \boldsymbol{\alpha}, \tilde{\mathbf{Q}}, \mathbf{u}, \mathbf{t}')$  # Algorithm 4.2.
12:  $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \dots, (\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) \leftarrow \mathbf{OPPOST}(\boldsymbol{\alpha}, \boldsymbol{\gamma}_1^*, \dots, \boldsymbol{\gamma}_r^*)$  # Algorithm 3.1.
13: return  $[\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \ \cdots \ \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)]^\top$ 

```

provided in Algorithm 4.2; in Section 5, the experiments use $n_s = 20$ posterior samples and set the stability margin constant to $\varphi = 5$. A similar strategy was deployed in our previous work [29]; however, in that work, only the MAP points $\hat{\mathbf{O}} = [\boldsymbol{\mu}_1 \ \cdots \ \boldsymbol{\mu}_r]^\top$ were used to evaluate candidate regularizers. Our approach, only slightly more computationally expensive, tests stability for multiple posterior samples and hence produces more robust models than the previous approach.

Executing Algorithm 4.2 is computationally inexpensive because all operations are in the reduced state space. However, minimizing the reconstruction error with respect to the prior variance vectors $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_r \in \mathbb{R}^{d(r,p)}$ constitutes an optimization of $r \cdot d(r,p)$ positive variables, which may require a large number of evaluations of Algorithm 4.2. To reduce the cost of the optimization, it is beneficial to parameterize the prior variances and thus reduce the total degrees of freedom designating the prior. First, setting $\boldsymbol{\gamma}_1 = \dots = \boldsymbol{\gamma}_r = \boldsymbol{\gamma} \in \mathbb{R}^{d(r,p)}$ reduces the number of unknowns to $d(r,p)$ positive constants. This is reasonable as long as the reduced-order modes have approximately the same magnitude. Second, $\boldsymbol{\gamma}$ may be designed so that each operator in the model is penalized individually. For example, for a quadratic ROM with operator matrix $\hat{\mathbf{O}} = [\hat{\mathbf{H}} \ \hat{\mathbf{B}}]$, we may define

$$\boldsymbol{\gamma} = \left[\overbrace{\gamma_1 \ \cdots \ \gamma_1}^{r^2 \text{ times}} \ \overbrace{\gamma_2 \ \cdots \ \gamma_2}^{p \text{ times}} \right]^\top, \quad (4.5)$$

so that $\gamma_1 > 0$ corresponds to a penalization for the quadratic operator $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}$ and $\gamma_2 > 0$ penalizes the input operator $\hat{\mathbf{B}} \in \mathbb{R}^{r \times p}$. This choice is similar to the regularization strategies of [33, 42, 51, 57] and results in a two-dimensional optimization, reasonably solvable via, e.g., the Nelder–Mead method [44]. In our numerical experiments, we found it convenient to make the further simplification $\boldsymbol{\gamma} = [\gamma \ \cdots \ \gamma]^\top$ for a single constant $\gamma > 0$, resulting in a one-dimensional optimization that can be carried out, e.g., via Brent’s method [8]. The numerical results were comparable when using the strategy of (4.5) or allowing each entry of $\boldsymbol{\gamma}$ to vary.

With all of the major ingredients now in hand, the entire computational procedure is detailed in Algorithm 4.3 and illustrated in Figure 1. To ensure that all reduced-order modes are of similar magnitudes, we center the (noisy)

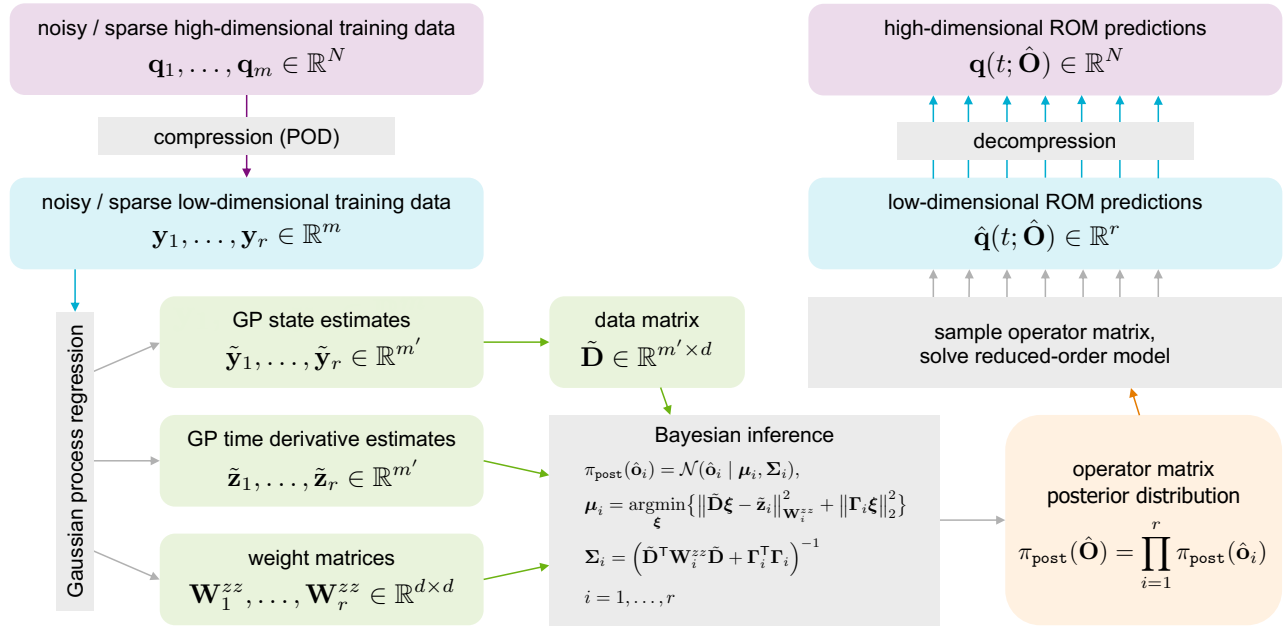


Figure 1: Flow diagram of GP-BayesOpInf via Algorithm 4.3.

snapshots $\mathbf{q}_0, \dots, \mathbf{q}_{m-1} \in \mathbb{R}^N$ by setting $\mathbf{Q} = [(\mathbf{q}_0 - \bar{\mathbf{q}}) \ \dots \ (\mathbf{q}_{m-1} - \bar{\mathbf{q}})] \in \mathbb{R}^{N \times m}$, where $\bar{\mathbf{q}} = \frac{1}{m} \sum_{j=0}^{m-1} \mathbf{q}_j \in \mathbb{R}^N$ is the mean snapshot. These centered snapshots are compressed to $r \ll N$ degrees of freedom in steps 3–4 by introducing the $N \times r$ basis matrix \mathbf{V} . We use proper orthogonal decomposition (POD) [7, 26, 63], in which \mathbf{V} is comprised of the first r left singular vectors of \mathbf{Q} . This choice is advantageous because POD has a hierarchical smoothing effect [17, 19, 71], the dominant modes being smoothed more than the high-frequency modes (see Figure 3). After compression, r GPs are trained via Algorithm 4.1, which are used in steps 8–10 to populate the generalized least-squares regression (3.14) for each reduced mode. Finally, the prior variances are chosen by minimizing the procedure of Algorithm 4.2, and the posterior for the operator matrix is produced by applying Algorithm 3.1 a final time. With only minor adjustments, Algorithms 4.2–4.3 can be adapted to cases where multiple data trajectories are available, as discussed in Section 3.3 (see Appendix B).

5 Numerical examples of GP-BayesOpInf

This section demonstrates GP-BayesOpInf via Algorithm 4.3 for two nonlinear PDEs. The first example uses a single data trajectory while the second incorporates multiple trajectories. To guarantee sufficient smoothness for derivative evaluations, in each of the examples we use the squared exponential (radial basis function) kernel

$$\kappa(t, t'; \boldsymbol{\theta}) = \sigma^2 \exp\left(-\frac{1}{2} \frac{(t - t')^2}{\ell^2}\right), \quad (5.1)$$

in which the signal variance $\sigma^2 > 0$ and the lengthscale $\ell > 0$ are the kernel hyperparameters. Codes to reproduce the experiments are available at <https://github.com/sandia-labs/GP-BayesOpInf>.

5.1 Compressible Euler equations

In this section, we show the effectiveness of the GP-BayesOpInf method as a probabilistic surrogate for compressible Euler equations with noisy and sparse data.

5.1.1 Governing equations

We consider the compressible Euler equations for an ideal gas over the one-dimensional spatial domain $[0, 2]$ with periodic boundary conditions, written in conservative form as

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho v \\ \rho e \end{bmatrix} = -\frac{\partial}{\partial x} \begin{bmatrix} \rho v \\ \rho v^2 + p \\ (\rho e + p)v \end{bmatrix}, \quad (5.2a)$$

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2, \quad (5.2b)$$

$$\rho(0, t) = \rho(2, t), \quad v(0, t) = v(2, t), \quad e(0, t) = e(2, t), \quad (5.2c)$$

where v is the velocity, ρ is the density, p is the pressure, ρv is the specific momentum, ρe is the total energy, and $\gamma = 1.4$ is the heat capacity ratio. The initial conditions are defined by setting $p(x, 0) = 10^5$ Pa and constructing cubic splines for the density and velocity with interpolation values

$$\begin{aligned} \rho(0, 0) &= 22 \text{ kg/m}^3, & \rho(\frac{2}{3}, 0) &= 20 \text{ kg/m}^3, & \rho(\frac{4}{3}, 0) &= 24 \text{ kg/m}^3, \\ v(0, 0) &= 95 \text{ m/s}, & v(\frac{2}{3}, 0) &= 105 \text{ m/s}, & v(\frac{4}{3}, 0) &= 100 \text{ m/s}. \end{aligned} \quad (5.3)$$

The initial energy $e(x, 0)$ is defined from the initial conditions of p , ρ , and v through the ideal gas law (5.2b).

To generate noisy and/or sparse data for this problem, we discretize the spatial derivatives in (5.2a) with first-order rightward finite differences using $n_x = 200$ degrees of freedom per variable, resulting in a system of $N = 3n_x = 600$ ordinary differential equations. For a given number of observations $m \in \mathbb{N}$ and a final observation time $t_{m-1} > 0$, we select observation times $\mathbf{t} = [t_0 \ t_1 \ \dots \ t_{m-1}]^\top \in \mathbb{R}^m$ by setting $t_0 = 0$ and sampling the $m - 2$ additional points (t_1, \dots, t_{m-2}) from the uniform distribution over the interval $(0, t_{m-1})$. Note that this results in non-uniform spacing between the sampling times. We then integrate the discretized system in time with the explicit adaptive-step Runge–Kutta method of order 5 [18] and employ quartic polynomial interpolation [59] to obtain the solution at times \mathbf{t} . The data after the initial condition are polluted with additive Gaussian noise. Specifically, for a given noise level $\xi \geq 0$, we observe

$$\mathbf{q}_j^{(c)} := \begin{bmatrix} \mathbf{q}_\rho(t_j) + \boldsymbol{\varepsilon}_\rho \\ \mathbf{q}_{\rho v}(t_j) + \boldsymbol{\varepsilon}_{\rho v} \\ \mathbf{q}_{\rho e}(t_j) + \boldsymbol{\varepsilon}_{\rho e} \end{bmatrix} \in \mathbb{R}^N, \quad \begin{aligned} \boldsymbol{\varepsilon}_\rho &\sim \mathcal{N}(\mathbf{0}, \nu_\rho^2 \mathbf{I}_{n_x}), \\ \boldsymbol{\varepsilon}_{\rho v} &\sim \mathcal{N}(\mathbf{0}, \nu_{\rho v}^2 \mathbf{I}_{n_x}), \\ \boldsymbol{\varepsilon}_{\rho e} &\sim \mathcal{N}(\mathbf{0}, \nu_{\rho e}^2 \mathbf{I}_{n_x}), \end{aligned} \quad j = 1, \dots, m-1, \quad (5.4a)$$

where $\mathbf{q}_\rho(t_j), \mathbf{q}_{\rho v}(t_j), \mathbf{q}_{\rho e}(t_j) \in \mathbb{R}^{n_x}$ respectively denote the spatially discrete numerical solutions for ρ , ρv , and ρe at time $t_j \in \mathbf{t}$, and where $\nu_\rho, \nu_{\rho v}, \nu_{\rho e} > 0$ is the observed range of the indicated variable scaled by the noise level. For example, collecting all numerical solutions for the density in the matrix $\mathbf{Q}_\rho = [\mathbf{q}_\rho(t_0) \ \dots \ \mathbf{q}_\rho(t_{m-1})]$, we set $\nu_\rho = \xi(\max(\mathbf{Q}_\rho) - \min(\mathbf{Q}_\rho))$. We also observe the given initial conditions without noise,

$$\mathbf{q}_0^{(c)} := \begin{bmatrix} \mathbf{q}_\rho(t_0) \\ \mathbf{q}_{\rho v}(t_0) \\ \mathbf{q}_{\rho e}(t_0) \end{bmatrix} \in \mathbb{R}^N. \quad (5.4b)$$

Figure 2 shows the initial conditions and, at select spatial locations, $m = 50$ snapshots with $t_{m-1} = 0.06$ s and $\xi = 1\%$ additive noise, along with the nonnoised numerical solution to the system.

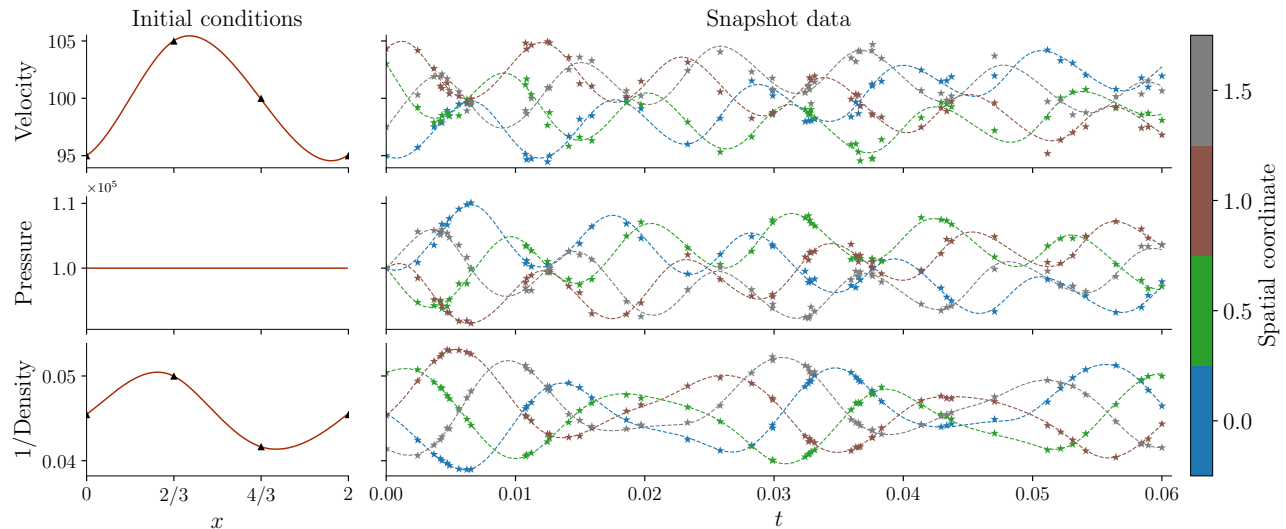


Figure 2: Initial conditions for the Euler system (5.2a)–(5.2c) (left) and noisy state observations (right) expressed in the specific volume variables (v, p, ζ) . The initial conditions are constructed with periodic splines interpolating the marked points. On the right, the dashed lines show the numerical solution without noise as a function of time at four fixed points in space; the markers indicate $m = 50$ sparsely observed states with $\xi = 1\%$ relative noise. Though data are only shown here at select spatial locations, they are observed at every point in the spatial discretization.

