PHYSICAL REVIEW FLUIDS **5**, 104401 (2020)

Probabilistic neural networks for fluid flow surrogate modeling and data recovery

Romit Maulik[*]
Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, Illinois 60439, USA

Kai Fukami[†]
School of Science for Open and Environmental Systems, Keio University, Yokohama 223-8522, Japan

Nesar Ramachandra[‡]
High Energy Physics Division, Argonne National Laboratory, Lemont, Illinois 60439, USA

Koji Fukagata[§]
Department of Mechanical Engineering, Keio University, Yokohama 223-8522, Japan

Kunihiko Taira[‖]
Department of Mechanical and Aerospace Engineering, University of California,
Los Angeles, Los Angeles, California 90095, USA

(Received 8 May 2020; accepted 16 September 2020; published 8 October 2020)

We consider the use of probabilistic neural networks for fluid flow surrogate modeling and data recovery. This framework is constructed by assuming that the target variables are sampled from a Gaussian distribution conditioned on the inputs. Consequently, the overall formulation sets up a procedure to predict the hyperparameters of this distribution which are then used to compute an objective function given training data. We demonstrate that this framework has the ability to provide for prediction confidence intervals based on the assumption of a probabilistic posterior, given an appropriate model architecture and adequate training data. The applicability of the present framework to cases with noisy measurements and limited observations is also assessed. To demonstrate the capabilities of this framework, we consider canonical regression problems of fluid dynamics from the viewpoint of reduced-order modeling and spatial data recovery for four canonical data sets. The examples considered in this study arise from (i) the shallow-water equations, (ii) a two-dimensional cylinder flow, (iii) the wake of a NACA0012 airfoil with a Gurney flap, and (iv) the NOAA sea surface temperature data set. The present results indicate that the probabilistic neural network not only produces a machine-learning-based fluid flow surrogate model but also systematically quantifies the uncertainty therein to assist with model interpretability.

DOI: 10.1103/PhysRevFluids.5.104401

I. INTRODUCTION

The uses of machine learning (ML) have been attracting attention for various applications within the fluid dynamics community. In particular, ML approaches hold great potential for extracting

---

[*] rmaulik@anl.gov

[†] kai.fukami@keio.jp

[‡] nramachandra@anl.gov

[§] fukagata@mech.keio.ac.jp

[‖] ktaira@seas.ucla.edu

©2020 American Physical Society

complex nonlinear relations embedded in fluid flow data [1–5]. For example, the successes of ML have been observed in several investigations into the development of closure models for large-eddy simulation [6–9] and Reynolds-averaged Navier-Stokes simulation [10–15] where direct numerical simulation (DNS) or experimental data have been used to improve conventional algebraic or differential-equation-based closures [16]. On many occasions, physics-informed turbulence closure models have proven superior for canonical flows when compared to classical turbulence models with their restrictive hypotheses. However, the generalization of such models for broader applicability and their uncertainty quantification is only recently beginning to be explored [17,18]. Another promising avenue for ML is for addressing the challenges of conventional reduced-order models (ROMs) [19,20]. Recent literature has demonstrated the capabilities of time-series methods from ML [21–26] for reduced-space temporal dynamics prediction as well as nonlinear subspace identification using image processing techniques [27–32]. A core motivation for exploring the use of ML methods for ROMs stems from the fact that intrusive ROMs depend on the resolution of a quadratic nonlinearity in proper orthogonal decomposition (POD) space [33], which is particularly computationally challenging in high-dimensional systems dominated by advection-type behavior. This leads to diminishing returns for equation-based ROMs when a large number of POD modes need to be resolved. There is a large volume of literature devoted to the stabilization (or closure) of intrusive ROMs when there is improper spectral support to resolve all frequencies [34–37]. Machine learning methods have also been used to learn these stabilizations to allow for hybrid modeling tasks where intrusive ROMs are corrected by data [38,39]. This allows the ROM user to balance the growing computational complexity of modal retention with data-driven corrections. Another limitation of POD-based ROMs comes into play when there is incomplete knowledge of the governing equations. This is encountered frequently when it is desirable to build emulators from experimental data. While less invasive ML methods may still be employed to act as corrections to equation-based models, there is considerable interest in directly building an efficient and accurate emulation framework using the data alone. These ROMs are often termed nonintrusive and have shown promising results for several applications where there may be a lack of underlying knowledge about the governing equations and data sets may suffer from incomplete observations [23,40–42].

In addition to turbulence closures and ROMs, we have also seen successful applications for ML in fluid data estimation and reconstruction problems [43–49]. These efforts have revealed that ML, due to its inherent ability to capture nonlinearity [50–52], is well suited to spatial flow data reconstruction and outperforms linear-reconstruction methods such as gappy POD [53] and linear stochastic estimation [54]. More recently, these investigated methods have also been applied to particle image velocimetry data [55–57] and understanding an interpretable relationship in between the predicted results and input data by focusing on vortical motions [58].

Despite the aforementioned efforts, there are limited studies on interpretable ML methods, particularly for the widely used neural networks. This is partly due to the fact that most ML methods are utilized as black boxes, which make point predictions. A maximum likelihood estimation of the loss function (usually a mean square error or cross-entropy cost function) via gradient descent methods is usually executed for training, resulting in any trained neural network predicting the target values in a deterministic fashion.

An alternative approach to obtain probabilistic outputs is to use Bayesian inference [59]. This approach relies on interpreting each trainable parameter of the network to be a random variable that may be sampled from (for each prediction). This leads to an output that can be characterized with a probability density function. Such Bayesian neural networks (for examples, see reviews in [60–63]) have been studied with various estimation schemes and applications. However, a Bayesian inference of the model parameters of the neural network is often prohibitively expensive in deeper architectures. Therefore, several approximations using variational inference [64–67], Monte Carlo dropouts [68,69], Gaussian process approximations [70,71], kernel analog forecasting [72], and maximum *a posteriori* estimation [73] are utilized to find the posterior of the weights during training. Most importantly, in the absence of such devices, a simple neural network result does not account for reliable uncertainty quantification, model selection, or convergence requirements. The view of reliability for predictions is further crucial for more practical applications, which require transparency and accountability.

In the present paper, we utilize a class of neural networks which assume that the predictions (in this case the observable) are generated from a distribution instead of providing just the best estimate. A broad class of such frameworks is referred to as mixture density networks (MDNs) [74]. Several variants of this method [75–77] have been developed over the years and many have been successfully applied in various scientific applications (for instance, in [78–80]). The probabilistic neural network (PNN) we incorporate is a special case of MDN when the estimate is a single Gaussian distribution, instead of a mixture of Gaussians.

We then consider its applicability for several canonical problems of fluid dynamics. While the nature of the distribution is user defined and may be interpreted as an addition hyperparameter, we utilize a unimodal Gaussian distribution. An attractive property of these networks is that a notion of uncertainty quantification is built into their architectures and any prediction for an observable is accompanied by an estimate for the corresponding uncertainty. This is obtained by assuming that the network predicts the mean and the variance of the posterior density function given the inputs it sees. The target is then compared to the mean of this density to compute a negative log-likelihood-based distance metric (which is minimized).

The major highlights of this investigation are summarized as follows.

(i) We propose the use of a probabilistic neural network architecture for efficiently embedding uncertainty quantification into data-driven tasks relevant to fluid flows.

(ii) We execute several tests to assess the strengths of the probabilistic framework for model order reduction, forecasting, and spatial field recovery across different applications.

(iii) We also assess the robustness of the network architecture in the presence of noisy data and incomplete information.

The present paper is organized as follows. The probabilistic neural network is introduced in Sec. II. In Sec. III we apply the framework to representative problem settings for the ROM and spatial fluid data recovery. We offer concluding remarks and future perspectives in Sec. IV.

II. PROBABILISTIC NEURAL NETWORKS

Conventional neural networks approximate an arbitrary mapping from inputs to outputs using a number of trainable parameters called weights. However, such networks provide discrete classes as either outputs (for classification problems) or continuous outputs (for regression). Generally, a full description of the output, i.e., an output estimated as a probability distribution function (PDF) for given inputs, is absent is such estimations. The MDNs, initially proposed in [74], aim to approximate this conditional probability distribution, where the PDFs of the output, conditioned on the inputs, are learned. The PDF is assumed to be a Gaussian mixture distribution [81]. This output is effectively a posterior estimate that quantifies the uncertainty of the estimation of a previously unseen data point.

Mixture density networks are not the only network architecture to provide uncertainty estimates. Bayesian networks remain an active field of research [60,62,82,83], with focus on scalability and interpretability of the posterior estimate. The probabilistic neural network framework we utilize is a special case of MDNs where the conditional probability estimate is assumed to a Gaussian. While MDNs with a large number of mixture components may approximate any arbitrary posterior estimate, it is achieved at a cost of computational expense and a larger requirement of training data.

Network framework

Predictions provided by neural networks for inputs $\boldsymbol{x}$ (such as the sensor measurements in Fig. 1) generally are determined from the minimization of a loss function as a function of weights $\mathcal{E}(\boldsymbol{w})$

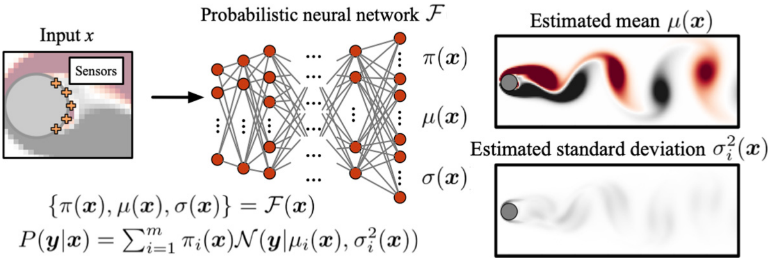


FIG. 1. Representative schematic of the probabilistic neural network. We consider the flow reconstruction from sensor measurements of a two-dimensional cylinder wake (Sec. III C 1) as an example problem.

(e.g., the mean square error). Such calculations between the truths $\boldsymbol{y}_t$ (such as the whole field in Fig. 1) and the predicted values from the network $\boldsymbol{y}_p(\boldsymbol{x})$ lack a probability distribution of $p(\boldsymbol{y}_p|\boldsymbol{x})$. Hence, the error bars on the estimates are typically absent in dense neural network outputs. To quantify uncertainty of the neural network estimates, the multidimensional surface of the loss function $\mathcal{E}(\boldsymbol{w})$ has to be explored in addition to finding the global minimum.

To obtain the posterior of estimates $p(\boldsymbol{y}_p|\boldsymbol{x})$, one may sample the network weights, thus providing a fully Bayesian inference on the parameter estimates. However, this method is computationally expensive unless the networks are shallow. For this reason, obtaining the probability distribution functions of the predictions in a computationally feasible manner is a challenging problem. To address this issue, we approach this problem by avoiding sampling altogether. Instead of mapping the neural network from inputs $\mathcal{F} : \boldsymbol{x} \to \boldsymbol{y}(\boldsymbol{x})$, one could define the mapping as $\mathcal{F} : \boldsymbol{x} \to (\mu, \sigma)$, where the mean μ and standard deviation σ parametrize a Gaussian probability distribution function $\mathcal{N}(\mu, \sigma)$. This approach ensures that the outputs $p(\boldsymbol{y}_p|\boldsymbol{x}) = \mathcal{N}(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$ are accompanied by uncertainty estimates assuming the errors are Gaussian.

For more complex probability distributions that depart from the Gaussian distribution, one may parametrize the mapping distribution accordingly. One straightforward extension is the Gaussian mixture model with the mapping $\mathcal{F} : \boldsymbol{x} \to (\pi_1, \mu_1, \sigma_1, \pi_2, \mu_2, \sigma_2, \ldots, \pi_N, \mu_N, \sigma_N)$, where π_i is the mixing probability for each Gaussian component satisfying the condition $\sum_{i=1}^{m} \pi_i = 1$. The distribution function in this model is a linear combination of several Gaussian components given by

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{m} \pi_i(\boldsymbol{x})\mathcal{N}(\mu_i(\boldsymbol{x}), \sigma_i(\boldsymbol{x})). \tag{1}$$

The value of m is generally prespecified based on the expectation of a posterior distribution. For example, if the output distribution is expected to be bimodal, one may select $m = 2$. The main advantage of this Gaussian mixture modeling, however, is that extremely complicated complex distributions can be approximated as a mixture of Gaussians with a large number of mixing components (large N). Alternatively, the number of mixing components can also be independently optimized along with the learning rate, decay rate, and other hyperparameters of the training scheme. It should be noted that the training of the MDN itself is supervised. However, mixture probabilities that are obtained from a fully trained network correspond to different clusters in the data that are learned in an unsupervised fashion.

