

# Operator inference with roll outs for learning reduced models from scarce and low-quality data

Wayne Isaac Tan Uy\*    Dirk Hartmann†    Benjamin Peherstorfer\*

August 31, 2023

Data-driven modeling has become a key building block in computational science and engineering. However, data that are available in science and engineering are typically scarce, often polluted with noise and affected by measurement errors and other perturbations, which makes learning the dynamics of systems challenging. In this work, we propose to combine data-driven modeling via operator inference with the dynamic training via roll outs of neural ordinary differential equations. Operator inference with roll outs inherits interpretability, scalability, and structure preservation of traditional operator inference while leveraging the dynamic training via roll outs over multiple time steps to increase stability and robustness for learning from low-quality and noisy data. Numerical experiments with data describing shallow water waves and surface quasi-geostrophic dynamics demonstrate that operator inference with roll outs provides predictive models from training trajectories even if data are sampled sparsely in time and polluted with noise of up to 10%.

**Keywords:** data-driven modeling; scientific machine learning; model reduction; scarce and noisy data

## 1 Introduction

Learning models from data has become a key building block in computational science and engineering. However, data in science and engineering applications often are scarce, perturbed and noisy. To cope with scarce low-quality data, we propose to combine non-intrusive model reduction via operator inference [1] with the dynamic training via roll outs of neural ordinary differential equations (Neural ODEs) [2]. We show that roll outs make operator inference more robust against scarce data and noise. Numerical experiments demonstrate that the proposed operator inference approach with roll outs learns reduced models even when traditional operator inference fails because of too little data and too much noise.

Learning reduced models from data is a widely studied problem. There are dynamic mode decomposition and Koopman-based methods [3, 4, 5, 6, 7, 8] as well as sparsity-based methods [9, 10, 11, 12]. Related are methods for data-driven closure modeling such as [13, 14]. A range of methods for non-intrusive model reduction learn coefficients of low-dimensional approximations from data [15, 16, 17, 18, 19, 20, 21]; see [22] for a software package. The systems and control community studies data-driven model reduction methods from the frequency-domain perspective, e.g., Loewner methods [23, 24, 25, 26, 27, 28, 29, 30, 31], vector fitting [32, 33], and eigensystem realization [34, 35]. All of these techniques are static in the sense that the prediction error of the to-be-learned model is penalized based only on the residual from a prediction of a single step into the future, because this typically leads to linear optimization problems.

---

\*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

†Siemens Industrial Software GmbH, Munich, Germany

Operator inference is a non-intrusive model reduction method that learns from time-domain data and is applicable to nonlinear systems [1]. It is a building block of other scientific machine learning techniques [36, 37, 38, 39, 40, 41, 42, 43, 44, 37] and has been applied to a range of challenging problems in engineering [45]. However, operator inference is prone to perturbations and noise in the data. The work [46] considers a perturbation analysis for operator inference, which shows that perturbations introduced via the approximation of the time derivatives can have large effects on the model predictions. The work [47] shows that even errors in the dynamics due to the projection onto low-dimensional subspaces can lead to large errors in the predicted states. Regularization is one remedy and is considered in [45, 43, 48]; however, determining regularizers that bias against noise and at the same time avoid losing the signal in the data that describe actual dynamics of the underlying systems remains an open problem. Operator inference from noisy data is also considered in the work [49], where an active learning approach is proposed for damping the effect of noise on the prediction error. All these approaches are based on the original formulation of operator inference that relies on the residual of the prediction one time step into the future only.

In contrast, we propose a formulation of operator inference that leverages ideas from differentiable programming and Neural ODEs. We include the accuracy of predictions of the to-be-learned model of multiple time steps into the future in the training objective. Our work is motivated by the preprint [50] that formulates operator inference in a differentiable setting and analytically derives gradients for the training. The training objective is motivated by Neural ODEs [2] that parametrize the right-hand side function of an ODE as a deep-neural network and then train the network parameters by integrating the corresponding ODE forward in time. Neural ODEs have been used in the context of model reduction in, e.g., [51, 52], where the network additionally depends on a physical input. There also is work on fitting Neural ODEs to the coefficients of projected data [53]. Similarly, the work [54] trains a network to predict the next state of reduced models; there also is [55] that use deep networks to make predictions into the future but no roll outs are used, which avoids differentiating through a time-integration scheme. Bayesian formulations of analogous concepts have been proposed as well [56]. There is a line of works [57, 58] that aims to identify a noise model of the data, which is different from our approach that aims to regularize against noise via roll outs rather than aiming to explicitly identify a model of the noise.

We leverage the insights gained from the works on Neural ODEs and related methods and combine them with the interpretability and scalability of operator inference. In particular, the parametrization of models via low-degree polynomials of traditional operator inference is beneficial also when using roll outs. Using polynomial models separates linear from nonlinear dynamics, which means that semi-implicit time-integration schemes can be applied. Furthermore, the polynomial structure allows estimating the stability radius of the domain of attraction of the learned models, which provides an analysis of the learned models beyond mere numerical experiments on test sets. Finally, interpolating between polynomial models to quickly derive new models for new physical inputs can be achieved in a structure preserving way by building on widely used techniques from model reduction for interpolating on matrix manifolds [59, 60, 61, 62].

With roll outs, we show that we gain an increase of robustness of operator inference. Roll outs lead to an implicit stability bias, which can be interpreted as a regularizer that helps when only few data samples are available. Similarly, if data are polluted with noise, roll outs help to prevent overfitting, as our numerical experiments will show. Finally, traditional operator inference typically estimates time derivatives via finite differences, which is prone to inaccuracies in the presence of already little noise and perturbations. Instead, by using roll outs, the time integration scheme is included in the learning process. Thus, the to-be-learned model is discretized in time and only state observations of the underlying systems are required, without time-derivative approximations of the observed states of the underlying high-dimensional system from which data are sampled; see also [63] for similar observations in the context of dictionary-based learning methods. We present numerical experiments that indicate that operator inference with roll outs can provide predictive models even from coarsely sampled training data trajectories and with noise of up to 10%.

The manuscript is organized as follows. In Section 2, preliminaries are reviewed, such as the traditional formulation of operator inference. Section 3 introduces operator inference with roll outs by building on Neural ODEs [2]. The section also discusses the benefits of polynomial models in terms of interpretability,

scalability, and generalization to new physical inputs as well as implementation aspects of operator inference with roll outs. Numerical results are demonstrated in Section 4 and conclusions are drawn in Section 5.

## 2 Preliminaries

We describe the data collection process in Section 2.1 and operator inference [1] for learning reduced models from data in Section 2.2. The problem formulation is given in Section 2.3 to highlight that the operator-inference formulation of [1] is static in the sense that the training loss penalizes the prediction of only a single time step into the future.

### 2.1 Collecting data and low-dimensional structure

Consider a process  $(\mathbf{q}_t(\boldsymbol{\mu}))_{t \geq 0}$  with state  $\mathbf{q}_t(\boldsymbol{\mu}) \in \mathbb{R}^N$  at time  $t$  that is obtained for a control trajectory  $(\mathbf{u}_t(\boldsymbol{\mu}))_{t \geq 0}$  with control  $\mathbf{u}_t(\boldsymbol{\mu}) \in \mathcal{U} \subset \mathbb{R}^p$  at time  $t$ , initial condition  $\mathbf{q}_0(\boldsymbol{\mu}) \in \mathcal{Q}_0 \subseteq \mathbb{R}^N$ , and input  $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d$ . Let  $\mathcal{D}_{\text{train}} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M\} \subset \mathcal{D}$  be a set of  $M$  training inputs. For each training input  $\boldsymbol{\mu}_i \in \mathcal{D}_{\text{train}}$ , there is given an initial condition  $\mathbf{q}_0(\boldsymbol{\mu}_i) \in \mathcal{Q}_0$  and a time-discrete control trajectory

$$\mathbf{U}(\boldsymbol{\mu}_i) = [\mathbf{u}_1(\boldsymbol{\mu}_i), \dots, \mathbf{u}_K(\boldsymbol{\mu}_i)] \in \mathbb{R}^{p \times K}$$

and the corresponding time-discrete state trajectory

$$\mathbf{Q}(\boldsymbol{\mu}_i) = [\mathbf{q}_1(\boldsymbol{\mu}_i), \dots, \mathbf{q}_K(\boldsymbol{\mu}_i)] \in \mathbb{R}^{N \times K}$$

at discrete time steps  $0 = t_0 < t_1 < \dots < t_K = T$ . For ease of exposition, we assume equidistant time steps with time-step size  $\delta t$ . However, the following methodology extends in a straightforward way to non-equidistant time steps. Additionally, in the following, we consider processes with a low-dimensional structure so that an accurate reduced model exists. Thus, there is a basis matrix  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{N \times n}$  of the space  $\mathcal{V}$  with dimension  $n \ll N$  that is obtained, for example, via the principal component analysis from the snapshots

$$\mathbf{Q} = [\mathbf{q}_0(\boldsymbol{\mu}_1), \mathbf{Q}(\boldsymbol{\mu}_1), \mathbf{q}_0(\boldsymbol{\mu}_2), \mathbf{Q}(\boldsymbol{\mu}_2), \dots, \mathbf{q}_0(\boldsymbol{\mu}_M), \mathbf{Q}(\boldsymbol{\mu}_M)] \in \mathbb{R}^{N \times M(K+1)}.$$

In the space  $\mathcal{V}$ , the states of the process  $(\mathbf{q}_t(\boldsymbol{\mu}))_{t \geq 0}$  can be approximated well; we refer to [64, 65, 66, 67] for model reduction in general.

### 2.2 Static learning of low-dimensional models via operator inference

Operator inference was introduced in [1] to learn low-dimensional models from data. It proceeds in three steps: First, the trajectories are projected onto the reduced space  $\mathcal{V}$

$$\bar{\mathbf{Q}}(\boldsymbol{\mu}_i) = [\bar{\mathbf{q}}_1(\boldsymbol{\mu}_i), \dots, \bar{\mathbf{q}}_K(\boldsymbol{\mu}_i)] = \mathbf{V}^T \mathbf{Q}(\boldsymbol{\mu}_i), \quad i = 1, \dots, M. \quad (1)$$

Second, the time derivatives of projected states  $\bar{\mathbf{q}}_j(\boldsymbol{\mu}_i)$  at time  $t_j$  for  $i = 0, \dots, M$  and  $j = 1, \dots, K$  are approximated via, e.g., finite differences, to obtain the approximate time derivatives

$$\bar{\mathbf{Q}}'(\boldsymbol{\mu}_i) = [\bar{\mathbf{q}}'_1(\boldsymbol{\mu}_i), \dots, \bar{\mathbf{q}}'_K(\boldsymbol{\mu}_i)], \quad i = 1, \dots, M. \quad (2)$$

Third, a low-dimensional model with polynomial structure is fitted to the projected states (1) and their approximate time derivatives (2) via a least-squares problem.

Operator inference was developed for fitting polynomial models of degree  $L \in \mathbb{N}$ ,

$$\frac{d}{dt} \hat{\mathbf{q}}(t; \boldsymbol{\mu}) = \sum_{\ell=1}^L \mathbf{A}_\ell(\boldsymbol{\mu}) \hat{\mathbf{q}}^\ell(t; \boldsymbol{\mu}) + \mathbf{B}(\boldsymbol{\mu}) \mathbf{u}(t; \boldsymbol{\mu})$$

with matrices  $\mathbf{A}_\ell(\boldsymbol{\mu}) \in \mathbb{R}^{n \times n_\ell}$  for  $\ell = 1, \dots, L$  and control matrix  $\mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times p}$ , where  $n_\ell = \binom{n+\ell-1}{\ell}$  because  $\hat{\mathbf{q}}^\ell(t; \boldsymbol{\mu})$  contains only combinations that are unique up to commutativity; see [1] for details. For all  $i = 1, \dots, M$  training inputs, the corresponding model operators  $\mathbf{A}_1(\boldsymbol{\mu}_i), \dots, \mathbf{A}_L(\boldsymbol{\mu}_i), \mathbf{B}(\boldsymbol{\mu}_i)$  are fitted via the following linear least-squares problem

$$\min_{\mathbf{A}_1, \dots, \mathbf{A}_L, \mathbf{B}} \frac{1}{K} \sum_{k=1}^K \left\| \sum_{\ell=1}^L \mathbf{A}_\ell \bar{\mathbf{q}}_k^\ell(\boldsymbol{\mu}_i) + \mathbf{B} \mathbf{u}_k(\boldsymbol{\mu}_i) - \bar{\mathbf{q}}_k' \right\|_2^2, \quad (3)$$

which gives the inferred operators

$$\mathbf{A}_1(\boldsymbol{\mu}_i), \dots, \mathbf{A}_L(\boldsymbol{\mu}_i), \mathbf{B}(\boldsymbol{\mu}_i). \quad (4)$$

For a new input  $\boldsymbol{\mu} \in \mathcal{D} \setminus \mathcal{D}_{\text{train}}$ , the inferred operators for all  $i = 1, \dots, M$  training inputs can be interpolated as shown in [1]; see also the work [37] that proposes operator inference for problems with an affine parameter dependence.