5.1.2 ROM structure

Before applying Algorithm 4.3, we apply a variable transformation to cast (5.2a) in a polynomial form that is suitable for OpInf [52]. Using the ideal gas law (5.2b), changing from the conservative variables $(\rho, \rho v, \rho e)$ to the specific volume variables $(v, p, 1/\rho)$ results in a quadratic system,

$$\frac{\partial}{\partial t} \begin{bmatrix} v \\ p \\ \zeta \end{bmatrix} = \begin{bmatrix} -v \frac{\partial v}{\partial x} - \zeta \frac{\partial p}{\partial x} \\ -\gamma p \frac{\partial v}{\partial x} - v \frac{\partial p}{\partial x} \\ -v \frac{\partial \zeta}{\partial x} + \zeta \frac{\partial v}{\partial x} \end{bmatrix}, \quad (5.5)$$

where $\zeta = 1/\rho$ is the specific volume. Note that the variables (v, p, ζ) obey periodic boundary conditions as in (5.2c). For each $j = 0, \dots, m-1$, the noisy conservative-variable snapshot $\mathbf{q}_j^{(c)}$ is transformed to the specific volume variables and nondimensionalized to produce the snapshot $\mathbf{q}_j \in \mathbb{R}^N$. After spatial discretization, the system (5.5) can be written in the quadratic form

$$\frac{d}{dt} \mathbf{q}(t) = \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)], \quad (5.6)$$

where $\mathbf{H} \in \mathbb{R}^{N \times N^2}$ and $\mathbf{q}(t) \in \mathbb{R}^N$ is the spatial discretization of the lifted variables (v, p, ζ) at time t . Using a centered-POD approximation,

$$\mathbf{q}(t) \approx \mathbf{V} \hat{\mathbf{q}}(t) + \bar{\mathbf{q}}, \quad \bar{\mathbf{q}} = \frac{1}{m} \sum_{j=0}^{m-1} \mathbf{q}_j \in \mathbb{R}^N, \quad (5.7)$$

where $\mathbf{V} \in \mathbb{R}^{N \times r}$ has orthonormal columns and $\hat{\mathbf{q}}(t) \in \mathbb{R}^r$, we obtain a ROM

$$\frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{c}} + \hat{\mathbf{A}} \hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)], \quad (5.8)$$

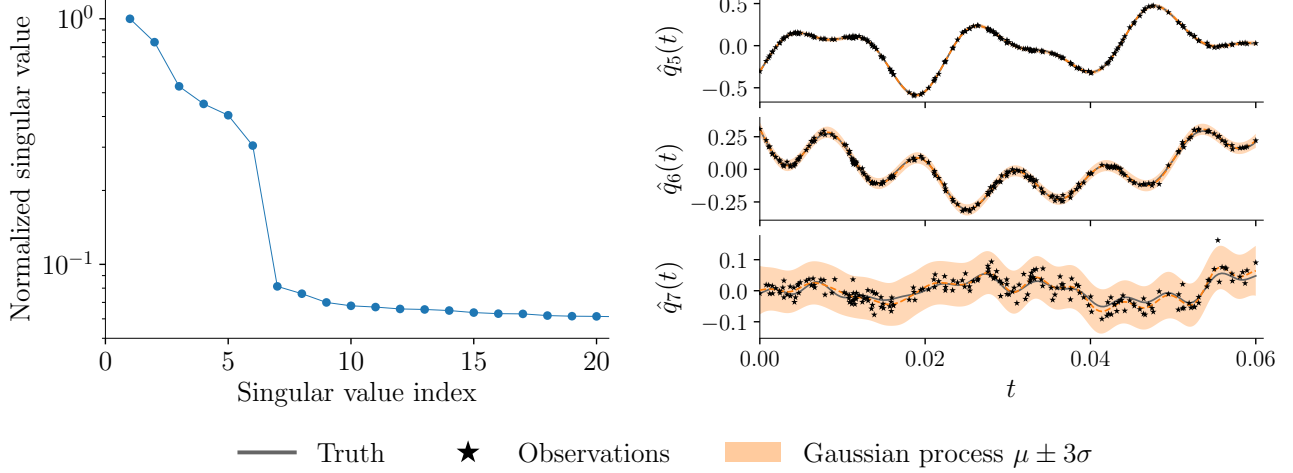


Figure 3: Singular value decay of $m = 200$ snapshots with $\xi = 3\%$ relative noise (left) and trained GPs for reduced modes 5, 6, and 7, compressed via centered POD (right). The shaded regions are within three standard deviations of each GP mean. The POD compression filters much of the noise in the dominant modes, hence each GP exhibits a tight fit to the data until the spectral gap after $r = 6$ modes.

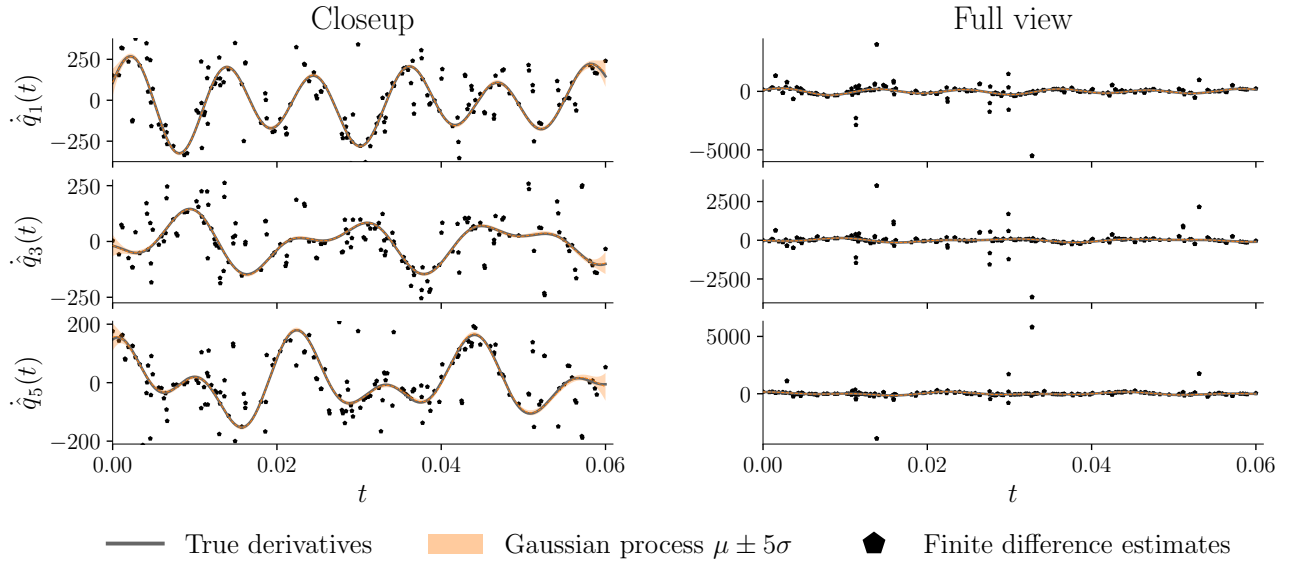


Figure 4: True reduced state time derivatives, as well as GP and finite difference estimates of the time derivatives, using $m = 200$ observations with $\xi = 3\%$ relative noise. The time derivative GPs have small variances and produce highly accurate estimates (left), unlike the finite difference estimates, which in some instances differ from the truth by an order of magnitude (right).

which is (2.5) with $\hat{\mathbf{O}} = [\hat{\mathbf{c}} \ \hat{\mathbf{A}} \ \hat{\mathbf{H}}] \in \mathbb{R}^{r \times d(r)}$, $\mathbf{d}(\hat{\mathbf{q}}) = [1 \ \hat{\mathbf{q}}^\top \ (\hat{\mathbf{q}} \otimes \hat{\mathbf{q}})^\top]^\top \in \mathbb{R}^{d(r)}$, and $d(r) = 1 + r + r^2$. The nonquadratic terms $\hat{\mathbf{c}} \in \mathbb{R}^r$ and $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$ appear due to the centering term $\bar{\mathbf{q}}$ [67]. Because the Kronecker product $\hat{\mathbf{q}} \otimes \hat{\mathbf{q}} = \text{vec}(\hat{\mathbf{q}}\hat{\mathbf{q}}^\top) \in \mathbb{R}^{r^2}$ has only $r(r+1)/2 < r^2$ degrees of freedom due to symmetry, in practice the quadratic operator $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}$ can be fully represented by an $r \times r(r+1)/2$ matrix that acts on a compressed version of the Kronecker product (see, e.g., Appendix B of [41]).

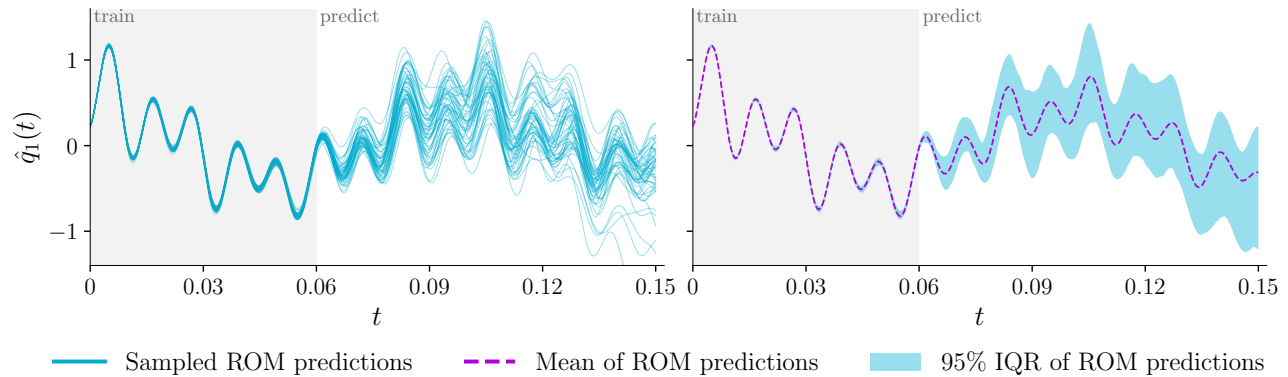


Figure 5: The dominant mode of the Bayesian ROM predictions for the Euler system (5.2a)–(5.2c) using $m = 200$ observations over the time domain $[0, 0.06]$ with $\xi = 3\%$ relative noise. Fifty samples are taken of the posterior operator distribution and the corresponding ROM predictions are displayed as a function of time (left). The uncertainty of the prediction distribution is characterized by computing the mean and the 95% interquartile range (IQR) of the sample predictions at each point in time (right).

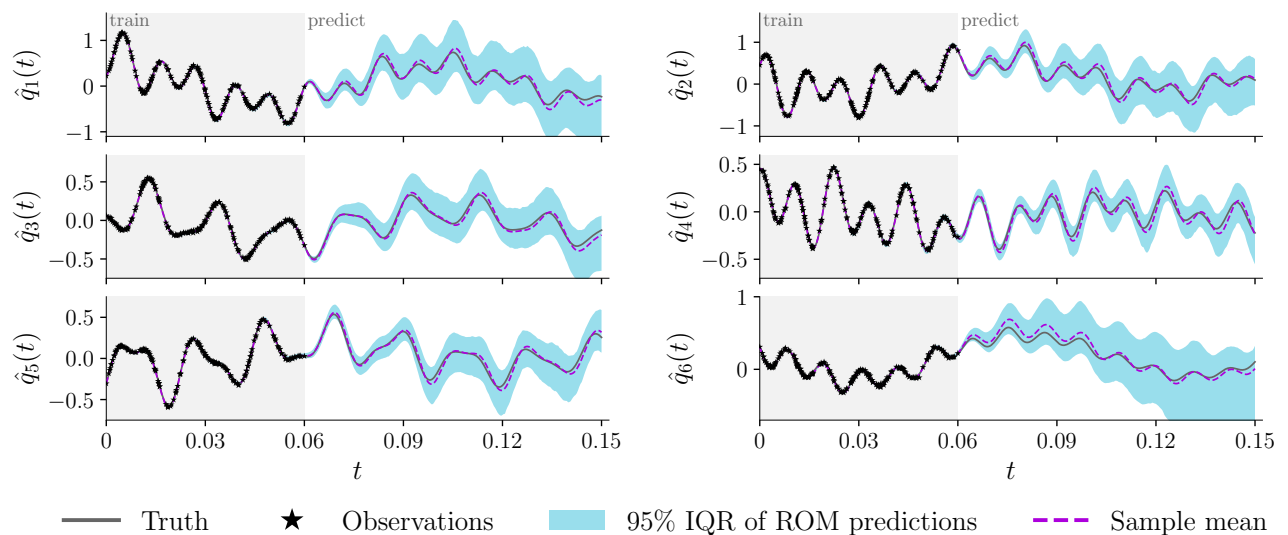


Figure 6: Bayesian ROM predictions for the Euler system (5.2a)–(5.2c) using $m = 200$ observations over the time domain $[0, 0.06]$ with $\xi = 3\%$ relative noise. The solid line shows the true numerical solution to the system without noise, expressed in the POD coordinates. The posterior operator distribution is sampled 500 times. The pointwise mean of the sample predictions gives an accurate estimate of the true solution, with uncertainty described by the 95% interquartile range.

5.1.3 Noisy training data

We consider first a scenario in which the observed data are relatively abundant but corrupted with significant noise. Let $m = 200$ with final observation time $t_{m-1} = 0.06$ s and relative noise level $\xi = 3\%$. This is a nontrivial amount of noise for this problem, especially given that the noise is applied to the conservative variables but the states are observed after a variable transformation. Our goal is to predict, with uncertainty, solutions of (5.2a)–(5.2c) from $t_0 = 0$ to the final time $t_f = 0.15$ s.

Figure 3 shows the POD singular value decay of the snapshot data, which decreases sharply and levels out after 6 modes. Compressed snapshot data for mode 7 and greater concentrate near zero and do not result in a tight GP fit, hence we set the ROM dimension to $r = 6$. The GP estimation times \mathbf{t}' are defined to be $m' = 400$ uniformly spaced times in the interval $[t_0, t_{m-1}]$. To further validate the quality of the training data produced by each GP regression, we solve (5.2) on a fine time grid and explicitly evaluate the right-hand side of (5.2a)

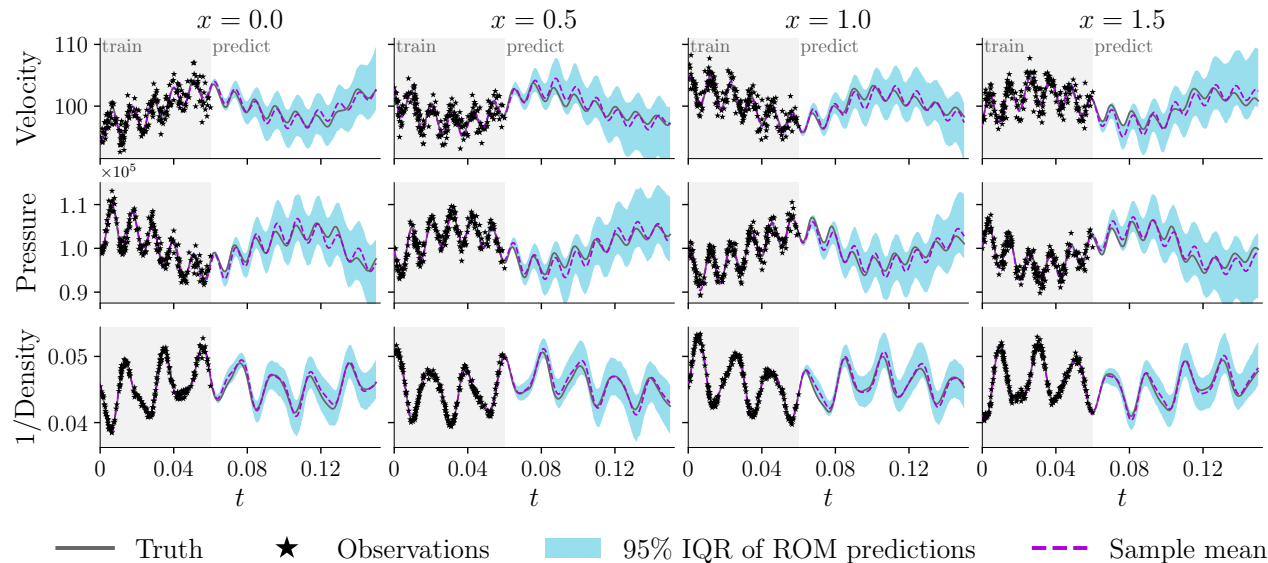


Figure 7: Bayesian ROM predictions of Figure 6 expressed in the specific volume variables ($v, p, 1/\rho$). The solid line shows the true numerical solution to the system without noise. Results are shown at four spatial locations.