Since our PNN provides a distribution of the estimation $p(\boldsymbol{y}_p|\boldsymbol{x})$ instead of a point prediction of $\boldsymbol{y}_p$, the loss function has to be chosen accordingly to utilize the full distribution of the prediction. This results in a crucial improvement of conventional neural networks. We choose to maximize the average likelihood of the training data, which captures the full information about the entropy between the distribution of the training data $\boldsymbol{y}_t$ and the corresponding prediction $\boldsymbol{y}_p$. Hence, in practice, our model $\mathcal{F}(\boldsymbol{x}; \boldsymbol{w})$ is trained to obtain the optimized weights $\boldsymbol{w}$ by minimizing the error

Algorithm 1. Training procedure for probabilistic neural network $\mathcal{F}$.

```
$\boldsymbol{w} \leftarrow$ Initialize network parameters
while $\mathcal{E}$ is not converged do
  Update probabilistic neural network
  $x \leftarrow$ Random minibatch from data set
  for $x_k$ in $x$ do
    Compute Gaussian mixture parameters: $(\pi_k, \mu_k, \sigma_k) \leftarrow \mathcal{F}(x_k; \boldsymbol{w})$
    Compute predictive distribution: $p(y_{k,p}|x_k) \leftarrow (\pi_k, \mu_k, \sigma_k)$
  end for
  Compute negative log-likelihood: $\mathcal{E} \leftarrow -\log \mathcal{L}$
  Update network parameters: $\boldsymbol{w} \leftarrow \text{Adam}(\nabla_{\boldsymbol{w}}\mathcal{E}, \boldsymbol{w})$
end while
```

function $\mathcal{E}$ given in terms of the average log-likelihood $\mathcal{L}$ such that

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\text{argmin}}[\mathcal{E}], \quad \text{where } \mathcal{E} \equiv -\log \mathcal{L} = -\sum_{k=1}^{K} p(y_{k,p}|x_k) \log p(y_{k,t}), \tag{2}$$

with k indicating each data point in the training data and K denoting a number of training samples. The term $p(y_{p,k}|x_k)$ in the error function is evaluated for each data point using the output of the network given in Eq. (1).

Cross entropy is a common error function used for classification problems using neural networks. In a continuous limit (i.e., in regression problems like ours), the cross entropy reduces to the mean square error between the outputs of the neural network and the targets. On the other hand, the negative log-likelihood shown in Eq. (2) also reduces to the mean square error when mean predictions alone are considered (i.e., in point predictions instead of probability distribution outputs). In other words, the log-likelihood and cross entropy are analogous cost functions in regression and classification problems, respectively [84]. Thus the likelihood maximizing model is equivalent to minimizing the cross entropy $H(p(\boldsymbol{y}_p|\boldsymbol{x}), p(\boldsymbol{y}_t))$. Hence our framework incorporates a generic loss prescription that can be applied to a wide variety of problems and specifies a representative uncertainty in the machine-learned estimates.

Our implementation of a PNN is shown for one of the applications (Sec. III A) in Fig. 1. In this example, the inputs $\boldsymbol{x} \in \mathbb{R}^5$ to the network model are the sensor measurements. These are mapped to the targets $\boldsymbol{y}_p$ whose truth values are a whole wake field of a two-dimensional cylinder $\boldsymbol{y}_t$ of dimension 13 440. The network for the example of Fig. 1 is a fully connected dense network with eight layers and the number of neurons per layer being $5 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1024 \rightarrow 2048 \rightarrow 13\,440 \times 3$. The last layer ($13\,440 \times 3$) corresponds to (π_1, μ_1, σ_1) for each grid point that parametrizes the predictive conditional distribution $p(\boldsymbol{y}_p|\boldsymbol{x})$ for the whole wake field, and the error function is calculated using Eq. (2). We remind the reader that we use only one Gaussian center in this investigation and focus on one value each of mean and variance (μ_1 and σ_1), which implies $\pi_1 = 1$ corresponding to each target. We utilize the same number of Gaussian centers (i.e., only one) for all our assessments hereafter. As an example, if three Gaussian centers were utilized, the final layer of the above map would have dimensions of $13\,440 \times 3 \times 3$ corresponding to (π_i, μ_i, σ_i) for $i = 1, 2, 3$ and $\sum_{i=1}^{3} \pi_i = 1$.

For updating the weights $\boldsymbol{w}$, we use the Adam optimizer [85]; the training procedure of the PNN is summarized in Algorithm 1. We note that we do not attempt to optimize the parameters of the network architecture for each problem setting in the present study. Our primary objective here is to demonstrate the applicability of the probabilistic model to quantify the uncertainty of machine-learned estimations for canonical fluid flow problems. One may consider the use of theoretical optimization methods such as Hyperopt [86] and Bayesian optimization [23,87] to

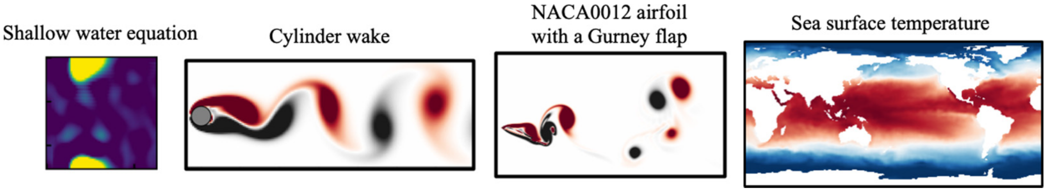


FIG. 2. Different canonical test cases considered in the present work.

further enhance the accuracy of the estimation which can be improved. A sample code for the present model is available online [88]. The code for the present study has been written in house by utilizing TENSORFLOW 2.2.0 on PYTHON 3.7. The present computation for the probabilistic neural network is performed on NVIDIA Tesla V100 GPU.

III. RESULTS

In this section, we demonstrate the capabilities of the PNN introduced in the preceding section through a variety of experiments. These experiments are (i) the prediction of POD coefficient evolution given initial conditions (Sec. III A) and (ii) the instantaneous estimation of POD coefficients from local sensor measurements (Sec. III B). Next we consider the applications to spatial fluid data recovery given by (a) the estimation of sensor measurements from other sensor placements and (b) the estimation of whole flow fields from local sensor measurements for three example problems: (1) a two-dimensional cylinder wake at $\mathrm{Re}_D = 100$ (Sec. III C 1), (2) a two-dimensional wake of a NACA0012 airfoil with a Gurney flap at $\mathrm{Re}_c = 1000$ (Sec. III C 2), and (3) the NOAA optimum interpolation sea surface temperature data set [89] (Sec. III C 3). The broad spread of potential applications can be observed through flow fields from our chosen data sets as seen in Fig. 2.

A. Parametric surrogates

In the following, we review the POD technique for the construction of a reduced basis [19,90–92] for our observed variable. This is for the express purpose of easily implementing a data-driven recovery of the system evolution in reduced-space. The POD procedure is tasked with identifying a space

$$\mathbf{X}^f = \mathrm{span}\{\boldsymbol{\vartheta}^1, \dots, \boldsymbol{\vartheta}^f\}, \tag{3}$$

which approximates snapshots $\boldsymbol{\vartheta}^i$ optimally with respect to the L_2 norm. The process of $\boldsymbol{\vartheta}$ generation commences with the collection of snapshots in the snapshot matrix

$$\mathbf{S} = \begin{bmatrix} \hat{\boldsymbol{q}}_h^1 & \hat{\boldsymbol{q}}_h^2 & \cdots & \hat{\boldsymbol{q}}_h^{N_s} \end{bmatrix} \in \mathbb{R}^{N_h \times N_s}, \tag{4}$$

where N_s is the number of snapshots and $\hat{\boldsymbol{q}}_h^i : \mathcal{T} \times \mathcal{P} \to \mathbb{R}^{N_h}$ corresponds to an individual snapshot in time of the discrete solution domain with the mean value removed, i.e.,

$$\hat{\boldsymbol{q}}_h^i = \boldsymbol{q}_h^i - \bar{\boldsymbol{q}}_h, \quad \bar{\boldsymbol{q}}_h = \frac{1}{N_s}\sum_{i=1}^{N_s} \boldsymbol{q}_h^i, \tag{5}$$

with $\bar{\boldsymbol{q}}_h : \mathcal{P} \to \mathbb{R}^{N_h}$ the time-averaged solution field. Our POD bases can then be extracted efficiently through the method of snapshots where we solve the eigenvalue problem on the correlation matrix $\mathbf{C} = \mathbf{S}^T\mathbf{S} \in \mathbb{R}^{N_s \times N_s}$. Then

$$\mathbf{C}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}, \tag{6}$$

where $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{N_s}\} \in \mathbb{R}^{N_s \times N_s}$ is the diagonal matrix of eigenvalues and $\mathbf{W} \in \mathbb{R}^{N_s \times N_s}$ is the eigenvector matrix. Our POD basis matrix can then be obtained by

$$\boldsymbol{\vartheta} = \mathbf{SW} \in \mathbb{R}^{N_h \times N_s}. \tag{7}$$

In practice, a reduced basis $\boldsymbol{\psi} \in \mathbb{R}^{N_h \times N_r}$ is built by choosing the first N_r columns of $\boldsymbol{\vartheta}$ for the purpose of efficient ROMs, where $N_r \ll N_s$. This reduced basis spans a space given by

$$\mathbf{X}^r = \mathrm{span}\{\boldsymbol{\psi}^1, \dots, \boldsymbol{\psi}^{N_r}\}. \tag{8}$$

The coefficients of this reduced basis (which capture the underlying temporal effects) may be extracted as

$$\mathbf{A} = \boldsymbol{\psi}^T \mathbf{S} \in \mathbb{R}^{N_r \times N_s}. \tag{9}$$

The POD approximation of our solution is then obtained via

$$\hat{\mathbf{S}} = [\tilde{\boldsymbol{q}}_h^1 \quad \tilde{\boldsymbol{q}}_h^2 \quad \cdots \quad \tilde{\boldsymbol{q}}_h^{N_s}] \approx \boldsymbol{\psi}\mathbf{A} \in \mathbb{R}^{N_h \times N_s}, \tag{10}$$

where $\tilde{\boldsymbol{q}}_h^i : \mathcal{T} \times \mathcal{P} \to \mathbb{R}^{N_h}$ corresponds to the POD approximation to $\hat{\mathbf{q}}_h^i$. The optimal nature of reconstruction may be understood by defining the relative projection error

$$\frac{\sum_{i=1}^{N_s} \|\hat{\boldsymbol{q}}_h^i - \tilde{\boldsymbol{q}}_h^i\|_{\mathbb{R}^{N_h}}^2}{\sum_{i=1}^{N_s} \|\hat{\boldsymbol{q}}_h^i\|_{\mathbb{R}^{N_h}}^2} = \frac{\sum_{i=N_r+1}^{N_s} \lambda_i^2}{\sum_{i=1}^{N_s} \lambda_i^2}, \tag{11}$$

which exhibits that with increasing retention of POD bases, increasing reconstruction accuracy may be obtained. Our first application of PNNs will be tasked with predicting $\tilde{\boldsymbol{q}}_h^i$ given information about parameters that control the evolution of the system.

1. *Inviscid shallow-water equations*

Our first example considers the two-dimensional inviscid shallow-water equations which are a prototypical system for geophysical flows. The governing equations are hyperbolic in nature and are

$$\frac{\partial(\rho\eta)}{\partial t} + \frac{\partial(\rho\eta u)}{\partial x} + \frac{\partial(\rho\eta v)}{\partial y} = 0, \tag{12a}$$

$$\frac{\partial(\rho\eta u)}{\partial t} + \frac{\partial}{\partial x}\left(\rho\eta u^2 + \frac{1}{2}\rho g\eta^2\right) + \frac{\partial(\rho\eta uv)}{\partial y} = 0, \tag{12b}$$

$$\frac{\partial(\rho\eta v)}{\partial t} + \frac{\partial(\rho\eta uv)}{\partial x} + \frac{\partial}{\partial y}\left(\rho\eta v^2 + \frac{1}{2}\rho g\eta^2\right) = 0. \tag{12c}$$

In the above set of equations, η corresponds to the total fluid column height and (u, v) is the horizontal flow velocity of the fluid averaged across the vertical column. Furthermore, g is gravitational acceleration and ρ is the fluid density fixed to be unity. Equation (12a) represents conservation of mass and Eqs. (12b) and (12c) define the conservation of momentum. For simplicity, we define $\boldsymbol{q} = [\rho\eta, \rho\eta u, \rho\eta v]^T$ and we use η and $\rho\eta$ interchangeably hereafter. We also point out that there is no linear diffusion term in the above shallow-water equations. This implies that our system evolution is solely advection dominated with any dissipation occurring as a result of numerical viscosity alone. Our initial conditions are given by

$$\rho\eta(x, y, t = 0) = 1 + \exp\left[-\left(\frac{(x-\bar{x})^2}{2(5e+4)^2} + \frac{(y-\bar{y})^2}{2(5e+4)^2}\right)\right],$$
$$\rho\eta u(x, y, t = 0) = 0, \quad \rho\eta v(x, y, t = 0) = 0, \tag{13}$$

while our two-dimensional domain is a square with periodic boundary conditions. These initial conditions $\boldsymbol{r} = [\bar{x}, \bar{y}]$ $(-0.5 \leqslant \{\bar{x}, \bar{y}\} \leqslant 0.5)$ correspond to the parameter controlling the spatiotemporal evolution of the dynamical system.