The optimization problem (3) is linear in the optimization variables  $\mathbf{A}_1, \dots, \mathbf{A}_L, \mathbf{B}$  and thus can be solved efficiently with standard linear algebra routines. This can be seen by reformulating (3) as

$$\mathbf{D}(\boldsymbol{\mu}_i) \mathbf{O}(\boldsymbol{\mu}_i) = \mathbf{R}(\boldsymbol{\mu}_i), \quad (5)$$

with matrices  $\mathbf{D}(\boldsymbol{\mu}_i) \in \mathbb{R}^{K \times \bar{n}}$ ,  $\mathbf{O}(\boldsymbol{\mu}_i) = [\mathbf{A}_1^T(\boldsymbol{\mu}_i), \dots, \mathbf{A}_L^T(\boldsymbol{\mu}_i), \mathbf{B}^T(\boldsymbol{\mu}_i)]^T \in \mathbb{R}^{\bar{n} \times n}$  and  $\mathbf{R}(\boldsymbol{\mu}_i) \in \mathbb{R}^{K \times n}$ , where  $\bar{n} = p + \sum_{\ell=1}^L n_\ell$ . The formulation also shows that at least  $\bar{n}$  rows in  $\mathbf{D}(\boldsymbol{\mu}_i)$  are required, i.e.,  $K \geq \bar{n}$  to obtain a full-rank matrix and thus to avoid an under-determined least-squares problem with non-unique solution; note that if the states are observed with the re-projection sampling scheme, then recovery guarantees can be established under assumptions on the data [47].

If fewer state observations are available, regularization can be helpful, see, e.g., [45, 43, 48]. In the following, we use the minimal-norm solution and thus solve

$$\begin{aligned} \min_{\mathbf{O}(\boldsymbol{\mu}_i)} \quad & \|\mathbf{O}(\boldsymbol{\mu}_i)\|_F^2, \\ \text{such that} \quad & \mathbf{D}(\boldsymbol{\mu}_i) \mathbf{O}(\boldsymbol{\mu}_i) = \mathbf{R}(\boldsymbol{\mu}_i), \end{aligned} \quad (6)$$

for  $i = 1, \dots, M$ , which can be solved with standard linear algebra routines.

## 2.3 Problem formulation

We refer to a formulation such as (3) as static in the following because the prediction of the learned model is taken into account only for one time step into the future during training. This means, the loss function only compares the residual of the learned model from one time step at a time, without predicting the states during the training process. This has several drawbacks that are inherent to such static formulations:

First, the problem formulation is affected by the sensitivity of least-squares regression to perturbation and noise, which is discussed in the context of operator inference in the work [46] and later in [49]. For example, the perturbation can come from the approximation of the time derivatives (2) and from partial state observations [39]. Another source of perturbations can be a wrong model frame (parametrization) in the sense that, for example, a quadratic model is fitted to data that are observed from a system with cubic dynamics. Noise can be introduced because the observed states are polluted with noise and other numerical perturbations such as early stopping in iterative solvers, as considered in [49]. These works show that even small perturbations can lead to large errors in the predictions of the learned models as well as to instabilities.

Second, without regularization, problem (3) has multiple solutions if too few data points are available to obtain a full-rank data matrix  $\mathbf{D}$  in (5). While regularization has been proposed for operator inference, it introduces a bias by design and the effect of the bias on the model prediction is still actively studied [45, 43, 48]. The sensitivity to perturbations in the data and the potentially high requirements of training data can be limitations when learning models with operator inference in practice.

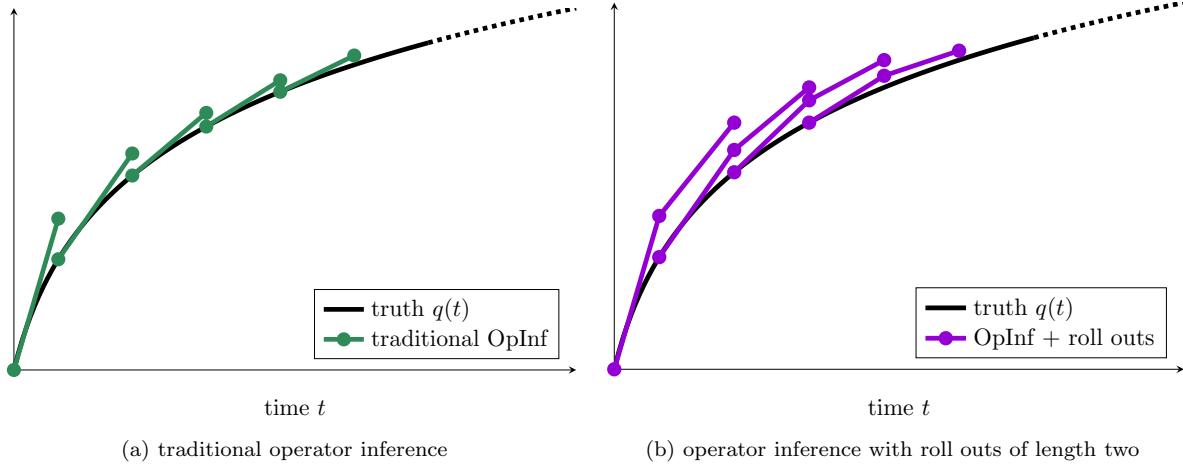


Figure 1: The loss of operator inference with roll outs penalizes model predictions of more than one time step into the future, which increases robustness against instabilities, noise, and perturbations in our numerical experiments. In contrast, traditional operator inference minimizes a loss that takes into account predictions of only one time step into the future and thus cannot penalize errors in long-term predictions of the learned models.

### 3 Operator inference with roll outs

We to roll out the learned model during the training process so that the error of the trajectory predicted over multiple time steps into the future by the learned model is penalized rather than the residual at the current time step alone; cf. Figure 1. Note that the concept of roll outs are extensively used in Neural ODEs [2]. Roll outs provide an inherent stability prior because models that are unstable during the training roll outs get strongly penalized. Similarly, our numerical experiments will demonstrate that the new formulation of operator inference is more robust against perturbations in the data because the whole predicted trajectory is considered in the training loss rather than only a single time step. The drawback is that the training problem becomes more challenging; however, we leverage the recent advances in automatic differentiation to efficiently compute the gradients of the learned model.

We formulate operator inference with rollouts in Section 3.1 and Section 3.2 and discuss polynomial models in the context of differentiable operator inference in Section 3.3. The stability bias introduced by the roll outs is analyzed in Section 3.4. The computational procedure is discussed in Section 3.5.

#### 3.1 General setup of differentiable non-intrusive model reduction

Consider a function  $\hat{f} : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \times \Theta \rightarrow \mathbb{R}^n$  that depends on a parameter  $\theta \in \Theta \subset \mathbb{R}^p$ , which we want to learn so that  $\hat{f}$  describes the dynamics of the observed data,

$$\frac{d}{dt} \hat{q}(t; \mu) = \hat{f}(\hat{q}(t; \mu), \mathbf{u}(t), \mu; \theta). \quad (7)$$

Notice that this setup follows, e.g., Neural ODEs where the function  $\hat{f}$  is given by a deep neural network [2]; see also [52, 53]. Based on model (7), define the discrete flow map

$$\hat{q}_k(\mu) = \hat{\Phi}_{\delta t}(\hat{q}_{k-1}(\mu), \mathbf{u}_{k-1}(\mu), \mu; \theta), \quad k = 1, \dots, K$$

with time-step size  $\delta t > 0$ , which depends on a time-discretization scheme. Explicit and implicit discretization schemes can be used and the discrete flow map can depend on a history of past time steps rather than just the previous one. The discrete flow map  $\hat{\Phi}_{\delta t}$  depends on the input  $\theta$  through the function  $\hat{f}$ .

Consider now a roll-out length  $R \in \mathbb{N}$  and the objective function  $J : \Theta \times \mathcal{D} \rightarrow \mathbb{R}$  defined as

$$J(\boldsymbol{\theta}; \boldsymbol{\mu}) = \sum_{k=0}^{K-R} \sum_{r=1}^R \|\bar{\mathbf{q}}_{k+r}(\boldsymbol{\mu}) - \hat{\mathbf{z}}_{k,r}(\boldsymbol{\mu})\|^2, \quad (8)$$

where

$$\hat{\mathbf{z}}_{k,r}(\boldsymbol{\mu}) = \hat{\Phi}_{\delta t}(\hat{\mathbf{z}}_{k,r-1}(\boldsymbol{\mu}), \mathbf{u}_{k+r-1}(\boldsymbol{\mu}), \boldsymbol{\mu}; \boldsymbol{\theta}), \quad \hat{\mathbf{z}}_{k,0}(\boldsymbol{\mu}) = \bar{\mathbf{q}}_k(\boldsymbol{\mu})$$

with  $\bar{\mathbf{q}}_k(\boldsymbol{\mu})$  the projected observed state at time step  $k$  with input  $\boldsymbol{\mu}$ ; cf. Eq. (1). Instead of penalizing the misfit between data and model predictions at all  $R$  time steps as in (8), we can also instead penalize the misfit at  $\hat{R}$  time increments  $r_1, \dots, r_{\hat{R}}$  such that  $r_1 \leq \dots \leq r_{\hat{R}} \leq R$ . The corresponding objective is

$$J_{\hat{R}}(\boldsymbol{\theta}; \boldsymbol{\mu}) = \sum_{k=0}^{K-R} \sum_{\ell=1}^{\hat{R}} \|\bar{\mathbf{q}}_{k+r_\ell}(\boldsymbol{\mu}) - \hat{\mathbf{z}}_{k,r_\ell}(\boldsymbol{\mu})\|^2. \quad (9)$$

The training problem of operator inference with roll outs is the optimization problem

$$\min_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}; \boldsymbol{\mu}), \quad (10)$$

which is solved for all training inputs  $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M$  to obtain the model parameters  $\boldsymbol{\theta}(\boldsymbol{\mu}_1), \dots, \boldsymbol{\theta}(\boldsymbol{\mu}_M)$ . The objective in (10) includes the flow map of the to-be-trained model and thus the objective takes into account the predictions of the to-be-trained model, in contrast to the traditional operator inference problem (3).

For each training input  $\boldsymbol{\mu}_i \in \mathcal{D}_{\text{train}}$ , one obtains a low-dimensional model (7) that one can integrate with new control  $\mathbf{u}(t) \in \mathbb{R}^p$  and new initial condition  $\hat{\mathbf{q}}_0 \in \mathbb{R}^n$ . To compute a low-dimensional model at a new input  $\boldsymbol{\mu} \in \mathcal{D} \setminus \mathcal{D}_{\text{train}}$ , the parameters  $\boldsymbol{\theta}(\boldsymbol{\mu}_1), \dots, \boldsymbol{\theta}(\boldsymbol{\mu}_M)$  are interpolated at  $\boldsymbol{\mu}$ ; note that this requires the model (7) to be chosen such that interpolating the model parameters indeed leads to models that describe well the dynamics at the new inputs; which we will discuss in the context of polynomial models in detail in Section 3.3.

### 3.2 Model predictive control and operator inference with roll outs

We formulate operator inference with rollouts as (10), which is a classical optimization problem that follows the concept of unrolling of Neural ODEs [2]; however, the formulation with objective (8) is equivalent to solving the following constrained optimization problem

$$\min_{\boldsymbol{\theta} \in \Theta} \sum_{k=0}^{K-R} \sum_{r=1}^R \|\bar{\mathbf{q}}_{k+r}(\boldsymbol{\mu}) - \hat{\mathbf{z}}_{k,r}(\boldsymbol{\mu})\|^2 \quad (11)$$

$$\text{such that } \hat{\mathbf{z}}_{k,r} = \hat{\Phi}_{\delta t}(\hat{\mathbf{z}}_{k,r-1}, \mathbf{u}_{k+r-1}(\boldsymbol{\mu}), \boldsymbol{\mu}; \boldsymbol{\theta}), \quad (12)$$

$$\hat{\mathbf{z}}_{k,0} = \bar{\mathbf{q}}_k(\boldsymbol{\mu}). \quad (13)$$

Such constrained optimization problems are well studied in the context of parameter calibration and optimal control [68], offering alternative solution strategies using Lagrange formulations. In particular, Lagrange formulations allow efficient realizations in software frameworks without automatic differentiation capabilities, with the corresponding adjoint formulations being straightforward to implement<sup>1</sup>. We also want to highlight the connection to so called solver-in-the-loop approaches, which currently gain attention in the context of enriching solvers of PDEs with two- and three-dimensional spatial domains with machine-learning models [69, 70], mostly leveraging the JAX library [71].

---

<sup>1</sup>Solving the adjoint formulation means solving a linear ODE backwards in time to determine the Lagrange multipliers.

### 3.3 Parametrizations via polynomials in model reduction

A wide range of parametrizations can be used for formulating the function  $\hat{\mathbf{f}}$  in (7) such as deep networks, as has been done in [2, 52]. However, in this work, we consider  $\hat{\mathbf{f}}$  to be a polynomial in the state and control, to mimic operator inference<sup>2</sup>. Thus, in the following, the function  $\hat{\mathbf{f}}$  is given by

$$\hat{\mathbf{f}}(\hat{\mathbf{q}}, \mathbf{u}, \boldsymbol{\mu}; \boldsymbol{\theta}) = \sum_{\ell=1}^L \mathbf{A}_\ell(\boldsymbol{\mu}) \hat{\mathbf{q}}^\ell + \mathbf{B}(\boldsymbol{\mu}) \mathbf{u},$$

where we have a linear dependence on the control  $\mathbf{u}$  and thus

$$\boldsymbol{\theta}(\boldsymbol{\mu}) = [\mathbf{A}_1(\boldsymbol{\mu}), \dots, \mathbf{A}_L(\boldsymbol{\mu}), \mathbf{B}(\boldsymbol{\mu})].$$

The use of polynomial functions has several advantages. First, interpolating an operator  $\mathbf{A}_i(\boldsymbol{\mu})$  between the parameters  $\boldsymbol{\theta}(\boldsymbol{\mu}_1), \dots, \boldsymbol{\theta}(\boldsymbol{\mu}_M)$  at a new input  $\boldsymbol{\mu} \in \mathcal{D} \setminus \mathcal{D}_{\text{train}}$  means to interpolate among the operators  $\{\mathbf{A}_i(\boldsymbol{\mu}_1), \dots, \mathbf{A}_i(\boldsymbol{\mu}_M)\}$  for each  $i = 1, \dots, \ell$ , which often leads to an interpolated parameter  $\boldsymbol{\theta}(\boldsymbol{\mu})$  that indeed leads to a predictive model of the process corresponding to  $\boldsymbol{\mu}$ ; this has been successfully leveraged in model reduction [59, 60, 61, 62] and operator inference [1] for a long time.

