AIAA JOURNAL

Prediction of Intermittent Fluctuations from Surface Pressure Measurements on a Turbulent Airfoil

Samuel H. Rudy* and Themistoklis P. Sapsis†
Massachusetts Institute of Technology, Cambridge, MA 02139

https://doi.org/10.2514/1.J061163

This work studies the effectiveness of several machine-learning techniques for predicting extreme events occurring in the flow around an airfoil at low Reynolds. For certain Reynolds numbers, the aerodynamic forces exhibit intermittent fluctuations caused by changes in the behavior of vortices in the airfoil wake. Such events are prototypical of the unsteady behavior observed in airfoils at low Reynolds, and their prediction is extremely challenging due to their intermittency and the chaotic nature of the flow. We seek to forecast these fluctuations in advance of their occurrence by a specified length of time. We assume knowledge only of the pressure at a discrete set of points on the surface of the airfoil, as well as offline knowledge of the state of the flow. Methods include direct prediction from historical pressure measurements, flow reconstruction followed by forward integration using a full-order solver, and data-driven dynamic models in various low-dimensional quantities. Methods are compared using several criteria tailored for extreme event prediction. We show that methods using data-driven models of low-order dynamic variables outperform those without dynamic models and that, unlike previous works, low-dimensional initializations do not accurately predict observables with extreme events such as drag.

Nomenclature

$\mathcal{C}$ = chord length
C_d, C_l = drag and lift coefficients
D = decoder network for flow reconstruction
dt = time step
E = encoder network for flow reconstruction
$F_{P/\psi/\xi}$ = nondynamic models for mapping $P/\psi/\xi$ to q
$F_{1,\mathrm{opt}}$ = optimal test set F_1 score
G = network for estimating proper orthogonal decomposition time series
$H_{P/\psi/\xi}$ = dynamic models for variables P, ψ, and ξ
$\mathcal{L}_\bullet$ = loss function used to train neural network $\bullet$
m = mass matrix from spectral element grid
P = airfoil surface-pressure time series
q = smoothed drag coefficient
r = rank of low-dimensional representations of flow
S, R, F_1, α = precision, recall, F_1 score, and area under S-R curve
$\boldsymbol{u}$, p = fluid velocity and pressure
u_∞ = inlet velocity
w = weights for proper orthogonal decomposition and flow reconstruction
α^* = maximum adjusted area under precision-recall curve
ξ = latent space representation of flowfield in full-field neural network
τ = load time for prediction of q
$\Phi\Sigma\psi(t)^T$ = components of proper orthogonal decomposition of flowfield at time t
$\Phi_r\Sigma_r\psi_r(t)^T$ = rank r truncation of proper orthogonal decomposition
ω = extreme event rate

I. Introduction

EXTREME events are common features in engineering and scientific disciplines including climate, ocean engineering, and fluid structure interaction that are characterized by observables of a dynamical system exhibiting heavy tails [1]. The outlier events populating these tails are of particular interest due to their effects on aerodynamics and fatigue or other potentially adverse consequences. However, the rarity and intermittency of such events also makes their prediction challenging. There has been significant recent interest in sampling strategies [2–4], optimization schemes [5], and tailored loss functions [6,7] for the prediction of extreme events. A common goal of many of the past works on extreme events and of the present work is the prediction of extreme events in advance of their occurrence.

The focus of this work is on the two-dimensional incompressible flow around an airfoil at low ($\mathcal{O}(10^4)$) Reynolds number. Dynamics of the flow around the airfoil at this Reynolds regime are highly nontrivial [8] and have been shown to be characteristically different from those at higher Reynolds [9]. Previous works using both experimental and computational tools have found that slow-moving airfoils exhibit a large range of wake behaviors, with qualitative changes in the nature of the flow occurring with small changes in angle of attack and Reynolds number [10–12]. Similar unstable behavior has been observed in the flow around a cylinder and in the so-called transitional regime between ordered and disordered behavior [13], as well as in vortex-induced vibrations of flexible cylinders [14]. Due to these instabilities, as well as an apparent lack of fidelity between computational and experimental results, it has been suggested that construction of rigid-winged slow-flying vehicles may be challenging if not impossible [15].

Despite apparent challenges, there has long been considerable interest in the study of low Reynolds airfoils [8]. In particular, recent works have explored the use of machine learning to estimate flowfield and aerodynamic data from sensors on the surface of the airfoil. These include methods for flow reconstruction from limited sensors using neural networks [16,17], filtering-based flow estimation [18,19], and prediction of aerodynamic coefficients [20]. Deep learning has also been used for estimating properties of the flow used in low-order vortex models. In [21,22], authors use neural network-based methods to estimate the leading-edge suction parameter (LESP). The model studied in this work has a Reynolds number of 17,500, substantially higher than other computational works that focus on Reynolds number in the range of $\mathcal{O}(10^2) - \mathcal{O}(10^3)$ [16,18,19,22]. This is closer to the lower end of the range considered by experimental work [17,20]. Unlike some other works [17,20–22], this work does not study the effects of pitching motions or

---

Received 11 August 2021; revision received 17 January 2022; accepted for publication 11 February 2022; published online 16 March 2022.

*Postdoctoral Associate, Department of Mechanical Engineering, 77 Massachusetts Ave.

†Associate Professor of Mechanical and Ocean Engineering, Mechanical Engineering, 77 Massachusetts Ave.

disturbances in the incident velocity. We instead focus on prediction of intermittent fluctuations in the aerodynamic coefficients of a stationary airfoil, which have not been the focus of previous works.

This work applies several machine-learning techniques to predict fluctuations in the drag coefficient of an airfoil in the transitional regime where we observe chaotic and intermittent behavior. It follows a broader trend of the application of tools from machine learning to problems in fluid dynamics. For a broader view, the interested reader may refer to a number of recent articles outlining and discussing the role of machine learning in fluid dynamics: [23] provides an excellent overview of many of the applications machine learning has seen in fluids; [24] provides an assessment of several common supervised learning methods applied to flow reconstruction, superresolution, and coefficient estimation; and [25] provides a discussion of the role of machine learning in fluids, as well as some pitfalls and concerns.

The paper is organized as follows: In Sec. II we formulate the problem of predicting aerodynamic fluctuations. Section III describes the methodology used in this work, including flow reconstruction methods in Sec. III.A, forecasting methods for aerodynamic fluctuations in Sec. III.B, and a discussion of performance metrics in Sec. III.C. Results are shown in Sec. IV with comparisons between all methods. In Sec. V we offer closing thoughts, pitfalls, and potential future directions based on this paper.

II. Problem Description

We consider a NACA 4412 airfoil at chord-length-based Reynolds number of $Re = 17{,}500$ and 5 deg angle of attack. Flow around the airfoil is simulated using the spectral element code Nek5000 [26] according to the incompressible Navier–Stokes equations given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\nabla p + \frac{1}{Re}\nabla^2\boldsymbol{u} \qquad \nabla\cdot\boldsymbol{u} = 0 \tag{1}$$

The computational grid uses 4368 elements with spectral order 7. Statistics of observables relevant to this work were found to be in agreement with those from a shorter simulation using a more resolved grid having 14,144 elements. A convective boundary condition is used for the outflow [27]. The spectral element grid without Gauss–Lobatto Legendre interpolation points and a snapshot of the vorticity are shown in the top row of Fig. 1. Further details of the numerical simulation are provided in Appendix A, and software for reproducing data used in this work is available online.

Pressure recordings along the surface of the airfoil $P(t)$ are taken at a discrete set of 50 points around the airfoil at intervals of $dt = 0.01$ throughout the simulation. These locations are shown in the bottom panel of Fig. 1. Aerodynamic force is computed using the pressure and skin friction. This decomposes into the streamwise x and crossflow y directions, defined by

$$\boldsymbol{F}(t) = \oint \boldsymbol{\tau}(t) - p(t)\boldsymbol{n}\,\mathrm{d}s = D(t)\boldsymbol{e}_x + L(t)\boldsymbol{e}_y \tag{2}$$

where $\boldsymbol{\tau}$, p, and $\boldsymbol{n}$ are the skin shear stress, pressure, and wall normal vector, and the integral is taken over the airfoil surface. Forces are then used to compute the nondimensional drag-coefficient C_d and lift coefficient C_l, defined as

$$C_d(t) = \frac{2D(t)}{\rho u_\infty^2 C}, \qquad C_l(t) = \frac{2L(t)}{\rho u_\infty^2 C} \tag{3}$$

where chord length $C = 1$, density $\rho = 1$, and freestream velocity $u_\infty = 1$.

The two-dimensional simulation yields a quasi-stable behavior in which intermittent fluctuations are observed in the aerodynamic coefficients, shown in the first two rows of Fig. 2 alongside the density functions of their absolute deviations. A more detailed view of these fluctuations, their frequency of occurrence, and the flow states associated with them is provided in Appendix B. We note that the density functions clearly exhibit the expected "heavy tails" associated with observables of dynamical systems with extreme events [1]. This regime of intermittent fluctuations is persistent for the entirety of the simulation used in this work but with altered conditions may exhibit mode switching to a state with more regular oscillations. Further details of this case are given in Appendix D. In this work we focus solely on the regime where intermittent fluctuations are observed.