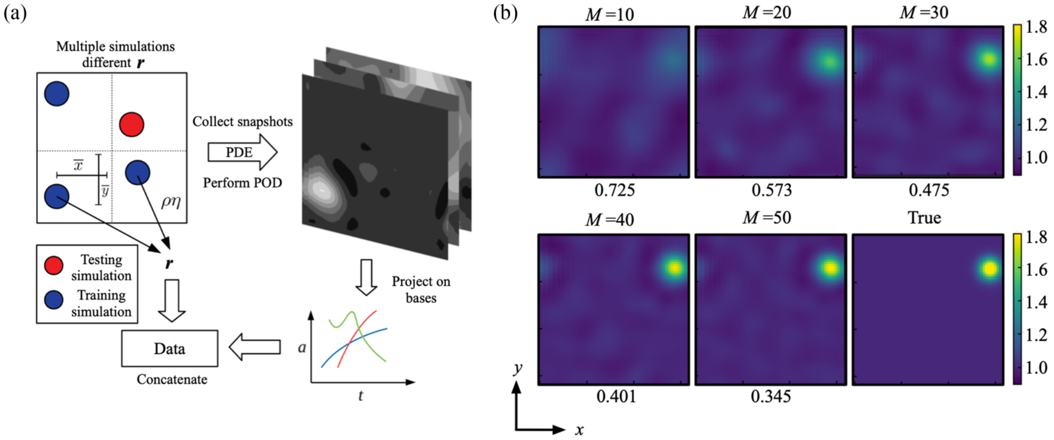


FIG. 3. Information of the data set for the shallow-water equation. (a) Schematic for the generation of training and testing data. Given inputs of $\boldsymbol{r}$ (i.e., the location of an initial Gaussian perturbation to η), the machine learning ROM is tasked with predicting the evolution of 40 POD coefficients over ten snapshots in time. The prediction is one shot. (b) Convergence of reduced representations to the true field with increasing M. The values underneath each field indicate the normalized L_1 error norm $\epsilon = |\eta_{\text{true}} - \eta_{\text{POD}}|/|\eta_{\text{true}} - 1|$.

Our data generation process utilizes full-order solutions of the above system of equations until $t = 0.1$ with a time step of 0.001. Our full-order model uses a fourth-order-accurate Runge-Kutta integration scheme and a fifth-order-accurate weighted essentially nonoscillatory (WENO) scheme [93] for computing state reconstructions at cell faces. The Rusanov Riemann solver is utilized for flux reconstruction after cell-face quantities are calculated. The reader is directed to [94] for further discussion of the temporal integration scheme and [95] for details on the WENO scheme and the Riemann solver implementation in two-dimensional problems.

The procedure for generating the training and testing data is illustrated in Fig. 3(a). A hundred different locations $\boldsymbol{r} = [\bar{x}, \bar{y}]$ are chosen for full-order simulations. We note that these 100 locations are chosen by Latin hypercube sampling. Out of the 100 simulations for which snapshots are obtained, 90 simulations are set aside for the purpose of training machine learning frameworks and for POD basis generation. Each simulation is sampled ten times temporally to obtain a total of 900 training snapshots for POD basis generation. Coefficients obtained by projecting these training simulations onto the POD bases are configured to be the target data for training the machine learning frameworks. For accurately capturing the evolution of the perturbation, we choose $M = 40$ as the number of POD coefficients with the normalized L_1 error norm $\epsilon = |\eta_{\text{true}} - \eta_{\text{POD}}|/|\eta_{\text{true}} - 1|$ of 0.401, as shown in Fig. 3(b). All our assessments for the machine learning frameworks are performed on snapshots obtained from the ten simulations kept aside for the purpose of testing. Also, our assessments are performed solely for the η flow field. We emphasize this point, since we assume that the other key variables $\rho\eta u$ and $\rho\eta v$ are not available for observation. This aligns with practical applications for fluid dynamics where constraints of cost or safety make it impractical to observe all the physics of a system. We also draw attention to the fact that the initial condition of the entire system is conditioned on a Gaussian excitation to $\rho\eta$ alone. Since the other conserved variables are never observed for generating training data, the framework cannot be expected to work if there are unseen perturbations to their initial conditions.

We first assess the ability of the proposed PNN for predicting the evolution of the shallow-water-equation dynamics given an input of $\boldsymbol{q} = [\bar{x}, \bar{y}]$ alone. For this task, we have a two-dimensional input space $\boldsymbol{q}$ and a 400-dimensional output space $\boldsymbol{a}_t = [\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^{10}]$ corresponding to 40 spatial POD coefficients $\boldsymbol{a}^t = [a_1^t, a_2^t, \ldots, a_{40}^t]$ over ten time steps so as to account for the entire trajectory

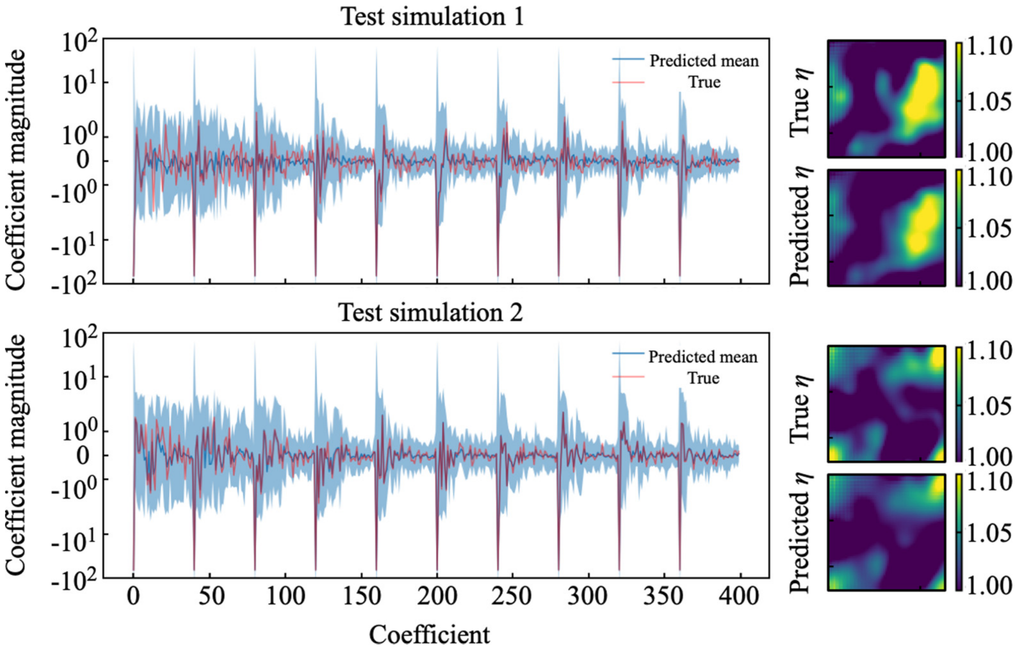


FIG. 4. Coefficient estimations of two representative test simulations by the present framework with the shallow-water equation showing error bars for two standard deviations around the mean as parametrized by a Gaussian distribution. Final time field reconstructions using estimated POD coefficient means are shown on the right-hand side of each coefficient evolution. True indicates the reference field with 40 POD modes.

of the η dynamics. Following Eq. (1) and Fig. 1, the problem setting here can be formulated as

$$\{\pi(\boldsymbol{q}), \mu(\boldsymbol{q}), \sigma(\boldsymbol{q})\} = \mathcal{F}(\boldsymbol{q}), \quad p(\boldsymbol{a}_t|\boldsymbol{q}) = \sum_{i=1}^{m} \pi_i(\boldsymbol{q})\mathcal{N}(\mu_i(\boldsymbol{q}), \sigma_i(\boldsymbol{q})). \tag{14}$$

For the purpose of effective training, all POD coefficients are scaled between 0 and 1. A schematic for this task and its assessment is shown in Fig. 3(a).

The coefficient estimations from the present framework for two representative simulations are shown in Fig. 4. The 400 coefficients are presented sequentially with 40 coefficients representing the information needed to reconstruct the flow field at one time instant. The coefficients are also accompanied by confidence intervals spanning two standard deviations on either side of the predicted means and represent the uncertainty quantification mechanism built into the probabilistic neural network. The dissipation of the coherent structures at later snapshots also leads to a reduced prediction of uncertainty by the framework, whereas at earlier snapshots, much larger uncertainty is observed due to the presence of coherent distortions of the solution field. The corresponding field reconstructions using the mean coefficients can be seen on the right-hand side of each coefficient evolution where qualitative agreement with the true simulations is clearly observable. We obtain a normalized L_1 error norm of 0.843 for the reconstructed field of test simulations.

At this juncture, we provide some remarks on the nature of the system that is emulated. The initial and boundary conditions for this particular shallow-water-equation experiment represent a tightly controlled traveling-wave problem that is translationally invariant. Different realizations of the initial condition lead to translationally shifted trajectories. We also note the presence of mirror symmetries with respect to $x = \bar{x}$ and $y = \bar{y}$ coupled with a rotational symmetry of $\pi/2$ rad about the origin. We caution the reader that the current ROM strategy is devoid of any symmetry-preserving mechanisms. In fact, the projection of the true field onto the 40 dominant POD modes breaks the

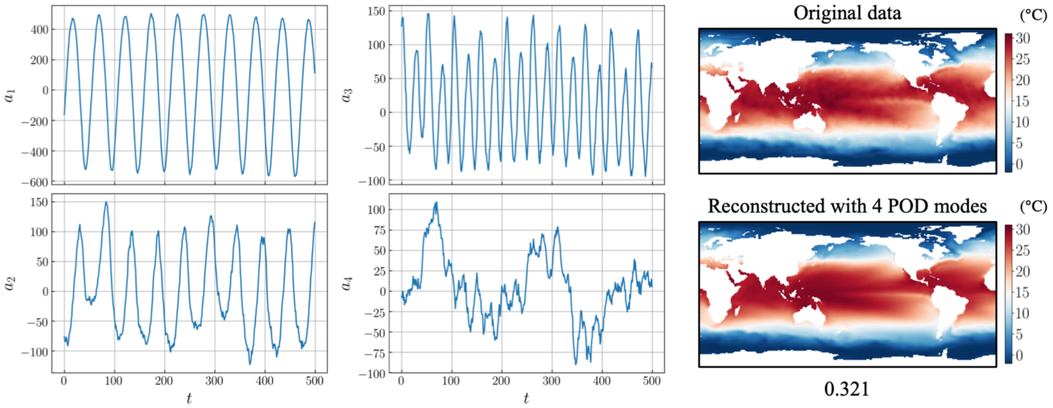


FIG. 5. POD coefficients and entire field of NOAA sea surface temperature. The value underneath the contour is the L_2 error norm $\epsilon_{\mathrm{POD}} = \|T_{\mathrm{ref}} - T_{\mathrm{POD}}\|_2 / \|T_{\mathrm{ref}}\|_2$.

aforementioned symmetries and thus leads to the lack of symmetry preservation within the training data itself. However, our motivation for a first assessment of our PNN on this system stems from the well-known fact that ROMs obtained from POD-based compression are severely limited in their ability to forecast on these simple traveling-wave systems [41,96] and require special treatment with intrinsic knowledge of the flow dynamics. Also, the surrogate model proposed here is conditioned on the initial flow field of $\rho\eta$ alone. Thus, this framework represents a promising approach for incomplete observations of geophysical dynamics where complete knowledge of physics is almost always impossible. We further this claim by actual experiments on remote sensing data sets later in this study.

2. *NOAA sea surface temperature*

To demonstrate the applicability of the PNN to practical applications, let us consider the NOAA sea surface temperature data set. The field data here have the spatial resolution of 360 × 180 based on a one degree grid and is obtained from satellite and ship observations without adequate knowledge of underlying governing equations. We use 20 years of data (1040 snapshots spanning years 1981 to 2001) as the training data set, while the test data set is prepared from 874 snapshots spanning from 2001 to 2018. This test setting is extrapolation in time but not physics, since the data set has influence of seasonal periodicity. The aforementioned problem setting follows the work of Callaham *et al.* [97], who attempted to reconstruct fluid flow fields from local sensors using sparse representations. Following Callaham *et al.* [97], the input sensors for the baseline model are chosen randomly from the region of 50° S to 50° N.

For the task of constructing a parametric surrogate for forecasting the sea surface temperature (SST), the PNN attempts to predict the temporal evolution of four POD coefficients $\boldsymbol{a}^t = [a_1^t, a_2^t, a_3^t, a_4^t]$ over 100 weeks $\boldsymbol{a}_t = [\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^{100}]$ from the local sensor measurements on the first week snapshot $\boldsymbol{s}^1$, i.e., initial information. The problem setting is expressed as

$$\{\pi(\boldsymbol{s}^1), \mu(\boldsymbol{s}^1), \sigma(\boldsymbol{s}^1)\} = \mathcal{F}(\boldsymbol{s}^1), \quad p(\boldsymbol{a}_t|\boldsymbol{s}^1) = \sum_{i=1}^{m} \pi_i(\boldsymbol{s}^1)\mathcal{N}(\mu_i(\boldsymbol{s}^1), \sigma_i(\boldsymbol{s}^1)). \tag{15}$$

As shown in Fig. 5, the reconstructed field with four spatial POD modes shows qualitative agreement with the original data globally, while the seasonal patterns at the global scale can be captured with modes 1, 2, and 3 and stochastic fluctuations may be represented with mode 4 [98].