Second, in case of polynomial models, the model parameter  $\boldsymbol{\theta}(\boldsymbol{\mu}) = [\mathbf{A}_1(\boldsymbol{\mu}), \dots, \mathbf{A}_\ell(\boldsymbol{\mu}), \mathbf{B}(\boldsymbol{\mu})]$  is interpretable. For example, the operators provide insights into the stability radius of the model; cf. [72, 73, 74, 75], which we will also demonstrate in the numerical experiments.

Third, the number of degrees of freedom of a polynomial model, i.e., the dimension of  $\boldsymbol{\theta}(\boldsymbol{\mu})$  is low, as long as the degree  $L$  of the polynomial stays low. As shown theoretically and numerically in [76, 36], low-degree polynomials are sufficient for describing the dynamics of a wide range of applications of interest in science and engineering. Additionally, it is wide engineering practice to approximate nonlinear behavior with low-degree polynomials derived from truncated Taylor expansions.

Fourth, polynomial models explicitly split the dynamics into linear and nonlinear dynamics via the matrices  $\mathbf{A}_1$  and  $\mathbf{A}_i, i > 1$ . This is helpful when applying semi-implicit time integration schemes that are explicit in the nonlinear dynamics and implicit in the linear ones. A decomposition into linear and nonlinear dynamics typically has to be explicitly enforced in other, more generic nonlinear parametrizations.

### 3.4 Stability of polynomial models obtained with operator inference with roll outs

To give insights into the stability bias imposed on the polynomial models by the roll outs, it is informative to look at the objective (8). For making the following arguments, it is sufficient consider an autonomous linear model of the form  $\hat{\mathbf{f}}(\hat{\mathbf{q}}, \mathbf{u}, \boldsymbol{\mu}; \boldsymbol{\theta}) = \mathbf{A}_1 \hat{\mathbf{q}}$ , where we assume that  $\mathbf{A}_1$  is symmetric. We use a forward Euler time discretization scheme with time-step size  $\delta t > 0$ . The corresponding loss with  $\boldsymbol{\theta} = [\mathbf{A}_1]$  is

$$J(\boldsymbol{\theta}) = \sum_{k=0}^{K-R} \sum_{r=1}^R \left\| \hat{\mathbf{q}}_{k+r} - (\mathbf{I} + \delta t \mathbf{A}_1)^r \hat{\mathbf{q}}_k \right\|_2^2. \quad (14)$$

The loss (14) shows that unstable  $\mathbf{A}_1$ —in the sense that  $\mathbf{I} + \delta t \mathbf{A}_1$  has eigenvalues with positive real parts—are penalized because of their unbounded grow with increasing roll-out length  $R$ . Similar observations can be made for other parametrizations than autonomous linear polynomial models, with an empirical demonstration provided by our numerical experiments in the next section.

### 3.5 Computational aspects

The objective (8) in the training problem is nonlinear in the optimization variable  $\boldsymbol{\theta}$ , which is in stark contrast to the linear objective of the traditional, static operator inference problem (3). In the following,

---

<sup>2</sup>In general, approximating nonlinear behavior with truncated Taylor expansion is wide engineering practice.

we use derivative-based optimization methods to numerically solve (10) and so to train the low-dimensional models. Thus, we have to differentiate through the flow map  $\Phi_{\delta t}$  with respect to the parameter  $\theta(\mu)$ . Gradients and also higher-order derivative information can be derived explicitly for polynomial models in many cases; see [50] for a derivation of the gradients in the context of operator inference. However, we instead opt to use the widely available automatic differentiation libraries such as [77, 78] for the computations in this work.

The training costs of differentiable operator inference can be higher than in traditional, static operator inference. First, the number of iterations required by derivative-based optimization methods such as gradient descent and Broyden-Fletcher-Goldfarb-Shanno to reduce the gradient norm below a threshold is a priori unknown in many cases. Second, at each objective evaluation, the to-be-learned model is integrated in time, which is computationally more expensive than computing the residual after one time step as in traditional operator inference. However, as we will empirically demonstrate with the numerical experiments in the following section, by investing higher training costs, one can obtain models with higher robustness against noise and perturbations; even from fewer training data, which is well in line with other works such as [79].

## 4 Numerical experiments

The numerical examples in this section demonstrate the following key points:

- Operator inference with roll outs compensates for scarce training data by penalizing the misfit of model predictions at times far out in the future, resulting oftentimes in more stable and accurate models than traditional operator inference that solves a possibly underdetermined least squares problem in the scarce data regime that leads to unreliable predictions.
- Increasing the roll-out length can improve the accuracy of the predictions of the learned models and help to mitigate the effect of noise by preventing overfitting when learning models from noisy data.
- If trajectories are sparsely sampled in time, then roll outs can provide meaningful values between sampled states and so lead to more accurate models, whereas traditional operator inference is required to take coarser approximations of the time derivatives.
- Computation of the stability radius of low-dimensional polynomially nonlinear models learned via operator inference with and without roll outs offer interpretability beyond mere numerical experiments on test data. The results show that roll outs help to increase upper bounds of stability radii, which indicates that roll outs lead to models that are more robust against disturbances and are applicable to a wider range of initial conditions.

Section 4.1 discusses the numerical implementation of the proposed approach as well as strategies for training the low-dimensional models with operator inference with roll outs. Numerical experiments are conducted with data sampled from models that describe shallow water waves in Section 4.2 and geophysical fluids in Section 4.3. The hyperparameters of the training procedure mentioned above are tuned using validation data. We chose the hyperparameter configuration that achieved the smallest error on the validation data in all of the learned models to ensure a fair comparison.

### 4.1 Numerical implementation

We implemented operator inference with roll outs in JAX<sup>3</sup> [78] to leverage automatic differentiation when solving the training problem (10). The Adam, adaptive moment estimation, optimizer [80] is used to minimize the objective function. We choose learning rates logarithmically spaced between  $10^{-5}$  and  $10^{-1}$ . For initial conditions to the non-convex optimization problem (10), we either start with zeros or use the model operators from traditional operator inference. For obtaining models with traditional operator inference, we solve the problem (6), which gives the minimal-norm solution in case of an underdetermined system.

---

<sup>3</sup><https://jax.readthedocs.io/en/latest/>

## 4.2 Dynamics of shallow water waves

We consider waves governed by the shallow water equations and demonstrate learning a model with operator inference with roll outs from noisy and scarce data.

### 4.2.1 Shallow water equations: Setup

Consider the two-dimensional spatial domain  $\Omega = (-4, 4)^2 \subset \mathbb{R}^2$  and the time domain  $\mathcal{T} = (0, T) \subset \mathbb{R}$  with  $T = 0.15$ . The shallow water equations are given by

$$\begin{aligned} \partial_t q_h + \nabla \cdot (q_h \nabla q_\phi) &= 0, & \text{in } \Omega \times \mathcal{T}, \\ \partial_t q_\phi + \frac{1}{2} |\nabla q_\phi|^2 + q_h &= 0, & \text{in } \Omega \times \mathcal{T}. \end{aligned}$$

The system consists of the scalar potential of the fluid given by  $q_\phi$  and the height of the free-surface  $q_h$ , which is normalized by its mean value.

The input  $\boldsymbol{\mu} = [\mu_1, \mu_2] \in \mathcal{D} = [0.2, 0.5] \times [1.1, 1.7]$  determines the initial condition

$$q_h^0(\mathbf{x}; \boldsymbol{\mu}) = 1 + \mu_1 e^{-\mu_2 \|\mathbf{x}\|_2^2}, \quad q_\phi^0(\mathbf{x}; \boldsymbol{\mu}) = 0,$$

where  $q_h(0, \mathbf{x}; \boldsymbol{\mu}) = q_h^0(\mathbf{x}; \boldsymbol{\mu})$  and  $q_\phi(0, \mathbf{x}; \boldsymbol{\mu}) = q_\phi^0(\mathbf{x}; \boldsymbol{\mu})$ . We impose periodic boundary conditions. The equations are discretized in space with second-order finite difference schemes on 60 equidistant grid points in each direction so that  $N = 7200$ . The time integration scheme used is the implicit midpoint rule for Hamiltonian systems with the time-step size being  $\delta t = 0.0004$ .

We learn a model of dimension  $n = 25$  in this example. To obtain the low-dimensional basis  $\mathbf{V}$ , we generate snapshot trajectories using 12 initial conditions with inputs  $[\mu_1^{\text{basis}}, \mu_2^{\text{basis}}]$  where  $\mu_1^{\text{basis}} \in \{0.2, 0.35, 0.5\}$  and  $\mu_2^{\text{basis}} \in \{1.1, 1.3, 1.5, 1.7\}$ . Training data are obtained by generating trajectories whose initial conditions correspond to three nested sets of training inputs representing increasing levels of information. The training set of inputs is constructed as follows:

$$\begin{aligned} \boldsymbol{\mu}^{\text{train-I}} &= \{(0.225, 1.15), (0.225, 1.65), (0.475, 1.15), (0.475, 1.65)\}, \\ \boldsymbol{\mu}^{\text{train-II}} &= \boldsymbol{\mu}^{\text{train-I}} \cup \{(0.225, 1.4), (0.35, 1.4), (0.475, 1.4)\}, \\ \boldsymbol{\mu}^{\text{train-III}} &= \boldsymbol{\mu}^{\text{train-II}} \cup \{(0.35, 1.65), (0.35, 1.15)\}. \end{aligned}$$

The hyperparameters are tuned using a validation data set, which contains trajectories whose initial conditions correspond to the validation inputs

$$\boldsymbol{\mu}^{\text{valid}} = \{(0.2875, 1.4), (0.35, 1.525), (0.35, 1.275), (0.4125, 1.4)\}.$$

The test data used to investigate the performance of the learned model is generated using initial conditions with inputs  $[\mu_1^{\text{test}}, \mu_2^{\text{test}}]$  where  $\mu_1^{\text{test}} \in \{0.2875, 0.4125\}$  and  $\mu_2^{\text{test}} \in \{1.275, 1.525\}$ .

An implicit-explicit scheme is utilized to integrate the learned low-dimensional model over  $t \in (0, T)$  with  $\delta t = 0.0004$  in which the linear and quadratic terms are evaluated implicitly and explicitly, respectively. The same implicit-explicit scheme with the same time-steps size is also used for the roll outs. In the experiments below, we chose (8) as the objective function. In addition to penalizing the misfit between the data and model predictions at all  $R$  time points as in (8), we will also consider the objective (9). In the latter, we measure the misfit at  $\hat{R} = 25$  time points that are equidistant between 1 and  $R$  in the logarithmic scale.

When training low-dimensional models using operator inference with roll outs, we select the hyperparameter configuration which results in the lowest validation error given by

$$\frac{1}{M_{\text{valid}}} \sum_{j=1}^{M_{\text{valid}}} \frac{\sum_{k=0}^K \|\mathbf{q}_k(\boldsymbol{\mu}_j^{\text{valid}}) - \mathbf{V} \hat{\mathbf{q}}_k(\boldsymbol{\mu}_j^{\text{valid}})\|_F}{\sum_{k=0}^K \|\mathbf{q}_k(\boldsymbol{\mu}_j^{\text{valid}})\|_F}, \quad (15)$$

where  $M_{\text{valid}}$  denotes the number of test parameters. The learned low-dimensional operators from this configuration are then used to assemble a model, which is used for predictions at test parameters and test initial conditions.

### 4.2.2 Shallow water equations: Scarce data

We demonstrate the performance of the low-dimensional models learned with traditional operator inference and operator inference with roll outs in the scarce data regime by varying the number of initial conditions utilized to generate state trajectories for training data. First, we only train with trajectories obtained from four initial conditions and we set the roll length to  $R = 100$ . Figure 2 shows ground-truth test data of the free-surface height computed with the truth model, the projection of the ground-truth data onto the low-dimensional subspace  $\mathcal{V}$ , which gives the best approximation that can be achieved, the predictions of the model with traditional, static operator inference, and the predictions of the model obtained with roll outs at time  $t = 0.15$  at the test parameter values. The plots show that the model learned with roll outs captures the complex patterns of the ground truth while static operator inference appears to overfit to certain patterns in this example.

We now investigate how the number of trajectories used for training as well as the roll-out length  $R$  influence the performance of operator inference. Figure 3 shows the relative error averaged over the test inputs over time of the low-dimensional models as a function of the number of training trajectories  $M_{\text{train}}$  and the roll-out length  $R$ . The plotted error is computed as

$$\frac{1}{M_{\text{test}}} \sum_{j=1}^{M_{\text{test}}} \frac{\sum_{k=0}^K \|\mathbf{q}_k(\boldsymbol{\mu}_j^{\text{test}}) - \mathbf{V}\hat{\mathbf{q}}_k(\boldsymbol{\mu}_j^{\text{test}})\|_F}{\sum_{k=0}^K \|\mathbf{q}_k(\boldsymbol{\mu}_j^{\text{test}})\|_F} \quad (16)$$

where  $M_{\text{test}}$  denotes the number of test parameters. The plots in Figure 3 show that roll outs during training help operator inference to learn models that make more accurate predictions on the test data set than traditional operator inference without roll outs, especially for the scarce data regime with a low number of training trajectories.