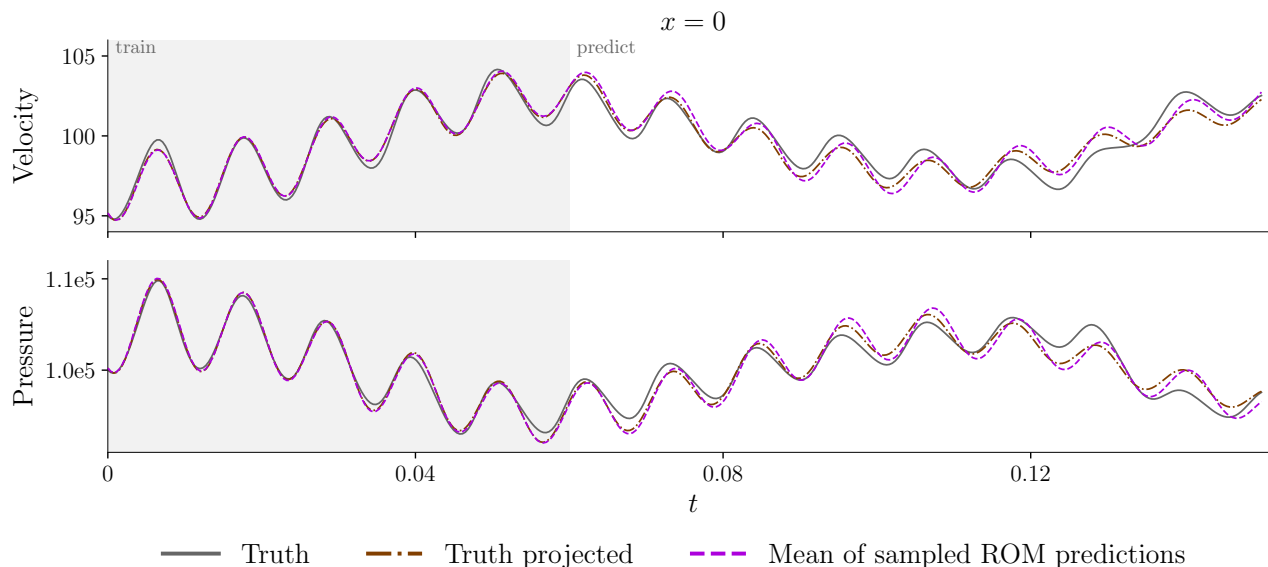


Figure 8: Closeup of the results in Figure 7 at $x = 0$, as well as the projection of the true numerical solution. In the training regime, the mean of the sampled ROM predictions tightly match the projection of the truth, a consequence of Algorithm 4.2. In the prediction regime, when the sample mean deviates from the truth it is generally close to the projection of the truth.

(without adding noise) to obtain time derivatives ($\partial_t \rho, \partial_t[\rho v], \partial_t[\rho e]$), which are then transformed to the specific-volume formulation ($\partial_t v, \partial_t p, \partial_t \xi$) and compressed using the POD basis. Figure 4 plots these true time derivatives together with the GP estimates for the time derivatives which, over the time grid \mathbf{t}' and for the i -th reduced state, are given by the mean vector $\tilde{\mathbf{z}}_i = \mathbf{K}_i^{zy}(\mathbf{K}_i^{yy})^{-1}\mathbf{y}_i$ with covariance $(\mathbf{W}_i^{zz})^{-1} = \mathbf{K}_i^{zz} - \mathbf{K}_i^{zy}(\mathbf{K}_i^{yy})^{-1}\mathbf{K}_i^{yz}$. The GP estimates are highly faithful to the true time derivatives, and the associated variances so small that the region within 10 standard deviations of the mean is only slightly visible. As a comparison, we also apply second-order finite differences to the noisy compressed snapshots to produce time derivative estimates, the standard practice for OpInf problems. Although the POD basis filters much of the noise from the high-dimensional observations

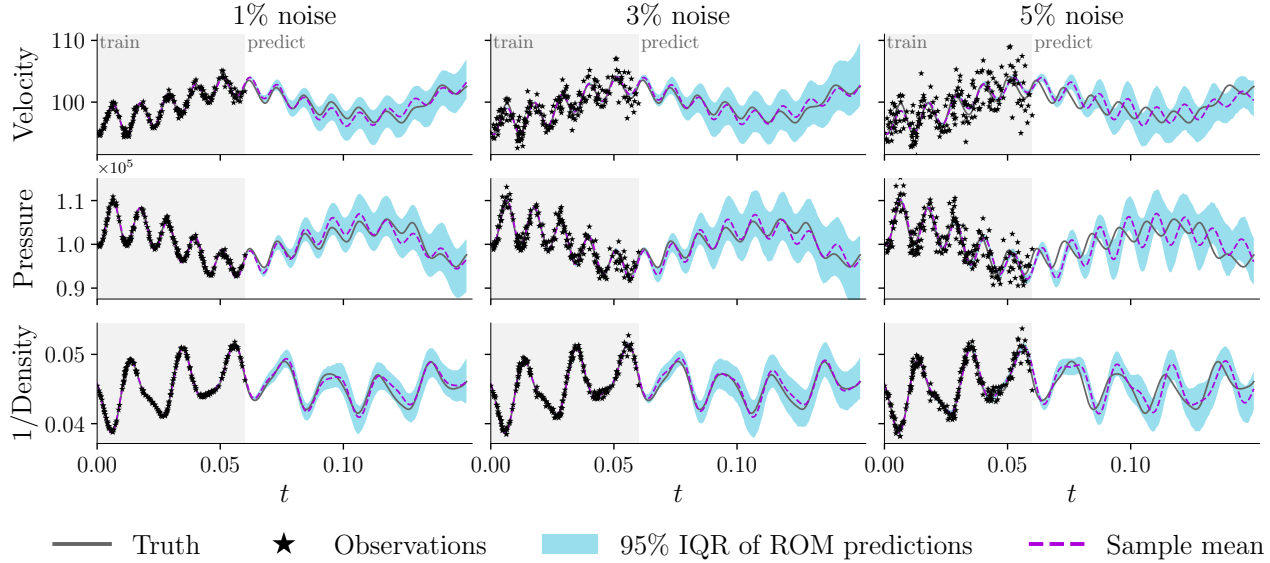


Figure 9: Pointwise Bayesian ROM predictions of the specific volume variables at $x = 0$ for the Euler system (5.2a)–(5.2c) using $m = 200$ observations over the time domain $[0, 0.06]$ with varying noise levels.

(see Figure 3), the remaining noise results in significant error in the finite difference estimates, rendering them unsuitable for populating the OpInf regression (2.7).

After applying Algorithm 4.3 to obtain a posterior distribution for the reduced operators $\hat{\mathbf{O}}$, we sample from the posterior 500 times and solve the corresponding ROMs (5.8) using the same Runge–Kutta scheme as before. The uncertainty stemming from the data noise propagates to the ROM solutions and forms a prediction distribution. To characterize the distribution, we compute the mean and the 2.5% and 97.5% quantiles of the samples at each time instance, see Figure 5. The individual draws are nearly identical in the training regime, but at the start of the prediction regime the width of the distribution increases noticeably and thereafter generally continues to increase in time at a relatively slow rate.

The results are compared to the true solution in Figure 6 for the reduced modes and in Figure 7 for the original state space in the (physically relevant) specific volume variables. In all variables and throughout the spatial domain, each ROM solution closely matches the true solution during the training regime, which is expected since Algorithm 4.3 minimizes the error of the sample mean and the GP estimates of the reduced training data. Although the ROM solutions are not accurate individually in the prediction regime, in aggregate they form an accurate estimate of the projection of the true solution. Figure 8 shows a closeup of results from Figure 7 at a single spatial location and includes the true solution $\mathbf{q}(t)$, the projection $\tilde{\mathbf{q}}(t) = \mathbf{V}\mathbf{V}^T(\mathbf{q}(t) - \bar{\mathbf{q}}) + \bar{\mathbf{q}}$, and the pointwise mean of the sampled ROM predictions. The error between the truth $\mathbf{q}(t)$ and the projection $\tilde{\mathbf{q}}(t)$ is due entirely to the low-dimensional state approximation (5.7), not the GP modeling or the OpInf regression. The sample mean is about as accurate as the projection of the truth, hence, in this experiment, the error in the sample mean is dominated by the fundamental limitations incurred by dimensionality reduction.

Figure 9 repeats the numerical experiment with $\xi = 1\%$ and $\xi = 5\%$ relative noise and reports the results at a single point in space. As the noise level ξ increases, so does the width of the distribution of predicted ROM solutions. The sample mean of the ROM predictions estimates the truth well for $\xi = 1\%$ and $\xi = 3\%$, but with $\xi = 5\%$ the ROM solutions are off phase from the truth in the prediction regime. In the latter case, the large amount of noise causes the ROM solutions in the training regime (and, though not shown, the projection of the truth) to be less accurate, and at times the true solution is not well represented by the prediction distribution. However, the solutions still roughly capture the trends of the system behavior, especially in the specific volume.

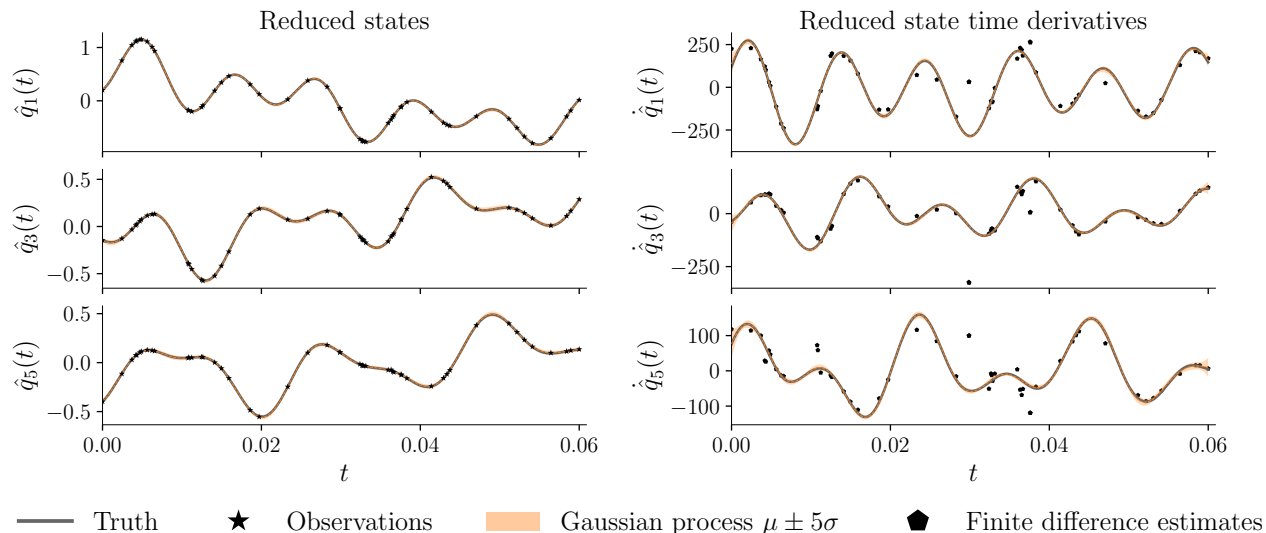


Figure 10: True reduced states, observations, and trained GPs for reduced modes 1, 3, and 5 (left), and corresponding reduced state time derivatives, GP estimates, and finite difference estimates (right).

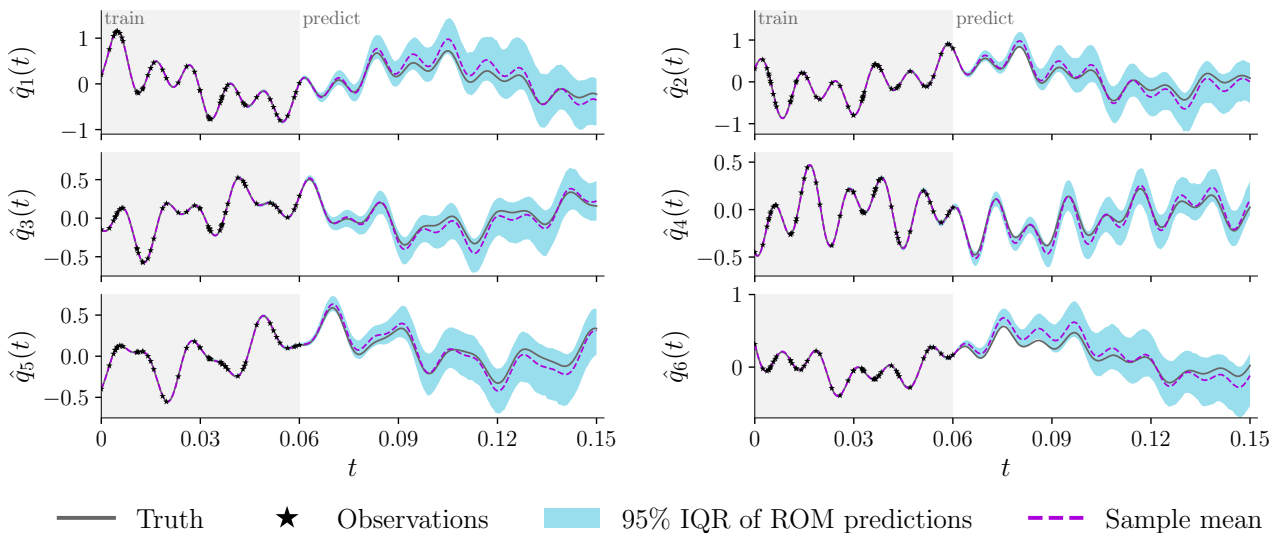


Figure 11: Bayesian ROM predictions for the Euler system (5.2a)–(5.2c) using $m = 50$ observations over the time domain $[0, 0.06]$ with $\xi = 1\%$ relative noise. The solid line shows the true numerical solution to the system without noise, expressed in the POD coordinates. The posterior is sampled 500 times. The pointwise mean of the sample predictions gives an accurate estimate of the true solution, with uncertainty described by the 95% interquartile range.