The baseline results predicted from ten input sensors are summarized in Fig. 6. The periodic trends represented by modes 1, 2, and 3 can be reasonably captured from only ten sensor

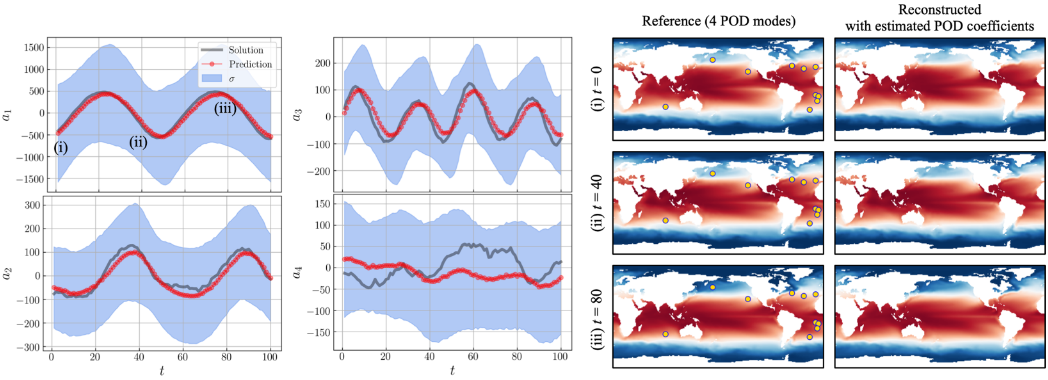


FIG. 6. Prediction of POD coefficient evolution from sensors at the first snapshot. Representative temporal evolution and the reconstructed field with POD eigenvectors at the first, 40th, and 80th weeks are shown.

measurements on the first snapshot while telling us the confidence interval of its estimation. Note, however, that the predicted temporal evolution of mode 4, which represents more stochasticity than that of the first three modes, exhibits further greater deviation compared to the reference. This result provides motivation to examine the influence on the number of input sensors, as discussed in the rest of this article.

We investigate the dependence of the prediction accuracy on the number of input sensors, as shown in Fig. 7. The curve starts to match with the reference data on increasing the number of input sensors, which can be particularly seen with mode 3 in Fig. 7(a). Similar trends can be confirmed in both Figs. 7(b) and 7(c), which show the reconstructed fields and L_2 error norms. Regarding the standard deviations presented in Fig. 7(d), the estimated uncertainties show no significant difference over the covered number of inputs, while decreasing by adding the training snapshots. This is due to the fact that the confidence interval of the present framework is not a barometer for error but a measure of the quality of the training data. This point can also be found in the later examples.

B. Field reconstructions through POD coefficient estimation for the shallow-water equations

In this section, we demonstrate the capability of the proposed framework to reconstruct the state of the field by random sampling in the domain. Similar to the preceding section, we task the probabilistic neural network with predicting the POD coefficients $\boldsymbol{a}$. Our inputs, however, are now given by sensor measurements $\boldsymbol{s}$ of the field in addition to a time stamp q_t that indicates the progress to the final time of the evolution. Hence, the problem setting here can be expressed as

$$\{\pi([\boldsymbol{s}, q_t]), \mu([\boldsymbol{s}, q_t]), \sigma([\boldsymbol{s}, q_t])\} = \mathcal{F}([\boldsymbol{s}, q_t]),$$
$$p(\boldsymbol{a}|[\boldsymbol{s}, q_t]) = \sum_{i=1}^{m} \pi_i([\boldsymbol{s}, q_t])\mathcal{N}(\mu_i([\boldsymbol{s}, q_t]), \sigma_i([\boldsymbol{s}, q_t])). \tag{16}$$

Note that this estimation is performed at the same instantaneous field between the input and output. We clarify that the random sensors that are measured for the purpose of field reconstruction are not perturbed by noise at the moment. We also compare the performance of the PNN against the well-known gappy POD technique for flow-field recovery.

Figure 8 shows the ability of the proposed framework to recover the coherent structures in the field for the shallow-water-equation system introduced in Sec. III A 1. Note that we present L_2 error norms of both estimated POD coefficients and reconstructed fields for Fig. 8(b). For reference, the average range of the 40 POD coefficients magnitude is 3.289 and the root mean square error of the POD coefficients is well within this range. A representative test simulation shows that increasing

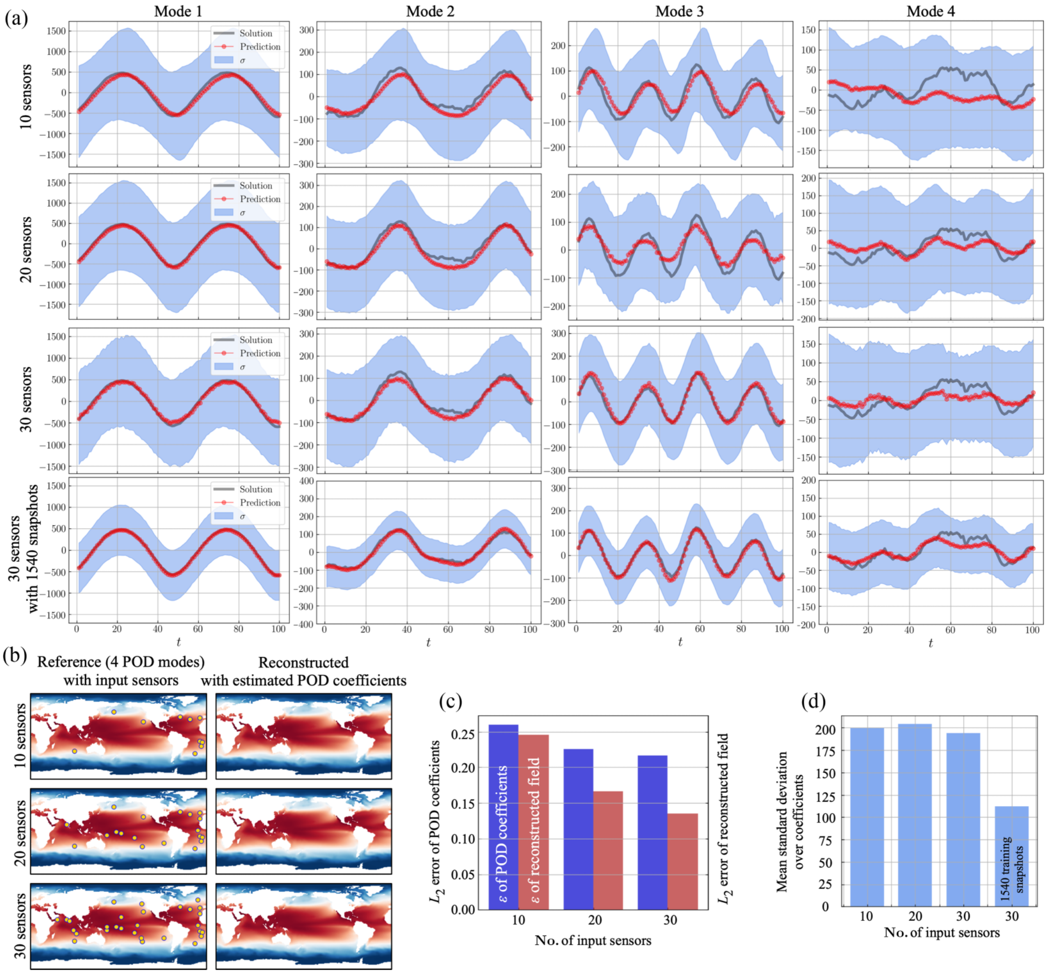


FIG. 7. Dependence of the POD coefficient prediction for sea surface temperature on the number of input sensors. (a) Temporal evolution of four POD coefficients. (b) Reconstructed temperature field with POD eigenvectors. (c) L_2 error norm of predicted POD coefficients and reconstructed field. (d) Ensemble average of estimated standard deviation taken over all POD coefficients.

the number of sensors leads to a reduction in errors. Note that, due to the random nature of sensor placement, at lower numbers, if point signals of coherent structures are not sampled effectively, larger errors may be obtained. This may explain why the lowest L_2 error norms appear at 30 sensors, as presented in Figs. 8(b) and 8(c). This suggests that an optimal selection of sensor locations based on regions of high uncertainty may improve convergence significantly. This inference can also be applied to the estimated standard deviations which do not show significant difference over the considered noise magnitude, as shown in Fig. 8(d).

For a thorough comparison of the PNN reconstruction with the well-known flow-field reconstruction methods, we show a comparison for flow reconstruction using 30 sensors for the gappy POD method [99] in Fig. 9. The number of sensors (30) is two orders of magnitude lower than the total number of degrees of freedom (4096) and therefore linear reconstruction methods are destined to be at a disadvantage for this test case. We also validate our gappy POD method by also comparing against improved linear reconstructions utilizing 3800 sensors. For all our testing simulations, the normalized L_1 error norms in field reconstruction are 0.155 for the PNN using 30

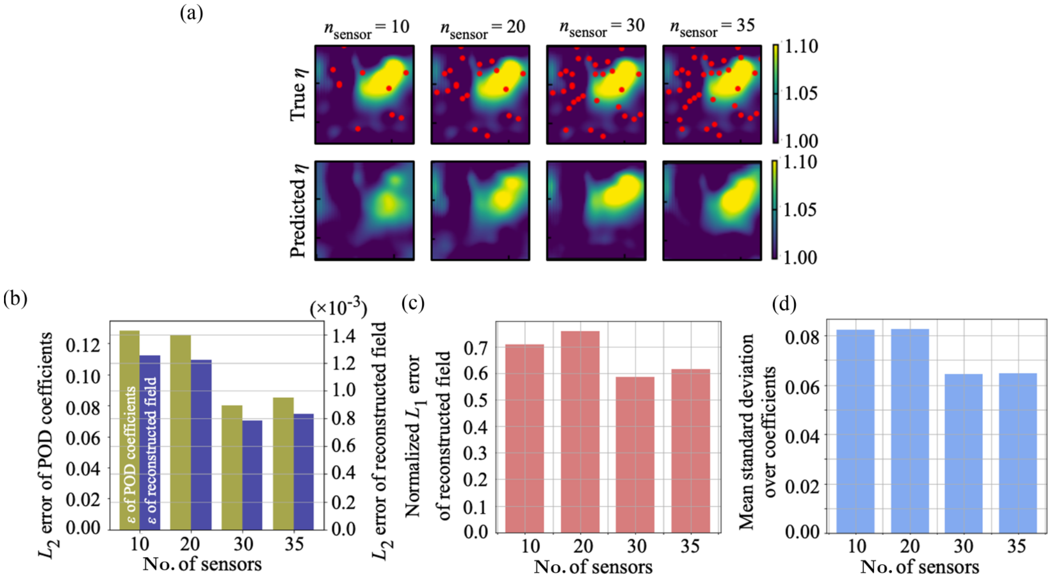
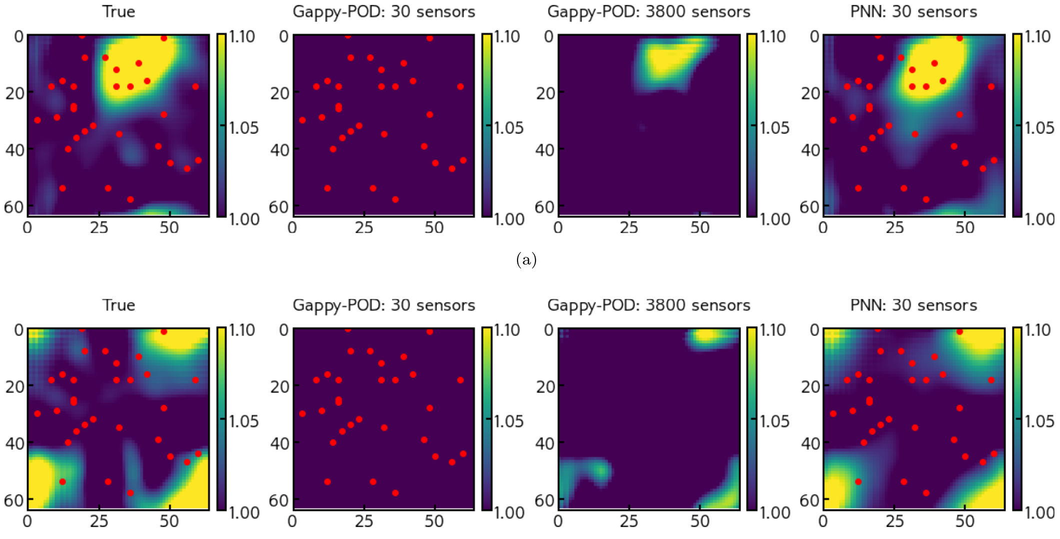


FIG. 8. Effect of the number of sensors for coefficient estimation of the shallow-water equation on the accuracy. (a) Field reconstructions with 10, 20, 30, and 35 sensors. Also shown is the relationship between the number of sensors and (b) L_2 errors of the estimated POD coefficient and reconstructed field, (c) the normalized L_1 error of the reconstructed field, and (d) the ensemble average of the estimated standard deviation taken over all POD coefficients. For reference, the average range of the POD coefficient magnitude is 3.289.