Parametrizing low-dimensional models via quadratic polynomials offers a tractable representation that permits us to study properties of the learned models such as stability. For quadratic models, we compute a bound of the stability radius [72, 73, 75] that is derived in [74] and given by

$$\gamma = \frac{\sigma_{\min}(\mathbf{L})}{2\sqrt{\|\mathbf{P}\|_F \|\mathbf{A}_2\|_F}}$$

where  $\mathbf{P} \in \mathbb{R}^{n \times n}$  satisfies the continuous Lyapunov equation  $\mathbf{A}_1^T \mathbf{P} + \mathbf{P} \mathbf{A}_1 = -\mathbf{L}\mathbf{L}^T$  for specified  $\mathbf{L} \in \mathbb{R}^{n \times n}$ . We generate 1000 realizations of a random matrix with independent standard normal entries for  $\mathbf{L}$  and then we compute the average stability radius of the learned quadratic models. The bounds of the stability radii of the learned low-dimensional models are shown in Figure 4. The results indicate that the stability radii of the learned models via operator inference with roll outs are orders of magnitude larger than those learned with traditional operator inference. These results agree with the observation discussed in Section 3 that roll outs impose a stability bias. The plots are also consistent with the fact that operator inference with roll outs offers more accurate predictions in this example than models learned with traditional operator inference.

### 4.2.3 Shallow water equations: Noisy and scarce data

We now consider noisy data that are generated as

$$\bar{\mathbf{q}}_k(\boldsymbol{\mu}) + \rho \boldsymbol{\varepsilon}_k \odot |\bar{\mathbf{q}}_k(\boldsymbol{\mu})|, \quad k = 1, \dots, K-1,$$

where  $\boldsymbol{\varepsilon}_k \sim N(0, \mathbf{I})$  is standard normal and  $\rho \in \{0.1\%, 1\%, 10\%\}$  controls the standard deviation of the noise. The  $\odot$  means component-wise multiplication. The plots in Figure 5 show the time-averaged relative error (16) of the model predictions across various noise levels for learning from  $M_{\text{train}} \in \{4, 7, 9\}$  noisy training trajectories with roll-out length  $R = 150$ . Despite having noisy training data, operator inference with roll outs still learns a numerically stable low-dimensional model. The error of the model learned with roll outs increases much slower than the model learned without roll outs, which indicates that the roll outs reduce the

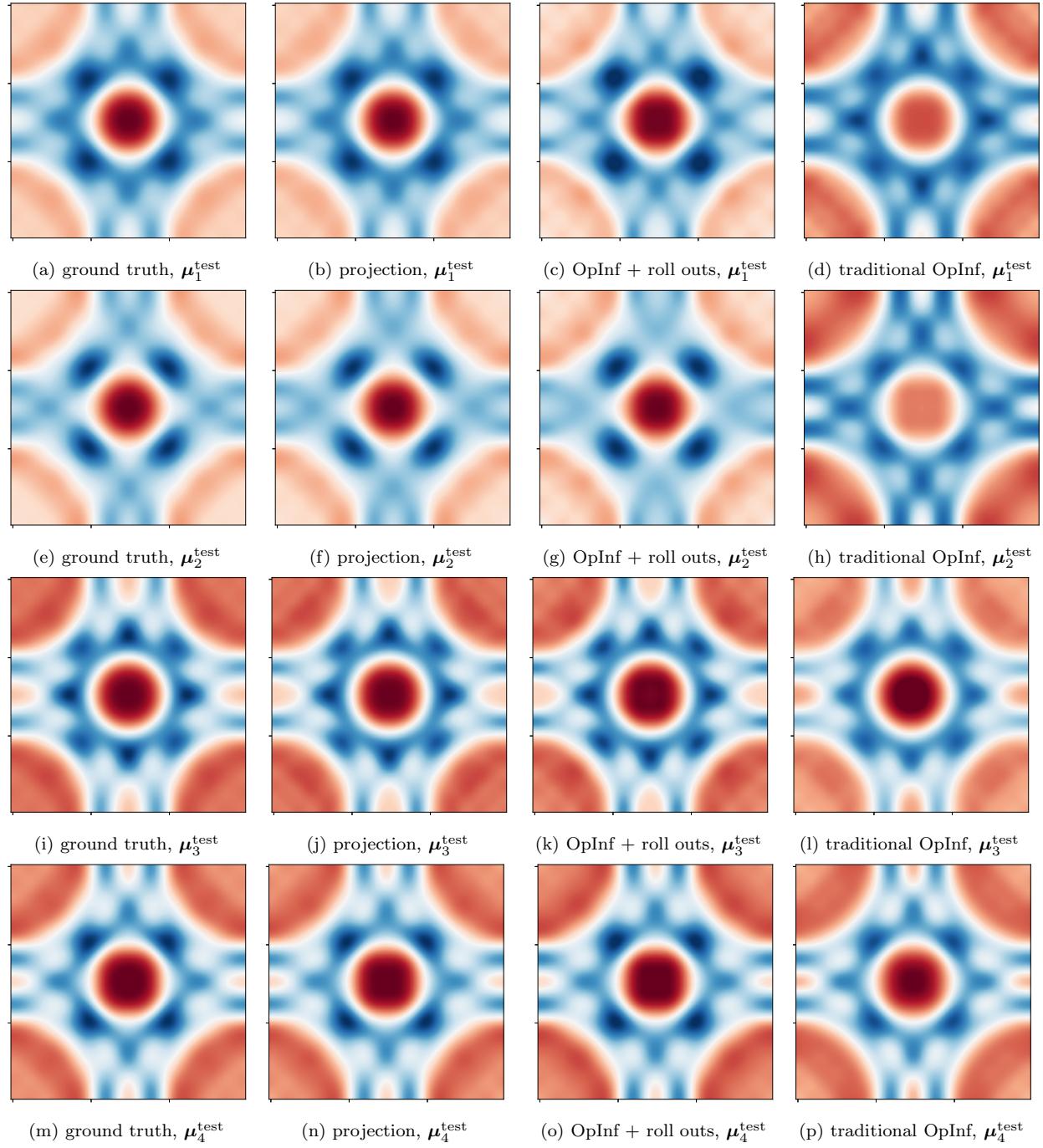


Figure 2: Shallow water equations (Section 4.2): In this experiment with training data set  $\mu^{\text{train-1}}$  and roll length  $R = 100$ , operator inference with roll outs leads to a predictive model on the test data set, whereas the model learned with traditional operator inference provides only inaccurate predictions. The plots corresponding to “projection” show the ground-truth data projected onto the reduced space  $\mathcal{V}$ , which provides the best approximation that any low-dimensional model seeking a linear approximation in  $\mathcal{V}$  can achieve.

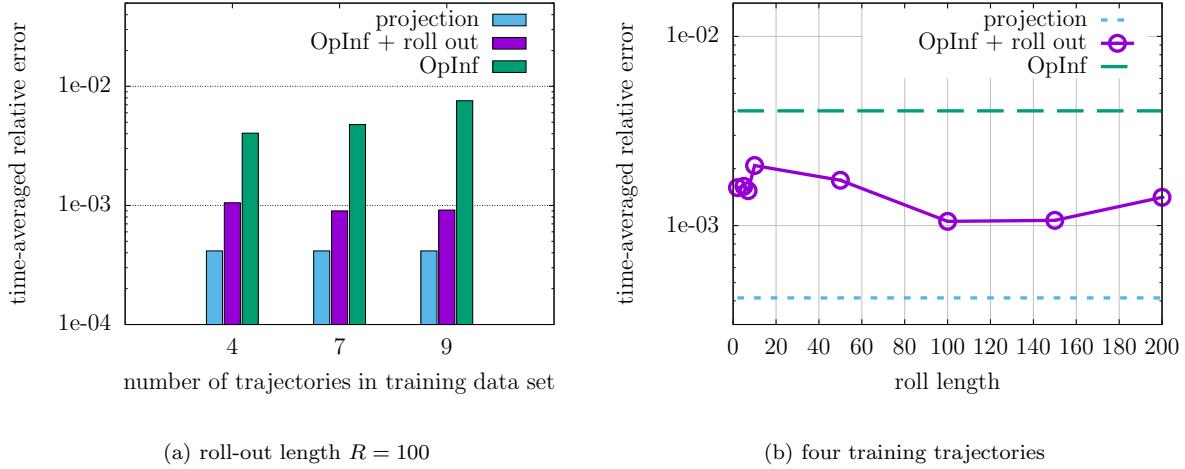


Figure 3: Shallow water equations (Section 4.2): The plots show that roll outs during training help operator inference to learn a model that makes more accurate predictions on the test data set than traditional operator inference without roll outs, especially in the scarce data regime with a low number of training trajectories.

effect of noise because they prevent the model from overfitting to the noise in the polluted training data. In contrast, the predictions from several models obtained with traditional operator inference are unstable (and thus replaced by the initial condition in the error computations).

Figure 6 shows the ground-truth free-surface height computed with the high-dimensional model and the various low-dimensional models at time  $t = 0.15$  for different test parameter values with  $M_{\text{train}} = 9, R = 150, \rho = 10\%$ . Observe that despite the high noise level, the model learned with roll outs is still able to capture the patterns in the ground truth. The model learned via traditional operator inference that does not use roll outs seems to have memorized a pattern for all parameter values in the test set, thereby failing to make meaningful predictions. Figure 7 shows the effect of the roll-out length on the prediction accuracy of the learned model. In this example, increasing the roll-out length aids in obtaining more accurate model predictions in the presence of noise. Increasing the roll-out length also increases the number of misfits penalized in the objective function (8), or more generally, it lengthens the time window over which we assess the misfit for the objective function (9), which makes the learned model more robust against overfitting.

We now compare the bounds of the stability radii of models learned from noisy data, see Figure 8. We vary the number of training trajectories as  $M_{\text{train}} \in \{4, 7, 9\}$  and set the roll-out length to  $R = 150$ . For noise with  $\rho = 0.1\%$ , the bounds of stability radii of the models learned with operator inference with roll outs is up to two orders of magnitude larger than the bounds corresponding to the models learned with traditional operator inference. This is in agreement with the error shown in Figure 5, where for  $M_{\text{train}} = 9$  training trajectories and noise with  $\rho = 0.1\%$ , the models learned with roll outs achieve about one order of magnitude higher accuracy. For high noise with  $\rho = 1.0\%$  and  $\rho = 10\%$ , the bounds of the stability radii of the models inferred under both approaches tend to be comparable in magnitude, which mirrors the increase in the error of the models learned with roll outs as  $\rho$  is increased to 10% in Figure 5.

### 4.3 Surface quasi-geostrophic dynamics

Surface quasi-geostrophic equations model the dynamics of buoyancy on horizontal boundaries and is a special case of the quasi-geostrophic equations which characterize fluid motion in the limit of strong rotation and stratification. We follow the problem as described in [81] and implemented in the PyQG code<sup>4</sup>.

<sup>4</sup><https://github.com/pyqg/pyqg>

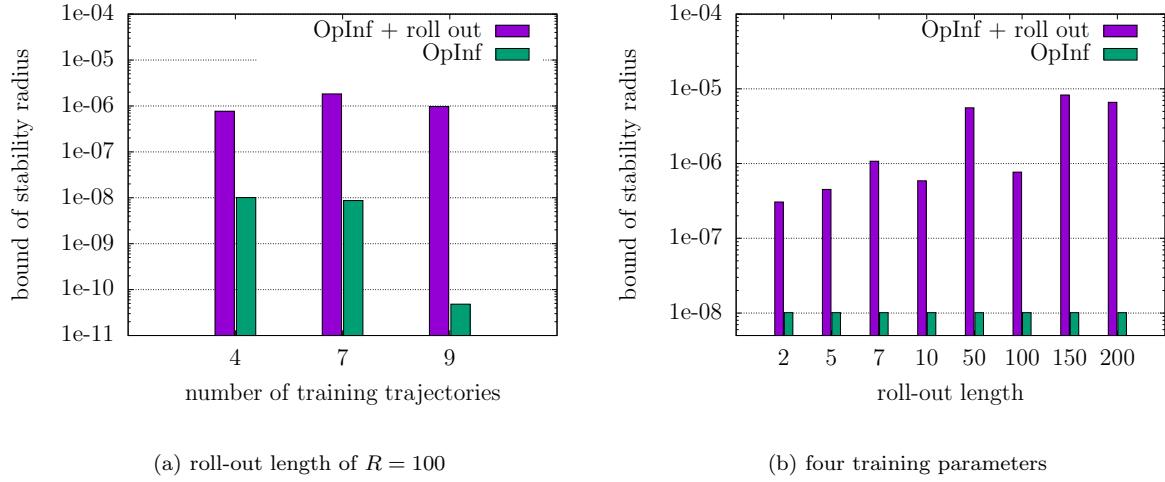


Figure 4: Shallow water equations (Section 4.2): The bounds of the stability radii of models learned with roll outs are orders of magnitude higher than for the models learned with traditional operator inference without roll outs. This suggests that training with roll outs promotes robustness in the learned low-dimensional model especially when training data are scarce.

