We consider the use of probabilistic neural networks for fluid flow model-order reduction 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 (1) the shallow water equations, (2) a two-dimensional cylinder flow, (3) the wake of NACA0012 airfoil with a Gurney flap, and (4) 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 model but also systematically quantifies the uncertainty therein to assist with model interpretability.

1 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 potentials for extracting complex nonlinear relations embedded in fluid flow data [1][2][3][4]. For example, the successes of ML have been observed in several investigations into the development of closure models for large eddy simulation (LES) [5][6][7][8] and Reynolds-Averaged Navier-Stokes (RANS) simulation [9][10][11][12][13][14] where DNS or experimental data have been used to improve conventional algebraic or differential equation based closures [15]. Another promising avenue for ML is for addressing the challenges of conventional reduced order models (ROMs) [16][17]. Recent literature has demonstrated the capabilities of time-series methods from ML [18][19][20][21][22][23] for reduced-space temporal dynamics prediction as well as nonlinear subspace identification using image processing techniques [24][25][26][27][28][29]. These methods have demonstrated promising results over conventional equation-based methods such as the proper-orthogonal decomposition (POD) based Galerkin-projection technique [30] which suffers from an inability to handle advection-dominated systems.
addition to turbulence closures and ROMs, we have also seen successful applications for ML in fluid data estimation and reconstruction problems \[^{[31, 32, 33, 34, 35, 36, 37]}\]. These efforts have revealed that ML, due to its inherent nonlinearity \[^{[33, 39]}\], is well-suited to spatial flow data reconstruction and outperforms linear-reconstruction methods such as Gappy POD \[^{[40]}\] and linear stochastic estimation \[^{[41]}\]. More recently, these investigated methods have also been applied to PIV data \[^{[42, 43, 44]}\] and understanding an interpretable relationship in between the predicted results and input data by focusing on vortical motions \[^{[45]}\].

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. What is common, is that a maximum likelihood estimation of the loss function (usually a mean-squared-error estimate) via gradient descent methods is usually executed for training. Subsequently, any trained neural network predicts target values in a deterministic fashion. One approach to obtain probabilistic outputs is to use Bayesian inference \[^{[46]}\]. 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 lends to an output that can be characterized with a probability density function. However, a Bayesian inference of the model parameters of the neural network is prohibitively expensive in deeper architectures. Therefore several approximations using variational inference \[^{[47, 48]}\], Monte-Carlo dropouts \[^{[49]}\], Gaussian process approximations \[^{[50, 51]}\] and maximum a-posteriori estimation \[^{[52]}\] 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 introduce the use of a class of probabilistic neural networks that assume our target data (in this case the observable) is generated from a distribution \[^{[53]}\]. We then consider its applicability for several canonical problems of fluid dynamics. While the nature of the distribution is user-defined, 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).

Our contributions for this investigation are summarized as:

1. We propose the use of a probabilistic neural network architecture for efficiently embedding uncertainty quantification into data-driven tasks relevant to fluid flows.
2. We execute several tests to assess the strengths of the probabilistic framework for model order reduction, forecasting and spatial field recovery across different applications.
3. 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 Section 2. In Section 3, we apply the framework to representative problem settings for ROM and spatial fluid data recovery. We offer concluding remarks and future perspectives in Section 4.

## 2 Probabilistic neural networks

Predictions provided by neural networks for inputs \(x\) (such as the sensor measurements in figure \[^{[1]}\]) generally are determined from the minimization of a loss function as a function of weights \(\mathcal{E}(w)\) (e.g., the mean squared error). Such calculations between the truths \(y_t\) (such as the whole field in figure \[^{[1]}\]) and the predicted values from the network \(y_p(x)\) lack a probability distribution of \(p(y_t|w)\). Hence, the error bars on the estimates are typically absent in dense neural network outputs. To quantify uncertainty of the neural network estimates, the multi-dimensional surface of the loss function \(\mathcal{E}(w)\) has to be explored in addition to finding the global minimum.

To obtain the posterior of estimates \(p(y_t|x)\), 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 \(\mathcal{F}: x \rightarrow y(x)\), one could define the mapping as \(\mathcal{F}: x \rightarrow (\mu(x), \sigma(x))\) where the mean \(\mu\) and standard deviation \(\sigma\) parametrize a Gaussian probability distribution function \(\mathcal{N}(\mu(x), \sigma(x))\). This approach ensures that the outputs \(p(y_t|x) = \mathcal{N}(\mu(x), \sigma(x))\) are accompanied by uncertainty estimates assuming the errors are Gaussian.
Figure 1: A representative schematic of the probabilistic neural network. We consider the flow reconstruction from sensor measurements of a two-dimensional cylinder wake (section 3.2.1) as an example problem.