(a)

(b)

FIG. 9. Comparison between the PNN and Gappy POD for POD coefficient reconstruction of two representative testing solutions. For all our testing simulations, the normalized L_1 error norms in field reconstruction are 0.155 for the PNN using 30 sensors, 3.18 for gappy POD using 30 sensors, and 0.277 for gappy POD using 3800 sensors. Sensor locations (shown in red) are not provided for the validation with 3800 sensors for visual clarity.

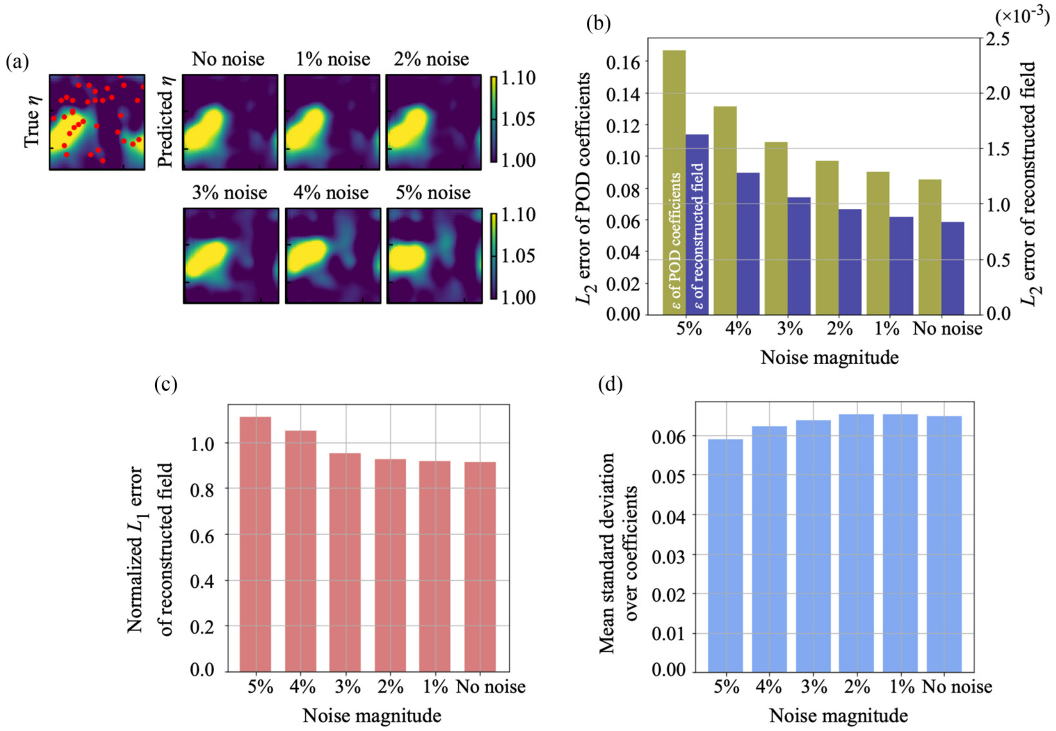


FIG. 10. Dependence of the accuracy for coefficient estimation of the shallow-water equation on the magnitude of noisy input. (a) Field reconstructions with 35 random sensors. Also shown is the relationship between the number of sensors and (b) L_2 errors of the estimated POD coefficient and reconstructed field, (c) the normalized L_1 error of the reconstructed field, and (d) the ensemble average of the estimated standard deviation taken over all POD coefficients. For reference, the average range of the POD coefficient magnitude is 3.289.

sensors, 3.18 for gappy POD using 30 sensors, and 0.277 for gappy POD using 3800 sensors. This result represents the advantages of using a nonlinear reconstruction procedure. Let us also recall that the gappy POD reconstruction for POD coefficients is deterministic whereas the PNN provides uncertainty predictions conditioned on the training data. The latter aspect will be explored in greater detail when the PNN is reformulated for direct spatial field recovery for challenging engineering and geophysical applications.

Following our assessments with reconstruction without the presence of sensor noise, we turn our attention to the effect of measurement noise at the random sampled locations looking at 35 randomly placed sensor locations. Results from training on noisy inputs are shown in Fig. 10. Figures 10(b) and 10(c) indicate that the framework becomes unsuitable for inputs perturbed with uniformly sampled noise corresponding to around 4% of the maximum value of the field. This may be due to the relatively low difference in magnitudes of the coherent structures in the field from the background flow. With regard to estimated standard deviations shown in Fig. 10(d), we clarify that the uncertainty estimation for the POD coefficients does not show a significant difference over the noise magnitude considered. We should note that this is because the estimated confidence interval is just for estimations and not a barometer for error. The reader should note that perturbation of inputs by noise causes errors but does not significantly affect the posterior, which assumes that the input is also a random variable. For both assessments in Figs. 8 and 10, this also indicates that a greater number of training snapshots is necessary to reduce the estimated uncertainty. The effect of number training snapshots will be assessed for a different problem later on in this study. Note that we have already assessed NOAA SST coefficient estimations via the PNN given field measurements, so a separate assessment is superfluous here.

C. Spatial fluid data recovery

In this section, we introduce the application of the PNN to spatial fluid flow reconstructions with connections to real engineering and geophysical applications. We consider a two-dimensional cylinder wake (Sec. III C 1), the wake of a NACA0012 airfoil with a Gurney flap (Sec. III C 2), and the NOAA optimum interpolation sea surface temperature data set (Sec. III C 3). We would like to note that the latter is constructed from satellite and ship observations and represents a real-world flow-field reconstruction task with no underlying information of governing equations. Another important remark here is that while previous flow-field reconstructions were via POD coefficient estimation, the PNN is used directly for spatial reconstruction.

For the examples covered in this section, we consider two types of problems: (1) estimation of the sensor measurement s_{target} from other sensors $\boldsymbol{s}_{\text{input}}$ and (2) estimation of the whole flow field $\boldsymbol{z}$ from local sensor measurements $\boldsymbol{s}_{\text{input}}$. These settings can be expressed as

$$\{\pi(\boldsymbol{s}_{\text{input}}), \mu(\boldsymbol{s}_{\text{input}}), \sigma(\boldsymbol{s}_{\text{input}})\} = \mathcal{F}(\boldsymbol{s}_{\text{input}}), \quad p(s_{\text{target}}|\boldsymbol{s}_{\text{input}}) = \sum_{i=1}^{m} \pi_i(\boldsymbol{s}_{\text{input}})\mathcal{N}(\mu_i(\boldsymbol{s}_{\text{input}}), \sigma_i(\boldsymbol{s}_{\text{input}})), \tag{17}$$

$$\{\pi(\boldsymbol{s}_{\text{input}}), \mu(\boldsymbol{s}_{\text{input}}), \sigma(\boldsymbol{s}_{\text{input}})\} = \mathcal{F}(\boldsymbol{s}_{\text{input}}), \quad p(z|\boldsymbol{s}_{\text{input}}) = \sum_{i=1}^{m} \pi_i(\boldsymbol{s}_{\text{input}})\mathcal{N}(\mu_i(\boldsymbol{s}_{\text{input}}), \sigma_i(\boldsymbol{s}_{\text{input}})) \tag{18}$$

for the sensor estimation and the whole field estimation, respectively.

1. Two-dimensional cylinder wake

Let us consider a two-dimensional cylinder wake at $\text{Re}_D = 100$ as an example of application to unsteady flows around a bluff body. The data set has been obtained by using a two-dimensional DNS [100,101]. The governing equations are the incompressible Navier-Stokes equations

$$\nabla \cdot \boldsymbol{u} = 0, \tag{19}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{\text{Re}_D}\nabla^2 \boldsymbol{u}, \tag{20}$$

where $\boldsymbol{u}$, p, and Re_D are the nondimensionalized velocity vector, pressure, and Reynolds number based on the cylinder diameter D, respectively. Five nested levels of multidomains are considered for numerical setup. The finest level of the domain considered here is $(x/D, y/D) = [-1, 15] \times [-8, 18]$ and the largest domain is $(x/D, y/D) = [-5, 75] \times [-40, 40]$. The time step for the present DNS is $\Delta t = 2.50 \times 10^{-3}$. As for the training data set, the domain around a cylinder body is extracted, i.e., $((x/D)^*, (y/D)^*) = [-0.7, 15] \times [-3.3, 3.3]$ and $(N_x, N_y) = (192, 70)$. The vorticity field is used as both input and output attributes in this case. Five sensor measurements located on a cylinder surface are chosen as input data $\boldsymbol{s}_{\text{input}}$ for both the sensor estimation and the whole field reconstruction. For both problem settings with the cylinder data set, we prepare 100 snapshots over approximately four periods in time as the training data. The assessments are conducted using test data set which also contains 100 snapshots over approximately four periods and is excluded from the training process.

For the problem settings of spatial fluid data recovery, we first apply the PNN to estimate local sensor measurements and then extend it to the concept of whole flow reconstruction. Here let us present the result of estimation on sensor 6 s_6 located on a wake region from the sensors located on the cylinder surface in Fig. 11. The estimations (red circles) are in excellent agreement with the reference data. This trend can also be seen from a quantitative assessment with an L_2 error norm $\epsilon =$

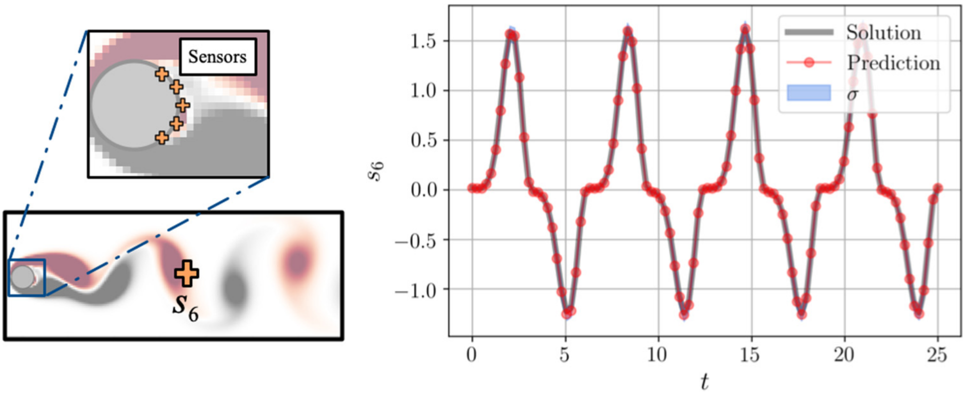


FIG. 11. Sensor estimation of the cylinder wake using the PNN. The left-hand side shows both input and output sensor locations. The L_2 error norm $\epsilon = \|s_{6,\mathrm{DNS}} - s_{6,\mathrm{ML}}\|_2/\|s_{6,\mathrm{DNS}}\|_2$ is 0.0137. Note here that the σ can be hardly seen due to the excellent agreement of the estimation with the reference.

$\|s_{6,\mathrm{DNS}} - s_{6,\mathrm{ML}}\|_2/\|s_{6,\mathrm{DNS}}\|_2$ of 0.0137. Since the cylinder wake at the present Reynolds number is of periodic nature in time, high-standard-deviation regions are not observed. This observation also enables us to have confidence in this estimation.

We then extend the model to the reconstruction of the whole wake, as shown in Fig. 12. Analogous to the local sensor estimation, we see that the estimated flow field shows nice agreement with the reference DNS data from both qualitative and quantitative assessments. What is notable is that the PNN provides us with the standard deviations of estimation as shown on the right-hand side of Fig. 12. In this case, the probabilistic machine learning model tells us that the standard deviation on regions of vortex shedding and a separated shear layer is larger than that on other portions. This observation coincides with the fact that these regions have higher fluctuation than the other regions without vortex shedding.

2. *Wake of a NACA0012 airfoil with a Gurney flap*

Next we consider the complex two-dimensional wake behind a NACA0012 airfoil with a Gurney flap. The flow field is also periodic in time analogous to the cylinder problem. However, the wake here is comprised of multiple dominant frequencies. The data set is generated using two-dimensional DNS at $\mathrm{Re}_c = 1000$, where c is the chord length [102]. It is known that various types of wakes can emerge depending on the angle of attack α and the Gurney-flap height h [102]. In the present paper, the case of $h/c = 0.1$ with $\alpha = 20^\circ$ which exhibits $2P$ wake is chosen for demonstration, as shown in Fig. 2(b). For the numerical setup, five nested levels of multidomains are considered as well as the cylinder problem. The finest domain range is $(x/c, y/c) = [-1, 1] \times [-1, 1]$ and the largest domain is $(x/c, y/c) = [-16, 16] \times [-16, 16]$. The time step is $\Delta t = 10^{-3}$. The size of the domain utilized and the number of grid points for the data set are $[-0.5, 7] \times [-2.5, 2.5]$ and $(N_x^*, N_y^*) = (352, 240)$, respectively. We use the vorticity field ω as the input and output attributes.