The goal of the present work is to predict these intermittent fluctuations in advance of their occurrence by some lead time τ using surface pressure information. To focus on nonperiodic behavior, we consider predictions on a smoothed time series derived from the drag coefficient. Specifically,

$$q(t) = (C_d * K)(t) \tag{4}$$

where $f_{\text{peak}} \approx 1.44$ is the peak frequency of the drag coefficient and Gaussian smoothing kernel K is given by a zero-mean normal

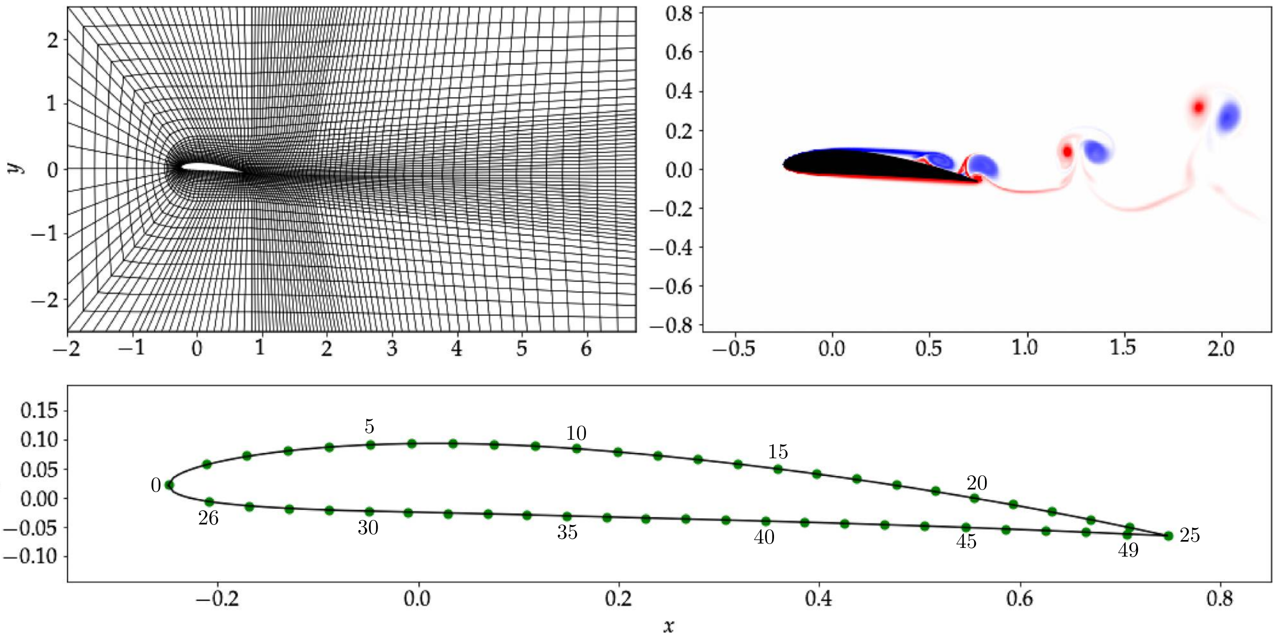

Fig. 1 Left: Domain of computational problems showing outlines of spectral elements without internal interpolation points. Right: Snapshot of vorticity. Bottom: Sensor placement showing indexing from 0–25 along top and 25–49 along bottom of airfoil. Axes on bottom figure not drawn to scale.

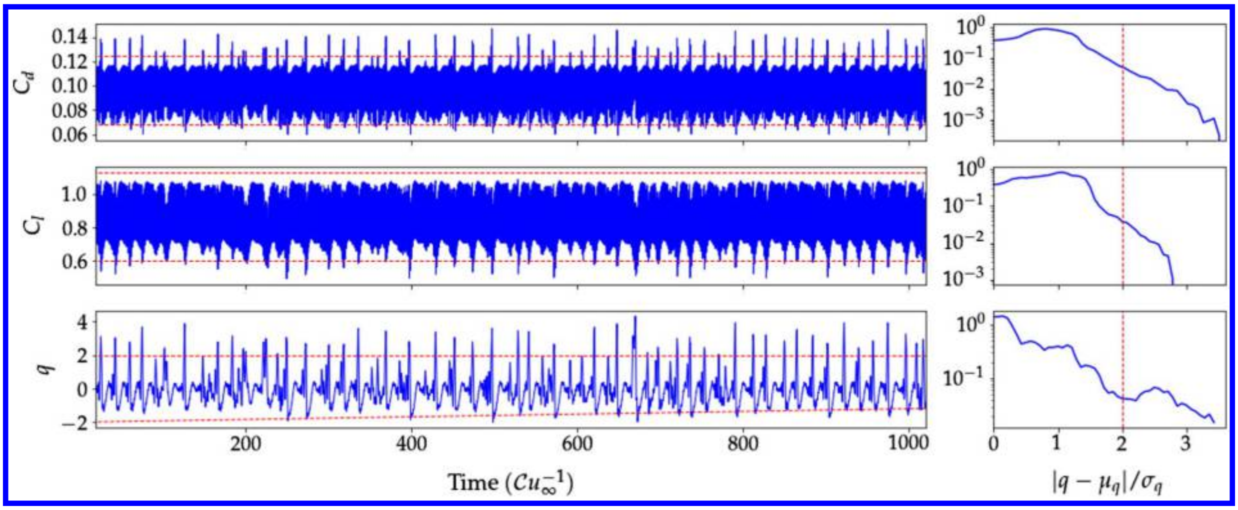

Fig. 2 Left: Aerodynamic coefficients C_d and C_l, and smoothed drag coefficient q as defined in Eq. (4). Red dashed lines indicate ±2 standard deviations away from the mean. Right: Histograms of the deviation of each quantity showing typical heavy tails of systems with extreme events.

distribution with standard deviation $1/(2f_{\mathrm{peak}})$. That is, $K(s) \propto \exp(-s^2/(2f_{\mathrm{peak}})^2)$. In practice, the convolution in Eq. (4) is computed using a compactly supported kernel having width $3/f_{\mathrm{peak}}$. The time series $q(t)$ is also normalized to be mean zero and unit variance. The quantity $q(t)$ captures the nonperiodic behavior of the drag coefficient, in particular maintaining extreme events and the heavy tailed deviation. The goal of this work is thus concisely stated as learning data-driven models for the prediction problem

$$P(s \le t) \to q(t+\tau) \tag{5}$$

at various $\tau \ge 0$.

III. Methods

In this section we formulate several data-driven models for Eq. (5). We assume knowledge of measurements of pressure at a discrete set of points along the surface of the airfoil up to the current time t and seek to predict the value of the extreme event indicator at time $t+\tau$. We denote the time series for pressure measurements up to the current time t as $P_t = P(s \le t)$. The minimal problem is thus learning a function directly from historical pressure measurements to the $q(t+\tau)$, which may be done using a variety of standard machine-learning tools.

A baseline predictor for Eq. (5) is found using a standard recurrent neural network, such as a long short-term memory networks (LSTM) [28], to interpolate a function directly from P_t to $q(t+\tau)$. Alternatively, we may try to improve forecasts of $q(t)$ through the offline use of flowfield data. Previous works [5] have used modal representations of a flowfield combined with adjoint equations to learn precursor states to extreme events. In this work we discuss two methods for compressing flowfield data, sensing expansion coefficients in the compressed basis, and exploiting this knowledge for potentially improved prediction of $q(t+\tau)$: the proper orthogonal decomposition and neural network-based flow reconstruction. In each case we study the predictive capability of initializing a flow solver with the reduced-order initial condition, predicting directly from historical representations in the reduced space, and neural network-based reduced-order models. We also consider data-driven dynamic models for the pressure measurements.

The methods considered in this work are summarized in Fig. 3. Methods are separated into an offline compression stage, a sensing stage where we infer the reduced-order state from point pressure measurements, reconstruction of the flowfield from the reduced-order state, and finally forecast of the quantity of interest $q(t)$. Learned functions G, E, and D are predictors of the POD mode time series, latent space encoder, and flow reconstruction network, respectively. The letter F has been used to denote LSTM predictions that do not use a dynamic model, with subscript indicating input. H is used to denote neural network-based reduced-order models. In each case, we use time series for pressure at discrete points on the surface of the airfoil as the starting point of the online prediction.

Details on each method are provided in the following sections. We assume familiarity with common implementations of neural

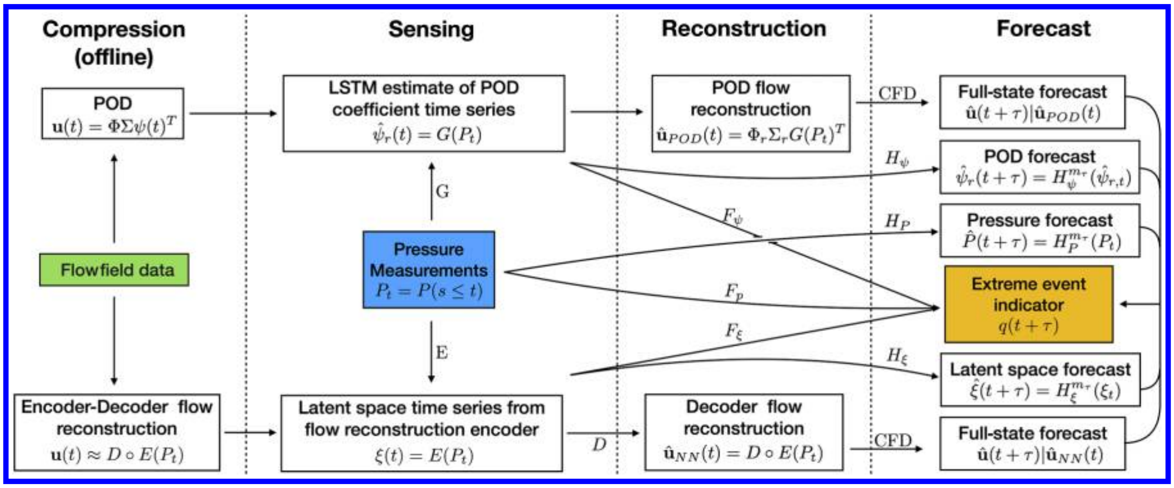

Fig. 3 Schematic of methods for extreme event prediction showing various flows of information from assumed knowledge of historical pressure measurements P_t to future value of quantity of interest $q(t+\tau)$.

networks and stochastic optimization. In particular, the work in this manuscript makes use of deep LSTMs [28] and the Adam method for optimization [29]. The unfamiliar reader may find an excellent reference in the free online textbook [30]. Further details on the structure and implementation of neural networks are given in Appendix C.

A. Flowfield Compression and Reconstruction

We begin with a discussion of the two offline methods for flow reconstruction: the proper orthogonal decomposition (POD) and an LSTM-based encoder–decoder pair. In this subsection we provide an overview of the methodology used to form the reduced rank representation for each of these two cases as well as methods for approximating time series associated with each POD mode.