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 $\mathcal{F}: x \rightarrow (\pi_1, \mu_1, \sigma_1, \pi_2, \mu_2, \sigma_2, \ldots, \pi_N, \mu_N, \sigma_N)$, where $\pi_i$ is the mixing probability for each Gaussian component satisfying the condition $\sum_{i=1}^{N} \pi_i = 1$. The distribution function in this model is a linear combination of several Gaussian components given by,

$$p(y|x) = \sum_{i=1}^{N} \pi_i(x) N(\mu_i(x), \sigma_i^2(x)). \quad (1)$$

This belongs to a class of mixture density networks (MDNs) introduced in [53] by combining a conventional neural network with a mixture density model to predict the conditional probabilities.

The value of $m$ is generally pre-specified based on the expectation of 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 learning rate, decay rate and other hyper-parameters of the training scheme. It has to be noted that the training of MDN itself is supervised. However, mixture probabilities that are obtained from a fully trained network corresponds to different clusters in the data that is learned in an unsupervised fashion.

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

```
1: $w \leftarrow$ Initialize network parameters
2: while $\mathcal{E}$ is not converged do
3: Update probabilistic neural network
4: $x \leftarrow$ Random mini-batch from data set
5: for $x_k$ in $x$ do
6: Compute Gaussian mixture parameters: $(\pi_k, \mu_k, \sigma_k) \leftarrow \mathcal{F}(x_k; w)$
7: Compute predictive distribution: $p(y_{k,p}|x_k) \leftarrow (\pi_k, \mu_k, \sigma_k)$
8: end for
9: Compute negative log-likelihood: $\mathcal{E} \leftarrow -\log L$
10: Update network parameters: $w \leftarrow \text{Adam}(\nabla_w \mathcal{E}, w)$
11: end while
```

Since our probabilistic neural network framework provides a distribution of the estimation $p(y_{p}|x)$ instead of a point prediction of $y_p$, the loss function has to chosen accordingly to utilize the full distribution of the prediction. This results in a crucial improvement in probabilistic neural networks over 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 $y_t$ and the corresponding prediction $y_{p}$. Hence in practice, our model $\mathcal{F}(x; w)$ is trained to obtain the optimized weights $w$ by minimizing the error function $\mathcal{E}$ given in terms of average log-likelihood $L$ such that

$$w = \text{argmin}_w [\mathcal{E}], \quad \text{where} \quad \mathcal{E} \equiv -\log L = -\sum_{k=1}^{K} p(y_{k,p}|x_k) \log p(y_{k,t}), \quad (2)$$
Figure 2: The different canonical test-cases considered in the present work.

with \( k \) indicating each data point in the training data and \( K \) denotes 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 equation (1). It may also be noted that the likelihood maximizing model is equivalent to minimizing the cross entropy \( H(p(y_{p}|x), p(y_t)) \). Thus such a framework is 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 probabilistic neural network is shown for one of the applications (section 3.1) in figure 1. In this example, the inputs \( x \in \mathbb{R}^5 \) to the network model are the sensor measurements. These are mapped to the targets \( y_p \) whose truth values are a whole wake field of two-dimensional cylinder \( y_t \) of dimension 13440. The network for the example of figure 1 is a fully connected dense network with 8 layers and number of neurons per layer having \( 5 \rightarrow 64 \rightarrow 128 \rightarrow 512 \rightarrow 1024 \rightarrow 13440 \times 3 \). The last layer \([13440 \times 3]\) corresponds to \((\pi_1, \mu_1, \sigma_1)\) for each grid point that parametrizes the predictive conditional distribution \( p(y_p|x) \) for the whole wake field, and the error function is calculated using equation (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 \((\mu_1, \sigma_1)\), which implies \( \pi_1 = 1 \) corresponding to each target. We utilize the same number of Gaussian centers (i.e., solely one) for all our assessments hereafter. As an example, if 3 Gaussian centers were utilized, the final layer of the above map would have dimensions of \([13440 \times 3 \times 3]\) corresponding to \((\pi_i, \mu_i, \sigma_i)\) for \( i = 1, 2, 3 \) and \( \sum_{i=1}^{3} \pi_i = 1 \).

For updating the weights \( w \), we use the Adam optimizer [54]. The training procedure of the present probabilistic neural network is summarized in algorithm 1. We note that we do not attempt to optimize parameters of mixture density network for each problem setting in the present study. Our primary objective here is to demonstrate the applicability of the probabilistic model to quantify uncertainty of machine-learned estimations for canonical fluid flow problems. One may also consider the use of theoretical optimization methods such as hyperopt [55] and Bayesian optimization [20, 56] to further enhance the accuracy of the estimation can be improved. A sample code for the present model is available online [57]. In the following, we refer to the probabilistic neural network as the 'PNN.'

3 Results

In this section, we demonstrate the capabilities of the PNN introduced in the previous section. We first apply the proposed formulation to the ROM of the shallow water equations. These experiments are:

- The prediction of POD coefficient evolution given the initial conditions (sec. 3.1.1),
- The instantaneous estimation of POD coefficients from local sensor measurements (sec. 3.1.2).