5.1.4 Sparse training data

Next, consider a scenario in which the observed data are sparse in time. Let $m = 50$, again with final observation time $t_{m-1} = 0.06$ s but now with only $\xi = 1\%$ relative noise. This is the scenario depicted in Figure 2. Figure 10 shows GP regressions of the compressed state data and predictions for the associated time derivatives. In this case, finite differences of the noisy snapshots result in estimates for the time derivatives that are much less unwieldy than in the previous experiment (see Figure 4), but they are still significantly less accurate than the GP time derivative estimates. Additionally, the GPs yield accurate training data for both the reduced state and the time derivative over the entire training time regime, not just at the sparse observation times.

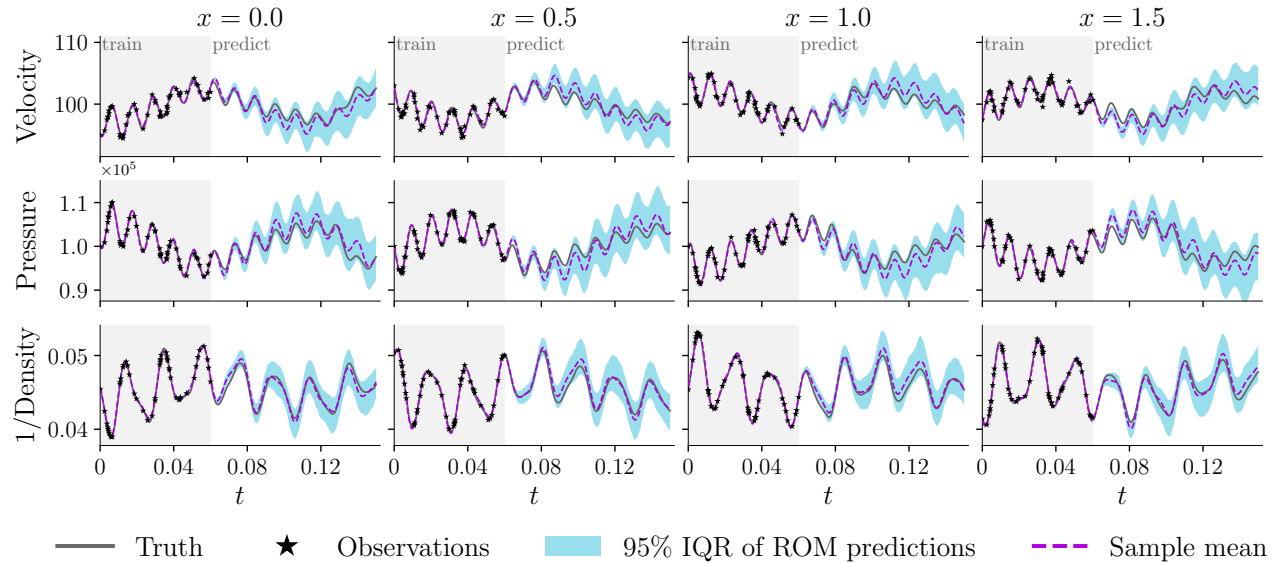


Figure 12: Bayesian ROM predictions of Figure 11 expressed in the specific volume variables ($v, p, 1/\rho$) at four spatial locations.

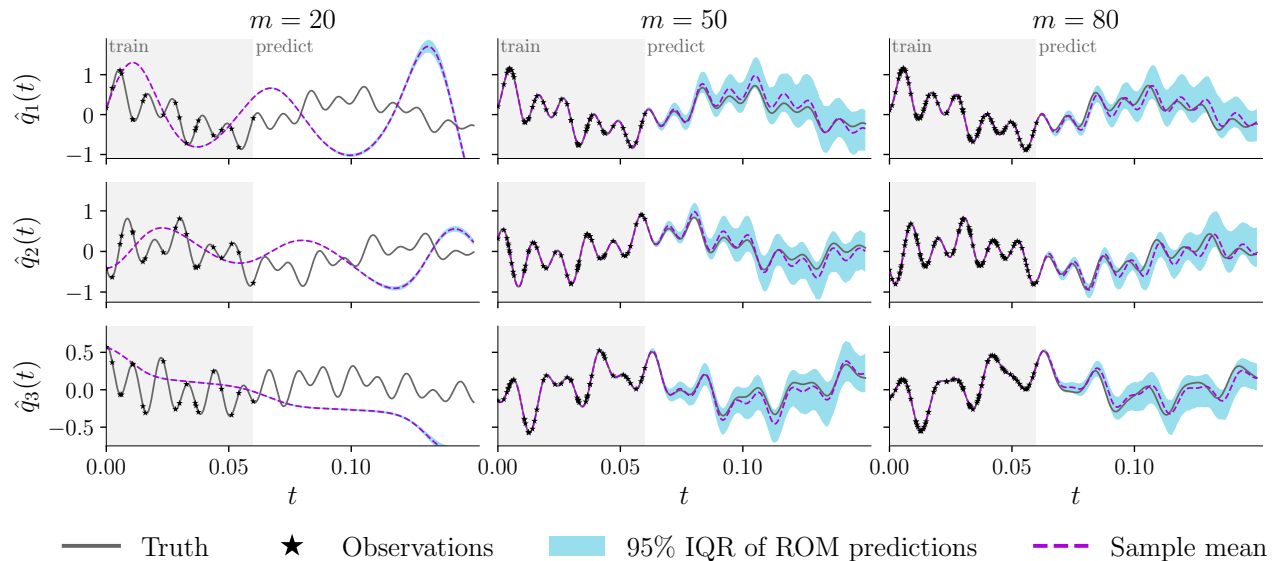


Figure 13: Bayesian ROM predictions for the Euler system (5.2a)–(5.2c) using a varied number of observations over the time domain $[0, 0.06]$ with $\xi = 1\%$ relative noise. With only $m = 20$ observations Algorithm 4.3 fails to produce an accurate model.

Results with 500 posterior samples are shown in Figure 11 in the reduced space and Figure 12 in the physical state space. The results are similar in quality to the previous experiments: the truth is reconstructed accurately in the training regime and the sample mean of the ROM predictions is a reasonable estimate of the truth in the prediction regime. To further probe the performance of our method on sparse data, we repeat the experiment with only $m = 20$ observations and again with $m = 80$ observations, all drawn uniformly over the same training time domain. Figure 13 displays the results at a single spatial location. With only $m = 20$ data points, GP-BayesOpInf fails to produce accurate solutions and also does not produce a wide prediction distribution. This is partially due to not having sufficient data to perform a good GP fit, but also because Algorithm 4.2 prioritizes stable but inaccurate ROM solutions over ROM solutions that are accurate for a

limited time but which become unstable before the final simulation time. On the other hand, using $m = 80$ training points produces a slightly more accurate sample mean and tighter prediction distribution than with $m = 50$, i.e., more training data leads to improved accuracy.

5.2 Nonlinear diffusion-reaction equation

In this section, we use GP-BayesOpInf to construct probabilistic surrogates for a nonlinear diffusion-reaction equation using data from multiple trajectories. Each trajectory uses the same initial conditions but a different instance of a parameterized nonlinear source term. The efficacy of the probabilistic surrogate is shown for prediction in both time and in parameters defining the source term with noisy, sparse training data.

5.2.1 Governing equations

Consider the following forced heat diffusion model with a cubic reaction term,

$$\begin{aligned} \frac{\partial q}{\partial t} &= \mu \frac{\partial^2 q}{\partial x^2} - q^3 + u, & x \in (0, 1), t \in (0, T], \\ q(0, t) &= 0, \quad q(1, t) = 1, & t \in (0, T], \\ q(x, 0) &= q_0(x), & x \in [0, 1], \end{aligned} \tag{5.9a}$$

where $\mu = 0.005$ is the diffusion constant and where the source term u , parameterized by constants $a, b \in \mathbb{R}$, is given by

$$u(x, t; a, b) = \frac{a \sin(2\pi t)}{1 + 100 \left(x - \frac{1}{4}\right)^2} + \frac{b \sin(4\pi t)}{1 + 100 \left(x - \frac{3}{4}\right)^2}. \tag{5.9b}$$

The initial conditions are defined to be

$$q_0(x) = x(1-x) \left(6(1-x)^2 e^{-x} - 10e^x \sin(x/6)\right) + x, \tag{5.9c}$$

a nonpolynomial function satisfying the boundary conditions. In the following experiments, we use training data from $\ell = 5$ trajectories obtained by solving (5.9a)–(5.9c) for the (a, b) pairs $(-2, 0)$, $(-1, -2)$, $(0, 1)$, $(1, -1)$, and $(2, 2)$, which form a coarse Latin square sampling of $[-2, 2] \times [-2, 2]$.

5.2.2 ROM structure

To choose a ROM structure for (5.9a), we introduce the auxiliary variable $w = q^2$ to expose quadratic dynamics [53]. As $\partial_t w = 2qq_t$, (5.9a) can be written equivalently as the PDE system

$$\frac{\partial q}{\partial t} = \mu \frac{\partial^2 q}{\partial x^2} - qw + u, \quad \frac{\partial w}{\partial t} = 2\mu q \frac{\partial^2 q}{\partial x^2} - w^2 + qu. \tag{5.10}$$

Let $\mathbf{q}(t) = [\mathbf{s}(t)^\top \ (\mathbf{s}(t) * \mathbf{s}(t))^\top]^\top \in \mathbb{R}^N$ be the spatial discretization of (q, w) , where $N = 2n_x = 1000$ and $*$ is the Hadamard (element-wise) product. The discretization of (5.10) is a system of ODEs that can be written as

$$\frac{d}{dt} \mathbf{q}(t) = \mathbf{A} \mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)] + \mathbf{B} \mathbf{u}(t) + \mathbf{N}[\mathbf{u}(t) \otimes \mathbf{q}(t)], \tag{5.11}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{H} \in \mathbb{R}^{N \times N^2}$, $\mathbf{B} \in \mathbb{R}^{N \times 2}$, $\mathbf{N} \in \mathbb{R}^{N \times 2N}$, and $\mathbf{u}(t; a, b) = (a \sin(2\pi t), b \sin(4\pi t))^\top \in \mathbb{R}^2$ gathers the time-dependent components of the source term u . Using a centered-POD approximation $\mathbf{q}(t) \approx \mathbf{V} \hat{\mathbf{q}}(t) + \hat{\mathbf{q}}$ as in (5.7), we obtain the ROM structure

$$\frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{c}} + \hat{\mathbf{A}} \hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] + \hat{\mathbf{B}} \mathbf{u}(t) + \hat{\mathbf{N}}[\mathbf{u}(t) \otimes \hat{\mathbf{q}}(t)], \tag{5.12}$$

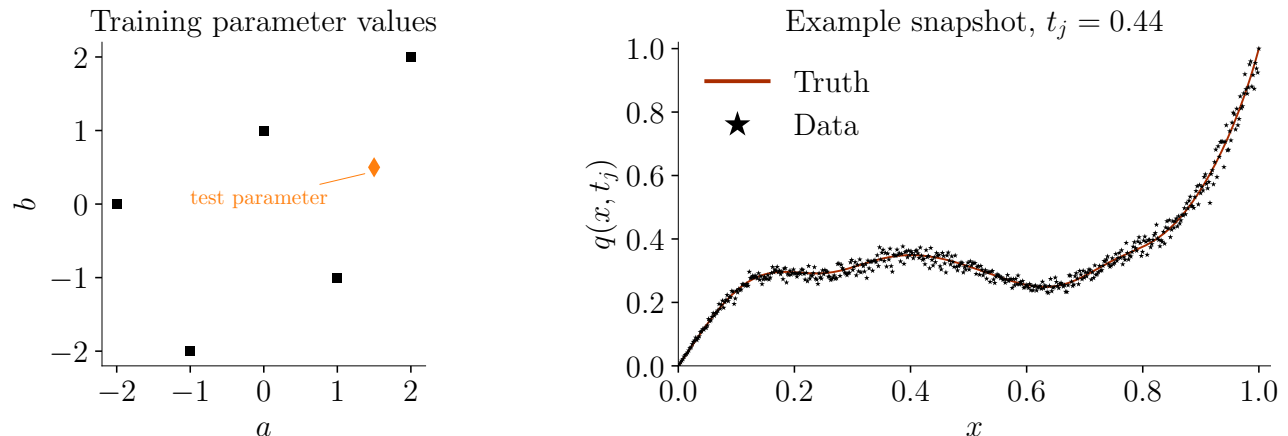


Figure 14: The five sets of input parameter values (a, b) parameterizing the input function $u(x, t; a, b)$ of (5.9b) used in generating training data (left) and an example snapshot with $\xi = 5\%$ noise for $a = 1$, $b = -2$ (right).

where $\hat{\mathbf{q}}(t) \in \mathbb{R}$, $\hat{\mathbf{c}} \in \mathbb{R}^r$, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r \times 2}$, and $\hat{\mathbf{N}} \in \mathbb{R}^{r \times 2r}$. This system is (2.5) with operator matrix $\hat{\mathbf{O}} = [\hat{\mathbf{c}} \ \hat{\mathbf{A}} \ \hat{\mathbf{H}} \ \hat{\mathbf{B}} \ \hat{\mathbf{N}}] \in \mathbb{R}^{r \times d(r,p)}$, data vector $\mathbf{d}(\hat{\mathbf{q}}, \mathbf{u}) = [1 \ \hat{\mathbf{q}}^\top \ (\hat{\mathbf{q}} \otimes \hat{\mathbf{q}})^\top \ \mathbf{u}^\top \ (\mathbf{u} \otimes \hat{\mathbf{q}})^\top]^\top \in \mathbb{R}^{d(r,p)}$, and dimension $d(r, p) = 1 + r + r^2 + p + rp$. The constant term $\hat{\mathbf{c}}$ arises due to the centering with $\bar{\mathbf{q}}$. Having selected this structure, the snapshots to be used in the procedure are $\mathbf{q}_j = [\mathbf{s}_j^\top \ (\mathbf{s}_j * \mathbf{s}_j)^\top]^\top \in \mathbb{R}^N$. Because noise is not applied at the spatial boundary, the centered snapshots $\mathbf{q}_j - \bar{\mathbf{q}}$ obey homogeneous Dirichlet boundary conditions, hence all ROM predictions $\mathbf{V}\hat{\mathbf{q}}(t) + \bar{\mathbf{q}}$ automatically satisfy the boundary conditions of (5.9a).

5.2.3 Data generation

Data for this experiment are generated by using second-order central finite differences to discretize the spatial derivative in (5.9a) with $n_x = 500$ degrees of freedom, then time stepping the system with an implicit multi-step backward difference method of variable order (1 to 5) with quasi-constant step size [10, 60]. In this experiment, we observe only $m = 20$ snapshots for each choice of (a, b) , totaling 100 snapshots, with different sampling times \mathbf{t} for each trajectory. In each case, we set $t_0 = 0$, $t_{m-1} = 1$, and sample the remaining points (t_1, \dots, t_{m-2}) from the uniform distribution over $(0, t_{m-1})$. The snapshots are corrupted with additive noise whose variance is determined by the magnitude of the state at each point in space, i.e.,

$$\mathbf{s}_j = \mathbf{s}(t_j) + \boldsymbol{\varepsilon}_j \in \mathbb{R}^{n_x}, \quad \boldsymbol{\varepsilon}_j \sim \mathcal{N}(\mathbf{0}, \text{diag}(\xi \mathbf{s}(t_j))^2), \quad (5.13)$$

where $\mathbf{s}(t_j)$ is the numerical solution for q at time $t_j \in \mathbf{t}$ and $\xi = 5\%$ is the noise level. Noise is not added at $t = t_0$ or to the spatial boundaries since the initial conditions and Dirichlet boundary conditions are given. Figure 14 visualizes the (a, b) parameter space and plots an example noised snapshot in space.