1. *Proper Orthogonal Decomposition and Sensing*

The proper orthogonal decomposition is a standard tool for decomposing a flowfield into spatial modes that are orthogonal with respect to a given inner product and whose time evolutions are also orthogonal [31,32]. We apply the POD to the velocity field around the airfoil using a weighted inner product. Specifically, the POD finds matrices Φ, Σ, and ψ, which are discretized solutions to

$$\boldsymbol{u}(\boldsymbol{x},t) - \bar{\boldsymbol{u}}(\boldsymbol{x}) = \Phi(\boldsymbol{x})\Sigma\psi(t)^T \tag{6}$$

where

$$\begin{aligned} \Sigma &= \mathrm{diag}(\sigma_1,\ldots,\sigma_r), \sigma_i \geq \sigma_{i+1} \geq 0 \\ \langle \phi_i, \phi_j \rangle_w &= \int_\Omega \phi_i(\boldsymbol{x})^T \phi_j(\boldsymbol{x}) w(\boldsymbol{x})\,\mathrm{d}x = \delta_{i,j} \\ \langle \psi_i, \psi_j \rangle_w &= \int_0^T \psi_i(t)\psi_j(t)\,\mathrm{d}t = \delta_{i,j} \end{aligned} \tag{7}$$

and $w(\boldsymbol{x}) > 0$ is a weight used to focus the inner product, and thus variation explained by POD, on regions of interest near the surface of the airfoil. A similar weighted approach was used in [5], where a wall-focused POD was used as a basis for predicting extreme dissipation events in channel flow. We use the family of weights given by

$$w(\boldsymbol{x}) = \frac{1-\epsilon}{1+e^{(d(x)-l)/\delta}} + \epsilon \tag{8}$$

where $d(\boldsymbol{x})$ is the distance from $\boldsymbol{x}$ to the surface of the airfoil. Equation (8) describes a smooth sigmoidal curve that decays from 1 (assuming $l \gg \delta$) at the airfoil surface to ϵ in the far field. For small δ, this transition is localized around $d = l$ and the weights are approximately 1 for $d < l - \delta$ and ϵ for $d > l + \delta$. We use parameter values $l = 1$, $\delta = 0.1$, and $\epsilon = 0.1$. The POD is therefore principally focused on describing variation in the velocity field within one chord length of the airfoil surface, with approximately one-tenth the weighting for variation outside this region.

In the case where $w(\boldsymbol{x}) = 1$, the POD is equivalent to the singular value decomposition of the mean subtracted data, also known as principal component analysis. For nonidentity weights, Σ and $\psi(t_j)$, $j = 1,\ldots,m$ are given by the eigenvalue decomposition of $\boldsymbol{U}^T \boldsymbol{W}_m \boldsymbol{U}$ where $\bar{\boldsymbol{U}} \in \mathbb{R}^{n\times m}$ is the mean-subtracted velocity data and $\boldsymbol{W}_m$ is a diagonal matrix with $w(\boldsymbol{x}_i)m(\boldsymbol{x}_i)$ along the diagonal where $m(\boldsymbol{x}_i)$ is the mass associated with that grid point for the spectral element grid [31]. Modes Φ are subsequently computed using their definition in Eq. (6). Alternative methods may compute Φ before Σ and ψ, but these suffer from numerical issues for $\epsilon \ll 1$. Mean flow, leading modes, and singular values are shown in Fig. 4. We also show the normalized residual energy, defined by

$$E_{\mathrm{res},j} = \frac{\sum_{i=j+1}^m \sigma_i^2}{\sum_{i=1}^m \sigma_i^2} \tag{9}$$

at various truncation ranks. Truncation ranks for $E_{\mathrm{res}} = 0.1$ and 0.01 are 60 and 391, respectively.

Applying POD to the airfoil data yields modes ϕ_i, shown in Fig. 4, singular values Σ, and time series $\psi_i(t)$ corresponding to each mode ϕ_i. In the online phase of any prediction method, we will only have access to P_t, not $\psi_i(t)$. The latter may be estimated from sparse or gappy measurements [33]. We therefore train a deep LSTM model to estimate the current POD representation of the flow from pressure measurements. Letting $\psi_r(t)$ be the rank r truncation of the time series of the POD expansion, we have

$$\hat{\psi}_r(t) = G(P_t) \tag{10}$$

The exact form of and training procedure for G are described in greater detail in Appendix C. The number of modes predicted by G may be tuned as appropriate. We test $r = 8$, 16, 32, and 64. For $r = 32$, true values of POD coefficients as well as their estimates from pressure via Eq. (10) are shown in Fig. 5. Note that reconstructed time series are filled in on a much denser grid than true values since they are computed from the finely sampled pressure time series. Temporal resolution on $\psi(t)$ is limited by the number of output files saved during numerical simulation as well as memory limitations in the computation of the POD.

It is worth noting that the time series $\psi(t)$ are normalized to unit variance and Σ is not considered in loss function. Hence, error in higher modes is treated the same as error in lower modes. Higher modes did still have higher error, perhaps because they tended to exhibit more chaotic behavior. The authors did not explore the loss function exhaustively since doing so would be a significant research endeavor on its own.

After prediction of $\hat{\psi}(t)$ using G, one may reconstruct an approximation of the full flowfield using Eq. (6). The lower two rows on Fig. 5 show the results of this in the streamwise direction as well as the true velocity and absolute error for two snapshots taken from the testing dataset, that is, snapshots not seen by the optimization algorithm used to learn G. As may be expected from Eq. (8), error in the wake is larger than that close to the airfoil. These reconstructions suffer from multiple sources of error. Expanding the difference between true and reconstructed fields, we get

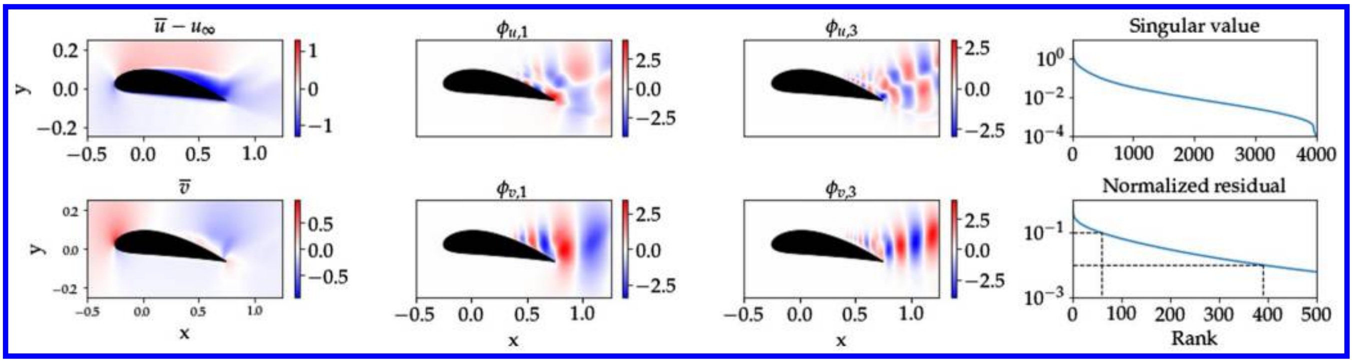

Fig. 4 The mean velocity field and three POD modes. Axes not drawn to scale.

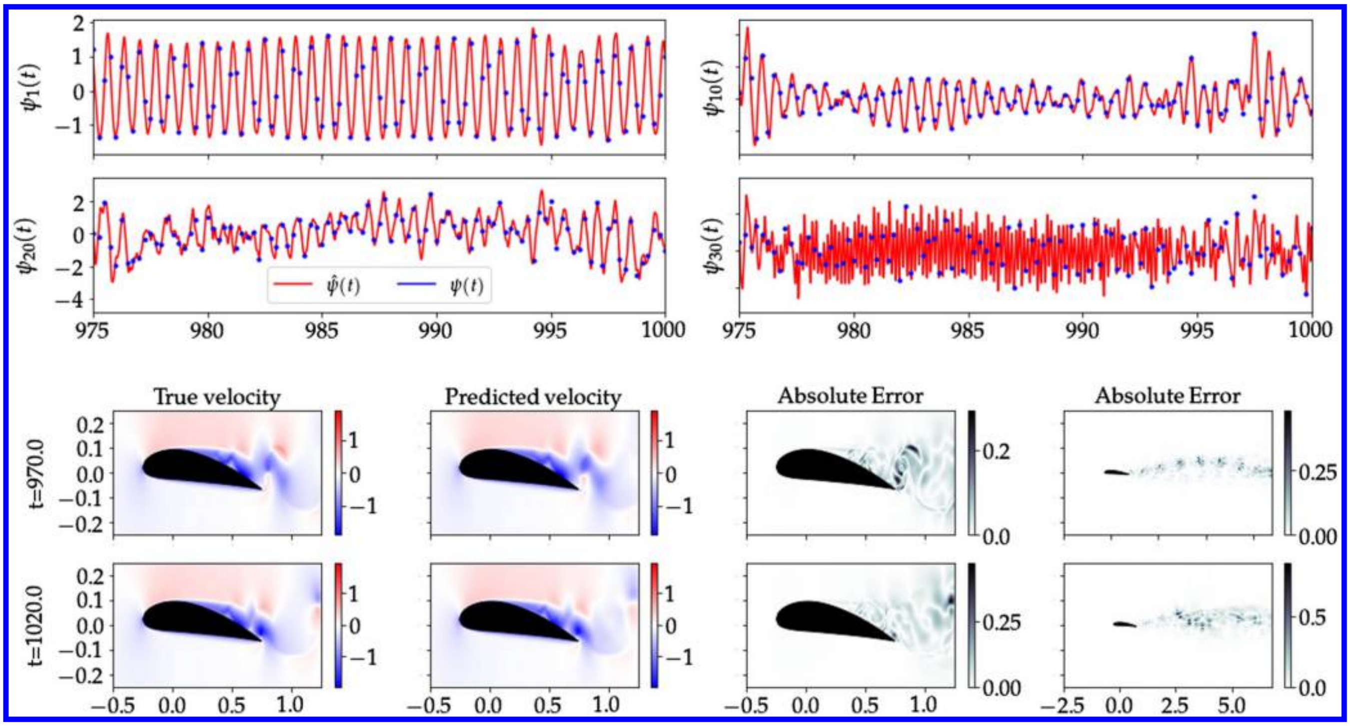

Fig. 5 Flowfield reconstruction using LSTM to predict time series for first 32 POD modes using pressure measurements P_t. Top [Operator $G(P_t)$]: LSTM prediction of POD time series for 1st, 10th, 20th, and 30th modes (red) and sparser true values (blue). Bottom [Operator $\Phi\Sigma G(P_t)^T$]: POD reconstruction of streamwise velocity compared to two true fields within test set data.

$$
\begin{aligned}
\boldsymbol{u}(t) - \hat{\boldsymbol{u}}_{\mathrm{POD}}(t) &= \Phi\Sigma\psi(t)^T - \Phi_r\Sigma_r G(P_t)^T \\
&= \underbrace{\Phi_{-r}\Sigma_{-r}\psi_{-r}(t)^T}_{\text{Unresolved}} + \Phi_r\Sigma_r\left(\underbrace{\psi_r(t) - G(P_t)}_{\text{NN Error}}\right)^T
\end{aligned}
\quad (11)
$$

where the subscript $-r$ denotes those components not in the first r. Error in the neural network increases for modes with high frequency and chaotic behavior, such as mode 30 shown in Fig. 5. However, significant error is also incurred from unresolved modes due to the slow decay of singular values Σ, as shown by Fig. 6. While spatially periodic translational behavior may be represented with pairs of modes [see Eq. (34) in [32]], the system studied in the present work tends to shed isolated pairs of vortices. Moreover, it also exhibits intermittency, which may be difficult to capture in a POD basis. The authors are not aware of efficient linear methods for representing translation of sparse structures such as the wake vortices observed in this data.