Next, we consider the applications to spatial fluid data recovery given by:

- The estimation of sensor measurements from other sensor placements,
- The estimation of whole flow fields from local sensor measurements.

for three example problems:

- A two-dimensional cylinder wake at \( \text{Re}_D = 100 \) (sec. 3.2.1),
- A two-dimensional wake of NACA0012 airfoil with a Gurney flap at \( \text{Re}_c = 1000 \) (sec. 3.2.2),
- The NOAA optimum interpolation sea surface temperature data set (sec. 3.2.3).

The broad spread of potential applications can be observed through flow fields from our chosen data sets as seen in figure 2.
Figure 3: (a) A schematic for the generation of training and testing data. Given inputs of \( r \) (i.e., the location of an initial Gaussian perturbation to \( \eta \)), the machine learning ROM is tasked with predicting the evolution of 40 POD coefficients over 10 snapshots in time. The prediction is one-shot. (b) Convergence of reduced representations to true field with increasing \( M \). The values underneath each field indicate \( L_2 \) error norm \( \epsilon = ||\eta_{\text{True}} - \eta_{\text{POD}}||_2 / ||\eta_{\text{True}}||_2 \).

3.1 The 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

\[
\begin{align*}
\frac{\partial (\rho \eta)}{\partial t} + \frac{\partial (\rho u \eta)}{\partial x} + \frac{\partial (\rho v \eta)}{\partial y} &= 0, \\
\frac{\partial (\rho u \eta)}{\partial t} + \frac{\partial}{\partial x} \left( \rho u^2 + \frac{1}{2} \rho g \eta^2 \right) + \frac{\partial (\rho uv \eta)}{\partial y} &= 0, \\
\frac{\partial (\rho v \eta)}{\partial t} + \frac{\partial (\rho uv \eta)}{\partial x} + \frac{\partial}{\partial y} \left( \rho v^2 + \frac{1}{2} \rho g \eta^2 \right) &= 0.
\end{align*}
\]

(3)

In the above set of equations, \( \eta \) 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 \( \rho \) is the fluid density fixed to be unity. The first equation represents conservation of mass and the remaining two denote the conservation of momentum. For simplicity, we denote \( q = [\rho \eta, \rho u \eta, \rho v \eta]^T \) and we use \( \eta \) 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

\[
\begin{align*}
\rho \eta(x, y, t = 0) &= \exp \left[ - \frac{(x - \bar{x})^2}{2(5e + 4)^2} \right] \left[ - \frac{(y - \bar{y})^2}{2(5e + 4)^2} \right], \\
\rho u \eta(x, y, t = 0) &= 0, \quad \rho v \eta(x, y, t = 0) = 0,
\end{align*}
\]

(4)

while our two-dimensional domain is a square with periodic boundary conditions. These initial conditions \( r = [\bar{x}, \bar{y}] \) \((-0.5 \leq \{\bar{x}, \bar{y}\} \leq 0.5)\) correspond to the parameter controlling the spatio-temporal evolution of the dynamical system. Our data generation process utilizes full-order solves of the above system of equations until \( t = 0.1 \) with a time step of 0.001. Our full-order model uses a 4th-order accurate Runge-Kutta integration scheme and a fifth-order accurate weighted essentially non-oscillatory scheme (WENO) [58] for computing state reconstructions at cell faces. The Rusanov Reimann solver is utilized for flux reconstruction after cell-face quantities are calculated. The reader is directed to [59] for greater discussion of the temporal integration scheme and [60] for details on WENO and the Riemann solver implementation in two-dimensional problems.

In the following, we review the POD technique for the construction of a reduced basis [16, 61, 62, 63] for our observed variable. The POD procedure is tasked with identifying a space

\[
\mathbf{X}^f = \text{span} \{ \varphi^1, \ldots, \varphi^f \},
\]

(5)
which approximates snapshots optimally with respect to the $L_2$ norm. The process of $\vartheta$ generation commences with the collection of snapshots in the snapshot matrix

$$
\mathbf{S} = [ \mathbf{q}_h^1 | \mathbf{q}_h^2 | \cdots | \mathbf{q}_h^{N_s} ] \in \mathbb{R}^{N_h \times N_s},
$$

(6)

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

$$
\mathbf{q}_h^i = \mathbf{q}_h^i - \bar{\mathbf{q}}_h, \quad \bar{\mathbf{q}}_h = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{q}_h^i.
$$

(7)

with $\bar{\mathbf{q}}_h : \mathcal{P} \to \mathbb{R}^{N_h}$ being 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

$$
\mathbf{C} \mathbf{w}_r = \mathbf{W}_r \lambda_r,
$$

(8)

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

$$
\vartheta = \mathbf{W} \varphi \in \mathbb{R}^{N_h \times N_r}.
$$

(9)

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

$$
\mathbf{X} = \text{span} \{ \varphi^1, \ldots, \varphi^{N_r} \}.
$$

(10)

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

$$
\mathbf{A} = \varphi^T \mathbf{S} \in \mathbb{R}^{N_r \times N_s}.
$$

(11)