5.2.4 Prediction at new input parameters

We apply Algorithm B.2 (the analog of Algorithm 4.3 for multiple trajectories) with $m' = 4m = 80$ GP estimates to obtain a posterior distribution for $\hat{\mathbf{O}}$ with $r = 5$ POD modes. We then sample the posterior 500 times and solve the corresponding ROM (5.12) with the same backwards-differences time stepper as the full-order model. For three choices of the input parameters (a, b) , Figure 15 shows results in the reduced space, while Figure 16 shows results in the original state space at select spatial locations. The accuracy of ROM solutions varies across the inputs because some trajectories are more easily captured by the low-dimensional state approximation than others. Note that the width of the prediction distribution increases whenever the sample mean is less accurate.

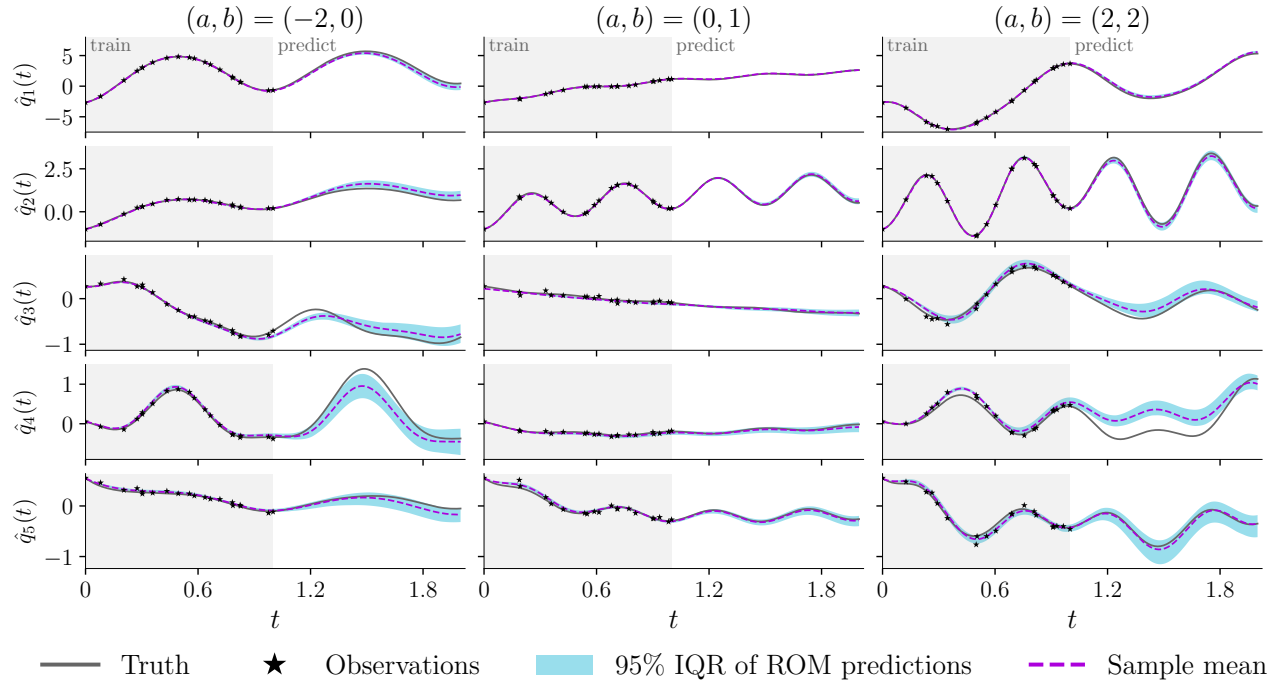


Figure 15: Bayesian ROM predictions for the nonlinear diffusion-reaction problem (5.9a)–(5.9c) for three values of the input parameters (a, b) . For each set of input parameters, data are observed at $m = 20$ points over the time domain $[0, 1]$ with $\xi = 5\%$ relative noise. Five hundred posterior samples are used to generate ROM solutions, and the resulting distribution is characterized by the pointwise sample mean and 95% interquartile range.

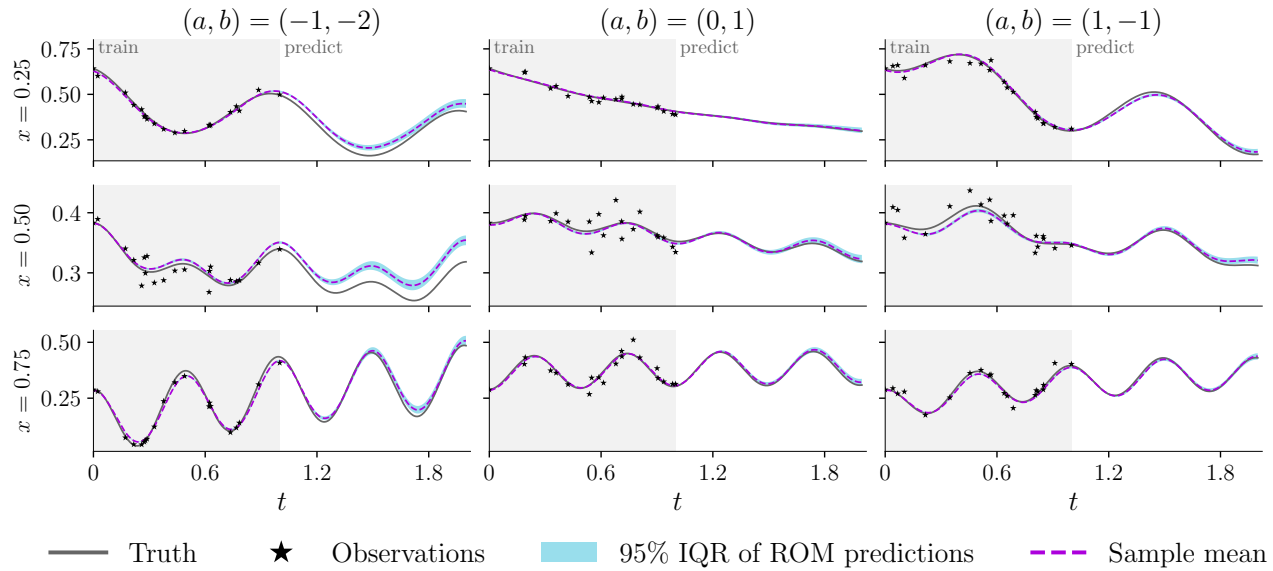


Figure 16: Bayesian ROM predictions of Figure 15 expressed in the original state variable q for three values of the input parameters (a, b) . Results are shown at three spatial locations.

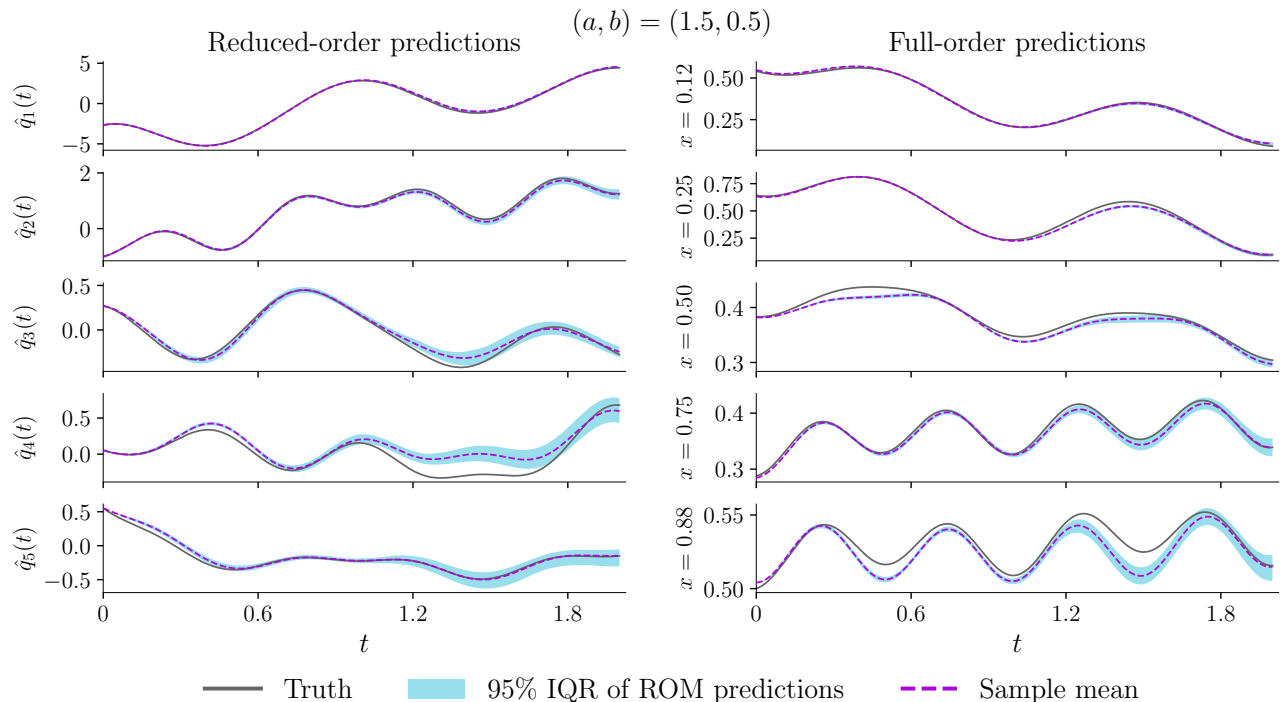


Figure 17: Bayesian ROM predictions for the nonlinear diffusion-reaction problem (5.9a)–(5.9c) for new input parameter values $(a, b) = (1.5, 0.5)$ not used in the training set, both in the reduced space (left) and pointwise in the original state space (right).

It is worth noting that, in this example, aggregating a few samples each from multiple trajectories tends to result in a more diverse and informative POD basis than when using many samples from a single trajectory, which is why the results are high quality even with the relatively small $m = 20$. This diversity is also helpful for generalization: Figure 17 shows results for posterior ROM samples using the new input parameter values $(a, b) = (1.5, 0.5)$, i.e., using an input function $\mathbf{u}(t)$ that was not part of the training set. This represents a prediction in the (a, b) input parameter space as well as extrapolation in time. As in previous experiments, the width of the prediction distribution increases in time, but here the transition from training regime to prediction regime is less pronounced.

6 GP-based parameter estimation for ODEs

The GP-BayesOpInf framework can be applied, with a minor adjustment, to a wide class of systems of ODEs that are not the result of spatially discretizing a PDE. Our approach allows for parameters to appear in multiple equations, treats all equations jointly, and leverages the prior variance selection framework developed in Section 4.

6.1 Derivation

We consider dynamical systems modeled by ODEs that may be nonlinear in state but which are linear in a set of scalar parameters, i.e.,

$$\frac{d}{dt} \hat{\mathbf{q}}(t) = \mathbf{S}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) \hat{\boldsymbol{\theta}}, \quad (6.1)$$

where $\hat{\mathbf{q}}(t) \in \mathbb{R}^r$ and $\mathbf{u}(t) \in \mathbb{R}^p$ are the state and the input (as in the PDE case), $\mathbf{S} : \mathbb{R}^r \times \mathbb{R}^p \rightarrow \mathbb{R}^{r \times d(r,p)}$ describes the (potentially nonlinear) system dynamics, and $\hat{\mathbf{o}} \in \mathbb{R}^{d(r,p)}$ are the system parameters. We use $\hat{\mathbf{q}}(t)$ to be consistent with previous notation, but in this setting the state $\hat{\mathbf{q}}(t)$ is *not* the result of compressing a high-dimensional state variable. Our objective is to construct a probability distribution for the parameter vector $\hat{\mathbf{o}}$ given (noisy) observations of the state and the corresponding inputs.

If data are available for $\hat{\mathbf{q}}(t)$, a Bayesian inference for $\hat{\mathbf{o}}$ can be formulated similar to (3.8). Let $\mathbf{t}_i \in \mathbb{R}^{m_i}$ denote the $m_i \in \mathbb{N}$ times at which the i -th component of $\hat{\mathbf{q}}(t)$ is observed as the (possibly noisy) vector $\mathbf{y}_i \in \mathbb{R}^{m_i}$, $i = 1, \dots, r$, and define the concatenated state data vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_r \end{bmatrix} \in \mathbb{R}^m, \quad m = \sum_{i=1}^r m_i. \quad (6.2)$$

As in (3.1), let $\hat{q}_i(t)$ denote a GP for the i -th component of $\hat{\mathbf{q}}(t)$. Define \mathbf{Y} to be the random m -vector whose entries are Gaussian random variables produced by evaluating \hat{q}_i over \mathbf{t}_i for each $i = 1, \dots, r$ and concatenating the results. Hence, \mathbf{y} is an observation of \mathbf{Y} . Similarly, for estimation times $\mathbf{t}'_i \in \mathbb{R}^{m'}$, let \mathbf{Z} be the random vector with rm' entries given by concatenating the time derivative GPs $\dot{\hat{q}}_i$ evaluated at \mathbf{t}' , $i = 1, \dots, r$. Finally, let $\tilde{\mathbf{D}} \in \mathbb{R}^{rm' \times d(r,p)}$ contain state and input data at the estimation times \mathbf{t}' following the structure of \mathbf{S} and define $\mathbf{R} = \mathbf{Z} - \tilde{\mathbf{D}}\hat{\mathbf{o}}$. We then have the parameter vector inference

$$\begin{aligned} \pi_{\text{post}}(\hat{\mathbf{o}}) &= p(\hat{\mathbf{o}} \mid \mathbf{Y} = \mathbf{y}, \mathbf{R} = \mathbf{0}, \tilde{\mathbf{D}}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_r, \boldsymbol{\gamma}) \propto \pi_{\text{prior}}(\hat{\mathbf{o}})\pi_{\text{like}}(\hat{\mathbf{o}}), \quad \text{with} \\ \pi_{\text{prior}}(\hat{\mathbf{o}}) &= p(\hat{\mathbf{o}} \mid \boldsymbol{\gamma}) \propto \exp\left(-\frac{1}{2}\|\text{diag}(\boldsymbol{\gamma})\hat{\mathbf{o}}\|_2^2\right), \\ \pi_{\text{like}}(\hat{\mathbf{o}}) &= p(\mathbf{Y} = \mathbf{y}, \mathbf{R} = \mathbf{0} \mid \hat{\mathbf{o}}, \tilde{\mathbf{D}}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_r), \end{aligned} \quad (6.3)$$

where, as before, $\boldsymbol{\theta}_i$ denotes the hyperparameters of the GP \hat{q}_i . This likelihood aligns with the gradient matching likelihood used in computational statistics [11, 68, 73].