2. *Neural Network Flow Reconstruction*

In light of some of the deficiencies of the POD, and motivated by the successful applications of neural networks to problems in fluid dynamics [23,24,34–36], we also consider neural network-based approaches to flow reconstruction. The approach in this work is to use an LSTM-based encoder coupled with a fully connected network predicting the velocity at each grid point. The fully connected decoder resembles that used in [16,37], though due to the large computational grid, we have not used the probabilistic formulation considered in the latter. We note, however, that neither of the aforementioned papers included the use of history terms in their prediction of the fluid state, as we will show is done by the LSTM encoder for the networks considered in this work. For brevity, we will call networks of this type full-field neural networks (FFNN), indicating that the output of the network is the values of fluid velocity at each grid point used by a solver.

As in the POD case, we represent the fluid velocity around an airfoil at time t using a low-dimensional representation $\xi(t) \in \mathbb{R}^r$. The time history of the pressure sensors is encoded to the state using an LSTM given by

$$\xi(t) = E(P_t) \quad (12)$$

Since the full state of the flowfield is encoded in $\xi(t)$, one may tune the dimension of ξ to acquire a desired rank for the reduced-order representation of $\boldsymbol{u}$. We found improvements in reconstruction accuracy up to approximately rank $r = 32$, with minimal improvement at high values. We therefore use $r = 32$ for the remainder of this work unless noted otherwise. The reconstructed velocity field is then given by

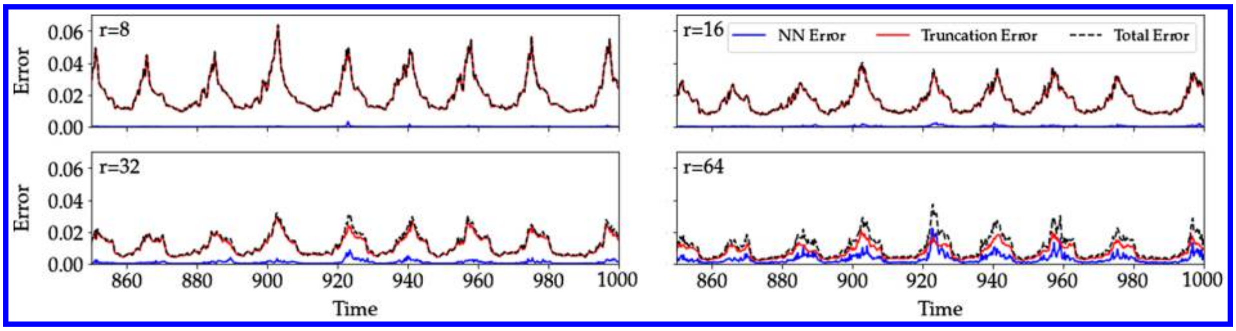

Fig. 6 Error due to truncation and neural network as measured by $\|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\mathrm{POD}}\|_{W_m}$.

$$\hat{\boldsymbol{u}}_{NN}(\boldsymbol{x}_i, t) = D(\xi(t))_i \tag{13}$$

where D is a standard fully connected neural network with the final layer being linear. Taken together, Eqs. (12) and (13) form a recurrent network from pressure sensor time histories to the fluid velocity given by

$$\hat{\boldsymbol{u}}_{NN}(x_i, t_j) = D \circ E(P_{t_j})_i \tag{14}$$

which are trained together using numerical simulation data. Further details on the network structure may be found in Appendix C.

We train the networks using a weighted square-error loss function designed to favor accurate reconstruction near the wall boundary

$$\mathcal{L}_{E,D} = \sum_j \int_\Omega \|\boldsymbol{u}(\boldsymbol{x}, t_j) - \hat{\boldsymbol{u}}_{NN}(\boldsymbol{x}, t_j)\|^2 w(\boldsymbol{x})\, \mathrm{d}x \tag{15}$$

where Ω is the computational domain and $w(\boldsymbol{x})$ is as defined in Sec. III.A.1 using $\epsilon = \delta = 0.1$ and $l = 1$. It is also possible to weigh the loss function to more heavily penalize errors in times immediately preceding an extreme event, but doing so was observed to have little effect on prediction performance. In practice, the integral in Eq. (15) is approximated using the mass matrix m obtained from the spectral element grid. The expression simplifies to a simple weighted sum of squares error given by

$$\mathcal{L}_{E,D} \approx \sum_j \sum_i \|\boldsymbol{u}(\boldsymbol{x}_i, t_j) - \hat{\boldsymbol{u}}_{NN}(\boldsymbol{x}_i, t_j)\|^2 w(\boldsymbol{x}_i) m(\boldsymbol{x}_i) \tag{16}$$

Snapshots of the true streamwise velocity, the FFNN-predicted streamwise velocity, and absolute error are shown in Fig. 7, which includes the same fields shown in Fig. 5 to illustrate POD reconstruction. As was the case with POD, error is higher in the wake than in regions close to the airfoil. This is unsurprising, given that the loss function used to train the network weighs error close to the airfoil more than that in the wake. Note that the maximum absolute error in the near airfoil region using neural networks is considerably lower than that of POD and that regions of high error are more localized. This indicates the superior performance of FFNN to POD for flow reconstruction in this particular case.

The choice to use a very large output layer of the neural network makes predictions specific to the particular grid used in training, though interpolation schemes could be used in other cases. The FFNN decoder's size also makes it highly memory-intensive, which limits batch size in training. It is plausible that similar networks for three-dimensional flows would require coordinate descent like algorithms where fractions of the output weights are updated on any given batch. This is in contrast to operator-type networks, where spatial coordinate x is given as input [35]. Methods based on the latter were implemented without physical constraints for flow reconstruction from pressure measurements but were found to underperform the full-field neural networks discussed in this work. We note that this could be in part due to using a neural encoder of pressure measurements rather than sparse function evaluations, as used in [35]. It is also possible that the use of physics-informed methods could improve prediction accuracy of the operator type networks, and this is noted as a potentially interested research question. However, we consider such an approach to be outside of the scope of this work.

B. Forecasting Aerodynamic Fluctuations

We now consider online methods for Eq. (5). We separate these methods into two broad categories; those that do not use dynamic models and those that use dynamic models such as the Navier–Stokes equations or data-driven dynamic models.

1. Nondynamic Methods

We first consider methods that do not employ any sort of dynamic model. These are simply interpolations of a function from the time history of an input quantity to the future value of $q(t)$. The general form is given by

$$q(t + \tau) = F_\bullet(\bullet_t) \tag{17}$$

where $\bullet \in [P, \psi, \xi]$ and F is a deep LSTM mapping some time series of historical measurements to the future quantity of interest. In the case where the reduced representation of the flow state is used, we have

$$\begin{aligned} q(t + \tau) &= F_\psi \circ G(P_t) \\ q(t + \tau) &= F_\xi \circ E(P_t) \end{aligned} \tag{18}$$

so the POD sensing network G and neural network encoder E may be considered as feature maps for the forecasting networks $F_{\psi/\xi}$. Each of $F_{P/\psi/\xi}$ are trained using the mean square error loss function. Other loss functions more specific to extreme events were considered but found to make negligible difference to the resulting trained network.

Nondynamic methods of the form given by Eq. (17) may be favorable for several reasons. They are simple, easy to train, and based on the ubiquitous and highly effective LSTM network structure. They may be of particular interest due to the computational

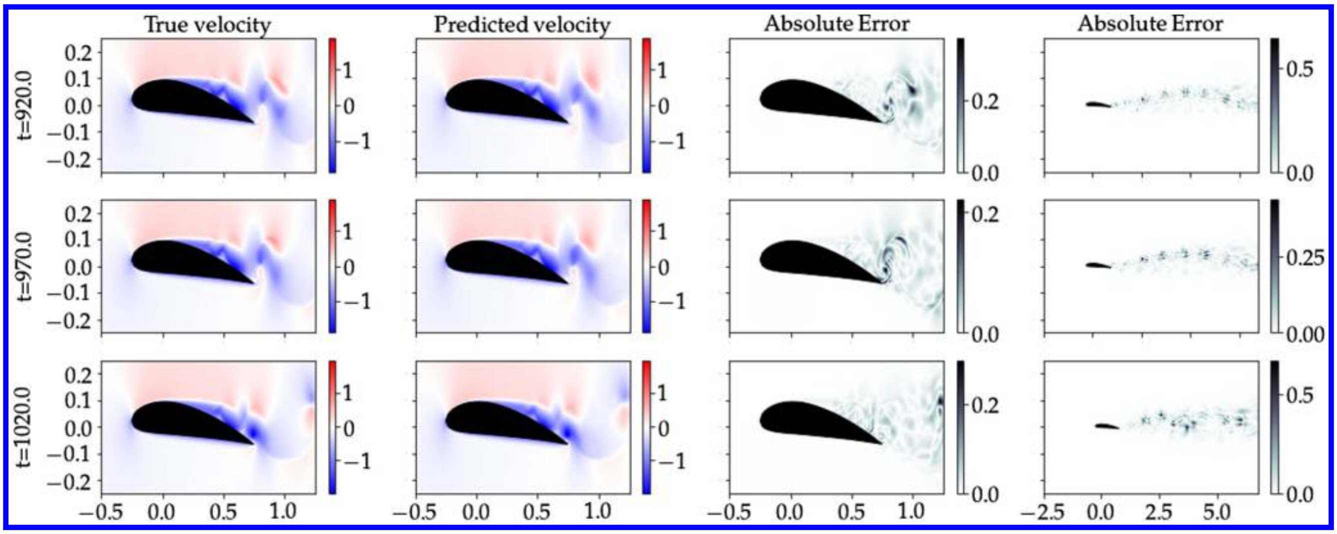

Fig. 7 [Operator $D \circ E(P_t)$] Reconstruction of streamwise velocity ($u_\infty = 1$ subtracted) using FFNN for flow compression with 32-dimensional latent space and LSTM to map the pressure measurements P_t to the latent variables ξ. Note maximum error is approximately half that of POD, and regions of high error are significantly more localized.