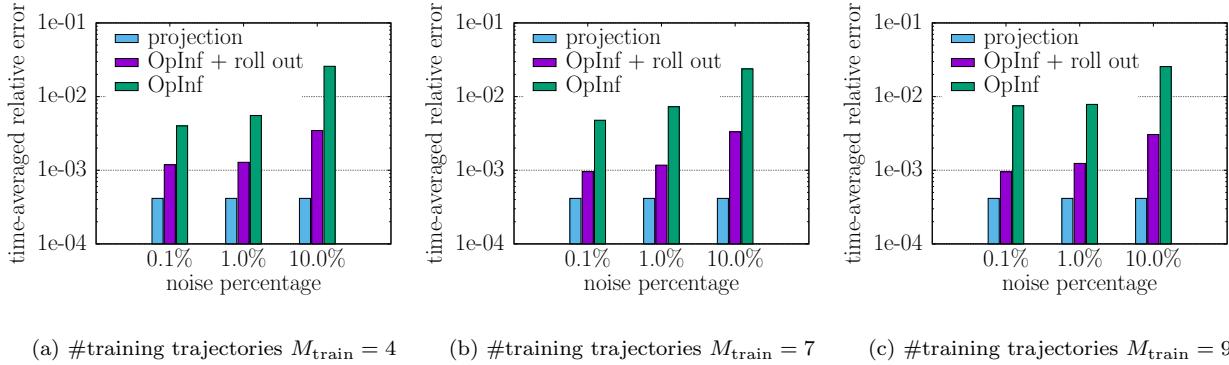


Figure 5: Shallow water equations (Section 4.2): Operator inference with roll outs learns models that achieve accurate predictions that are close to the best approximation (projection) in this example even when training data are polluted by 10% noise. In contrast, traditional operator inference without roll outs fails to make meaningful predictions especially for large noise in the training data.

#### 4.3.1 Surface quasi-geostrophic dynamics: Setup

For  $z \in (-\infty, 0]$  and  $x, y \in \Omega$  where  $\Omega = (-\pi, \pi) \times (-\pi, \pi)$ , let  $\psi$  be the stream function such that  $b(t, x, y) = \frac{\partial \psi}{\partial z}(t, x, y, 0)$  is the buoyancy. The governing equation of the surface buoyancy for  $t \in (0, T) \subset \mathbb{R}$  is given by

$$\frac{\partial b}{\partial t} + \left( -\frac{\partial \psi}{\partial y} \Big|_{z=0}, \frac{\partial \psi}{\partial x} \Big|_{z=0} \right) \cdot \nabla b = 0 \quad (17)$$

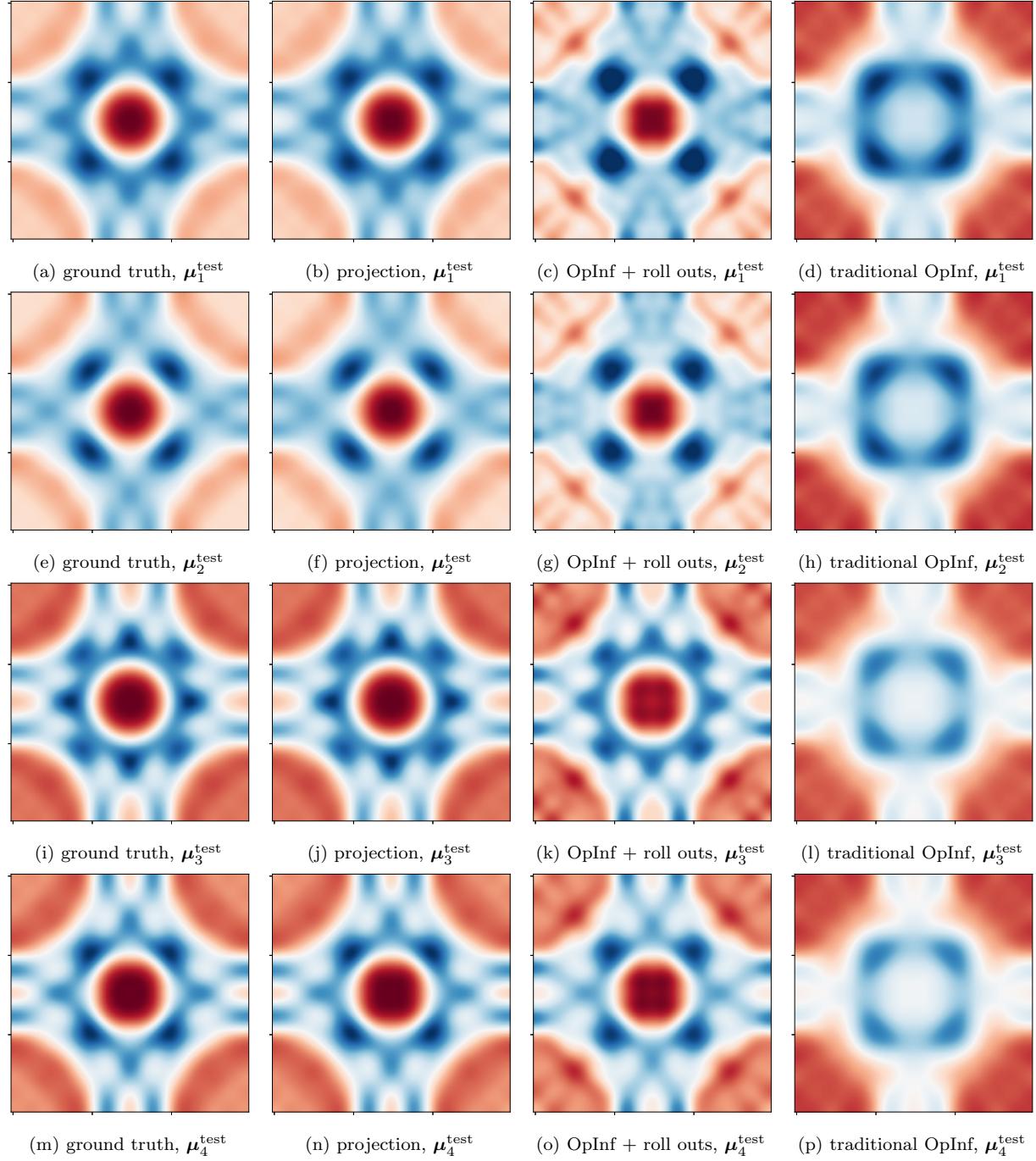


Figure 6: Shallow water equations (Section 4.2): In this experiment with nine training parameters and roll length  $R = 150$  and noise  $\rho = 10\%$ , operator inference with roll outs leads to a predictive model for  $q_h$  for the test parameters, whereas the model learned traditional operator inference provides inaccurate predictions.

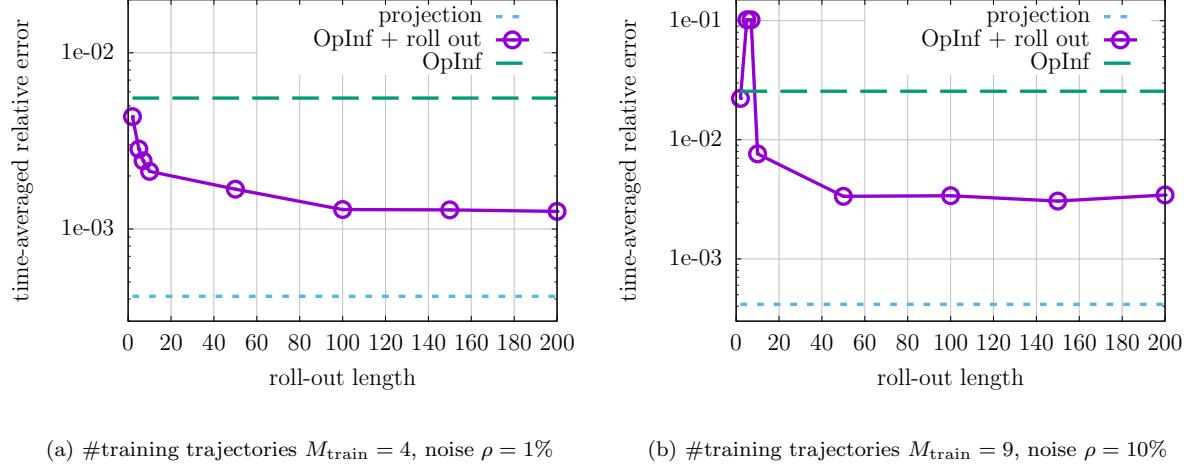


Figure 7: Shallow water equations (Section 4.2): Increasing the roll-out length improves the prediction accuracy of the learned model when training on noisy data.

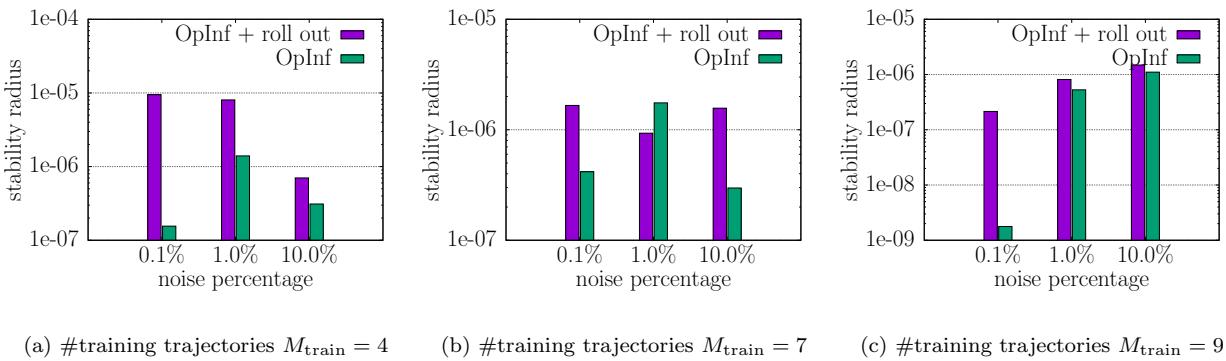


Figure 8: Shallow water equations (Section 4.2): For small noise intensity, the stability radii of the learned model with roll outs is up to two orders of magnitude higher than the model obtained with traditional operator inference. For larger noise intensity, the stability radii of the two approaches are comparable, owing to the increase in the learned model errors.

such that

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2} = 0, \\ \lim_{z \rightarrow -\infty} \frac{\partial \psi}{\partial z} = 0.$$

The initial condition is

$$b(0, x, y) = -\exp\left\{-\frac{9}{4\pi^2}((\mu_1(x-0.3))^2 + (\mu_2 y)^2)\right\} - \exp\left\{-\frac{9}{4\pi^2}((\mu_1(x+0.3))^2 + (\mu_2 y)^2)\right\}$$

with inputs  $\boldsymbol{\mu} = [\mu_1, \mu_2] \in \mathcal{D}$  where  $\mathcal{D} = [2.5, 4] \times [3, 4.5] \cup [4, 5.5] \times [1.5, 3]$ .

We generate data using the PyQG code which discretizes (17) using a spectral method and advances the corresponding system of ordinary differential equations in time using a third-order Adams-Bashford scheme. We set  $\delta t = 0.005$  and  $T = 5$ . The square domain is partitioned into a regular grid with 200 equidistant points along each dimension such that at time step  $k$ , the components of the high-dimensional state  $\mathbf{q}_k(\boldsymbol{\mu}) \in \mathbb{R}^N$ ,  $N = 40000$ , correspond to the buoyancy at each point in the grid. We use an implicit-explicit scheme for integrating the low-dimensional models over  $t \in (0, T)$  with  $\delta t = 0.005$ . The linear term is handled implicitly while the quadratic term is evaluated explicitly.

We aim to learn a low-dimensional model of dimension  $n = 60$  for this example. The low-dimensional basis  $\mathbf{V}$  of the space  $\mathcal{V}$  is computed from snapshot trajectories that are obtained from nine different initial conditions corresponding to the inputs

$$\boldsymbol{\mu}^{\text{basis}} \in \{(2.5, 3), (4, 4.5), (4, 1.5), (5.5, 3), (4, 3), (3.25, 3.75), (4.75, 2.25), (5.5, 1.5), (2.5, 4.5)\}.$$

Training data correspond to three nested sets of inputs given by

$$\begin{aligned} \boldsymbol{\mu}^{\text{train-I}} &= \{(2.5, 3.75), (5.5, 2.25), (4, 3.75), (4, 2.25)\}, \\ \boldsymbol{\mu}^{\text{train-II}} &= \boldsymbol{\mu}^{\text{train-I}} \cup \{(3, 3), (3.5, 4.5), (4.5, 3), (5, 1.5)\}, \\ \boldsymbol{\mu}^{\text{train-III}} &= \boldsymbol{\mu}^{\text{train-II}} \cup \{(3, 4.5), (3.5, 3), (4.5, 1.5), (5, 3)\}. \end{aligned}$$

Hyperparameters are tuned using trajectories on the validation data set corresponding to the inputs

$$\begin{aligned} \boldsymbol{\mu}^{\text{valid}} \in & \{(2.75, 3.375), (3.75, 3.375), (2.75, 4.125), (3.75, 4.125), \\ & (4.25, 1.875), (5.25, 1.875), (4.25, 2.625), (5.25, 2.625)\}. \end{aligned} \quad (18)$$

We then assess the performance of the learned low-dimensional models using test data corresponding to the inputs  $\boldsymbol{\mu}^{\text{test}} \in \{(3.25, 4.125), (3.25, 3.375), (4.75, 1.875), (4.75, 2.625)\}$ .