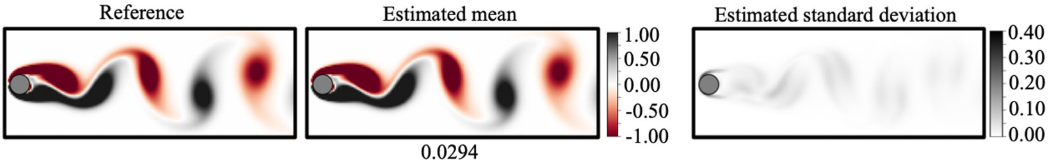


FIG. 12. Wake reconstruction of the cylinder flow using the PNN. The value below the estimated mean field μ indicates the L_2 error norm $\epsilon = \|\omega_{\mathrm{DNS}} - \omega_{\mathrm{ML}}\|_2/\|\omega_{\mathrm{DNS}}\|_2$.

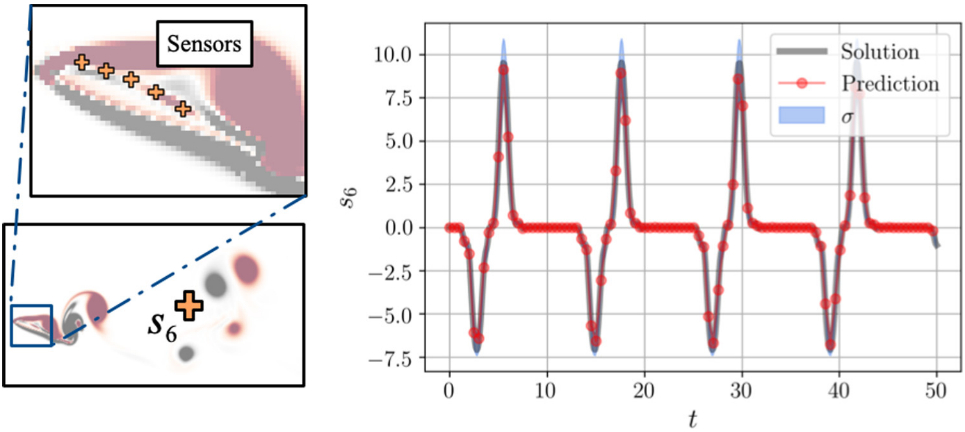


FIG. 13. Sensor estimation of a NACA0012 wake with a Gurney flap using the PNN. The left-hand side shows both input and output sensor locations. The L_2 error norm $\epsilon = \|s_{6,\mathrm{DNS}} - s_{6,\mathrm{ML}}\|_2/\|s_{6,\mathrm{DNS}}\|_2$ is 0.0429.

Five sensor measurements located on the surface of an airfoil are chosen as input data for both the sensor estimation and the whole field reconstruction. The number of snapshots n_{snapshot} used for the baseline model are the same as that in the cylinder problem such that there are 100 snapshots over approximately four periods for both training and test data. Note that we also investigate the dependence on the number of snapshots using the example of a NACA0012 wake.

The PNN is applied to the local sensor estimation of a NACA0012 wake, as shown in Fig. 13. As can be seen, the estimated plots are in great agreement with the reference data. Regions of noticeable standard deviation can be seen near the peaks, although these were not observed with sensor estimation for the cylinder problem. This indicates that the true peaks in the curve may show some variation from the predicted values given the training of the network. This is due to the difference in the complexity of the flows, i.e., the frequency contents contained on the wakes, as mentioned above.

The estimated whole fields from the sensor measurements located on the surface of the airfoil are shown in Fig. 14. With $n_{\mathrm{snapshot}} = 100$, the location and size of vortex structures are captured reasonably well by using the present model. However, the L_2 error norm $\epsilon = \|\omega_{\mathrm{DNS}} - \omega_{\mathrm{ML}}\|_2/\|\omega_{\mathrm{DNS}}\|_2$ is 0.514, which is substantially larger than that in the case of the cylinder wake, despite the reasonable estimation. This is because the L_2 error norm is known as a strict measurement of difference and

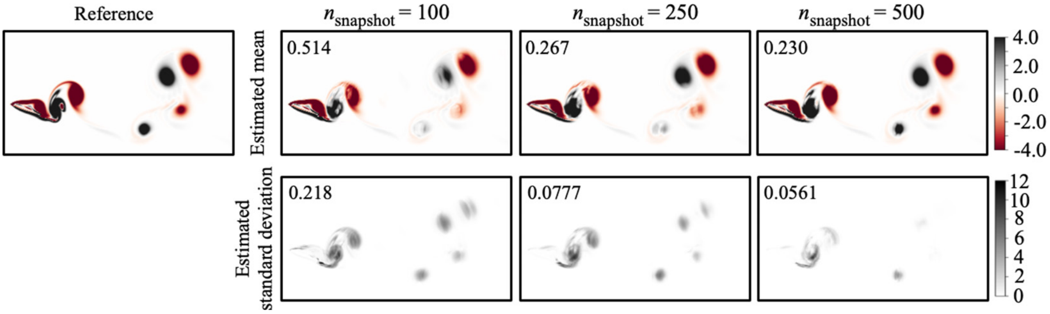


FIG. 14. Wake reconstruction of a NACA0012 airfoil with a Gurney flap depending on the number of snapshots for training. The values inside the estimated mean field indicate the L_2 error norm $\epsilon = \|\omega_{\mathrm{DNS}} - \omega_{\mathrm{ML}}\|_2/\|\omega_{\mathrm{DNS}}\|_2$. The values inside the estimated standard deviation express the ensemble-averaged standard deviation over the field.

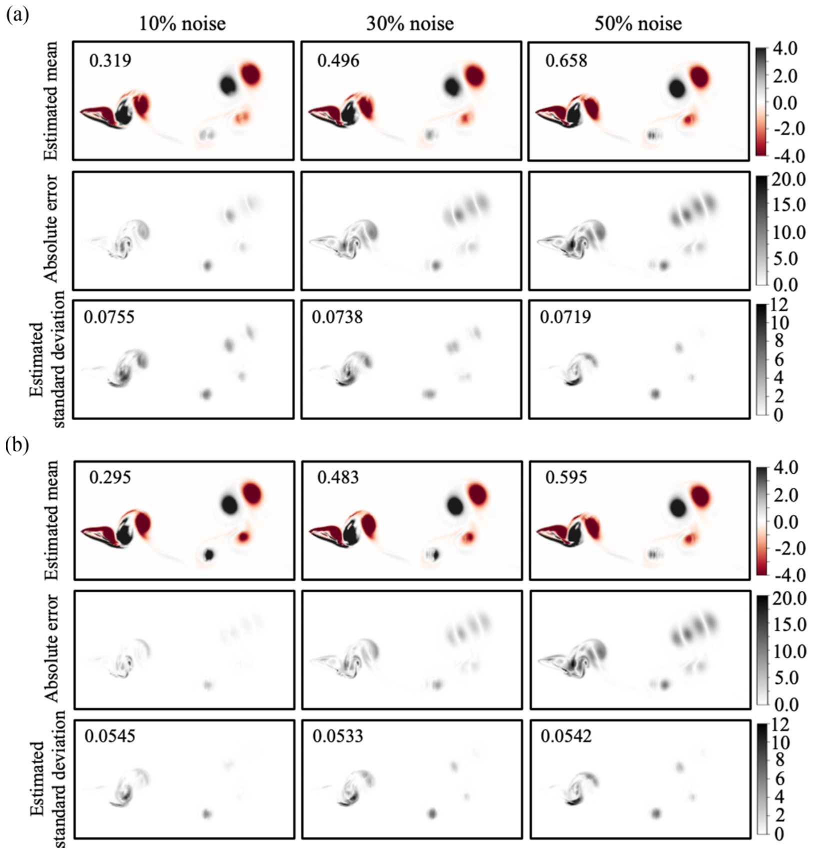


FIG. 15. Dependence of estimation accuracy for the whole wake field of a NACA0012 airfoil with a Gurney flap on the magnitude of noisy input using the machine-learned model with (a) $n_{\text{snapshot}} = 250$ and (b) $n_{\text{snapshot}} = 500$. The values inside the estimated mean field indicate the L_2 error norm. The values inside the estimated standard deviation express the ensemble-averaged standard deviation over the field.

does not account for translational or rotational similarities [4,45]. With regard to the standard deviation distribution, the low confidence interval region is concentrated in the regions of vortical structures, similar to the cylinder problem. The dependence on the number of snapshots used for training is also examined considering $n_{\text{snapshot}} = \{100 \text{ (baseline)}, 250, 500\}$, as shown in Fig. 14. We would like to draw attention to the fact that regions with high uncertainty shrink with the increasing number of snapshots. This highlights the true benefit of the PNN, namely, the better estimate due to the improvement in training data (i.e., greater number of snapshots) is properly quantified as increased confidence. These results with the NACA0012 airfoil suggest that the present probabilistic framework can perform well in reconstructing complex flows while providing feedback about the viability of the training data for learning.

Next let us examine the robustness for noisy input in Fig. 15, which is analogous to the assessment with the shallow-water equation presented in Fig. 10. The noisy input is designed with a Gaussian distribution based on the ratio, i.e., 10%, 30%, and 50%, for the maximum absolute value of the field. Adding noise to the input leads to increasing the error, as can be seen by both L_2 error norms and absolute error maps in Fig. 10. Note that the standard deviations have no significant difference among the noise level considered with the same number of snapshots due to the fact that the estimated confidence interval is not a barometer for error. This trend coincides with the example of the shallow-water equation shown in Fig. 10. By comparing Figs. 15(a) and 15(b), we

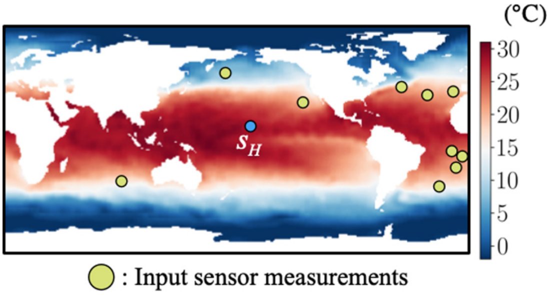


FIG. 16. Instantaneous temperature field on the sea surface. Green circles are sensors used as input data.

can find that the error decreases by increasing the number of training snapshots, as we expected from the observation with the example of the shallow-water equation in Fig. 10. We reiterate, therefore, that the proposed formulation does not absolve the user of the best practices of ML methods (for instance, introduced in [4]). Adequate validation with held-out testing data sets cannot be precluded for the purpose of error diagnostics. However, the effect of improved learning can be directly correlated with the physics of the predictive task, in this case flow-field recovery.

3. NOAA sea surface temperature

We here consider the NOAA sea surface temperature data set so as to assess the applicability of the PNN to practical applications of spatial field recovery. The setup for training and test data is basically the same as that for the parametric surrogate discussed in Sec. III A 2. Analogous to them, the input sensors for the baseline model are placed in the region of 50° S to 50° N as shown in Fig. 16. We present the results of local sensor estimation from ten other sensors in Fig. 17. The reader may observe that the machine learning model can capture the seasonal periodicity quite well. What is notable here is that the interval of standard deviation is wider than that with cases of numerical simulation data, i.e., the cylinder wake and NACA0012 airfoil with a Gurney flap. One possible reason is the complexity of the environmental processes which have been added on seasonal temperature variance, e.g., global warming, since the test data is set as extrapolation in time against the range used for training.

Let us also consider the estimation of a global temperature field, as shown in Fig. 18. The reconstructed field shows a reasonable match with the reference field and this can also be found by an L_2 norm assessment. The benefit of the proposed framework can be seen when the variance predicted by the framework is correlated to the global grid. We note that the presence of uncertainty is most likely influenced by the choice of input sensor locations. We can also see a relatively higher

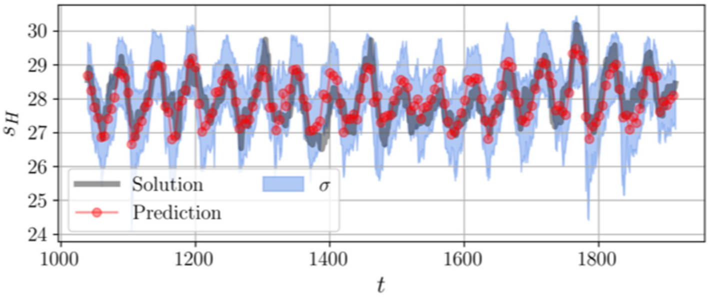


FIG. 17. Sensor estimation of sea surface temperature using the PNN. The L_2 error norm $\epsilon = \|s_{H,\mathrm{ref}} - s_{H,\mathrm{ML}}\|_2 / \|s_{H,\mathrm{ref}}\|_2$ is 0.0163.