savings offered by avoiding dynamic models. Approximation of $q(t+\tau)$ may be rapidly computed from the state of the LSTM, which is updated online from streaming data. However there are also downsides to the nondynamic approach. As τ becomes large, we are approximating larger steps of a chaotic time series. The problem thus becomes very sensitive to inputs, and balancing sufficient model complexity with overfitting becomes challenging. Moreover, prediction of $q(t+\tau)$ for any given lead time τ requires its own trained network.

2. Full-Order Dynamical Model-Based Prediction Methods

Previous work predicting extreme events for turbulent flows has used a low-dimensional representation of the flow as an initial condition for a fluid simulation and employed the adjoint to evaluate the gradient of the future quantity of interest with respect to coefficients in the low-dimensional expansion [5]. This is possible in cases where the low-order initial condition is sufficiently close to the full-order state to track its behavior. Here, we evaluate whether this is the case for low-dimensional initialization from both the POD basis and FFNN. Numerical simulations are initialized using either the FFNN-based reconstructed velocity or the low-dimensional reconstruction using the known POD modes and singular values, along with the estimated temporal coefficients $\hat{\psi}(t) = G(P_t)$. Future values of $q(t)$ are computed using the same convolution as in Eq. (4):

$$\begin{aligned}\hat{q}(t+\tau) &= (\hat{C}_d(s;\hat{\boldsymbol{u}}) * K(s))(t+\tau)\\ \text{where } \hat{\boldsymbol{u}}(t) &= D(\xi) \text{ or } \hat{\boldsymbol{u}}(t) = \Phi\Sigma G(P_t)^T\end{aligned} \tag{19}$$

where $\hat{\boldsymbol{u}}(s), s \in [t, t+\tau]$ is found via numerical integration of the Navier–Stokes equations and K is as in Eq. (4). Since the time series for $\hat{C}_d$ is much shorter than the full simulation used for training neural networks, the common practice of zero padding distorts the values of $\hat{q}$ near the endpoints. We instead truncate and renormalize the smoothing kernel. For valid comparison, the same is done for the true drag coefficient truncated to the interval $[t, t+\tau]$. Following Eq. (19), time series for q and $\hat{q}$ are normalized using the mean and variance of the full-length time series defined in Eq. (4).

3. Data-Driven Dynamical Model-Based Prediction Methods

We also consider forecasts of either 1) the pressure signal $P(t)$ or 2) the reduced-order state of the fluid flow, using data-driven dynamic models. In the latter case we formulate and evaluate reduced-order dynamical models for both the reduced POD state and the latent representation learned by the FFNN. For the POD reduced order model (ROM), we train the dynamic model on the estimated POD coefficients $\hat{\psi} = G(P_t)$, instead of the true coefficients ψ, since the former may be found exactly from airfoil surface pressure. This eliminates error due to imprecise initial conditions.

Forecasting models in each case are LSTM networks mapping historical measurements of a given quantity $(P/\hat{\psi}/\xi)$ to its value κ timesteps in the future. They are represented as

$$P(t_{j+\kappa}) = H_P(P(t_j), P(t_{j-\kappa}), \dots)) \tag{20}$$

with similar networks for $\hat{\psi}$ and ξ. Here κ is taken to be three in order to alleviate some of the numerical difficulties with training data-driven dynamic models in the small time-step limit [38]. Thus, the LSTM maps historical measurements of the dynamic quantity P to the value of that quantity three steps in the future: $P(t_{j+3})$. Since P is measured every $dt = 0.01$ time units, the LSTM is effectively a dynamic model for P with time step equal to $\kappa dt = 0.03$. Networks are trained using the mean square error loss over a prediction window of 20 steps:

$$\mathcal{L}_{H_P} = \sum_i \sum_{j=1}^{20} \|H_P^j(P_i) - P(t_{i+\kappa j})\|^2 \tag{21}$$

where we define composition of H_P with itself by

$$\begin{aligned}H_P^1(P_i) &= H_P(P(t_i), P(t_{i-\kappa}), P(t_{i-2\kappa}), \dots),\\ H_P^2(P_i) &= H_P(H_P^1(P_i), P(t_i), P(t_{i-\kappa}), \dots),\\ H_P^3(P_i) &= H_P(H_P^2(P_i), H_P^1(P_i), P(t_i), \dots)\end{aligned} \tag{22}$$

The sum over index i in Eq. (21) is taken over all initial times in the training dataset. The same network structure and loss function is used for the full set of 50 pressure measurements as well as the 32-dimensional reduced-order representations of the flowfield. Layer sizes are scaled to account for the difference in dimension between P and ψ/ξ.

For any $\tau \geq 0$, we obtain the estimated forecast

$$\hat{P}(t+\tau) = H_P^{m_\tau}(P_t) \tag{23}$$

where $m_\tau = \tau/(\kappa dt)$. Following the LSTM-based forecast of $P/\hat{\psi}/\xi$, we may use $F_{P/\hat{\psi}/\xi}$ trained for zero lead time to evaluate $\hat{q}(t+\tau) = F_P \circ H_P^{m_\tau}(P_t)$ and likewise for $\hat{\psi}/\xi$. Since prediction of q with zero lead time is a much simpler problem, we use standard feed-forward neural networks in place of the LSTMs used for nondynamic predictions with nonzero lead time.

C. Error Metrics

Each of the neural networks in this work is trained using the mean square error (MSE). The MSE is differentiable and may be evaluated on subsets of the training data, allowing for the use of stochastic optimization schemes run on a graphics processing unit. However, the MSE may not be a good indicator of success in predicting extreme events. We have therefore adapted several extreme-event-tailored error metrics to compare the various methods considered in this work. Specifically, results are compared using the batch relative entropy loss, the maximum adjusted area under the precision-recall curve, the extreme event rate dependent area under the precision recall curve, and the optimal F_1 score.

The batch relative entropy (BRE) loss is inspired by the work in [7], where authors use a relative entropy loss function to train convolutional neural networks that are capable of making accurate predictions of a system governed by the truncated Korteweg–de Vries (tKdV) equation in regimes with extreme events. Their work uses empirical partition functions similar to the soft-max activation commonly used in neural networks to transform high-dimensional predictions into probability distribution functions highlighting outlier values. Loss is subsequently measured loss using the Kullback–Leibler-divergence. This approach was shown to significantly improve prediction accuracy over the MSE on the tKdV problem. A similar approach was adapted for the present work using partition functions over minibatches rather than output dimensions. Specifically, we define the batch relative entropy loss as

$$\text{BRE} = \sum_i z_i \log\left(\frac{z_i}{\hat{z}_i}\right) \tag{24}$$

where z_i, $\hat{z}_i$ are given by the empirical partition functions

$$z_i = \frac{e^{q(t_i)}}{\sum_j e^{q(t_j)}}, \qquad \hat{z}_i = \frac{e^{\hat{q}(t_i)}}{\sum_j e^{q(t_j)}} \tag{25}$$

and where the sum is taken over a minibatch. We note that this work considers partition functions that weigh positive outliers, as opposed to the symmetric variant used in [7], since all events of interest to this work skew positive. The BRE loss was tested as a means of training the neural networks described in previous sections, but taken over minibatches was found to perform comparably with the mean square error. For evaluation, we use the batch relative entropy loss taken over the full testing dataset.

The performance of a predictor of extreme events may also be measured by the ability of that predictor to classify events based on a threshold value of the quantity of interest. The maximum adjusted area under the precision-recall curve (α^*) was proposed in [6] as a loss function for predicting extreme events and shown to perform well for predicting extreme dissipation events in Kolmogorov flow and rogue waves in the Majda–McLaughlin–Tabak model. To define α^*, we first introduce the quantity ω for the extreme event rate and the corresponding threshold $\hat{a}(\omega)$, such that $p(q > \hat{a}(\omega)) = \omega$. Introducing a second threshold $\hat{b}$ for the prediction $\hat{q}$ yields a classifier for which we can compute the precision (S = true positives divided by predicted positives) and recall (R = true positives divided by total positives). Noting that precision is uniquely determined by the extreme event rate and recall, [6] computes the area under the precision recall curve

$$\alpha(\omega) = \int_0^1 S(R,\omega)\,\mathrm{d}R = \int_{\mathbb{R}} S(\hat{b},\omega)\left|\frac{\partial R(\hat{b},\omega)}{\partial \hat{b}}\right|\mathrm{d}\hat{b} \tag{26}$$

and the maximum adjusted area under the precision-recall curve as

$$\alpha^* = \max_{\omega}\alpha(\omega) - \omega \tag{27}$$

We compute the integral in Eq. (26) using a finite grid of $\hat{\beta}$ values spread evenly between $\min(\hat{q})$ and $\max(\hat{q})$ with finite difference scheme to evaluate the derivative term. Its value is bounded between 0 and 1 but occasionally falls slightly outside this range due to numerical issues and is subsequently clipped. The maximization in Eq. (27) is taken over a discrete set of samples evenly spaced ω between 0.01 and 0.25.