The POD approximation of our solution is then obtained via

$$
\hat{\mathbf{S}} = [ \hat{\mathbf{q}}_h^1 | \hat{\mathbf{q}}_h^2 | \cdots | \hat{\mathbf{q}}_h^{N_s} ] \approx \varphi \mathbf{A} \in \mathbb{R}^{N_h \times N_r},
$$

(12)

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

$$
\frac{\sum_{i=1}^{N_s} ||\mathbf{q}_h^i - \hat{\mathbf{q}}_h^i||_{\mathbb{R}^{N_h}}^2}{\sum_{i=1}^{N_s} ||\mathbf{q}_h^i||_{\mathbb{R}^{N_h}}^2} = \frac{\sum_{i=N_s+1}^{N_r} \lambda_r^2}{\sum_{i=1}^{N_s} \lambda_r^2},
$$

(13)

which exhibits that with increasing retention of POD bases, increasing reconstruction accuracy may be obtained.

The procedure for generating the training and testing data is illustrated in figure 3(a). A hundred different locations $\mathbf{r} = [\bar{x}, \bar{y}]$ are chosen for full-order simulations using finite-volume discretizations of the shallow-water equations as outlined in section 3.1. 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 10 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.

The POD approximation of our solution is then obtained via

$$
\hat{\mathbf{S}} = [ \hat{\mathbf{q}}_h^1 | \hat{\mathbf{q}}_h^2 | \cdots | \hat{\mathbf{q}}_h^{N_s} ] \approx \varphi \mathbf{A} \in \mathbb{R}^{N_h \times N_r},
$$

(12)

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

$$
\frac{\sum_{i=1}^{N_s} ||\mathbf{q}_h^i - \hat{\mathbf{q}}_h^i||_{\mathbb{R}^{N_h}}^2}{\sum_{i=1}^{N_s} ||\mathbf{q}_h^i||_{\mathbb{R}^{N_h}}^2} = \frac{\sum_{i=N_s+1}^{N_r} \lambda_r^2}{\sum_{i=1}^{N_s} \lambda_r^2},
$$

(13)

which exhibits that with increasing retention of POD bases, increasing reconstruction accuracy may be obtained.

The procedure for generating the training and testing data is illustrated in figure 3(a). A hundred different locations $\mathbf{r} = [\bar{x}, \bar{y}]$ are chosen for full-order simulations using finite-volume discretizations of the shallow-water equations as outlined in section 3.1. 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 10 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 $L_2$ error norm $\epsilon = ||\eta_{\text{True}} - \eta_{\text{POD}}||_2 / ||\eta_{\text{True}}||_2$ of $5.79 \times 10^{-4}$, as shown in figure 3(b). All our assessments for the machine learning frameworks are performed on snapshots obtained from the 10 simulations kept aside for the purpose of testing. Also, our assessments are performed solely for the $\eta$ flow field. We emphasize on this point, since we assume that the other key variables $\rho u$ and $\rho v$ 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 would also draw attention to the fact that the initial condition of the entire system is conditioned on a Gaussian excitation to $\rho u$ 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.

### 3.1.1 Parametric surrogates

We first assess the ability of the proposed PNN for predicting the evolution of the shallow-water equation dynamics given an input of $\mathbf{q} = [\bar{x}, \bar{y}]$ alone. In this task, we have a two-dimensional input space $\mathbf{q}$ and a 400-dimensional output space $\mathbf{a}_s = [\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^{40}]$ corresponding to 40 spatial POD coefficients $\mathbf{a}^i = [a^i_1, a^i_2, \ldots, a^i_{40}]$ over 10 time steps.
Figure 4: Coefficient estimations of two representative test simulation by the present framework showing error bars for two standard deviations around the mean as parameterized by a Gaussian distribution. Final time field reconstructions using estimated POD coefficient means are shown in the right side of each coefficient evolution. True indicates the reference field with 40 POD modes.

as to account for the entire trajectory of the $\eta$ dynamics. Following equation and figure, the problem setting here can be formulated as

$$\{\pi(q), \mu(q), \sigma(q)\} = F(q), \quad p(a_t|q) = \sum_{i=1}^{m} \pi_i(q) N(\mu_i(q), \sigma_i(q)).$$

For assessing the robustness of the framework, all POD coefficients (i.e., 400 for each simulation) are perturbed with additive noise obtained from a random number generator sampling from a zero-mean Gaussian distribution. Additionally, the variance of this distribution is further sampled from a uniformly distributed random number between 0 and 2. This adds a significant complexity to the training data set - namely that the measurement of $\eta$ is affected by noise across scales (since perturbations are occurring in global POD space) and that the variance of this noise is not fixed. Such situations call for probabilistic learning frameworks that may learn distributions conditioned on the initial conditions instead of a deterministic map. A schematic for this task and its assessment is shown in figure (a).