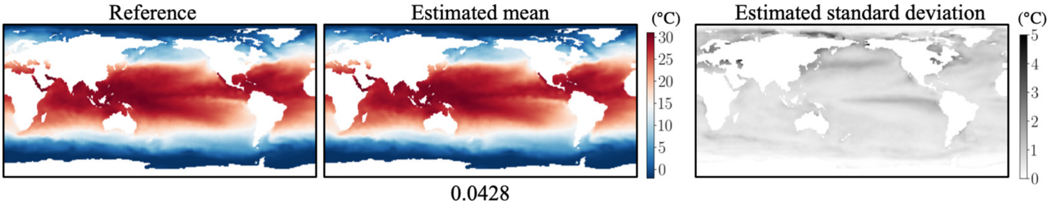


FIG. 18. Whole field reconstruction of sea surface temperature. Note that the estimated standard deviation is the time-ensemble average value. The value below the estimated mean field indicates the L_2 error norm $\epsilon = \|T_{\rm ref} - T_{\rm ML}\|_2/\|T_{\rm ref}\|_2$.

standard deviation region across the Pacific Ocean near South America. This is because of El Niño (for which the sea temperature is higher than usual) and La Niña (lower than usual), which emerge every few years in this area [41]. Hence, the temperature fluctuation in this area is larger than that in the other areas.

The results of the present study educate us about the possible applications of probabilistic neural networks so that we may be able to assess not only the reliability of estimated results but also the characteristics of a given training data set by focusing on estimated uncertainties. Furthermore, we can also utilize the estimated field to place additional sensors efficiently since we can observe high standard deviation areas through our probabilistic predictions. Here we also assess this viewpoint by placing five additional sensors, as shown in Fig. 19. We consider two cases by adding five sensors randomly [Fig. 19(a)] and based on the estimated standard deviation in Fig. 18 [Fig. 19(b)]. As presented, the latter outperforms the case of randomly chosen sensors in terms of the L_2 error norm. In addition, the sensor selection based on the estimated confidence interval leads to lower standard deviation, e.g., on the Pacific Ocean near South America where El Niño and La Niña can be observed as mentioned above. Through the results in our test cases, we see great potential of the present probabilistic framework for various studies in fluid dynamics.

IV. CONCLUSION

In this investigation, we have introduced probabilistic neural networks for addressing questions related to the uncertainty quantification of data-driven surrogate model applications to fluid flows. Probabilistic neural networks bestow the ability to parametrize the output as a sample from a Gaussian or a mixture of Gaussian distributions. Consequently, every prediction is accompanied

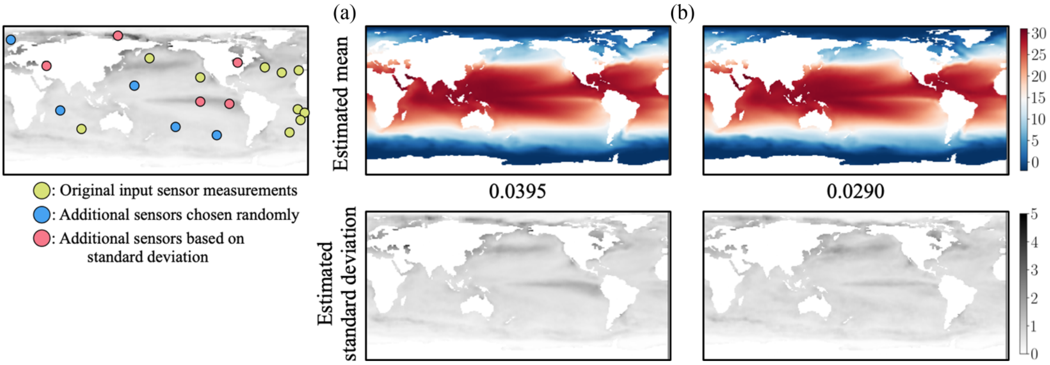


FIG. 19. Influence of estimation accuracy for the whole field of sea surface temperature on additional sensors. Five sensors are added (a) randomly and (b) based on the estimated standard deviation in Fig. 18. The values underneath the estimated mean fields indicate the L_2 error norm.

by a confidence interval that may guide the user of the framework about potential errors due to insufficient training data. This represents an improvement on the majority of data-driven studies in fluid mechanics which formulate a deterministic prediction requirement for their surrogates. To demonstrate the viability of the proposed framework, we first deployed it for a parametric model-order reduction task where a surrogate model was constructed for the shallow-water equations. Our surrogate modeling task was particularly complicated by the presence of measurement noise as well as incomplete observations of all relevant conserved variables. The proposed framework was seen to predict, nonintrusively, the evolution of a Gaussian excitation of the field with uncertainty estimates. Following this task, we deployed the framework to a data-recovery problem where measurements at random sensor locations present in a flow field were used to reconstruct the entire field in the presence of input noise. We also obtained favorable comparisons for our probabilistic flow-field reconstruction against gappy POD, a linear reconstruction technique.

Following assessments for the shallow-water equations, the probabilistic neural network was then applied to the spatial reconstruction of flow fields from data sets with connections to engineering and geophysical applications such as for two-dimensional cylinder flow data, flow around the NACA0012 airfoil with a Gurney flap, and the NOAA optimal interpolation sea-surface temperature data set. This reconstruction procedure relied on a direct prediction of flow fields without an intermediate POD-based compression method. For the problems of the two-dimensional cylinder wake and the NACA0012 airfoil with a Gurney flap, we found that the present probabilistic model estimated sensor values and reconstructed entire flow fields well while providing uncertainty estimates for the machine-learned estimation. For examples of cases without a modeled governing equation, i.e., the NOAA sea surface temperature data set, our results indicated that the reliability of estimated results as well as the characteristics of a given training data set could be analyzed with the present probabilistic model which shows estimated uncertainties. We note that the direct reconstruction of spatial fields may lead to the violation of physical laws unless the neural network architectures are actively penalized for these violations via constraints. One approach to deploy this is through the use of partial differential equation (PDE) constraints as regularization terms in the optimization statement such as the well-known physics-informed neural networks [103]. However, extensions to data sets without knowledge of the underlying PDEs such as NOAA SST are not trivial and would require other strategies such as data augmentation and constrained network design to satisfy symmetries such as the use of projection-basis networks [10]. This is an active area of investigation.

To address the issue of *what* uncertainty is being quantified, we assessed our frameworks for different noise perturbations and training data sizes. We were able to ascertain that the addition of noise to inputs (after training) caused errors in the mean but did not affect confidence intervals. In contrast, improved training data (by utilizing more snapshots) directly led to reducing standard deviations because confidence in outputs was improved. With noisy targets, the framework was adept at characterizing useful confidence intervals. In addition, by plotting standard deviations on the computational grid for the latter case, the regions contributing to the learning difficulty could easily be ascertained for further sampling. Therefore, our recommendation for the use of such frameworks is for applications where both targets and input data are noisy *before* training, a common phenomenon. In the absence of such training data, we suggest the addition of artificial noise (within a certain range) to improve the robustness of the predictions in terms of mean errors. Armed with this knowledge, these probabilistic predictors may be utilized for greater interpretability in data-driven forecasting, reconstruction, or model-order-reduction tasks with the potential for establishing a feedback loop to improve training data on the fly. We hint towards this using intelligent sampling for the NOAA sea surface temperature reconstruction task. Notably, the utility of the approximation of the target distribution also has direct applications for the assessment of model quality in the absence of training data.

We remark that the confidence intervals are merely representations of the standard deviation around an expected value, i.e., both the standard deviation and the expectation are learned in a semisupervised manner to maximize the likelihood of the targets given the inputs. For instance, the grayed out regions of each line plot indicate, with confidence equivalent to one standard deviation, the probability of the true prediction deviating from the mean. Therefore, the viability of using a prediction (in the form of the mean and the intervals) is intimately associated with the target application. We do not extend the interpretation of the confidence intervals beyond the above characterization, i.e., we do not claim that our probabilistic estimates are the "true" posterior distributions for the targets. Such posterior approximations from an inference method (like variational inference techniques) can only be verified against a fully Bayesian approach, like a Markov chain Monte Carlo (MCMC) sampling. However, for several problems of interest in deep learning, including our applications, doing a full MCMC sampling over all the parameters of the network is prohibitively expensive due the large number of model parameters, i.e., the weights of the network. The departure from being a full posterior arises from the fact that (i) the model parameters (such as the weights of the network) are not sampled from a distribution, (ii) priors are implicit, i.e., defined by the training set rather than being explicitly defined, and (iii) the estimated distribution is constrained to be a parametrized conditional density distribution $p(\boldsymbol{y}|\boldsymbol{x})$ as shown in Eq. (1), instead of a generic posterior distribution. However, this does not mean that the estimated confidence intervals are uninformative. The error bars shown in our results represent the uncertainty quantification in our estimates that arise from the network, under a given training data and a Gaussian approximation. This is clearly seen in our results, where a less-informative training data priors, with either a reduced number of snapshots (in Fig. 14) in the training phase or a reduced number of sensors [in Fig. 7(a)], results in a broader conditional density estimation $p(\boldsymbol{y}|\boldsymbol{x})$, reflecting a lower confidence in the prediction when training sampling is poor.

The PNN is a pragmatic solution for the problem of estimating uncertainty in deep neural networks, although it assumes a Gaussian form for the predictive probability distribution. If the predictions are not expected to be Gaussian distributions (for example, in situations with multiple subpopulations or mixed data), then one may simply use the more generic mixture density networks with an appropriate number of mixture components. We have verified this for the case of shallow-water equations, where the addition of components does not make any difference (i.e., all but one mixing coefficient end up with near-zero weight after training) in the reconstruction. This validates the Gaussian assumption and corresponding use of PNNs.

We have considered the use of probabilistic neural networks with a fully connected structure. However, the fully connected model often suffers due to the curse of dimensionality since the number of weights is drastically increased with the connected nodes in the model. This suggests that a convolutional neural network formulation [104], which is able to deal with high-dimensional data through the concept of filter sharing, may be investigated for the next step. In addition, these probabilistic models, which can express confidence intervals for predictions, may also be enhanced if they can be applied efficiently to unstructured data that are seen in various applications for fluid dynamics. To that end we are studying the feasibility of graph neural networks [105] and generalized moving least-squares [106] frameworks.

ACKNOWLEDGMENTS

This material was based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Contract No. DE-AC02-06CH11357. This research was funded in part and used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract No. DE-AC02-06CH11357. R.M. acknowledges support from the ALCF Margaret Butler Fellowship. K. Fukami and K. Fukagata are grateful for support from Japan Society for the Promotion of Science (KAKENHI Grant No. 18H03758). K.T. acknowledges generous support from the U.S. Army Research Office (Grant No. W911NF-17-1-0118) and U.S. Air Force Office of Scientific Research (Grant No. FA9550-16-1-0650). The authors acknowledge Dr. M. Gopalakrishnan Meena (University of California, Los Angeles) for sharing his DNS data.

[1] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. **52**, 477 (2020).

[2] M. P. Brenner, J. D. Eldredge, and J. B. Freund, Perspective on machine learning for advancing fluid mechanics, Phys. Rev. Fluids **4**, 100501 (2019).

[3] J. N. Kutz, Deep learning in fluid dynamics, J. Fluid Mech. **814**, 1 (2017).

[4] K. Fukami, K. Fukagata, and K. Taira, Assessment of supervised machine learning for fluid flows, Theor. Comput. Fluid Dyn. **34**, 497 (2020).

[5] S. L. Brunton, M. S. Hemanti, and K. Taira, Special issue on machine learning and data-driven methods in fluid dynamics, Theor. Comput. Fluid Dyn. **34**, 333 (2020).

[6] R. Maulik, O. San, J. D. Jacob, and C. Crick, Sub-grid scale model classification and blending through deep learning, J. Fluid Mech. **870**, 784 (2019).

[7] M. Gamahara and Y. Hattori, Searching for turbulence models by artificial neural network, Phys. Rev. Fluids **2**, 054604 (2017).

[8] R. Maulik and O. San, A neural network approach for the blind deconvolution of turbulent flows, J. Fluid Mech. **831**, 151 (2017).

[9] A. Vollant, G. Balarac, and C. Corre, Subgrid-scale scalar flux modeling based on optimal estimation theory and machine-learning procedures, J. Turbul. **18**, 854 (2017).

[10] J. Ling and J. Templeton, Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty, Phys. Fluids **27**, 085103 (2015).

[11] E. J. Parish and K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, J. Comput. Phys. **305**, 758 (2016).

[12] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modeling using deep neural networks with embedded invariance, J. Fluid Mech. **807**, 155 (2016).

[13] J.-L. Wu, H. Xiao, and E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Phys. Rev. Fluids **3**, 074602 (2018).

[14] R. Matai and P. A. Durbin, Zonal eddy viscosity models based on machine learning, Flow Turbul. Combust. **103**, 93 (2019).