### 4.3.2 Surface quasi-geostrophic dynamics: Scarce data because of sparsely sampled trajectories in time

In this example, we will also demonstrate another advantage of operator inference with roll outs: the objective function is flexible enough to accommodate the situation where the training data is not observed at all time points. This would pose a challenge for traditional operator inference when it approximates time derivatives via finite differences, for example. In particular, operator inference with roll outs is well-equipped to handle data at irregularly-spaced time points. To keep computations manageable, in this work, we will consider scarce training data that are obtained at equidistant time points; however, the methodology applies to non-equidistant time points as well. The sampling period is  $\xi \in \mathbb{N}$ , which corresponds to the number of time steps between two consecutive measurements of training states. Thus, with a sampling period  $\xi$ , the training data are collected at  $t \in \{\xi \Delta t | 0 \leq \xi \Delta t \leq T, \xi \in \mathbb{Z}, \xi > 0\}$ . Setting  $\xi = 1$  means that data are not sparsified. For traditional operator inference, we change the objective function as

$$\min_{\mathbf{A}_1, \dots, \mathbf{A}_L, \mathbf{B}} \frac{1}{[K/\xi]} \sum_{k=1}^{[K/\xi]} \left\| \sum_{\ell=1}^L \mathbf{A}_\ell \bar{\mathbf{q}}_{k\xi}^\ell(\boldsymbol{\mu}_i) + \mathbf{B} \mathbf{u}_{k\xi}(\boldsymbol{\mu}_i) - \bar{\mathbf{q}}'_{k\xi} \right\|_2^2,$$

where the derivative  $\bar{\mathbf{q}}'_{k\xi}$  is approximated via finite difference using the scarce data, i.e.

$$\bar{\mathbf{q}}'_{k\xi} = \frac{\bar{\mathbf{q}}_{(k+1)\xi} - \bar{\mathbf{q}}_{k\xi}}{\xi \Delta t}.$$

For operator inference with roll outs, the objective function is now

$$J(\boldsymbol{\theta}; \boldsymbol{\mu}) = \sum_{k=0}^{\lfloor (K-R)/\xi \rfloor} \sum_{r=1}^{\lfloor R/\xi \rfloor} \left\| \bar{\mathbf{q}}_{(k+r)\xi}(\boldsymbol{\mu}) - \hat{\mathbf{z}}_{k\xi, r\xi}(\boldsymbol{\mu}) \right\|^2.$$

#### 4.3.3 Surface quasi-geostrophic dynamics: Results for learning from scarce data

In the following, training data are scarce with respect to two properties: the number of training trajectories and the sampling period  $\xi$ , which corresponds the number of time steps that lie between two consecutive measurements.

Consider training data obtained with sampling period  $\xi = 10$  and  $M_{\text{train}} = 12$  training trajectories. We set the roll-out length to  $R = 200$ . The state predictions are shown in Figure 9. Operator inference with roll outs trains models that accurately predict the states corresponding to the test inputs. In contrast, the model obtained with traditional operator inference poorly generalizes to the test inputs. In particular, traditional operator inference leads to models that show instabilities, e.g., Figure 9d. This is because when the training data is only available at sparsely sampled time points, recall that the sampling period is  $\xi = 10$  in this experiment, then the approximation of time derivatives via finite differences becomes less reliable.

To quantify the accuracy of the learned models, Figure 10 plots the test error (16) over the number of training trajectories for sampling periods  $\xi = 5$  and  $\xi = 10$ . The error of the model obtained with roll outs decreases as the amount of training data is increased. For traditional operator inference, several of the obtained models resulted in unstable predictions, and so we replaced their predictions by the initial conditions in computing the shown errors. The results indicate that the lack of training data is compensated by roll outs via solving a modified optimization problem that penalizes discrepancies between the data and model predictions at future times. In Figure 11, we fix the number of training trajectories to  $M_{\text{train}} = 12$  and plot the error over the roll length for sampling periods  $\xi = 5$  and  $\xi = 10$ . In this example, increasing the roll-out length helps to decrease the prediction error of the learned model.

Figure 12 plots the bound of the stability radii, averaged over 1000 realizations of the random matrix  $\mathbf{L}$  with standard normal entries. We set  $R = 200$ . When the training data are very scarce, e.g.,  $M_{\text{train}} = 4$ , then the bound of the stability radius of the learned models from operator inference with roll outs is larger by up to three orders of magnitude compared to the models learned with traditional operator inference. However, as more training data are used, e.g., by increasing the sampling period and by increasing the number of trajectories, the bounds of the stability radii of the models learned with traditional operator inference improves. Indeed, the model learned via traditional operator inference in this example appears to be stable despite being inaccurate; see Figure 9.

#### 4.3.4 Surface quasi-geostrophic dynamics: Scarce data that are also noisy

We now add noise to the training data in the same way as described in Section 4.2.3, with  $\rho$  controlling the standard deviation of the noise. We plot the buoyancy obtained with the learned models in Figure 13, where  $R = 200$ ,  $M_{\text{train}} = 12$ ,  $\xi = 5$ ,  $\rho = 10\%$ . The plots demonstrate that in the scarce and noisy data regime, operator inference with roll outs offers an approximation that closely resembles the projected states, which is the best approximation possible in the low-dimensional subspace. Meanwhile, traditional operator inference fails to provide meaningful predictions across various test inputs.

Figure 14 shows the test error (16) of the model predictions with respect to the standard deviation given by  $\rho$  of the noise. The roll-out length is set to  $R = 200$  and the number of training trajectories is  $M_{\text{train}} = 12$ . As can be seen, despite the increase in the standard deviation of the noise, we only observe a modest increase

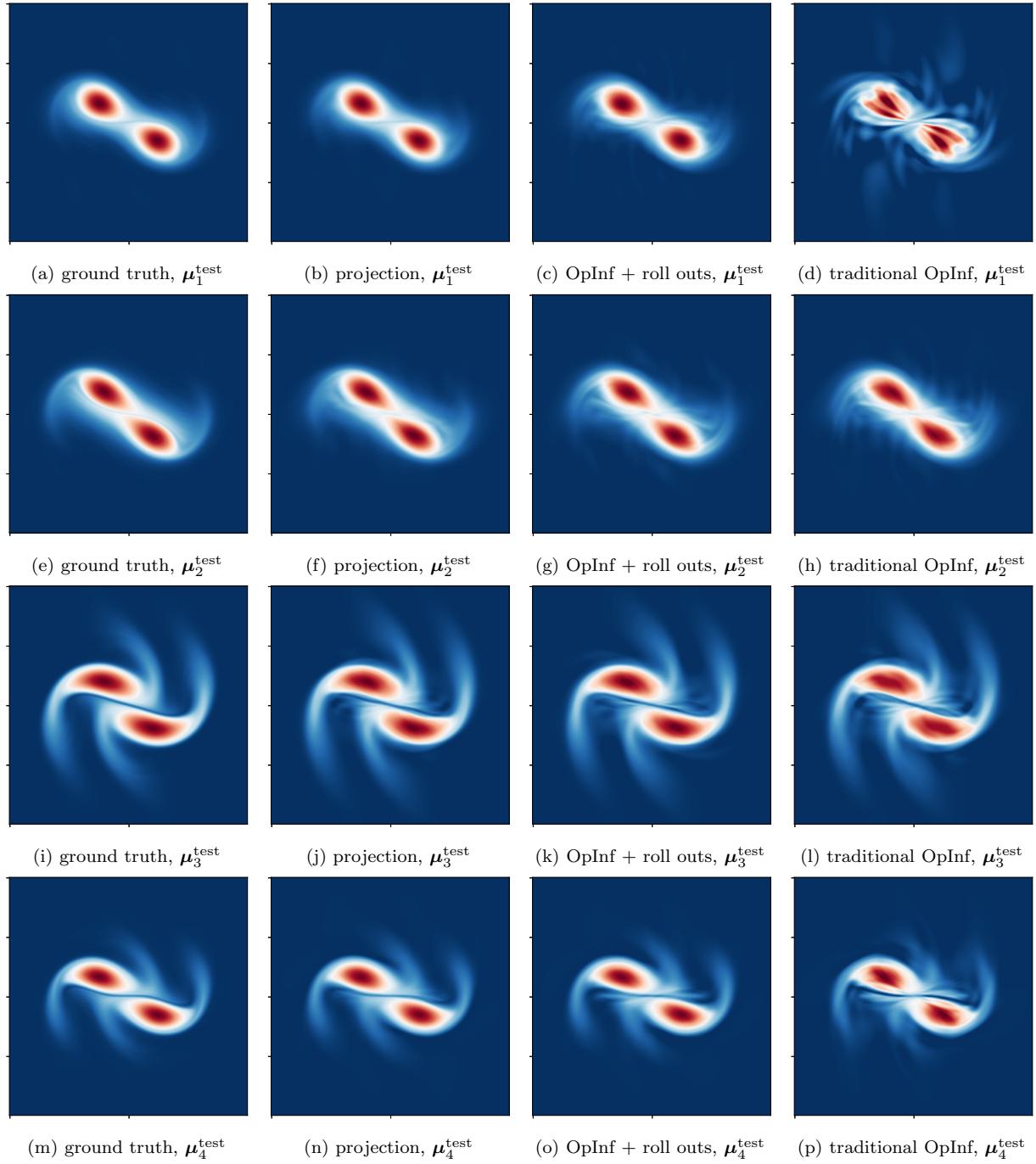


Figure 9: Surface quasi-geostrophic dynamics (Section 4.3): Operator inference with roll outs with sampling period  $\xi = 10$  leads to models that accurately predict the surface buoyancy at the test inputs. In contrast, traditional operator inference without roll outs results in almost unstable predictions for several test inputs and leads to visibly higher errors in the state predictions.

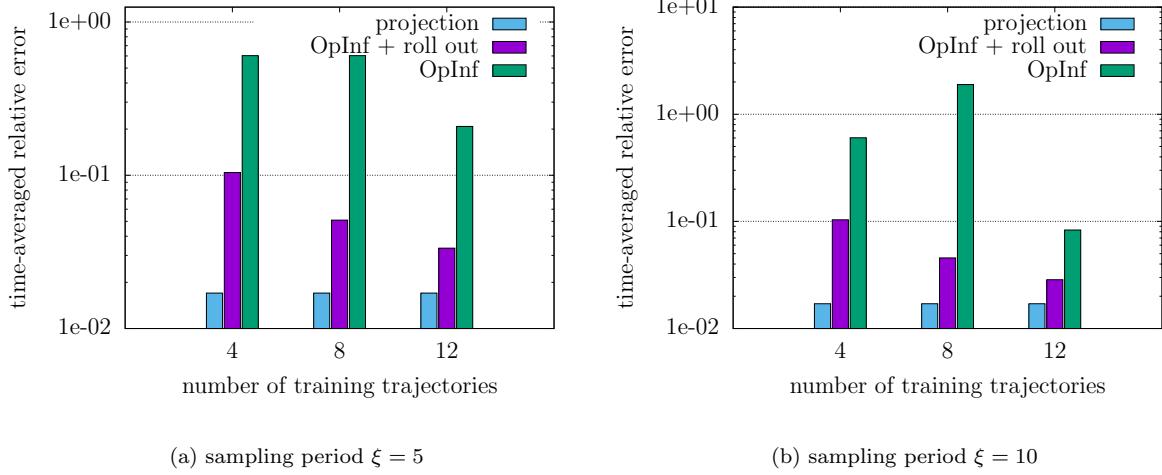


Figure 10: Surface quasi-geostrophic dynamics (Section 4.3): For training data obtained with a large sampling period and from few trajectories only, roll outs compensate to some extent for the information loss by penalizing the misfit in the model predictions at future times. The error of the model obtained with roll outs approaches the projection error as the amount of information is increased. When training with traditional operator inference, the model predictions can be numerically unstable and inaccurate.

in the prediction error of the model learned with roll outs. In the scarce and noisy data regime, traditional operator inference leads to numerically unstable models quickly. In these cases, for computing the error, the initial condition serves as a surrogate for the model prediction at all times.

In Figure 15, we visualize the error over the roll length for  $\rho = 25\%$  and  $M_{\text{train}} = 12$ . The results shown in the plots demonstrate that in this example, increasing the roll length improves the accuracy and stability of the learned model by up to two orders of magnitude. Increasing the roll length penalizes the misfit between data and predictions at more time points, thus, the learned model is less likely to overfit to the noise in the data.

Let us now investigate the stability radii of the learned models; we follow the same computation of the bounds of the stability radii as in the previous examples. The results are shown in Figure 16 over the standard deviation given by  $\rho$  for the sampling periods  $\xi = 5$  and  $\xi = 10$ , with fixed  $R = 200, M_{\text{train}} = 12$ . The models learned with roll outs achieve bounds of the stability radii that are up to two orders of magnitude larger than those of the models learned with traditional operator inference. This is in agreement with our previous numerical experiments, where traditional operator inference leads to numerically unstable models when noise is added and data are scarce.

## 5 Conclusions

Operator inference with roll outs combines the interpretability, scalability, and structure preservation of traditional operator inference with the stability and robustness of dynamic training given by the roll outs of Neural ODEs. The roll outs are especially helpful to obtain stable and robust models when data are noisy and scarce. Because we insist on the polynomial structure of the models, the learned models are amenable for analysis and interpretation. In particular, the stability bias imposed by the roll outs can be observed in increased bounds of the stability radii of the domain of attraction of the learned models. Furthermore, linear dynamics can be interpolated in a structure preserving way to make predictions at new, previously unseen physical inputs. Finally, the polynomial structure is also in agreement with wide engineering practice, where non-linear behavior is often approximated via polynomials corresponding to truncated Taylor expansions.