The coefficient estimations from the present framework for two representative simulations are shown in figure. 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 present of coherent distortions of the solution field. The corresponding field reconstructions using the mean coefficients can be seen in the right side of each coefficient evolution where qualitative agreement with the true simulations is clearly observable. We stress, once again, that 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.

3.1.2 Field reconstructions

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 previous section, we task the probabilistic neural network in predicting the POD coefficients $a$. Our inputs, however, are now given by sensor measurements $s$ 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

\[
\{ \pi([s, q_t]), \mu([s, q_t]), \sigma([s, q_t]) \} = \mathcal{F}([s, q_t]), \quad p(a|[s, q_t]) = \sum_{i=1}^{n_{\text{samples}}} \pi_i([s, q_t])\mathcal{N}(\mu_i([s, q_t]), \sigma_i([s, q_t])). \tag{15}
\]

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. Figure 5 shows the ability of the proposed framework to recover the coherent structures in the field. Note that we present \(L_2\) error norms of both estimated POD coefficients and reconstructed fields for figure 5\((b)\). A representative test simulation shows that increasing the number of sensors leads to 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 explains why the lowest \(L_2\) error norms are appeared at 30 sensors as presented in figure 5\((b)\) despite figure 5\((a)\) showing one snapshot with absent coherent structure reconstruction. It suggests that an optimal selection of sensor locations (by clustering) may improve convergence significantly. This inference can be also applied to the estimated standard deviations which do not show significant difference over the considered noise magnitude, as shown in figure 5\((c)\).

We then look at 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 figure 6\((b)\). Figure 6\((b)\) indicates 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 figure 6\((c)\), we clarify that the uncertainty estimation for the POD coefficients do not show significant difference over the considered noise magnitude. We should note that this is because the estimated confidence interval is just for estimations and not a barometer for error. The reader must 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 figures 5 and 6, 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.
3.2 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. 3.2.1), the wake of a NACA0012 airfoil with a Gurney flap (sec. 3.2.2), and the NOAA Optimum Interpolation Sea Surface Temperature data set (sec. 3.2.3). 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.

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

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

(16)

$$\{\pi(s_{\text{input}}), \mu(s_{\text{input}}), \sigma(s_{\text{input}})\} = F(s_{\text{input}}), \quad p(z|s_{\text{input}}) = \sum_{i=1}^{m} \pi_i(s_{\text{input}}) N(\mu_i(s_{\text{input}}), \sigma_i(s_{\text{input}})),$$

(17)

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

3.2.1 Two-dimensional cylinder wake

Let us consider a two-dimensional cylinder wake at $Re_D = 100$ as an example of application to unsteady flows around a bluff body. The data set has been obtained by using a two-dimensional direct numerical simulation (DNS) \[64, 65\]. The governing equations are the incompressible Navier–Stokes equations,$$
\nabla \cdot \mathbf{u} = 0, \quad \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re_D} \nabla^2 \mathbf{u},$

(18)

(19)

where $\mathbf{u}$, $p$ and $Re_D$ are the non-dimensionalized velocity vector, pressure and Reynolds number based on the cylinder diameter $D$, respectively. Five nested levels of multi-domains are considered for numerical set up. The finest
Figure 7: Sensor estimation of cylinder wake using mixture density network. The left side shows both input and output sensor locations. $L_2$ error norm $\epsilon = \frac{||s_{6,\text{DNS}} - s_{6,\text{ML}}||_2}{||s_{6,\text{DNS}}||_2}$ is 0.0137. Note here that the $\sigma$ can be hardly seen due to an excellent agreement of estimation with the reference.

Figure 8: Wake reconstruction of cylinder flow using mixture density network. The value below the estimated mean field $\mu$ indicates $L_2$ error norm $\epsilon = \frac{||\omega_{\text{DNS}} - \omega_{\text{ML}}||_2}{||\omega_{\text{DNS}}||_2}$.

level of the considered domain 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 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 $s_{\text{input}}$ 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 4 periods in time as the training data. The assessments are conducted using test data set which also contains 100 snapshots over approximately 4 periods and excludes from the training process.

For the problem settings of spatial fluid data recovery, we first apply the mixture density network 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 cylinder surface in figure 7. The estimations (red circles) are in excellent agreement with the reference data. This trend can be also seen from a quantitative assessment with $L_2$ error norm $\epsilon = \frac{||s_{6,\text{DNS}} - s_{6,\text{ML}}||_2}{||s_{6,\text{DNS}}||_2}$ of 0.0137. Since the cylinder wake at the present Reynolds number is on periodic nature in time, high standard deviation regions are not observed. This observation also enables us to have confidence for this estimation.

We then extend the model to the reconstruction of the whole wake, as shown in figure 8. 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 mixture density network provides us with the standard deviations of estimation as shown on the right side of figure 8. In this case, the probabilistic machine learning model tells us that the standard deviation on regions of vortex shedding and separated shear layer is relatively 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.