[15] L. Zhu, W. Zhang, J. Kou, and Y. Liu, Machine learning methods for turbulence modeling in subsonic flows around airfoils, Phys. Fluids **31**, 015105 (2019).

[16] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. **51**, 357 (2019).

[17] N. Geneva and N. Zabaras, Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks, J. Comput. Phys. **383**, 125 (2019).

[18] N. Geneva and N. Zabaras, Multi-fidelity generative deep learning turbulent flows, arXiv:2006.04731.

[19] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. Dawson, and C. Yeh, Modal analysis of fluid flows: Applications and outlook, AIAA J. **58**, 998 (2020).

[20] K. Taira, M. S. Hemati, and L. S. Ukeiley, Modal analysis of fluid flows: Introduction to the virtual collection, AIAA J. **58**, 991 (2020).

[21] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, Proc. R. Soc. A **474**, 20170844 (2018).

[22] S. E. Ahmed, O. San, A. Rasheed, and T. Iliescu, A long short-term memory embedding for hybrid uplifted reduced order models, Physica D: Non. Phen. **409**, 132471 (2020).

[23] R. Maulik, A. Mohan, B. Lusch, S. Madireddy, and P. Balaprakash, Time-series learning of latent-space dynamics for reduced-order model closure, Physica D **405**, 132368 (2020).

[24] A. T. Mohan, D. Daniel, M. Chertkov, and D. Livescu, Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence, arXiv:1903.00033.

[25] A. T. Mohan and D. V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, arXiv:1804.09269.

[26] Q. Wang, N. Ripamonti, and J. S. Hesthaven, Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, J. Comput. Phys. **410**, 109402 (2020).

[27] B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nat. Commun. **9**, 4950 (2018).

[28] F. J. Gonzalez and M. Balajewicz, Learning low-dimensional feature dynamics using deep convolutional recurrent autoencoders, arXiv:1808.01346.

[29] J. Xu and K. Duraisamy, Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics, Comput. Methods Appl. Mech. Eng. **372**, 113379 (2020).

[30] R. Maulik, B. Lusch, and P. Balaprakash, Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, arXiv:2002.00470.

[31] M. Cheng, F. Fang, C. C. Pain, and I. M. Navon, An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling, arXiv:2003.10547.

[32] K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata, Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes, Theor. Comput. Fluid Dyn. **34**, 367 (2020).

[33] C. W. Rowley and S. T. M. Dawson, Model reduction for flow analysis and control, Annu. Rev. Fluid Mech. **49**, 387 (2017).

[34] J. Borggaard, T. Iliescu, and Z. Wang, Artificial viscosity proper orthogonal decomposition, Math. Comput. Model. **53**, 269 (2011).

[35] X. Xie, M. Mohebujjaman, L. G. Rebholz, and T. Iliescu, Data-driven filtered reduced order modeling of fluid flows, SIAM J. Sci. Comput. **40**, B834 (2018).

[36] E. Rezaian and M. Wei, A hybrid stabilization approach for reduced-order models of compressible flows with shock-vortex interaction, Int. J. Numer. Methods Eng. **121**, 1629 (2020).

[37] O. San and J. Borggaard, Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows, Int. J. Numer. Methods Fluids **78**, 37 (2015).

[38] O. San and R. Maulik, Extreme learning machine for reduced order modeling of turbulent geophysical flows, Phys. Rev. E **97**, 042322 (2018).

[39] S. Sheriffdeen, J. C. Ragusa, J. E. Morel, M. L. Adams, and T. Bui-Thanh, Accelerating PDE-constrained inverse solutions with deep learning and reduced order models, arXiv:1912.08864.

[40] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, Philadelphia, 2016).

[41] J. N. Kutz, X. Fu, and S. L. Brunton, Multiresolution dynamic mode decomposition, SIAM J. Appl. Dyn. Syst. **15**, 713 (2016).

[42] O. San and R. Maulik, Neural network closures for nonlinear model order reduction, Adv. Comput. Math. **44**, 1717 (2018).

[43] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, Synthetic turbulent inflow generator using machine learning, Phys. Rev. Fluids **4**, 064603 (2019).

[44] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, Shallow neural networks for fluid flow reconstruction with limited sensors, Proc. R. Soc. A **476**, 20200097 (2020).

[45] K. Fukami, K. Fukagata, and K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. **870**, 106 (2019).

[46] K. Fukami, K. Fukagata, and K. Taira, Proceedings of the 11th International Symposium on Turbulence and Shear Flow Phenomena (TSFP11), Southampton, 2019, available at http://www.tsfp-conference.org/proceedings/2019/208.pdf (unpublished).

[47] R. Onishi, D. Sugiyama, and K. Matsuda, Super-resolution simulation for real-time prediction of urban micrometeorology, Sci. Online Lett. Atmos. **15**, 178 (2019).

[48] B. Liu, J. Tang, H. Huang, and X.-Y. Lu, Deep learning methods for super-resolution reconstruction of turbulent flows, Phys. Fluids **32**, 025105 (2020).

[49] K. Fukami, K. Fukagata, and K. Taira, Machine learning based spatio-temporal super resolution reconstruction of turbulent flows, arXiv:2004.11566.

[50] M. Milano and P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. **182**, 1 (2002).

[51] T. Murata, K. Fukami, and K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, J. Fluid Mech. **882**, A13 (2020).

[52] K. Fukami, T. Nakamura, and K. Fukagata, Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, Phys. Fluids **32**, 095110 (2020).

[53] T. Bui-Thanh, M. Damodaran, and K. Willcox, Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, AIAA J. **42**, 1505 (2004).

[54] T. Suzuki and Y. Hasegawa, Estimation of turbulent channel flow at $\mathrm{Re}_\tau = 100$ based on the wall measurement using a simple sequential approach, J. Fluid Mech. **830**, 760 (2006).

[55] M. Morimoto, K. Fukami, and K. Fukagata, Experimental velocity data estimation for imperfect particle images using machine learning, arXiv:2005.00756.

[56] S. Cai, S. Zhou, C. Xu, and Q. Gao, Dense motion estimation of particle images via a convolutional neural network, Exp. Fluids **60**, 60 (2019).

[57] Z. Deng, C. He, Y. Liu, and K. C. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, Phys. Fluids **31**, 125111 (2019).

[58] J. Kim and C. Lee, Prediction of turbulent heat transfer using convolutional neural networks, J. Fluid Mech. **882**, A18 (2020).

[59] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis* (Wiley, New York, 2011), Vol. 40.

[60] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer Science + Business Media, New York, 2012), Vol. 118.

[61] B. Mohebali, A. Tahmassebi, A. Meyer-Baese, and A. H. Gandomi, in *Handbook of Probabilistic Models*, edited by P. Samui, D. Tien Bui, S. Chakraborty, and R. C. Deo (Butterworth-Heinemann, Oxford, 2020), pp. 347–367.

[62] Y. Gal, Uncertainty in deep learning, Ph.D. thesis, University of Cambridge, 2016.

[63] E. Goan and C. Fookes, Bayesian neural networks: An introduction and survey, Case Stud. App. Bay. Data Sci. **2259**, 45 (2020).

[64] G. E. Hinton and D. van Camp, in *Proceedings of the Sixth Annual Conference on Computational Learning Theory, Santa Cruz, 1993*, edited by L. Pitt (Association for Computing Machinery, New York, 1993), pp. 5–13.

[65] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, in *Proceedings of the 32nd International Conference on Machine Learning*, edited by F. Bach and D. Blei (JMLR, Brookline, 2015), Vol. 37, pp. 1613–1622.

[66] A. Graves, in *Proceedings of the 24th International Conference on Advances in Neural Information Processing Systems*, edited by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Curran, Red Hook, 2011), pp. 2348–2356.

[67] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, arXiv:1312.6114.

[68] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation, arXiv:1506.02157.

[69] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014).

[70] C.-K. Lu, S. C.-H. Yang, X. Hao, and P. Shafto, Interpretable deep Gaussian processes with moments, arXiv:1905.10963.

[71] A. Damianou and N. Lawrence, *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, Scottsdale, 2013* (JMLR, Brookline, 2013), Vol. 31, pp. 207–215.

[72] R. Alexander and D. Giannakis, Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques, Physica D **409**, 132520 (2020).

[73] S. Santini and A. D. Bimbo, Recurrent neural networks can be trained to be maximum a posteriori probability classifiers, Neural Netw. **8**, 25 (1995).

[74] C. M. Bishop, Mixture density networks, Aston University, Neural Computing Research Group Report No. NCRG/94/004, 1994 (unpublished).

[75] D. Ormoneit and V. Tresp, in *Proceedings of the Ninth International Conference on Advances in Neural Information Processing Systems, Denver, 1996*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche (MIT Press, Cambridge, 1997), pp. 542–548.

[76] F. Yang, J. Soriano, T. Kubo, and K. Ikeda, in *Proceedings of the 24th International Conference on Neural Information Processing, Guangzhou, 2017*, edited by D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Lecture Notes in Computer Science Vol. 10637 (Springer, Cham, 2017), pp. 878–885.

[77] O. Makansi, E. Ilg, O. Cicek, and T. Brox, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, 2019), pp. 7144–7153.

[78] C. P. Martin and J. Torresen, RoboJam: A musical mixture density network for collaborative touchscreen interaction, Lect. Notes Comput. Sci. **10783**, 161 (2018).

[79] A. D'Isanto and K. L. Polsterer, Photometric redshift estimation via deep learning. Generalized and pre-classification-less, image based, fully probabilistic redshifts, Astron. Astrophys. **609**, A111 (2018).

[80] M. Lucey, Y.-S. Ting, N. S. Ramachandra, and K. Hawkins, From the inner to outer Milky Way: A photometric sample of 2.6 million red clump stars, Mon. Not. R. Astron. Soc. **495**, 3087 (2020).

[81] G. J. McLachlan and K. E. Basford, Mixture models: Inference and applications to clustering J. R. Stat. Soc. C **38**, 384 (1989).

[82] D. J. C. Mackay, Probable networks and plausible predictions — A review of practical Bayesian methods for supervised neural networks, Netw.: Comput. Neural Syst. **6**, 469 (1995).

[83] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, How good is the Bayes posterior in deep neural networks really? arXiv:2002.02405.

[84] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination, San Francisco, 2015), Vol. 2018.

[85] D. P. Kingma and J. Ba, Proceedings of the Third International Conference for Learning Representations, San Diego, 2015 (unpublished).

[86] J. Bergstra, D. Yamins, and D. Cox, in *Proceedings of the 30th International Conference on Machine Learning, Atlanta, 2013*, edited by S. Dasgupta and D. McAllester (JMLR, Brookline, 2013), Vol. 28.

[87] E. Brochu, V. Cora, and N. de Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, University of British Columbia Report No. TR-2009-023, 2009 (unpublished).

[88] R. Maulik, K. Fukami, N. Ramachandra, K. Fukagata, and K. Taira, https://github.com/Romit-Maulik/Probabilistic_ML_Fluids.

[89] Available at https://www.esrl.noaa.gov/psd/.

[90] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, Modal analysis of fluid flows: An overview, AIAA J. **55**, 4013 (2017).

[91] D. D. Kosambi, Statistics in function space, J. Indian Math. Soc. **7**, 76 (1943).

[92] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. **25**, 539 (1993).

[93] X.-D. Liu, S. Osher, and T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. **115**, 200 (1994).

[94] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer Series in Computational Mathematics Vol. 8 (Springer, Berlin, 1993).

[95] R. Maulik and O. San, Resolution and energy dissipation characteristics of implicit LES and explicit filtering models for compressible turbulence, Fluids **2**, 14 (2017).

[96] A. Mendible, S. L. Brunton, A. Y. Aravkin, W. Lowrie, and J. N. Kutz, Dimensionality reduction and reduced order modeling for traveling wave physics, Theor. Comput. Fluid Dyn. **34**, 385 (2020).

[97] J. L. Callaham, K. Maeda, and S. L. Brunton, Robust flow reconstruction from limited measurements via sparse representation, Phys. Rev. Fluids **4**, 103907 (2019).

[98] R. Maulik, R. Egele, B. Lusch, and P. Balaprakash, Recurrent neural network architecture search for geophysical emulation, arXiv:2004.10928.

[99] R. Everson and L. Sirovich, Karhunen-Loeve procedure for gappy data, J. Opt. Soc. Am. A **12**, 1657 (1995).

[100] K. Taira and T. Colonius, The immersed boundary method: A projection approach, J. Comput. Phys. **225**, 2118 (2007).

[101] T. Colonius and K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, Comput. Methods Appl. Mech. Eng. **197**, 2131 (2008).

[102] M. G. Meena, K. Taira, and K. Asai, Airfoil wake modification with gurney flap at low Reynolds number, AIAA J. **56**, 1348 (2018).

[103] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations, arXiv:1711.10566.

[104] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[105] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020), doi:10.1109/TNNLS.2020.2978386.

[106] N. Trask, R. G. Patel, B. J. Gross, and P. J. Atzberger, GMLS–Nets: A framework for learning from unstructured data, arXiv:1909.05371.