Finally, we consider the extreme event rate dependent optimal F_1 score, defined as the F_1 score on the testing dataset using the threshold that optimizes the F_1 score on the training and validation datasets. That is,

$$F_{1,\mathrm{opt}}(q,\hat{q},\omega) = F_1((q_{\mathrm{test}} > \hat{a}(\omega)), (\hat{q}_{\mathrm{test}} > \hat{b}_{\mathrm{opt}})) \tag{28}$$

where

$$\hat{b}_{\mathrm{opt}} = \arg\max_{\hat{b}} F_1((q_{\mathrm{train/val}} > \hat{a}(\omega)), (\hat{q}_{\mathrm{train/val}} > \hat{b})) \tag{29}$$

Taken together this yields five metrics of predictor performance. Mean square error, batch relative entropy, and α^* are independent of extreme event rate and yield simple scalar metrics of performance, though they do not indicate performance at a particular extreme event rate. The extreme event rate dependent $\alpha(\omega)$ and $F_{1,\mathrm{opt}}(\omega)$ each seek to measure a balance between precision and recall at variable extreme event rates. The dependency on extreme event rate allows for a more descriptive quantification of error, since we can compare methods for a variety of extreme event rates.

IV. Results

In this section we present results for each of the methods for Eq. (5). Results for prediction using the full-order model with reduced-order initial condition are kept distinct from those using neural networks. This is due to the weak performance by the former, as well as its considerable computational expense, which limits the number of trials we use to evaluate it. For predictions of $q(t)$ using dynamic methods, we apply a smoother to remove higher frequency oscillations. This smoother weights previous predictions with exponentially decreasing importance and does not use any future prediction beyond τ.

A. Simulations with Reduced-Order Initial Conditions

We first present results showing the failure of approaches taken by the authors to forecast aerodynamic fluctuations using the Navier–Stokes solver and reduced-order initial condition. For each of a variety of ranks, reduced-order initial conditions were formed using the estimated POD time series $\hat{\psi}(t)$ and FFNN flow reconstruction $D \circ E(P_t)$ at 50 evenly spaced times throughout the portion on data reserved for testing. Examples of the resulting smoothed drag coefficient are shown in Fig. 8, which compares predicted time series for $\hat{q}(t)$ using several ranks of POD reconstructions (left column) as well as FFNN-based reconstructions (right column) to the true value $q(t)$ (black curves). Each row represents a different initial condition from within the testing dataset. We note that all the initializations using different reduction methods/orders exhibit poor agreement with the true time series in at least one of the three cases shown.

The mean absolute errors in the drag coefficient and smoothed drag coefficient q for each rank are shown in Fig. 9. There are several noteworthy features; the approximated values of C_d for both methods are initially fairly accurate, falling within 10% of the true values, but very quickly diverge. By the end of the 10.5 time unit simulations, the error in each quantity is roughly as large as the standard deviation of the true data. Initial error in $q(t)$ is higher, as is expected since this quantity includes some information from future estimates of C_d. We also note the lack of noticeable correlation between the rank of the latent space used for flow reconstruction and prediction accuracy. In the POD case, it appears that the lowest rank $r = 8$ performs poorly compared to higher ranks, but $r = 64$ is not uniformly better than $r = 16$ or $r = 32$. A plausible explanation for

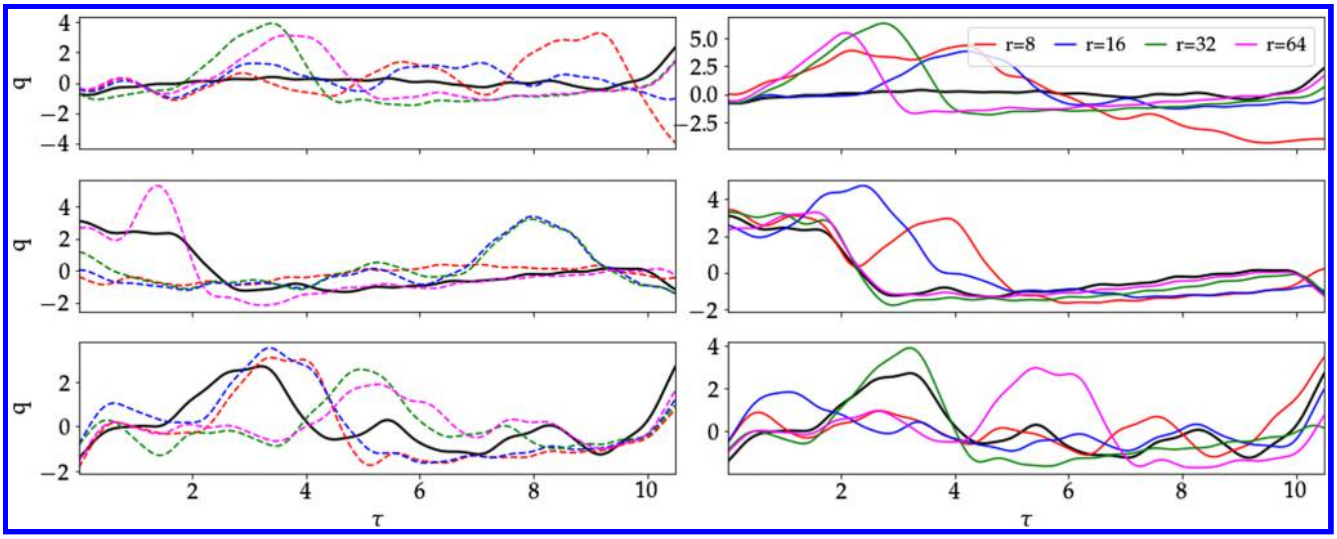

Fig. 8 Examples of forecasts of $q(t+\tau)$ using Eq. (19) with various ranks of POD (left column) and FFNN (right column). Examples show prediction from initial conditions are spaced evenly throughout the testing dataset. True value is shown as solid black line. Each row represents a different realization of the flow.

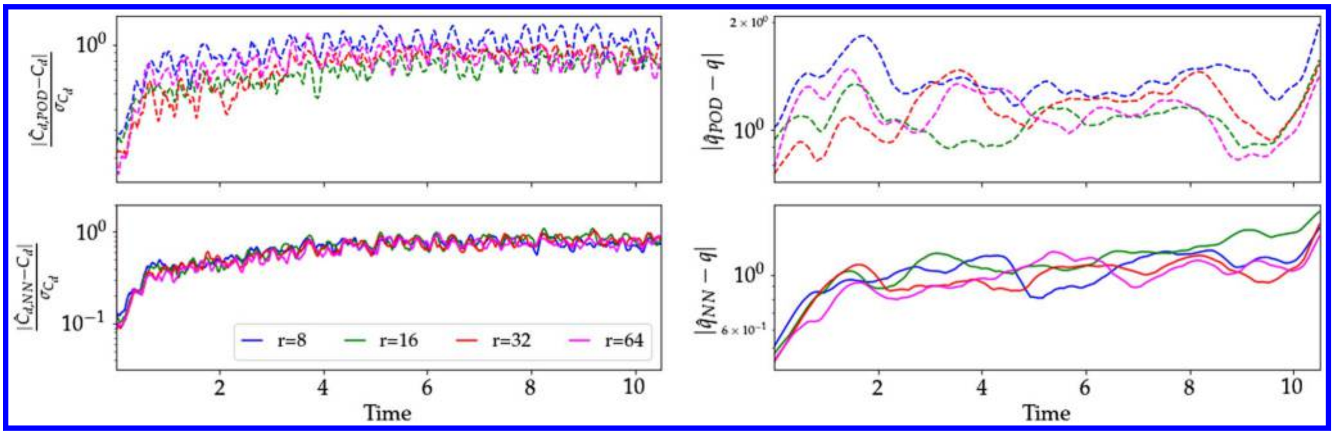

Fig. 9 **Mean absolute error, normalized by standard deviations of true time series for the drag coefficient and $q(t)$ using various ranks of reduced-order initial conditions from the estimated POD time series and FFNN.**

this is that as the rank is increased, so too is the difficulty of the sensing problem outlined at the end of Sec. III.A.1. Indeed, using the true (nonreconstructed) $\psi(t)$, we see improved agreement with full-order results if r is let to become large. The connection between rank and accuracy for the FFNN examples is more opaque. While higher-rank initial conditions are slightly more accurate at predicting the initial drag, the difference is small and decays quickly as the simulation progresses.

While the flow reconstruction methods presented in this work were found to be inadequate for initializing a simulation capable of capturing bursts in drag, it is plausible that future work may find methods able to do so. However, this problem will likely become even more challenging at higher Reynolds numbers or in three dimensions. We include them to show the sensitivity of the problem to small changes in the initial condition (note that the FFNN was able to accurately reconstruct the fluid velocity) and to contrast to other examples in fluids where a representation in low rank basis was found to be an effective precursor to extreme events [5].

B. Comparison of Reduced-Order Dynamic Methods

We now consider the data-driven dynamic models for the pressure signals and reduced-order representations of the flowfield. Figure 10 shows the time evolution of the mean square error in the dynamic quantity and absolute error in prediction of q for each of the three methods $H_{P/\psi/\xi}$. The dynamic quantities in each case have been normalized to unit variance, and so direct comparisons of error magnitudes across methods and across indices are meaningful. Error statistics have been binned for each timestep and normalized such that vertical slices of any subplot are density functions across all examples from the testing dataset. Apparent recurring features in the error plots for q are likely due to the methods missing the same feature from multiple closely sampled initial conditions.

In each of the three models, the bulk of the error remains low throughout the forecast window, as is shown by the curves for mean error. Error statistics for the dynamic model for P remain low for approximately three time units before some trajectories diverge. Outlier errors for the dynamic model for ψ grow more rapidly than others initially and by the end of the 10.5 convective unit interval are noticeably larger, on average, than the other two. Mean error in the dynamic quantity, shown by red curves in the left panel of Fig. 10, is lowest for H_ξ, with H_P having slightly larger error and H_ψ considerably larger than H_ξ and H_P. Shading of error density indicates more positive outliers in the dynamic quantity error for H_P than H_ξ, especially at lower lead times. Differences in mean absolute error for prediction of q, shown on the right-hand side of Fig. 10, are less pronounced, with H_ξ still the most accurate.

For the models in P and ψ, where indices in the dynamic variable carry meaning, it is also instructive to see what variables accumulate error at what rates. Figure 11 shows the mean square error for each index in each dynamic variable taken across test set examples.