The mean and covariance of the posterior $\hat{\mathbf{o}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are given by

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\tilde{\mathbf{D}}^\top\mathbf{W}\tilde{\mathbf{z}}, \quad \boldsymbol{\Sigma} = \left(\tilde{\mathbf{D}}^\top\mathbf{W}\tilde{\mathbf{D}} + \boldsymbol{\Gamma}^\top\boldsymbol{\Gamma}\right)^{-1}, \quad (6.4a)$$

in which

$$\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{\mathbf{z}}_1 \\ \vdots \\ \tilde{\mathbf{z}}_r \end{bmatrix} \in \mathbb{R}^{rm'}, \quad \mathbf{W} = \text{diag}(\mathbf{W}_1^{zz}, \dots, \mathbf{W}_r^{zz}) \in \mathbb{R}^{rm' \times rm'}, \quad \boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}) \in \mathbb{R}^{d(r,p) \times d(r,p)}, \quad (6.4b)$$

where $\mathbf{W}_i^{zz} \in \mathbb{R}^{m' \times m'}$ and $\tilde{\mathbf{z}}_i \in \mathbb{R}^{m'}$ are defined by the GP \hat{q}_i as in (3.9) and (3.13), respectively. The generalized least-squares problem (3.14) for the posterior mean $\boldsymbol{\mu}$ is

$$\min_{\boldsymbol{\eta}} \left\| \tilde{\mathbf{D}}\boldsymbol{\eta} - \tilde{\mathbf{z}} \right\|_{\mathbf{W}}^2 + \|\boldsymbol{\Gamma}\boldsymbol{\xi}\|_2^2. \quad (6.5)$$

6.2 SEIRD epidemiological model

In this section, we consider a system of five ODEs with five scalar parameters that defines the SEIRD compartmental model for infectious disease spreading through a population [37]:

$$\begin{aligned} \frac{d}{dt}\hat{q}_S(t) &= -\beta\hat{q}_S(t)\hat{q}_I(t), & \frac{d}{dt}\hat{q}_R(t) &= (1-\alpha)\gamma\hat{q}_I(t), \\ \frac{d}{dt}\hat{q}_E(t) &= \beta\hat{q}_S(t)\hat{q}_I(t) - \delta\hat{q}_E(t), & \frac{d}{dt}\hat{q}_D(t) &= \alpha\rho\hat{q}_I(t), \\ \frac{d}{dt}\hat{q}_I(t) &= \delta\hat{q}_E(t) - (1-\alpha)\gamma\hat{q}_I(t) - \alpha\rho\hat{q}_I(t). \end{aligned} \quad (6.6)$$

The state variables represent the proportions of the population that are susceptible to the disease ($\hat{q}_S(t)$), exposed but not yet infected ($\hat{q}_E(t)$), currently infected ($\hat{q}_I(t)$), recovered (and immune) post-infection ($\hat{q}_R(t)$), and deceased post-infection ($\hat{q}_D(t)$). The system parameters describe the rates at which individuals transition from one state to another: $\beta > 0$ is the expected number of people an infected person exposes to the disease per day, $\delta > 0$ is the incubation period for transition from exposed to infected, $\gamma > 0$ is the recovery rate of an infected individual, $\alpha > 0$ is the infection fatality rate, and $\rho > 0$ is the inverse of the average number of days for an infected person to die if they do not recover. Our goal is to give a probabilistic estimate of the system parameters given noisy and/or sparse state observations.

To use the methodology of Section 6.1, we recognize that (6.6) can be written as

$$\frac{d}{dt} \begin{bmatrix} \hat{q}_S(t) \\ \hat{q}_E(t) \\ \hat{q}_I(t) \\ \hat{q}_R(t) \\ \hat{q}_D(t) \end{bmatrix} = \begin{bmatrix} -\hat{q}_S(t)\hat{q}_I(t) & 0 & 0 & 0 \\ \hat{q}_S(t)\hat{q}_I(t) & -\hat{q}_E(t) & 0 & 0 \\ 0 & \hat{q}_E(t) & -\hat{q}_I(t) & -\hat{q}_I(t) \\ 0 & 0 & \hat{q}_I(t) & 0 \\ 0 & 0 & 0 & \hat{q}_I(t) \end{bmatrix} \begin{bmatrix} \beta \\ \delta \\ (1-\alpha)\gamma \\ \alpha\rho \end{bmatrix}, \quad (6.7)$$

which is (6.1) with parameter vector $\hat{\mathbf{o}} = (\beta, \delta, (1-\alpha)\gamma, \alpha\rho)^\top \in \mathbb{R}^4$. Note that an estimate of $\hat{\mathbf{o}}$ does not explicitly provide estimates for α , γ , or ρ , but it does enable prediction in time and for new initial conditions.

6.3 Data generation

To generate data for this problem, we set initial conditions $\hat{\mathbf{q}}(0) = (0.994, 0.005, 0.001, 0, 0)^\top$ and true parameter values $\beta = 0.25$, $\delta = 0.1$, $\gamma = 0.1$, $\alpha = 0.05$, and $\rho = 0.05$. Hence, the true parameter vector is $\hat{\mathbf{o}} = (0.25, 0.1, 0.095, 0.0025)^\top$. The system (6.6) is then solved with the explicit adaptive-step Runge–Kutta time stepping method of order 5 over the time domain $[0, 199]$ days. We conduct experiments with different numbers of observations m , final observation times t_{m-1} , and noise levels $\xi > 0$. The state variables are observed at integer sample times chosen uniformly from $\mathbb{Z} \cap [t_0, t_{m-1}]$ (i.e., sample times are rounded to the nearest day). Thus, if $m = t_{m-1} + 1$, all state variables are observed daily; when $m < t_{m-1} + 1$, each state variable can be observed over a potentially different set of m days within time t_{m-1} . Letting $y_i(t_j) \in \mathbb{R}$ denote the true solution of the i -th state variable at sample time t_j , for each (i, j) pair we record noisy observations drawn from a truncated normal distribution centered at $y_i(t_j)$ with magnitude-dependent variance $\xi y_i(t_j)$ and truncation limits $[0, 1]$. Hence, the observations are contained in the interval $[0, 1]$, with more noise applied to states of larger magnitude. We thereby obtain snapshot vectors $\mathbf{y}_i = (y_{i,0}, y_{i,1}, \dots, y_{i,m-1})^\top$ for each state variable, which are used as inputs to Algorithm B.4 (the analog of Algorithm 4.3 for this setting).

6.4 Parameter estimation results

We first set $t_{m-1} = m - 1$ and relatively high noise level $\xi = 10\%$ (i.e., one noisy observation per day). We then apply Algorithm B.4 with $m = 60, 90$, and 120 days of data, using $m' = 4m$ GP estimates in each case.

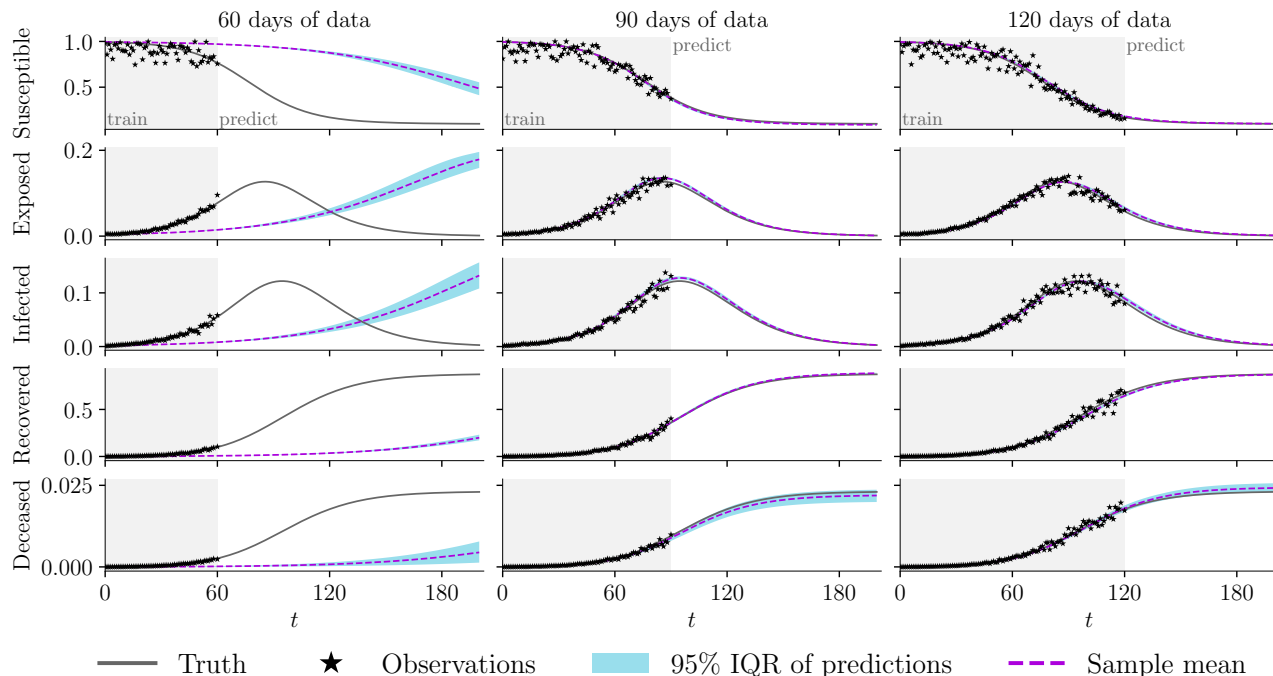


Figure 18: Bayesian ODE predictions for the SEIRD system (6.6) with varying amounts of data and $\xi = 10\%$ relative noise.

			β	δ	$(1 - \alpha)\gamma$	$\alpha\rho$	
Figure 18			Truth:	0.250	0.100	0.095	0.0025
$m = 60$	$t_{m-1} = m - 1$	$\xi = 10\%$	0.097	0.029	0.0269	0.00061	
$m = 90$	$t_{m-1} = m - 1$	$\xi = 10\%$	0.252	0.096	0.0922	0.00228	
$m = 120$	$t_{m-1} = m - 1$	$\xi = 10\%$	0.243	0.098	0.0926	0.00259	
Figure 19			Truth:	0.250	0.100	0.095	0.0025
$m = 10$	$t_{m-1} + 1 = 60$	$\xi = 5\%$	0.123	0.080	0.0769	0.00162	
$m = 10$	$t_{m-1} + 1 = 90$	$\xi = 5\%$	0.222	0.098	0.0725	0.00253	
$m = 10$	$t_{m-1} + 1 = 120$	$\xi = 5\%$	0.245	0.099	0.0915	0.00244	

Table 1: Parameter posterior means for the SEIRD system (6.6).

Sampling the posterior distributions for $\hat{\mathbf{o}}$ 500 times and solving the corresponding SEIRD system produces the results shown in Figure 18. Using data for only 60 days leads to poor results and the proposed method is not able to learn the dynamics of the underlying system due to the high level of noise. However, as the number of days for which data are available increases to 90 and 120, the predictions in the test regime become increasingly accurate. Table 1 reports the mean of the posterior distribution for each case and shows a similar trend. This shows training data through a certain amount of time t_{m-1} is required to capture the dynamics and accurately extrapolate in time.

To further analyze the effect of t_{m-1} on the proposed GP-based parameter estimation method, we limit the number of observations to $m = 10$ and reduce noise level to $\xi = 5\%$ while changing the time over which data are collected to $t_{m-1} + 1 = 60, 90,$ and 120 days. Note that data observation times are not aligned across the states. In each case $m' = 4(t_{m-1} + 1)$ GP estimates are employed in Algorithm B.4. The means of the posterior parameter distribution are shown in Table 1 and sampled predictions are provided in Figure 19. The results of the Bayesian fit are accurate with even $m = 10$ observations as long as they are spread over 120 days, but are

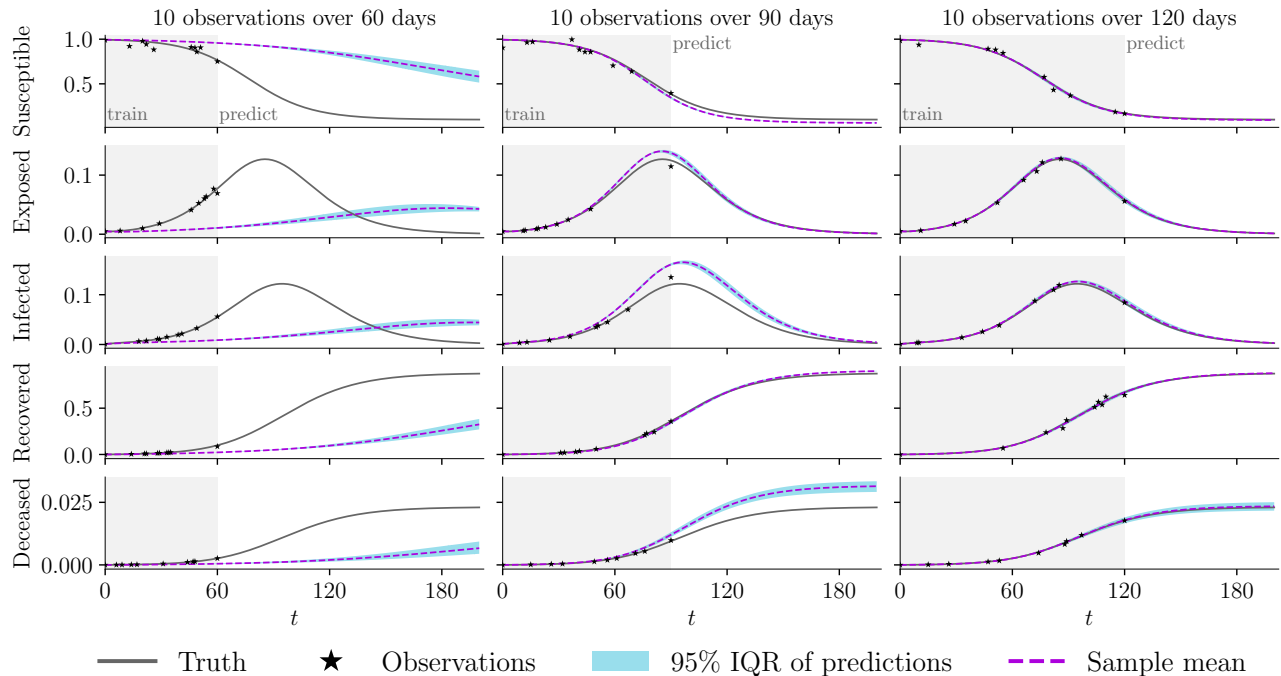


Figure 19: Bayesian ODE predictions for the SEIRD system (6.6) with $m = 10$ observations, $\xi = 5\%$ relative noise, and varying observation periods. Data observation times are different for each state.

inaccurate when training data are only available for 60 days. Hence, for this experiment, the performance of the method is more sensitive to the quality of the training data than to the amount of data.

Remark 6.1 (Equation fitting vs. trajectory fitting). For Bayesian parameter estimation in ODEs, the likelihood setting in this work essentially fits given equations to data. Time derivatives on the left-hand side of the ODEs are estimated using GPs, and the parameters on the right-hand side are estimated through a linear regression problem. This is fundamentally different from a likelihood that compares the evolution of the dynamical model—time integration of the ODEs—with the observed data, i.e., fitting the trajectories. The latter may also improve the robustness to data noise and sparsity, but the training process often demands more computational effort. A typical example is the (Bayesian) neural ODE approach [13, 16], which requires solving adjoint state equations to evaluate the gradients of loss functions.

7 Concluding remarks

This work presents a new Bayesian method for OpInf that uses a probabilistic formulation for the reduced-order operators, thereby combining physics-based and data-driven modeling. In particular, GP emulation incorporates both reduced state data and their differential equation structure into the definition of the likelihood function for the reduced operators. The power of GPs in smoothing and interpolating noisy and/or sparse data enables probabilistic ROMs in scenarios where standard OpInf is not feasible. The closed-form posterior estimates of the reduced operators provide an interpretable low-dimensional representation of complex physics-based models.

The numerical results demonstrate that GP-BayesOpInf is effective at constructing probabilistic surrogates for high-dimensional dynamical systems with polynomial structure given only data and knowledge of the model structure. The method was shown to perform well with abundant but noisy data, as well as in low-noise scenarios with sparse data. GP-BayesOpInf also applies to problems with multiple trajectories stemming from

different initial conditions or input functions, as well as to a large class of differential systems with parameters shared across the equations.