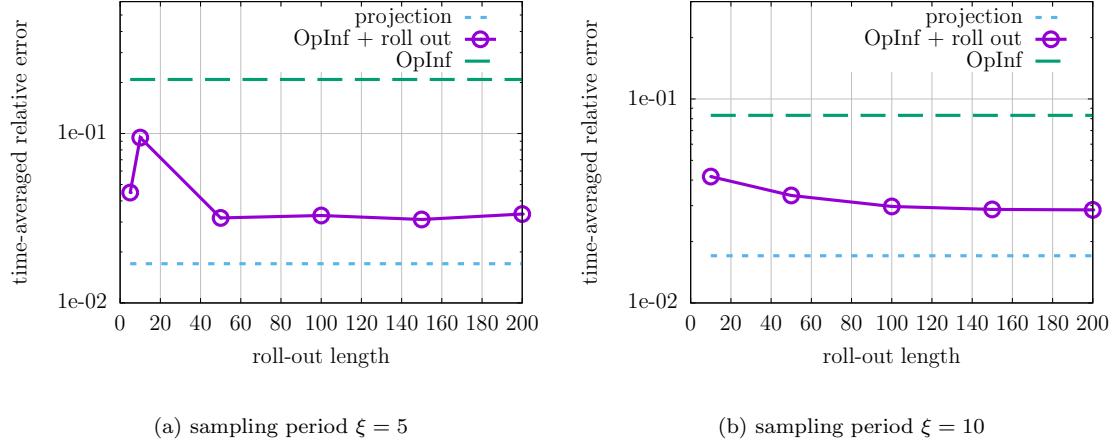


Figure 11: Surface quasi-geostrophic dynamics (Section 4.3): In this example, increasing the roll-out length improves the prediction accuracy of the learned model.

While the training is more involved, operator inference with roll outs is straightforward to implement with automatic differentiation and tractable in large settings due to advances in stochastic gradient descent methods. The numerical experiments with shallow water wave and surface quasi-geostrophic dynamics demonstrate the increased robustness and stability of the models learned with roll outs.

## Acknowledgements

The first and third author acknowledge partially supported by US Department of Energy, Office of Advanced Scientific Computing Research, Applied Mathematics Program (Program Manager Dr. Steven Lee), DOE Award DESC0019334. The third author also acknowledges funding from the National Science Foundation under Grant No. 1901091. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

## References

- [1] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, *Computer Methods in Applied Mechanics and Engineering* 306 (2016) 196–215.
- [2] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 31, Curran Associates, Inc., 2018, pp. 1–13.
- [3] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics* 656 (2010) 5–28.
- [4] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. Henningson, Spectral analysis of nonlinear flows, *Journal of Fluid Mechanics* 641 (2009) 115–127.
- [5] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, J. N. Kutz, On dynamic mode decomposition: Theory and applications, *Journal of Computational Dynamics* 1 (2) (2014) 391–421.
- [6] J. N. Kutz, S. L. Brunton, B. W. Brunton, J. L. Proctor, *Dynamic mode decomposition: Data-driven modeling of complex systems*, SIAM, 2016.

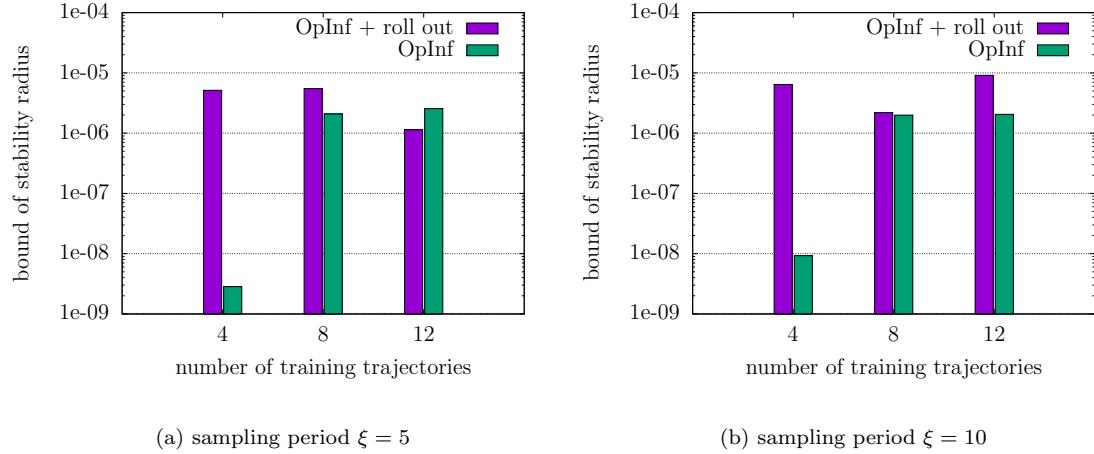


Figure 12: Surface quasi-geostrophic dynamics (Section 4.3): If data are scarce, e.g., the number of training trajectories is four, then roll outs help operator inference to achieve an orders of magnitude larger bound of stability radius in this example. As more training data are used, e.g., by increasing the number of training trajectories, operator inference without roll outs achieves comparable bounds of stability radii.

- [7] M. O. Williams, I. G. Kevrekidis, C. W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition, *Journal of Nonlinear Science* 25 (6) (2015) 1307–1346.
- [8] D. Burov, D. Giannakis, K. Manohar, A. Stuart, Kernel analog forecasting: Multiscale test problems, *Multiscale Modeling & Simulation* 19 (2) (2021) 1011–1040.
- [9] S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences* 113 (15) (2016) 3932–3937.
- [10] H. Schaeffer, R. Caflisch, C. D. Hauck, S. Osher, Sparse dynamics for partial differential equations, *Proceedings of the National Academy of Sciences* 110 (17) (2013) 6634–6639.
- [11] H. Schaeffer, G. Tran, R. Ward, Extracting sparse high-dimensional dynamics from limited data, *SIAM Journal on Applied Mathematics* 78 (6) (2018) 3279–3295.
- [12] S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations, *Science Advances* 3 (4) (2017).
- [13] X. Xie, M. Mohebujjaman, L. G. Rebholz, T. Iliescu, Data-driven filtered reduced order modeling of fluid flows, *SIAM Journal on Scientific Computing* 40 (3) (2018) B834–B857.
- [14] C. Mou, B. Koc, O. San, L. G. Rebholz, T. Iliescu, Data-driven variational multiscale reduced order models, *Computer Methods in Applied Mechanics and Engineering* 373 (2021) 113470.
- [15] J. S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, *Journal of Computational Physics* 363 (2018) 55–78.
- [16] N. Demo, M. Tezzele, G. Rozza, A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces, *Comptes Rendus Mécanique* 347 (11) (2019) 873–881.
- [17] M. Guo, J. S. Hesthaven, Reduced order modeling for nonlinear structural analysis using Gaussian process regression, *Computer Methods in Applied Mechanics and Engineering* 341 (2018) 807–826.

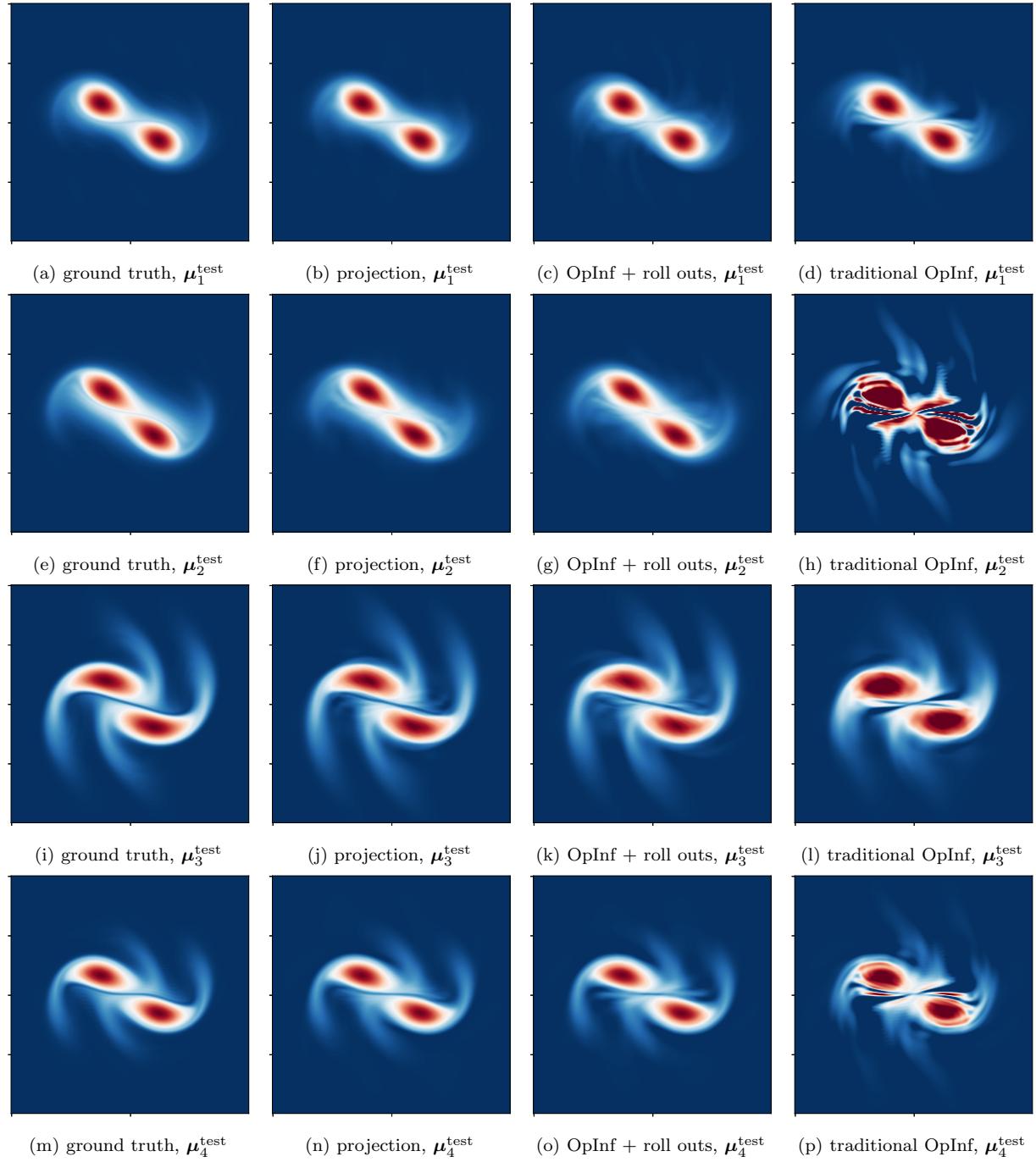


Figure 13: Surface quasi-geostrophic dynamics (Section 4.3): Operator inference with roll outs ( $M_{\text{train}} = 12, R = 200, \xi = 5$ ) trained on data with  $\rho = 10\%$  noise, leads to models that accurately predict the surface buoyancy at the test parameters. In contrast, traditional operator inference without roll outs fails to make meaningful predictions across various test parameter values.

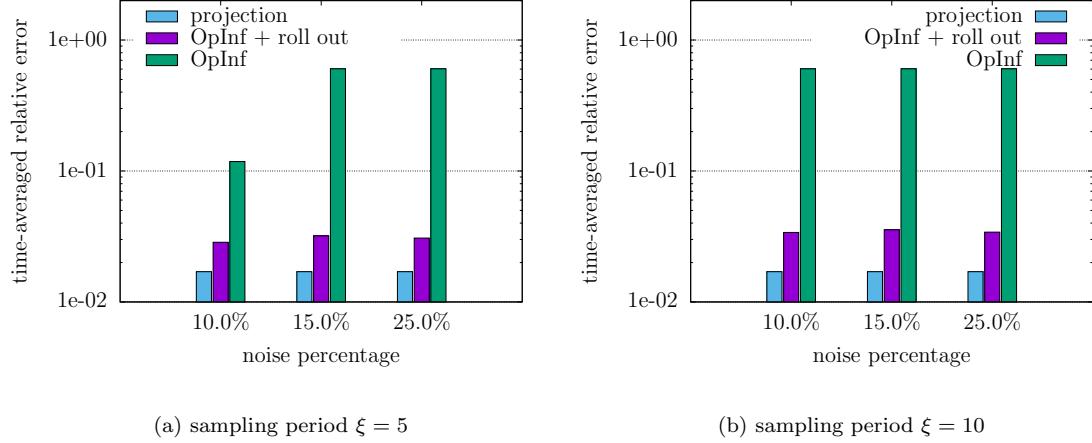


Figure 14: Surface quasi-geostrophic dynamics (Section 4.3): Operator inference with roll outs leads to models that are predictive even when the training data are polluted by 25% noise. The accuracy achieved by operator inference with roll outs is close to the projection error of the ground-truth data on the reduced space. In contrast, traditional operator inference mostly learns unstable models that cannot make meaningful predictions at test inputs.

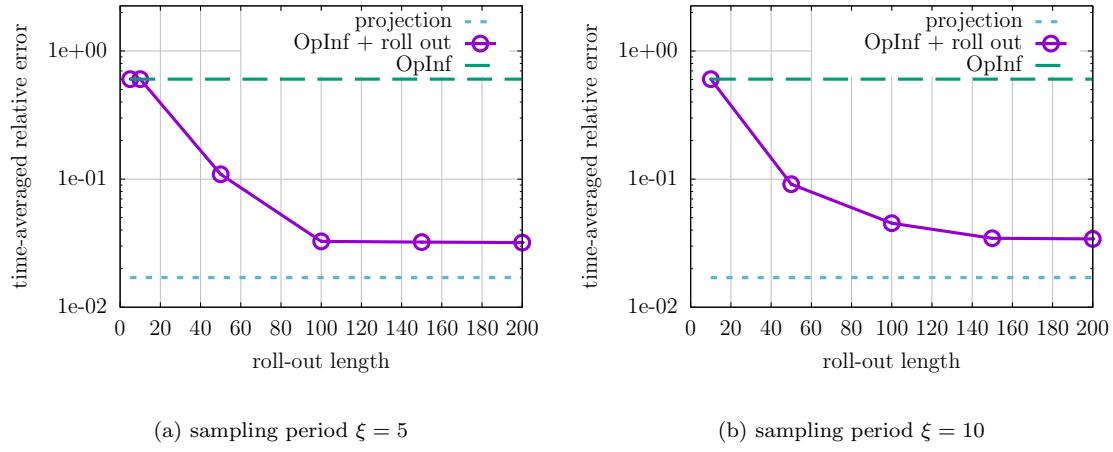


Figure 15: Surface quasi-geostrophic dynamics (Section 4.3): When training data are both scarce and noisy, increasing the roll length leads to more accurate models because the objective function accounts for the model misfit at multiple times for larger roll-out lengths. Thus, operator inference with roll outs prevents overfitting to the noise and achieves a prediction error that is close to the projection error.