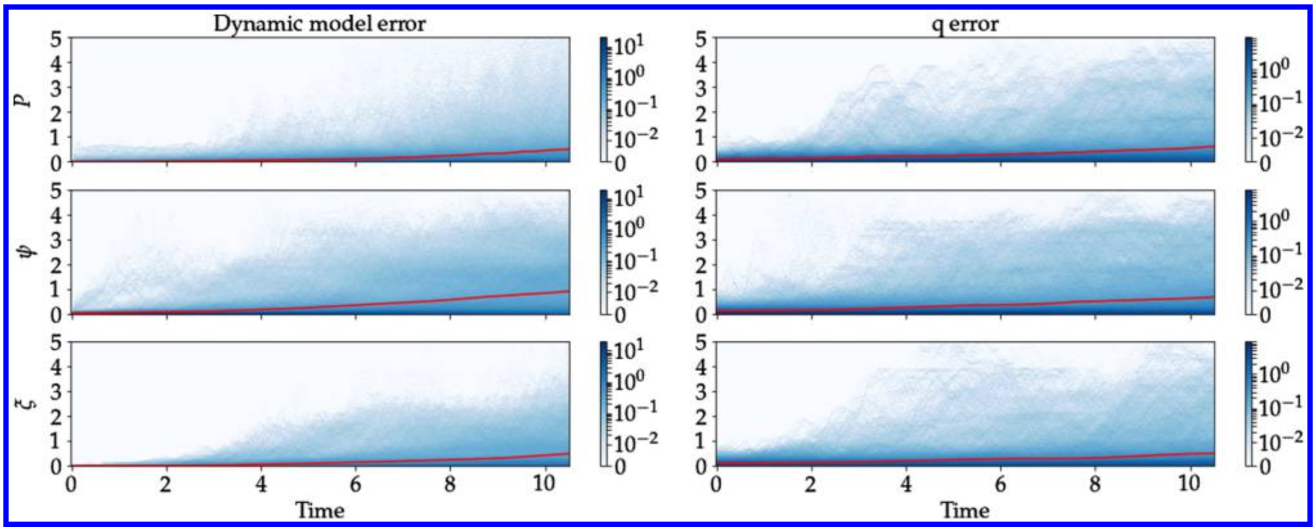

Fig. 10 **Test set empirical probability density functions of the mean square error of the LSTM forecast (left) and absolute error in predicting $q(t+\tau)$ for 50-dimensional data-driven forecast of pressure measurements (top), and 32-dimensional reduced-order models using POD modes (middle) and FFNN latent space (bottom). Mean error quantities are shown by red curves.**

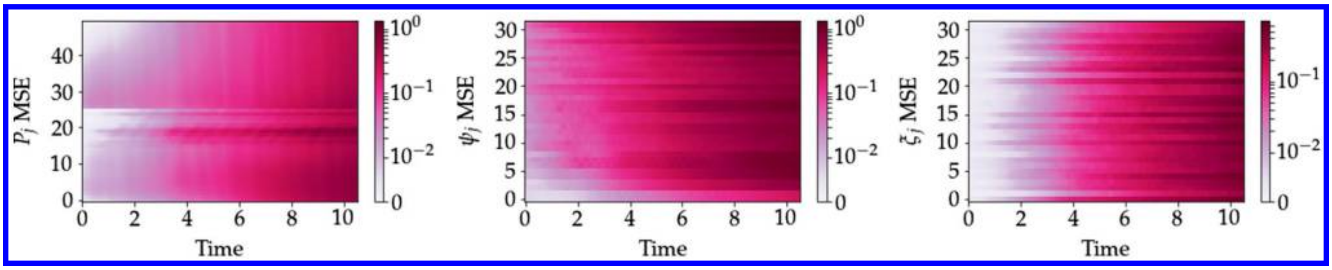

Fig. 11 Time evolution of mean (across test set samples) square error for each of 50 pressure sensors (left), 32 POD modes (middle), and dimensions of FFNN latent space (right) using LSTM-based dynamic models.

The sharp discontinuity in the error for pressure sensors is a consequence of their position on the airfoil, shown in the lower panel of Fig. 1. The first 26 pressure sensors are equally placed on the top of the airfoil from front to tail including endpoints, and the subsequent 24 are evenly spaced from front to tail on the lower side. We see that error towards the tail of the airfoil on both the suction and pressure sides is initially lower than towards the front. However, as time progresses, there is considerable error in pressure sensors towards the rear of the suction edge of the airfoil.

Error in the low-order POD modes is lower than in the high-order, less energetic, modes. This may be explained by the fact that these low-order modes tend to track large-scale features of the flow and have slower and less chaotic trajectories. It is worth noting that, while the left-hand column of Fig. 10 makes clear that error in ψ grows much more rapidly in the initial forecast than P or ξ, the same is not as obvious for low τ prediction of $q(t+\tau)$ using the dynamic model for $\hat{\psi}$. This suggests that the value of q may largely be a function of the low-order energetic POD modes, which Fig. 11 shows are accurately tracked for longer lead times. Indices for the FFNN latent space variable ξ do not have meaning, and there is no correlation between index and the rate at which error increases.

C. Forecasting Aerodynamic Fluctuations

In this section we compare the six neural network-based prediction strategies using the metrics outlined in Sec. III.C. Examples of predictions for lead time $\tau = 7.0$ using all methods are shown in Fig. 12. Blue curves indicate true values $q(t)$ and red dashed curves show predictions with lead time $\tau = 7.0$ using the three nondynamic and three dynamic methods. Plots for other lead times are shown in Appendix E.

Scalar values (MSE, α^*, and BRE) of prediction accuracy for each method and lead times τ ranging from 0 to 10.5 convective time units in intervals of $0.7 \approx 1/f_{\text{peak}}$ are shown in Fig. 13. Metrics for prediction via nondynamic methods are shown as solid lines, and those for dynamic models are shown as dashed lines.

Figure 13 clearly indicates some qualitative differences between dynamic and nondynamic methods. In particular, error statistics for the dynamic models at a particular value of τ are highly correlated with those for similar lead times. The curves indicating their performance as a function of lead time are therefore smooth. The same is not true for nondynamic models, where predictions for each lead time are performed via their own trained neural network. It is plausible that more care could be taken to yield consistency across lead times for

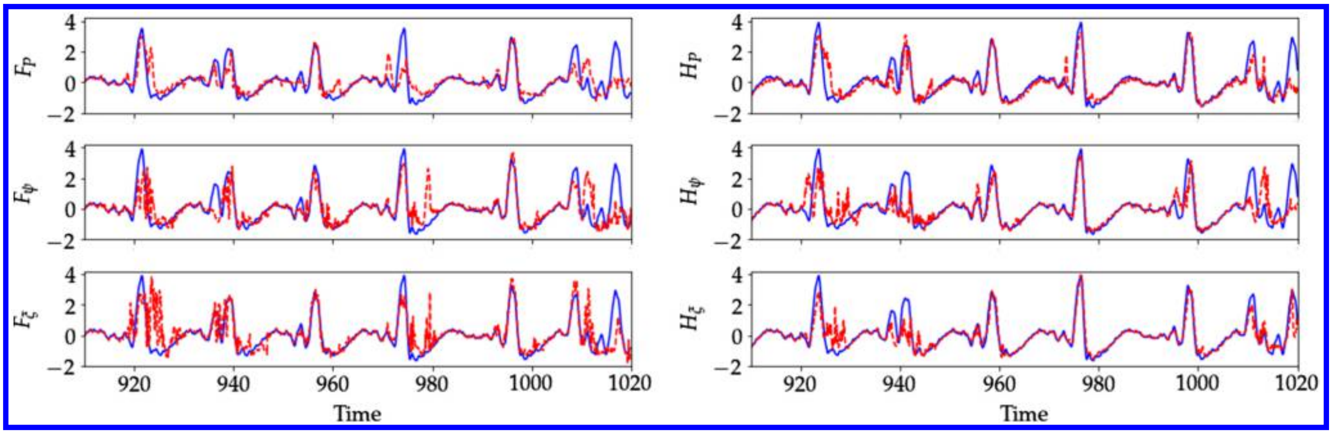

Fig. 12 True (blue) and predicted (red) time series for $q(t)$ with lead time $\tau = 7$.

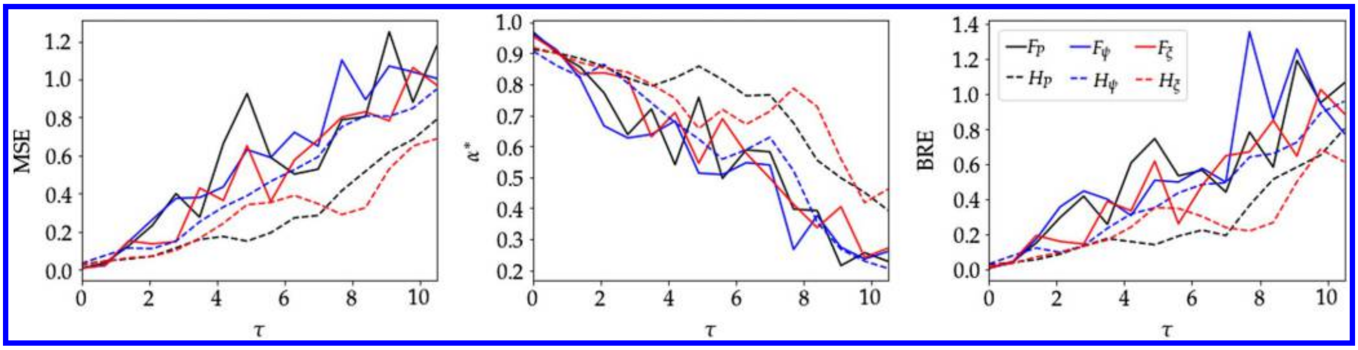

Fig. 13 Mean square error, maximum adjusted area under the precision-recall curve (α^*), and test set relative entropy for each of the six neural network-based prediction methods. Evaluation is performed on partition of data reserved for method comparison (test set).

nondynamic models, but methods for doing so are not immediately apparent.