There are several avenues for future work related to GP-BayesOpInf. Our experiments used squared-exponential GP kernels and linear dimensionality reduction by POD, but the methodology is flexible and can be applied with any sufficiently differentiable GP kernel and with any type of compression (including nonlinear manifold approximators such as in [3, 22, 23, 34, 58]). It would be interesting to explore if embedding desirable properties in the kernel can improve model performance for problems with specific characteristics (for example, periodicity). In the presence of large amounts of noise, the number of reliable reduced modes that can be obtained through POD is limited. To reduce the projection error by including more reduced modes in GP-BayesOpInf, it may be advantageous to leverage filtering strategies as shown in [20] prior to applying dimensionality reduction. We also plan to explore more advanced GP techniques, such as sparse GPs [64, 65] to reduce the trajectory data needed for regression (i.e., selecting m' automatically), multifidelity GPs [2, 12, 40, 45, 49, 50] to fuse data from different sources, as well as a single vector-valued GP for all reduced modes to incorporate physically meaningful spatial correlations. Another promising direction is to leverage the quantified uncertainties to guide the active selection of initial conditions and/or input functions for parsimoniously generating informative training trajectories.

Acknowledgments

S.M. was supported by the John von Neumann postdoctoral fellowship, a position sponsored by the Applied Mathematics Program of the U.S. Department of Energy Office of Advanced Scientific Computing Research. This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. 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.

A.C. and K.E.W. acknowledge support from DARPA Automating Scientific Knowledge Extraction and Modeling (ASKEM) program under Contract No. HR0011262087 (award number 653002). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

This work was in part carried out when M.G. held a position at the University of Twente (NL), for which he acknowledges financial support from *Sectorplan Bèta* under the focus area *Mathematics of Computational Science*.

Appendix A Gaussian process regression

A Gaussian process is a collection of random variables, any finite number of which obeys a joint normal distribution. In Gaussian process regression, the prior on the regression function is assumed to follow a Gaussian process, i.e., for $(\mathbf{x}, \mathbf{x}') \in \Omega \times \Omega$,

$$f(\mathbf{x}) \sim \mathcal{GP}(b(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')), \quad (\text{A.1})$$

where Ω is the input domain of interest, $b : \Omega \rightarrow \mathbb{R}$ is a given mean function, and $\kappa : \Omega \times \Omega \rightarrow \mathbb{R}$ is a covariance function which often depends on kernel hyperparameters $\boldsymbol{\theta}$. We also assume that the observation of $f(\mathbf{x})$, denoted by $y(\mathbf{x})$, may be corrupted by an independent white noise term, i.e., $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \chi)$.

Given a finite number m of training input locations $\mathbf{X} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_m]^\top$ in the domain Ω , a joint Gaussian is thus defined for the corresponding observed outputs $\mathbf{y} = \{y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_m)\}^\top$:

$$\mathbf{y}|\mathbf{X} \sim \mathcal{N}(b(\mathbf{X}), \mathbf{K}), \quad \mathbf{K} = \kappa(\mathbf{X}, \mathbf{X}) + \chi \mathbf{I}_m. \quad (\text{A.2})$$

Using the standard rule for conditioning Gaussians, the (predictive) posterior distribution for any new test input $\mathbf{s} \in \Omega$ conditioned on the training data can be formulated as a new Gaussian process $f^*(\mathbf{s})$:

$$\begin{aligned} f^*(\mathbf{s})|\mathbf{X}, \mathbf{y} &\sim \mathcal{GP}(b^*(\mathbf{s}), \kappa^*(\mathbf{s}, \mathbf{s}')), & \text{with} \\ b^*(\mathbf{s}) &= b(\mathbf{s}) + \kappa(\mathbf{s}, \mathbf{X})\mathbf{K}^{-1}(\mathbf{y} - b(\mathbf{X})), & \text{and} \quad \kappa^*(\mathbf{s}, \mathbf{s}') = \kappa(\mathbf{s}, \mathbf{s}') - \kappa(\mathbf{s}, \mathbf{X})\mathbf{K}^{-1}\kappa(\mathbf{X}, \mathbf{s}'). \end{aligned} \quad (\text{A.3})$$

The values of hyperparameters can be tuned to optimize the predictive performance. A widely used technique is the empirical Bayes approach of maximizing marginal likelihood. With a standard gradient-based optimizer, the optimal hyperparameters $\boldsymbol{\theta}^*$ can be estimated through the following maximization problem:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \\ &= \arg \max_{\boldsymbol{\theta}} \left\{ -\frac{1}{2}(\mathbf{y} - b(\mathbf{X}))^\top \mathbf{K}^{-1}(\boldsymbol{\theta})(\mathbf{y} - b(\mathbf{X})) - \frac{1}{2} \log |\mathbf{K}(\boldsymbol{\theta})| - \frac{m}{2} \log(2\pi) \right\}. \end{aligned} \quad (\text{A.4})$$

Here, $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ is the marginal likelihood function of $\boldsymbol{\theta}$, which is given by the density function value of the observations \mathbf{y} over the input locations \mathbf{X} given the hyperparameters $\boldsymbol{\theta}$.

Appendix B Algorithmic extensions

Algorithms 3.1–4.3 are written for problems in the PDE setting where a single trajectory of data is available for training. This appendix generalizes the algorithms to the special cases discussed in Section 3.3 and Section 6.

B.1 Multiple trajectories

If data are available from $\ell > 1$ trajectories due to varying initial conditions or inputs, as discussed in Section 3.3, the algorithmic procedure requires minor modifications. First, one GP is trained per reduced mode per trajectory, resulting in $r \cdot \ell$ total executions of Algorithm 4.1. Second, for each reduced mode, GP estimates from all ℓ trajectories are concatenated as described in (3.23). Third, the prior variances are selected by minimizing the average reconstruction error across all ℓ trajectories. Algorithm B.1 and Algorithm B.2 adapt Algorithm 4.2 and Algorithm 4.3, respectively, to this setting. Algorithm B.2 uses $\mathbf{t}^{(l)} \in \mathbb{R}^{m^{(l)}}$ to denote the $m^{(l)} \in \mathbb{N}$ observation times for the l -th trajectory, which may be different for each l .

B.2 Parameter estimation for ordinary differential equations

The parameter estimation setting of Section 6 requires small algorithmic adjustments and simplifications. Algorithm B.3 and Algorithm B.4 adapt Algorithm 4.2 and Algorithm 4.3, respectively, to this scenario.

Algorithm B.1 Estimate the mean error of the Bayesian OpInf model over multiple trajectories.

```

1: procedure OPINFERRORMULTI(
    Prior variances  $\gamma_1, \dots, \gamma_r \in \mathbb{R}^d$ ;
    Regression arguments  $\alpha = \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_r, \mathbf{W}_1^{1/2}, \dots, \mathbf{W}_r^{1/2}\}$ ,
    GP state estimates  $\tilde{\mathbf{Q}}^{(1)}, \dots, \tilde{\mathbf{Q}}^{(\ell)} \in \mathbb{R}^{r \times m'}$ 
    Input functions  $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(\ell)} : \mathbb{R} \rightarrow \mathbb{R}^p$ ,
    GP estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Stability margin parameter  $\varphi > 0$ ,
    Number of samples  $n_s \in \mathbb{N}$ ,
    Final prediction time  $t_f > t'_0$ 
)
2:    $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \dots, (\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) \leftarrow \text{OPPOST}(\boldsymbol{\alpha}, \gamma_1, \dots, \gamma_r)$ 
3:   # Check stability and compute the training error for each trajectory.
4:   for  $l = 1, \dots, \ell$  do
5:      $\hat{\mathbf{q}}_0^{(l)} \leftarrow \tilde{\mathbf{Q}}_{:,0}^{(l)}$ 
6:      $B^{(l)} \leftarrow \varphi \cdot \max_{ij} |\tilde{\mathbf{Q}}_{ij}^{(l)}|$ 
7:     for  $k = 1, \dots, n_s$  do
8:        $\hat{\mathbf{O}}_k^{(l)} \leftarrow \text{sample from } [\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \cdots \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)]^\top$ 
9:        $\check{\mathbf{Q}} \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{O}}_k^{(l)} \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}^{(l)}(t)), \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0^{(l)} \text{ over } t \in [t'_0, t_f]$ 
10:      if  $\max_{ij} |\check{\mathbf{Q}}_{ij}| > B$  then
11:        return  $\infty$ 
12:       $\hat{\mathbf{Q}}_k^{(l)} \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \hat{\mathbf{O}}_k^{(l)} \mathbf{d}(\hat{\mathbf{q}}(t), \mathbf{u}^{(l)}(t)), \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0^{(l)} \text{ over } t \in \mathbf{t}'$ 
13:       $\hat{\mathbf{Q}}^{(l)} = \frac{1}{n_s} \sum_{k=1}^{n_s} \hat{\mathbf{Q}}_k^{(l)}$ 
14:   return  $\frac{1}{n_s} \sum_{l=1}^{\ell} \|\tilde{\mathbf{Q}}^{(l)} - \hat{\mathbf{Q}}^{(l)}\|$  # Average error compared to GP state estimates.
```

Algorithm B.2 Bayesian Operator Inference with Gaussian processes for multiple data trajectories.

```

1: procedure GPBAYESOPINFMULTI(
    Observation times  $\mathbf{t}^{(l)} \in \mathbb{R}^{m^{(l)}}$ ,  $l = 1, \dots, \ell$ ,
    Snapshots  $\mathbf{Q}^{(l)} \in \mathbb{R}^{N \times m^{(l)}}$ ,  $l = 1, \dots, \ell$ ,
    Reduced dimension  $r \ll N$ ,
    Estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Numerical regularization  $0 < \tau \ll 1$ ,
    Input functions  $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(\ell)} : \mathbb{R} \rightarrow \mathbb{R}^p$ 
)
2:   # GP regression on compressed training data.
3:    $\mathbf{V} \leftarrow \text{basis}(\mathbf{Q}^{(1)}, \dots, \mathbf{Q}^{(\ell)}; r)$  # Global rank- $r$  basis (e.g., POD).
4:   for  $l = 1, \dots, \ell$  do
5:      $[\mathbf{y}_1^{(l)} \ \dots \ \mathbf{y}_r^{(l)}]^\top \leftarrow \mathbf{V}^\top \mathbf{Q}^{(l)}$ 
6:     for  $i = 1, \dots, r$  do
7:        $\tilde{\mathbf{y}}_i^{(l)}, \tilde{\mathbf{z}}_i^{(l)}, (\mathbf{W}_i^{zz^{(l)}})^{1/2} \leftarrow \text{GPFIT}(\mathbf{t}^{(l)}, \mathbf{y}_i^{(l)}, \mathbf{t}', \tau)$ 
8:       # Prepare regression data for the  $l$ -th trajectory.
9:        $\tilde{\mathbf{Q}}^{(l)} \leftarrow [\tilde{\mathbf{y}}_1^{(l)} \ \dots \ \tilde{\mathbf{y}}_r^{(l)}]^\top$ 
10:       $\tilde{\mathbf{D}}^{(l)} \leftarrow \text{data\_matrix}(\tilde{\mathbf{Q}}^{(l)}, \mathbf{u}^{(l)}(\mathbf{t}'))$ 
11:      # Concatenate regression data across all trajectories.
12:       $\tilde{\mathbf{D}} \leftarrow [(\tilde{\mathbf{D}}^{(1)})^\top \ \dots \ (\tilde{\mathbf{D}}^{(\ell)})^\top]^\top$  # Concatenate data matrices.
13:      for  $i = 1, \dots, r$  do
14:         $\tilde{\mathbf{z}}_i \leftarrow [(\tilde{\mathbf{z}}_i^{(1)})^\top \ \dots \ (\tilde{\mathbf{z}}_i^{(\ell)})^\top]^\top$  # Concatenate time derivatives.
15:         $\mathbf{W}_i \leftarrow \text{diag}(\mathbf{W}_i^{zz^{(1)}}, \dots, \mathbf{W}_i^{zz^{(\ell)}})$  # Block diagonal weights.
16:       $\boldsymbol{\alpha} \leftarrow \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_r, \mathbf{W}_1^{1/2}, \dots, \mathbf{W}_r^{1/2}\}$  # Regression arguments.
17:      # Operator inference regression to construct a probabilistic reduced model.
18:       $\boldsymbol{\gamma}_1^*, \dots, \boldsymbol{\gamma}_r^* \leftarrow \text{argmin}_{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_r} \text{OPINFERRORMULTI}(\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_r; \boldsymbol{\alpha}, \tilde{\mathbf{Q}}^{(1)}, \dots, \tilde{\mathbf{Q}}^{(\ell)}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(\ell)}, \mathbf{t}')$ 
19:       $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \dots, (\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) \leftarrow \text{OPPOST}(\boldsymbol{\alpha}, \boldsymbol{\gamma}_1^*, \dots, \boldsymbol{\gamma}_r^*)$ 
20:      return  $[\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \ \dots \ \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)]^\top$ 

```

Algorithm B.3 Estimate the mean error of the Bayesian OpInf model for ODE parameter estimation.

```

1: procedure OPINFERRORODES(
    Prior variance  $\gamma \in \mathbb{R}^d$ ;
    Regression arguments  $\alpha = \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}, \mathbf{W}^{1/2}\}$ ,
    GP state estimates  $\tilde{\mathbf{Q}} \in \mathbb{R}^{r \times m'}$ 
    Input functions  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^p$ ,
    GP estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Stability margin parameter  $\varphi > 0$ ,
    Number of samples  $n_s \in \mathbb{N}$ ,
    Final prediction time  $t_f > t'_0$ 
)
2:    $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{OPPOST}(\alpha, \gamma)$  # Algorithm 3.1 with  $r = 1$ .
3:    $\hat{\mathbf{q}}_0 \leftarrow \tilde{\mathbf{Q}}_{:,0}$ 
4:    $B \leftarrow \varphi \cdot \max_{ij} |\tilde{\mathbf{Q}}_{ij}|$ 
5:   for  $k = 1, \dots, n_s$  do
6:      $\hat{\mathbf{o}}_k \leftarrow \text{sample from } \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 
7:      $\hat{\mathbf{Q}} \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \tilde{\mathbf{D}}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) \hat{\mathbf{o}}_k, \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0 \text{ over } t \in [t'_0, t_f]$ 
8:     if  $\max_{ij} |\hat{\mathbf{Q}}_{ij}| > B$  then
9:       return  $\infty$ 
10:     $\hat{\mathbf{Q}}_k \leftarrow \text{solve } \frac{d}{dt} \hat{\mathbf{q}}(t) = \tilde{\mathbf{D}}(\hat{\mathbf{q}}(t), \mathbf{u}(t)) \hat{\mathbf{o}}_k, \hat{\mathbf{q}}(t'_0) = \hat{\mathbf{q}}_0 \text{ over } t \in \mathbf{t}'$ 
11:    $\hat{\mathbf{Q}} = \frac{1}{n_s} \sum_{k=1}^{n_s} \hat{\mathbf{Q}}_k$ 
12:   return  $\|\tilde{\mathbf{Q}} - \hat{\mathbf{Q}}\|$ 

```

Algorithm B.4 Bayesian parameter estimation for ODEs with Gaussian processes.

```

1: procedure GPBAYESODES(
    Observation times  $\mathbf{t}_i \in \mathbb{R}^{m_i}, i = 1, \dots, r$ ,
    Snapshots  $\mathbf{y}_i \in \mathbb{R}^{m_i}, i = 1, \dots, r$ ,
    Estimation times  $\mathbf{t}' \in \mathbb{R}^{m'}$ ,
    Numerical regularization  $0 < \tau \ll 1$ ,
    Input functions  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^p$ 
)
2:   # GP regression on state observations.
3:   for  $i = 1, \dots, r$  do
4:      $\tilde{\mathbf{y}}_i, \tilde{\mathbf{z}}_i, (\mathbf{W}_i^{zz})^{1/2} \leftarrow \text{GPFIT}(\mathbf{t}, \mathbf{y}_i, \mathbf{t}', \tau)$ 
5:   # Prepare regression data.