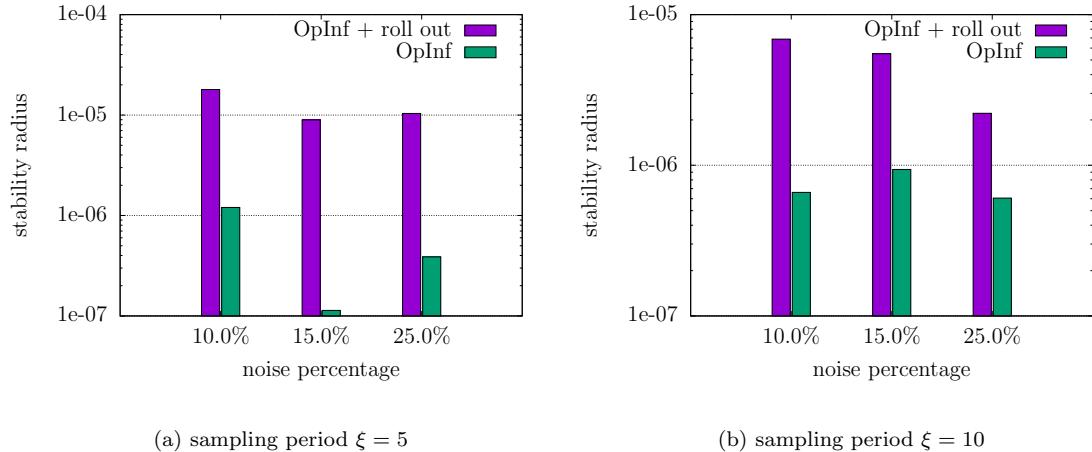


Figure 16: Surface quasi-geostrophic dynamics (Section 4.3): Models learned with roll outs achieve larger bounds of the stability radii than the models obtained via traditional operator inference in this example.

- [18] D. Forti, G. Rozza, Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems, *International Journal of Computational Fluid Dynamics* 28 (3-4) (2014) 158–169.
- [19] M. Guo, J. S. Hesthaven, Data-driven reduced order modeling for time-dependent problems, *Computer Methods in Applied Mechanics and Engineering* 345 (2019) 75–99.
- [20] A. Lario, R. Maulik, O. T. Schmidt, G. Rozza, G. Mengaldo, Neural-network learning of SPOD latent dynamics, *Journal of Computational Physics* 468 (2022) 111475.
- [21] S. M. Rahman, S. Pawar, O. San, A. Rasheed, T. Iliescu, Nonintrusive reduced order modeling framework for quasigeostrophic turbulence, *Phys. Rev. E* 100 (2019) 053306.
- [22] N. Demo, M. Tezzele, G. Rozza, EZyRB: Easy reduced basis method, *Journal of Open Source Software* 3 (24) (2018) 661.
- [23] A. C. Antoulas, B. D. O. Anderson, On the scalar rational interpolation problem, *IMA Journal of Mathematical Control & Information* 3 (2-3) (1986) 61–88.
- [24] S. Lefteriu, A. C. Antoulas, A new approach to modeling multiport systems from frequency-domain data, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 29 (1) (2010) 14–27.
- [25] A. J. Mayo, A. C. Antoulas, A framework for the solution of the generalized realization problem, *Linear Algebra and its Applications* 425 (2–3) (2007) 634–662.
- [26] C. Beattie, S. Gugercin, Realization-independent  $\mathcal{H}_2$ -approximation, in: *Proc. IEEE Conf. Decis. Control, Maui, HI, USA, 2012*, pp. 4953–4958.
- [27] A. C. Antoulas, I. V. Gosea, A. C. Ionita, Model reduction of bilinear systems in the Loewner framework, *SIAM Journal on Scientific Computing* 38 (5) (2016) B889–B916.
- [28] I. V. Gosea, A. C. Antoulas, Data-driven model order reduction of quadratic-bilinear systems, *Numerical Linear Algebra with Applications* 25 (6) (2018) e2200.

[29] A. C. Ionita, A. C. Antoulas, Data-driven parametrized model reduction in the Loewner framework, *SIAM Journal on Scientific Computing* 36 (3) (2014) A984–A1007.

[30] Z. Drmač, B. Peherstorfer, Learning low-dimensional dynamical-system models from noisy frequency-response data with Loewner rational interpolation, in: C. Beattie, P. Benner, M. Embree, S. Gugercin, S. Lefteriu (Eds.), *Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas*, Springer International Publishing, Cham, 2022, pp. 39–57.

[31] B. Peherstorfer, S. Gugercin, K. Willcox, Data-driven reduced model construction with time-domain Loewner models, *SIAM Journal on Scientific Computing* 39 (5) (2017) A2152–A2178.

[32] B. Gustavsen, A. Semlyen, Rational approximation of frequency domain responses by vector fitting, *IEEE Transactions on Power Delivery* 14 (3) (1999) 1052–1061.

[33] Z. Drmač, S. Gugercin, C. Beattie, Vector fitting for matrix-valued rational approximation, *SIAM Journal on Scientific Computing* 37 (5) (2015) A2346–A2379.

[34] J.-N. Juang, R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, *Journal of Guidance, Control, and Dynamics* 8 (5) (1985) 620–627.

[35] B. Kramer, S. Gugercin, Tangential interpolation-based eigensystem realization algorithm for MIMO systems, *Mathematical and Computer Modelling of Dynamical Systems* 22 (4) (2016) 282–306.

[36] E. Qian, B. Kramer, B. Peherstorfer, K. Willcox, Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems, *Physica D: Nonlinear Phenomena* 406 (2020) 132401.

[37] S. A. McQuarrie, P. Khodabakhshi, K. E. Willcox, Non-intrusive reduced-order models for parametric partial differential equations via data-driven operator inference, *arXiv* 2110.07653 (2021).

[38] P. Khodabakhshi, K. E. Willcox, Non-intrusive data-driven model reduction for differential-algebraic equations derived from lifting transformations, *Computer Methods in Applied Mechanics and Engineering* 389 (2022) 114296.

[39] W. I. T. Uy, B. Peherstorfer, Operator inference of non-Markovian terms for learning reduced models from partially observed state trajectories, *Journal of Scientific Computing* 88 (3) (2021) 91.

[40] W. I. T. Uy, B. Peherstorfer, Probabilistic error estimation for non-intrusive reduced models learned from data of systems governed by linear parabolic partial differential equations, *ESAIM: Mathematical Modelling and Numerical Analysis (M2AN)* 55 (3) (2021) 735–761.

[41] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, K. Willcox, Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, *Computer Methods in Applied Mechanics and Engineering* 372 (2020) 113433.

[42] H. Sharma, Z. Wang, B. Kramer, Hamiltonian operator inference: Physics-preserving learning of reduced-order models for Hamiltonian systems, *Physica D: Nonlinear Phenomena* 431 (2022) 133122.

[43] S. A. McQuarrie, C. Huang, 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) (2021) 194–211.

[44] M. Guo, S. A. McQuarrie, K. E. Willcox, Bayesian operator inference for data-driven reduced-order modeling, *Computer Methods in Applied Mechanics and Engineering* (2022) 115336.

[45] R. Swischuk, B. Kramer, C. Huang, K. Willcox, Learning physics-based reduced-order models for a single-injector combustion process, *AIAA Journal* 58 (6) (2020) 2658–2672.

[46] E. Qian, A scientific machine learning approach to learning reduced models for nonlinear partial differential equations, Ph.D. thesis, Massachusetts Institute of Technology (2021).

[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 (2020) A3489–A3515.

[48] N. Sawant, B. Kramer, B. Peherstorfer, Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference, *arXiv* 2107.02597 (2021).

[49] W. I. T. Uy, Y. Wang, Y. Wen, B. Peherstorfer, Active operator inference for learning low-dimensional dynamical-system models from noisy data, *arXiv* 2107.09256 (2021).

[50] D. Hartmann, L. Failer, A differentiable solver approach to operator inference, *arXiv* 2107.02093 (2021).

[51] Y. Rubanova, R. T. Q. Chen, D. K. Duvenaud, Latent ordinary differential equations for irregularly-sampled time series, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019, pp. 1–11.

[52] K. Lee, E. J. Parish, Parameterized neural ordinary differential equations: applications to computational physics problems, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 477 (2253) (2021) 20210162.

[53] J. Baker, E. Cherkaev, A. Narayan, B. Wang, Learning POD of complex dynamics using heavy-ball neural ODEs, *arXiv* 2202.12373 (2022).

[54] O. San, R. Maulik, M. Ahmed, An artificial neural network framework for reduced order modeling of transient flows, *Communications in Nonlinear Science and Numerical Simulation* 77 (2019) 271–287.

[55] S. Fresca, L. Dede', A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs, *Journal of Scientific Computing* 87 (2) (2021) 61.

[56] Y. Yang, M. Aziz Bhouri, P. Perdikaris, Bayesian differential programming for robust systems identification under uncertainty, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476 (2243) (2020) 20200290.

[57] S. H. Rudy, J. Nathan Kutz, S. L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *Journal of Computational Physics* 396 (2019) 483–506.

[58] P. Goyal, P. Benner, Neural ODEs with irregular and noisy data, *arXiv* 2205.09479 (2022).

[59] D. Amsallem, J. Cortial, K. Carlberg, C. Farhat, A method for interpolating on manifolds structural dynamics reduced-order models, *International Journal for Numerical Methods in Engineering* 80 (9) (2009) 1241–1258.

[60] H. Panzer, J. Mohring, R. Eid, B. Lohmann, Parametric model order reduction by matrix interpolation, *at – Automatisierungstechnik* 58 (8) (2010) 475–484.

[61] J. Degroote, J. Vierendeels, K. Willcox, Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis, *International Journal for Numerical Methods in Fluids* 63 (2) (2010) 207–230.

[62] D. Amsallem, C. Farhat, An online method for interpolating linear parametric reduced-order models, *SIAM Journal on Scientific Computing* 33 (5) (2011) 2169–2198.

[63] P. Goyal, P. Benner, Discovery of nonlinear dynamical systems using a Runge-Kutta inspired dictionary-based sparse regression approach, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 478 (2262) (2022) 20210883.

[64] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Society for Industrial and Applied Mathematics, 2005.

[65] G. Rozza, D. Huynh, A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Archives of Computational Methods in Engineering* 15 (3) (2008) 1–47.

[66] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Review* 57 (4) (2015) 483–531.

[67] B. Peherstorfer, Breaking the Kolmogorov barrier with nonlinear model reduction, *Notices of the American Mathematical Society* 69 (2022) 725–733.

[68] J. B. Rawlings, D. Q. Mayne, M. Diehl, *Model predictive control: theory, computation, and design*, Vol. 2, Nob Hill Publishing Madison, 2017.

[69] K. Um, R. Brand, Y. R. Fei, P. Holl, N. Thuerey, Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers, *Advances in Neural Information Processing Systems* 33 (2020) 6111–6122.

[70] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, *Proceedings of the National Academy of Sciences* 118 (21) (2021) e2101784118.

[71] S. Schoenholz, E. D. Cubuk, Jax MD: a framework for differentiable physics, *Advances in Neural Information Processing Systems* 33 (2020) 11428–11441.

[72] R. Genesio, A. Tesi, Stability analysis of quadratic systems, *IFAC Proceedings Volumes* 22 (3) (1989) 195–199.

[73] A. Tesi, F. Villorese, R. Genesio, On stability domain estimation via a quadratic Lyapunov function: convexity and optimality properties for polynomial systems, in: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, Vol. 2, 1994, pp. 1907–1912.

[74] B. Kramer, Stability domains for quadratic-bilinear reduced-order models, *SIAM Journal on Applied Dynamical Systems* 20 (2) (2021) 981–996.

[75] A. A. Kaptanoglu, J. L. Callaham, C. J. Hansen, A. Aravkin, S. L. Brunton, Promoting global stability in data-driven models of quadratic nonlinear dynamics, *arXiv* 2105 (01843) (2021) 1–29.

[76] C. Gu, QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30 (9) (2011) 1307–1320.

[77] R. Frostig, M. Johnson, C. Leary, Compiling machine learning programs via high-level tracing, in: *ML-Sys*, 2018, pp. 1–3.  
URL <https://mlsys.org/Conferences/doc/2018/146.pdf>

[78] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Nuclea, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs (2018).  
URL <http://github.com/google/jax>

- [79] A. Rahman, J. Drgoňa, A. Tuor, J. Strube, Neural ordinary differential equations for nonlinear system identification, arXiv preprint arXiv:2203.00120 (2022).
- [80] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization., in: Y. Bengio, Y. LeCun (Eds.), ICLR (Poster), 2015, pp. 1–1.
- [81] I. M. Held, R. T. Pierrehumbert, S. T. Garner, K. L. Swanson, Surface quasi-geostrophic dynamics, Journal of Fluid Mechanics 282 (1995) 1–20.