3.2.2 Wake of 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 $Re_c = 1000$, where $c$ is the chord length [68].
Figure 9: Sensor estimation of NACA0012 wake with a Gurney flap using mixture density network. The left side shows both input and output sensor locations. $L_2$ error norm $\epsilon = ||s_{6,DNS} - s_{6,ML}||_2/||s_{6,DNS}||_2$ is 0.0429.

Figure 10: Wake reconstruction of NACA0012 with a Gurney flap depending on the number of the snapshots for training. The values inside the estimated mean field indicates $L_2$ error norm $\epsilon = ||\omega_{DNS} - \omega_{ML}||_2/||\omega_{DNS}||_2$. The values inside the estimated standard deviation express the ensemble-averaged standard deviation over the field.

It is known that various types of wakes can emerge depending on the angle of attack $\alpha$ and the Gurney-flap height $h$ [66]. 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 figure 2(b). For the numerical setup, five nested levels of multi-domains 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 utilized domain and the number of grid points for data set are $[-0.5, 7] \times [-2.5, 2.5]$ and $(N_x, N_y) = (352, 240)$. We use the vorticity field $\omega$ as the input and output attributes. Five sensor measurements located on surface of an airfoil are chosen as input data for both the sensor estimation and the whole field reconstruction. The number of snapshots $n_{\text{snapshot}}$ used for the baseline model are the same as that in the cylinder problem such that 100 snapshots over approximately 4 periods for both training and test data. Note that we also investigate the dependence on the number of the snapshots using the example of NACA0012 wake.

The mixture density network is applied to the local sensor estimation of NACA0012 wake, as shown in figure 9. As it 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 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 contained frequency contents on the wakes, as mentioned above.

The estimated whole fields from the sensor measurements located on the surface of airfoil are shown in figure 10. With $n_{\text{snapshot}} = 100$, the location and size of vortex structures are captured reasonably well by using the present model. However, the $L_2$ error norm $\epsilon = ||\omega_{DNS} - \omega_{ML}||_2/||\omega_{DNS}||_2$ is 0.514, which is substantially larger than that in the case of cylinder wake, despite the reasonable estimation. This is because the $L_2$ error norm is known as a...
Figure 11: Dependence of estimation accuracy for the whole flow field 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 indicates $L_2$ error norm. The values inside the estimated standard deviation express the ensemble-averaged standard deviation over the field.

Next, let us examine the robustness for noisy input in figure 11, which is analogous to the assessment with the shallow water equation presented in figure 6. 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 figure 6. Note that the standard deviations have no significant difference among the considered noise level with same number of snapshots due to the fact that the estimated confidence interval is not a barometer for error. This trends coincide with the example of shallow water equation shown in figure 6. By comparing figures 11 (a) and (b), we can find that the error decreases by increasing the number of training snapshots, as we expected from the observation with the example of shallow water equation in figure 6. We reiterate, therefore, that
Figure 12: Instantaneous temperature field on the sea surface. Green circles are used sensors as input data.

Figure 13: Sensor estimation of sea surface temperature using mixture density network. $L_2$ error norm $\epsilon = \frac{||s_{H,\text{Ref}} - s_{H,\text{ML}}||_2}{||s_{H,\text{Ref}}||_2}$ is 0.0163.

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.

3.2.3 NOAA sea surface temperature

To demonstrate the applicability of mixture density network to practical applications, let us consider the NOAA sea surface temperature data set [67]. The field data here has the spatial resolution of $360 \times 180$ based on a one degree grid and is obtained from satellite and ship observations without adequate knowledge of underlying governing equations. We use the 20 years of data (1040 snapshots spanning 1981 to 2001) as the training data set, while the test data set is prepared from 874 snapshots spanning from year 2001 to 2018. Note in passing that 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. [68] who attempted to reconstruct fluid flow fields from local sensors using sparse representations. Following Callaham et al. [68], the input sensors for the baseline model are chosen randomly from the region of $50^\circ$S to $50^\circ$N, as shown in figure 12.

We present the result of local sensor estimation from 10 other sensors in figure 13. 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., cylinder wake and NACA0012 airfoil with a Gurney flap. One of possible reasons is the complexity 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 used range for training.

Let us also consider to the estimation of global temperature field, as shown in figure 14. The reconstructed field shows reasonable match with the reference field and this can be also found by $L_2$ 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 relatively higher standard deviation region across the Pacific Ocean near South America. This is because of El Niño (sea temperature is higher than usual) and La Niña (lower than usual), which emerge every few years in this area [69]. Hence, the temperature fluctuation in this area is larger than that in the other areas.
Figure 14: Whole field reconstruction of sea surface temperature. The value below the estimated mean field indicates \( L_2 \) error norm \( \epsilon = \frac{||T_{\text{Ref}} - T_{\text{ML}}||_2}{||T_{\text{Ref}}||_2} \).

Figure 15: Influence of estimation accuracy on additional sensors. Added 5 sensors (a) randomly and (b) based on the estimated standard deviation in figure 14. The values underneath the estimated mean fields indicate \( L_2 \) error norm.