6:    $\tilde{\mathbf{Q}} \leftarrow [\tilde{\mathbf{y}}_1 \ \cdots \ \tilde{\mathbf{y}}_r]^\top$ 
7:    $\tilde{\mathbf{D}} \leftarrow \text{data\_matrix}(\tilde{\mathbf{Q}}, \mathbf{u}(\mathbf{t}'))$ 
8:    $\tilde{\mathbf{z}} \leftarrow [\tilde{\mathbf{z}}_1^\top \ \cdots \ \tilde{\mathbf{z}}_r^\top]^\top$  # Concatenate time derivatives.
9:    $\mathbf{W} \leftarrow \text{diag}(\mathbf{W}_1^{zz}, \dots, \mathbf{W}_r^{zz})$  # Block diagonal weights.
10:   $\alpha \leftarrow \{\tilde{\mathbf{D}}, \tilde{\mathbf{z}}, \mathbf{W}^{1/2}\}$  # Regression arguments.
11:  # Operator inference regression to construct a probabilistic reduced model.
12:   $\gamma^* \leftarrow \text{argmin}_\gamma \text{OPINFERRORODES}(\gamma; \alpha, \tilde{\mathbf{Q}}, \mathbf{u}, \mathbf{t}')$  # Algorithm B.3.
13:   $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{OPPOST}(\alpha, \gamma^*)$  # Algorithm 3.1 with  $r = 1$ .
14:  return  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 

```

Appendix C Main Nomenclature

Full order state

$N \in \mathbb{N}$	full-order state dimension
$\mathbf{q}(t) \in \mathbb{R}^N$	full-order state vector
$\mathbf{u}(t) \in \mathbb{R}^p$	input function (time-dependent forcing / boundary conditions, etc.)

Reduced order state

$r \in \mathbb{N}$	reduced-order state dimension, number of basis vectors
$\hat{\mathbf{q}}(t) \in \mathbb{R}^r$	reduced-order state vector with entries $(\hat{q}_1(t), \dots, \hat{q}_r(t))$
$\mathbf{V} \in \mathbb{R}^{N \times r}$	rank- r basis, i.e., $\mathbf{q}(t) \approx \mathbf{V}\hat{\mathbf{q}}(t) + \bar{\mathbf{q}}$
$\mathbf{d} : \mathbb{R}^r \times \mathbb{R}^p \rightarrow \mathbb{R}^d$	data vector defining reduced-order model structure
$\hat{\mathbf{O}} \in \mathbb{R}^{r \times d}$	operator matrix defining reduced-order model structure
$\hat{\mathbf{o}}_i \in \mathbb{R}^d$	operators for the i -th reduced mode, the i -th row of $\hat{\mathbf{O}}$

Training data

$m \in \mathbb{N}$	number of available snapshots
$\mathbf{t} \in \mathbb{R}^m$	times t_0, \dots, t_{m-1} corresponding to the observed snapshots
$\mathbf{q}_j \in \mathbb{R}^N$	possibly noisy full-order snapshot at time $t = t_j$, i.e., $\mathbf{q}_j \approx \mathbf{q}(t_j)$
$\bar{\mathbf{q}} \in \mathbb{R}^N$	average training snapshot
$\hat{\mathbf{q}}_j \in \mathbb{R}^r$	reduced-order snapshot at time $t = t_j$, i.e., $\hat{\mathbf{q}}_j = \mathbf{V}^\top(\mathbf{q}_j - \bar{\mathbf{q}})$
$\mathbf{y}_i \in \mathbb{R}^m$	reduced-order snapshot data for the i -th reduced mode, i.e., the i -th entry of each $\hat{\mathbf{q}}_j$, $j = 0, \dots, m-1$

Gaussian process

$\kappa_i : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$	kernel function for the i -th reduced mode with hyperparameters $\boldsymbol{\theta}_i$
$m' \in \mathbb{N}$	number of GP reduced state / time derivative estimates
$\mathbf{t}' \in \mathbb{R}^{m'}$	times corresponding to the GP estimates
$\tilde{\mathbf{y}}_i \in \mathbb{R}^{m'}$	GP reduced state estimate for the i -th reduced mode
$\tilde{\mathbf{z}}_i \in \mathbb{R}^{m'}$	GP reduced time derivative estimate for the i -th reduced mode
$\mathbf{K}_i^{yy} \in \mathbb{R}^{m \times m}$	kernel matrix for \mathbf{y} - \mathbf{y} correlation (observation times \mathbf{t})
$\mathbf{K}_i^{zz} \in \mathbb{R}^{m' \times m'}$	kernel matrix for \mathbf{z} - \mathbf{z} correlation (estimation times \mathbf{t}')
$\mathbf{K}_i^{zy} \in \mathbb{R}^{m' \times m}$	kernel matrix for \mathbf{z} - \mathbf{y} correlation (\mathbf{t}' vs \mathbf{t})
$\mathbf{W}_i^{zz} \in \mathbb{R}^{m' \times m'}$	block of the inverse of the kernel matrix for observation times

Operator inference

$\mathbf{D} \in \mathbb{R}^{m \times d}$	data matrix for standard OpInf
$\tilde{\mathbf{D}} \in \mathbb{R}^{m' \times d}$	GP-enhanced data matrix
$\boldsymbol{\mu}_i \in \mathbb{R}^d$	posterior mean of $\hat{\mathbf{o}}_i$
$\boldsymbol{\Sigma}_i \in \mathbb{R}^{d \times d}$	posterior covariance matrix of $\hat{\mathbf{o}}_i$
$\boldsymbol{\Gamma}_i \in \mathbb{R}^{d \times d}$	prior covariance for $\hat{\mathbf{o}}_i$
$\boldsymbol{\gamma}_i \in \mathbb{R}^d$	diagonals of the prior covariance $\boldsymbol{\Gamma}_i$

References

- [1] K. M. Abadir and J. R. Magnus. *Matrix Algebra*, volume 1. Cambridge University Press, 2005.
- [2] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.
- [3] J. Barnett and C. Farhat. Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction. *Journal of Computational Physics*, 464:111348, 2022.
- [4] T. Beckers, Q. Wu, and G. J. Pappas. Physics-enhanced Gaussian process variational autoencoder. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 521–533. Pmlr, 2023.
- [5] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox. Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. *Computer Methods in Applied Mechanics and Engineering*, 372:113433, 2020.
- [6] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- [7] G. Berkooz, P. Holmes, and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993.
- [8] R. P. Brent. *Algorithms for Minimization Without Derivatives*. Dover Publications, New York, NY, 2002.
- [9] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [10] G. D. Byrne and A. C. Hindmarsh. A polyalgorithm for the numerical solution of ordinary differential equations. *ACM Transactions on Mathematical Software*, 1(1):71–96, 1975.
- [11] B. Calderhead, M. Girolami, and N. Lawrence. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. *Advances in Neural Information Processing Systems*, 21, 2008.
- [12] A. Chaudhuri, A. N. Marques, and K. Willcox. mfegra: Multifidelity efficient global reliability analysis through active learning for failure boundary location. *Structural and Multidisciplinary Optimization*, 64(2):797–811, 2021.
- [13] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.
- [14] Y. Chen, B. Hosseini, H. Owhadi, and A. M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.
- [15] L. Cicci, S. Fresca, M. Guo, A. Manzoni, and P. Zunino. Uncertainty quantification for nonlinear solid mechanics using reduced order models with Gaussian process regression. *Computers & Mathematics with Applications*, 149:1–23, 2023.
- [16] R. Dandekar, K. Chung, V. Dixit, M. Tarek, A. Garcia-Valadez, K. V. Vemula, and C. Rackauckas. Bayesian neural ordinary differential equations. *arXiv preprint arXiv:2012.07244*, 2020.
- [17] D. del Castillo-Negrete, D. A. Spong, and S. P. Hirshman. Proper orthogonal decomposition methods for noise reduction in particle-based transport calculations. *Physics of Plasmas*, 15(9):092308, 2008.

- [18] J. R. Dormand and P. J. Prince. A family of embedded Runge–Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.
- [19] B. P. Epps and A. H. Techet. An error threshold criterion for singular value decomposition modes extracted from PIV data. *Experiments in fluids*, 48(2):355–367, 2010.
- [20] I. Farcaş, R. Munipalli, and K. E. Willcox. On filtering in non-intrusive data-driven reduced-order modeling. In *AIAA Aviation 2022 Forum*, 2022. AIAA paper 2022-3487.
- [21] I. Farcaş, R. Gundeveia, R. Munipalli, and K. E. Willcox. Parametric non-intrusive reduced-order models via operator inference for large-scale rotating detonation engine simulations. In *AIAA SCITECH 2023 Forum*, 2023. AIAA Paper 2023-0172.
- [22] R. Geelen, L. Balzano, S. Wright, and K. Willcox. Learning physics-based reduced-order models from data using nonlinear manifolds. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(3), 2024.
- [23] R. Geelen, S. Wright, and K. Willcox. Operator inference for non-intrusive model reduction with quadratic manifolds. *Computer Methods in Applied Mechanics and Engineering*, 403:115717, 2023.
- [24] Y. Geng, J. Singh, L. Ju, B. Kramer, and Z. Wang. Gradient preserving operator inference: Data-driven reduced-order models for equations with gradient structure. *arXiv preprint arXiv:2401.12138*, 2024.
- [25] O. Ghattas and K. Willcox. Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554, 2021.
- [26] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. Part I—Open-loop model development. *International Journal for Numerical Methods in Engineering*, 44(7):945–972, 1999.
- [27] A. Gruber and I. Tezaur. Canonical and noncanonical Hamiltonian operator inference. *Computer Methods in Applied Mechanics and Engineering*, 416:116334, 2023.
- [28] M. Guo and J. S. Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 345:75–99, 2019.
- [29] M. Guo, S. McQuarrie, and K. Willcox. Bayesian operator inference for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 402:115336, 2022.
- [30] T. Guo, O. Rokoš, and K. Veroy. Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 384:113924, 2021.
- [31] D. Hansen, D. C. Maddix, S. Alizadeh, G. Gupta, and M. W. Mahoney. Learning physical models that can respect conservation laws. *Physica D: Nonlinear Phenomena*, 457:133952, 2024.
- [32] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.
- [33] P. Jain, S. McQuarrie, and B. Kramer. Performance comparison of data-driven reduced models for a single-injector combustion process. In *AIAA Propulsion and Energy 2021 Forum*, 2021. Paper AIAA-2021-3633.
- [34] S. Jain, P. Tiso, J. B. Rutzmoser, and D. J. Rixen. A quadratic manifold for model order reduction of nonlinear structural dynamics. *Computers & Structures*, 188:80–94, 2017.

- [35] M. Kast, M. Guo, and J. S. Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 364:112947, 2020.
- [36] T. Koike and E. Qian. Energy-preserving reduced operator inference for efficient design and control. *AIAA SciTech Forum*, 2024. AIAA paper 2024-1012.
- [37] I. Korolev. Identification and estimation of the SEIRD epidemic model for COVID-19. *Journal of Econometrics*, 220(1):63–85, 2021.
- [38] B. Kramer, B. Peherstorfer, and K. E. Willcox. Learning nonlinear reduced models from data with operator inference. *Annual Review of Fluid Mechanics*, 56(1):521–548, 2024.
- [39] B. Kramer and K. Willcox. Balanced truncation model reduction for lifted nonlinear systems. In C. Beattie, P. Benner, M. Embree, S. Gugercin, and S. Lefteriu, editors, *Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas*, pages 157–174, Cham, 2022. Springer International Publishing.
- [40] L. Le Gratiet. *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis, Université Paris-Diderot-Paris VII, 2013.
- [41] S. A. McQuarrie. *Data-driven parametric reduced-order models: Operator inference for reactive flow applications*. PhD thesis, The University of Texas at Austin, 2023.
- [42] S. A. McQuarrie, C. Huang, and K. E. Willcox. Data-driven reduced-order models via regularised operator inference for a single-injector combustion process. *Journal of the Royal Society of New Zealand*, 51(2):194–211, 2021.
- [43] S. A. McQuarrie, P. Khodabakhshi, and K. E. Willcox. Non-intrusive reduced-order models for parametric partial differential equations via data-driven operator inference. *SIAM Journal on Scientific Computing*, 45(4):A1917–a1946, 2023.
- [44] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [45] A. O’Hagan and M. C. Kennedy. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.
- [46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [47] B. Peherstorfer. Sampling low-dimensional Markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference. *SIAM Journal on Scientific Computing*, 42(5):A3489–a3515, 2020.
- [48] B. Peherstorfer and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.
- [49] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751, 2017.
- [50] M. Poloczek, J. Wang, and P. Frazier. Multi-information source optimization. *Advances in neural information processing systems*, 30, 2017.

- [51] E. Qian, I.-G. Farcas, and K. Willcox. Reduced operator inference for nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 44(4):A1934–a1959, 2022.
- [52] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.
- [53] E. Y. Qian. *A scientific machine learning approach to learning reduced models for nonlinear partial differential equations*. PhD thesis, Massachusetts Institute of Technology, 2021.
- [54] A. Quarteroni, G. Rozza, et al. *Reduced order methods for modeling and computational reduction*, volume 9. Springer, 2014.
- [55] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.
- [56] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.
- [57] N. Sawant, B. Kramer, and B. Peherstorfer. Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference. *Computer Methods in Applied Mechanics and Engineering*, 404:115836, 2023.
- [58] P. Schwerdtner and B. Peherstorfer. Greedy construction of quadratic manifolds for nonlinear dimensionality reduction and nonlinear model reduction. *arXiv preprint arXiv:2403.06732*, 2024.
- [59] L. F. Shampine. Some practical Runge–Kutta formulas. *Mathematics of Computation*, 46(173):135–150, 1986.
- [60] L. F. Shampine and M. W. Reichelt. The MATLAB ode suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.
- [61] H. Sharma and B. Kramer. Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale dynamical systems. *Physica D: Nonlinear Phenomena*, 462:134128, 2024.
- [62] H. Sharma, Z. Wang, and B. Kramer. Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems. *Physica D: Nonlinear Phenomena*, 431:133122, 2022.
- [63] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.
- [64] A. Smola and P. Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.
- [65] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.
- [66] R. Swischuk, B. Kramer, C. Huang, and K. Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA Journal*, 58(6):2658–2672, 2020.
- [67] R. C. Swischuk. Physics-based machine learning and data-driven reduced-order modeling. Master’s thesis, Massachusetts Institute of Technology, 2019.

- [68] F. Tronarp, H. Kersting, S. Särkkä, and P. Hennig. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: A new perspective. *Statistics and Computing*, 29:1297–1315, 2019.
- [69] W. I. T. Uy, D. Hartmann, and B. Peherstorfer. Operator inference with roll outs for learning reduced models from scarce and low-quality data. *Computers & Mathematics with Applications*, 145:224–239, 2023.
- [70] W. I. T. Uy, Y. Wang, Y. Wen, and B. Peherstorfer. Active operator inference for learning low-dimensional dynamical-system models from noisy data. *SIAM Journal on Scientific Computing*, 45(4):A1462–a1490, 2023.
- [71] D. Venturi. On proper orthogonal decomposition of randomly perturbed fields with applications to flow past a cylinder and natural convection over a horizontal plate. *Journal of Fluid Mechanics*, 559:215–254, 2006.
- [72] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and Contributors, SciPy 1.0. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [73] D. Ye and M. Guo. Gaussian process learning of nonlinear dynamics. *Communications in Nonlinear Science and Numerical Simulation*, 138:108184, 2024.
- [74] Z. Zhang, M. Guo, and J. S. Hesthaven. Model order reduction for large-scale structures with local nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 353:491–515, 2019.