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 reliability of estimated results but also the characteristics of 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, let us also assess this viewpoint by putting five additional sensors, as shown in figure 15. We consider two cases by adding 5 sensors (a) randomly and (b) based on the estimated standard deviation in figure 14. As presented, the latter outperforms the case of randomly chosen sensors in terms of \( L_2 \) 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.

4 Concluding remarks

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 parameterize the output as a sample from a Gaussian or a mixture of Gaussian distributions. Consequently, every prediction is accompanied 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, non-intrusively, 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.
Following assessments for the shallow water equations, the probabilistic neural network was then applied to 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 for the NOAA optimal interpolation sea-surface temperature data set. For the problems of two-dimensional cylinder wake and 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 modelled 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.

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.

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 [70], 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 [71] and generalized moving least squares [72] frameworks.

**Acknowledgement**

This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Contract 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 DE-AC02-06CH11357. R.M. acknowledges support from the ALCF Margaret Butler Fellowship. K. Fukami and K. Fukagata thank the support from Japan Society for the Promotion of Science (KAKENHI grant number: 18H03758). K. Taira acknowledges the generous support from the US Army Research Office (grant number: W911NF-17-1-0118) and US Air Force Office of Scientific Research (grant number: FA9550-16-1-0650). The authors acknowledge M. Gopalakrishnan Meena (University of California, Los Angeles) for sharing his DNS data.

**References**

[1] S. L. Brunton, B. R. Noack, and P. Kouloutsakos. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.*, 52:477–508, 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.*, 814:1–4, 2017.

[4] K. Fukami, K. Fukagata, and K. Taira. Assessment of supervised machine learning methods for fluid flows. *Theor. Comp. Fluid Dyn.* (available online), https://doi.org/10.1007/s00162-020-00518-y, 2020.

[5] R. Maulik, O. San, J. D. Jacob, and C. Crick. Sub-grid scale model classification and blending through deep learning. *J. Fluid Mech.*, 870:784–812, 2019.

[6] M. Gamahara and Y. Hattori. Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids*, 2(5):054604, 2017.
[7] R. Maulik and O. San. A neural network approach for the blind deconvolution of turbulent flows. *J. Fluid Mech.*, 831:151–181, 2017.

[8] A. Vollant, G. Balarac, and C. Corre. Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures. *J. Turbul.*, 18(9):854–878, 2017.

[9] J. Ling and J. Templeton. Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Phys. Fluids*, 27(8):085103, 2015.

[10] E. J. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comp. Phys.*, 305:758–774, 2016.

[11] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.*, 807:155–166, 2016.

[12] J-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids*, 3(7):074602, 2018.

[13] R. Matai and P. A. Durbin. Zonal eddy viscosity models based on machine learning. *Flow, Turbulence and Combustion*, 103(1):93–109, 2019.

[14] L. Zhu, W. Zhang, J. Kou, and Y. Liu. Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Phys. Fluids*, 31(1):015105, 2019.

[15] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annu. Rev. Fluid. Mech.*, 51:357–377, 2019.

[16] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. Dawson, and CA Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA J.*, 58(3):998–1022, 2020.

[17] K. Taira, M. S. Hemati, and L. S. Ukeiley. Modal analysis of fluid flows: Introduction to the virtual collection. *AIAA J.*, 58(3):991–993, 2020.

[18] 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. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[19] S. E. Ahmed, O. San, A. Rasheed, and T. Iliescu. A long short-term memory embedding for hybrid uplifted reduced order models. arXiv preprint, arXiv:1912.06756, 2019.

[20] 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: Nonlinear Phenomena*, 405:132368, 2020.

[21] 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 preprint, arXiv:1903.00033, 2019.

[22] 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 preprint, arXiv:1804.09269, 2018.

[23] 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. Comp. Phys.*, 410:109402, 2020.

[24] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Comm.*, 9(1):4950, 2018.

[25] F. J. Gonzalez and M. Balajewicz. Learning low-dimensional feature dynamics using deep convolutional recurrent autoencoders. arXiv preprint, arXiv:1808.01346, 2018.

[26] J. Xu and K. Duraisamy. Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. arXiv preprint, arXiv:1912.11114, 2019.

[27] R. Maulik, B. Lusch, and P. Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. arXiv preprint, arXiv:2002.00470, 2020.

[28] M. Cheng, F. Fang, C. C. Pain, and I. M. Navon. An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling. arXiv preprint, arXiv:2003.10547, 2020.

[29] 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. Comp. Fluid Dyn. (available online)*, https://doi.org/10.1007/s00162-020-00528-w, 2020.

[30] C. W. Rowley and S. T. M. Dawson. Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, 49:387–417, 2017.
[31] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata. Synthetic turbulent inflow generator using machine learning. 
Phys. Rev. Fluids, 4:064603, 2019.

[32] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz. Shallow learning for fluid 
flow reconstruction with limited sensors and limited data. arXiv:1902.07358, 2019.

[33] K. Fukami, K. Fukagata, and K. Taira. Super-resolution reconstruction of turbulent flows with machine learning. 
J. Fluid Mech., 870:106–120, 2019.

[34] K. Fukami, K. Fukagata, and K. Taira. Super-resolution analysis with machine learning for low-resolution flow 
data. In 11th International Symposium on Turbulence and Shear Flow Phenomena (TSFP11), Southampton, UK, 
number 208, 2019.

[35] R. Onishi, D. Sugiyama, and K. Matsuda. Super-resolution simulation for real-time prediction of urban micrometeorology. 
Scientific Online Letters on the Atmosphere (SOLA), 15:178–182, 2019.

[36] 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.

[37] K. Fukami, K. Fukagata, and K. Taira. Machine learning based spatio-temporal super resolution reconstruction of 
turbulent flows. arXiv:2004.11566, 2020.

[38] M. Milano and P. Koumoutsakos. Neural network modeling for near wall turbulent flow. J. Comput. Phys., 
182:1–26, 2002.

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

[40] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper 
orthogonal decomposition. AIAA J, 42(8), 2004.

[41] T. Suzuki and Y. Hasegawa. Estimation of turbulent channel flow at Reτ = 100 based on the wall measurement 
using a simple sequential approach. J. Fluid Mech., 830:760–796, 2006.

[42] M. Morimoto, K. Fukami, and K. Fukagata. Experimental velocity data estimation for imperfect particle images 
using machine learning. arXiv:2005.00756, 2020.

[43] S. Cai, S. Zhou, C. Xu, and Q. Gao. Dense motion estimation of particle images via a convolutional neural 
network. Exp. Fluids, 60:60–73, 2019.

[44] 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.

[45] J. Kim and C. Lee. Prediction of turbulent heat transfer using convolutional neural networks. J. Fluid Mech., 
882:A18, 2020.

[46] G. E. P. Box and G. C. Tiao. Bayesian inference in statistical analysis, volume 40. John Wiley & Sons, 2011.

[47] G. E. Hinton and D. van Camp. Keeping the neural networks simple by minimizing the description length of the 
weights. In Proceedings of the Sixth Annual Conference on Computational Learning Theory, COLT ’93, page 
5–13, New York, NY, USA, 1993. Association for Computing Machinery.

[48] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. arXiv 
preprint, arXiv:1505.05424, 2015.

[49] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation. arXiv preprint, arXiv:1506.02157, 2015.

[50] C.-K. Lu, S. C.-H. Yang, X. Hao, and P. Shafto. Interpretable deep Gaussian processes with moments. arXiv 
preprint, arXiv:1905.10963, 2019.

[51] A. Damianou and N. Lawrence. Deep Gaussian Processes. In Artificial Intelligence and Statistics, pages 207–215, 
2013.

[52] S. Santini and A. Del Bimbo. Recurrent neural networks can be trained to be maximum a posteriori probability 
classifiers. Neural Networks, 8(1):25 – 29, 1995.

[53] C. M. Bishop. Mixture density networks. 1994.

[54] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint, arXiv:1412.6980, 2014.

[55] J. Bergstra, D. Yamins, and D. Cox. Making a science of model search: Hyperparameter optimization in hundreds 
of dimensions for vision architectures. International Conference on Machine Learning, pages 115–123, 2013.

[56] E. Brochu, V. Cora, and de N. Freitas. A tutorial on bayesian optimization of expensive cost functions, with 
application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-023, 
University of British Columbia, 2009.
[57] R. Maulik, K. Fukami, N. Ramachandra, K. Fukagata, and K. Taira. https://github.com/Romit-Maulik/Probabilistic_ML_Fluids

[58] X-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. J. Comp. Phys., 115(1):200–212, 1994.

[59] E. Hairer, S. P. Nørsett, and G. Wanner. Solving ordinary differential equations. 1, Nonstiff problems. Springer-Vlg, 1991.

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

[61] 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(12):4013–4041, 2017.

[62] D. D. Kosambi. Statistics in function space. J Indian Math. Soc., 7:76–88, 1943.

[63] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech., 25(1):539–575, 1993.

[64] K. Taira and T. Colonius. The immersed boundary method: A projection approach. J. Comput. Phys., 225(2):2118–2137, 2007.

[65] 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., 197:2131–2146, 2008.

[66] M. Gopalakrishnan Meena, K. Taira, and K. Asai. Airfoil wake modification with gurney flap at low Reynolds number. AIAA J., 56(4):1348–1359, 2018.

[67] Available on https://www.esrl.noaa.gov/psd/.

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

[69] J. N. Kutz, X. Fu, and S. L. Brunton. Multi-resolution dynamic mode decomposition. SIAM Journal of Applied Dynamical Systems, 15(2):713–735, 2016.

[70] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proc. IEEE, 86(11):2278–2324, 1998.

[71] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[72] N. Trask, R. G. Patel, B. J. Gross, and P. J. Atzberger. Gmls-nets: A framework for learning from unstructured data. arXiv preprint, arXiv:1909.05371, 2019.