The scalar error metrics also clearly show that the dynamic models in P and ξ, H_P and H_ξ are the most accurate of the six models. Differences between the other four are less pronounced, though H_ψ does, on average, slightly outperform nondynamic methods. This difference is most notable in the MSE though still apparent for the extreme event-specific metrics. This suggests that a nonnegligible portion of the improvement H_ψ has over nondynamic methods manifests in the nonextreme values of q, though the scalar valued metrics do not resolve this feature.

Values for the remaining extreme event rate dependent evaluation metrics are shown in Fig. 14 for values of ω between 0.01 and 0.25. The top row shows values for $F_{1,\mathrm{opt}}$, and the bottom row shows the area under the precision-recall curve. Within each subplot, values towards the bottom of the image indicate the prediction accuracy of the model for classifying very rare events (1%) while those at the very top show more common events, with lead time increasing across the horizontal axis. While differences in values of prediction accuracy using Fig. 14 may not be immediately clear, the same qualitative features seen in Fig. 13 are again apparent. In particular, errors for the dynamic methods are much smoother in time than those of the nondynamic methods.

Differences between methods become much more apparent when plotted directly. Figure 15 shows the differences in values between all methods for both $F_{1,\mathrm{opt}}$ (left) and the area under the precision-recall curve α (right). Values within each subfigure indicate the metric evaluated with the method on the corresponding row minus the metric for the method on the corresponding column. Hence, blue (negative) indicates the method corresponding to that column performed better while red (positive) indicates the method corresponding to the row is better.

Figure 15 shows H_P and H_ξ outperforming other methods across all lead times τ and extreme event rates ω except for a small number of data points clustered around $\tau = 4$ and low ω. Comparisons between other methods are less easily summarized. There do not appear to be discernible trends in the comparison between nondynamic methods using P, ψ, or ξ as input. From this it seems reasonable to surmise that the accuracy of nondynamic methods is not improved in a meaningful manner by exploiting flowfield data. The dynamic model for ψ outperforms nondynamic methods for midrange lead times and lower extreme event rates but underperforms in prediction of less rare

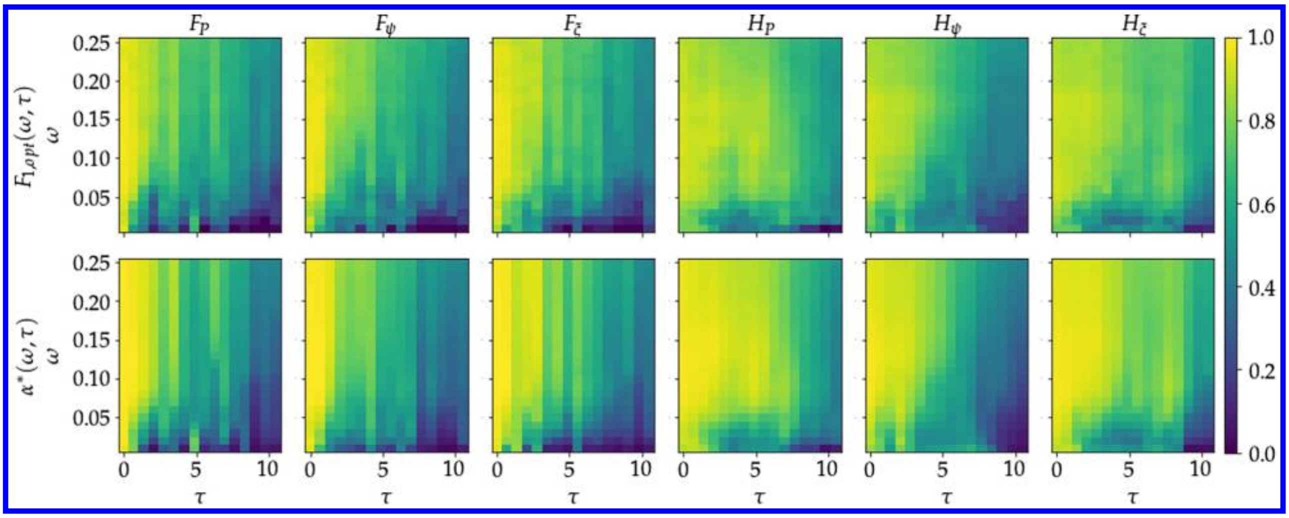

Fig. 14 Extreme event rate dependent area under the precision-recall curve ($\alpha(\omega)$ and $F_{1,\mathrm{opt}}$) for each method evaluated on test set.

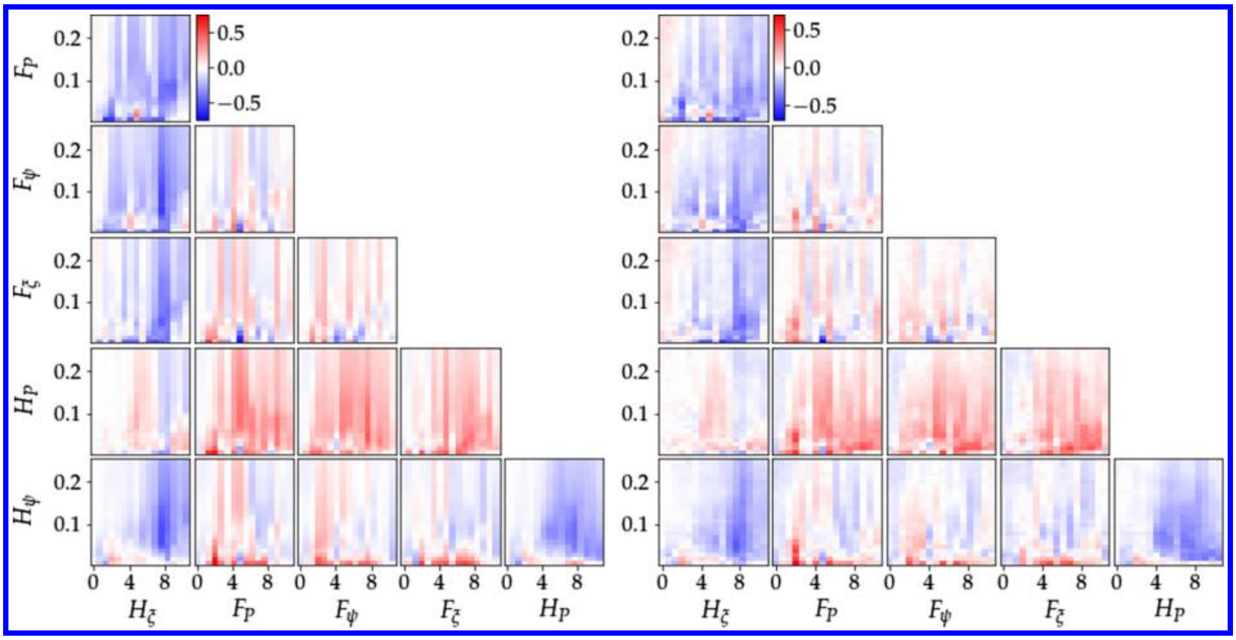

Fig. 15 Comparison of α (left) and $F_{1,\mathrm{opt}}$ (right) across various methods, lead times τ, and extreme event rates ω. Each plot shows $\alpha/F_{1,\mathrm{opt}}$ for method corresponding to row minus $\alpha/F_{1,\mathrm{opt}}$ for method corresponding to column. Blue (negative) indicates column method outperforms row method. Hence, lower left plot indicates H_ξ has higher α (favorable performance) than H_ψ in most cases.

events (higher ω) at longer lead times. Differences between methods are largely consistent between $F_{1,\mathrm{opt}}$ and α.

V. Conclusions

The authors have investigated the feasibility of a variety of methods for forecasting aerodynamic fluctuations occurring in a simulation of two-dimensional incompressible flow around a NACA 4412 airfoil using only pressure recordings along the surface of the airfoil. Some of these methods also exploited, in an offline manner, knowledge of the flowfield to construct low dimensional representations that were either used as inputs to machine-learned predictors or variables in dynamic models. Comparison between the methods considered in this work suggests that the complexity of the flow, though low Reynolds and only two-dimensional, precludes the use of a full-order computational fluid dynamic model with low dimensional input. Other methods were largely comparable, except for the learned dynamic models of the pressure measurements and of the latent variable of the FFNN-based flow reconstruction, which performed notably better than others.

The authors highlight that neural networks allow for tremendous freedom with regards to structure, hyperparameters, regularization, and other factors that may affect performance. Several architectures have been tested and the best result selected for this paper, but the authors acknowledge that their work falls considerably short of testing across all sizes and training procedures. Doing so would require extensive computational resources, and results would have no guarantee of generality beyond the problem specifically discussed in this work. Rather, they sought to investigate the use of various classes of models (nondynamic, ROM, full-order simulation) and representations of data (POD modes, FFNN) for the purpose of forecasting a quantity exhibiting extreme events. They also note that the work contained in this manuscript has been done without the inclusion of artificial noise, as is common in works applying machine learning to synthetic datasets. The authors suspect that including noise would not change the results in a meaningful way, since flow reconstruction methods have been shown to be robust to noise [16,37] and quantities with dynamic models may be estimated with filtering. Nonetheless, further work studying the effect of noise would be necessary to confirm this.

In Sec. IV it was suggested that low-order initializations of the Navier–Stokes solver failed to accurately predict fluctuations. This motivated the use of data-driven dynamic models. This is not surprising, given the complexity and nonperiodic nature of the flow. It is possible that a mixed strategy of projecting the governing Navier–Stokes equations onto POD modes coupled with an LSTM closure model would outperform both approaches, as suggested in [38]. This approach, however, would be highly nontrivial when using the latent space of the FFNN. Given a sufficiently expressive latent space and decoder, it may be possible to use a physics-based nonlinear reduced-order model for ξ such as [39]. This approach may be an interesting direction for future research.

The work included in this manuscript is based on numerical simulations of an airfoil, and it is worth noting differences that may occur in an application setting. Numerical simulation for the current work was implemented with a blockage ratio of 3.18%. This falls within the range that might be expected for experiments on bluff bodies [40,41] but is higher than some works studying the aerodynamic properties of airfoils. Several works have noted dependence of aerodynamic properties including the drag coefficient, Strouhal frequency, and certain critical Reynolds numbers on the blockage ratio [42]. This effect was observed in the case used for the present work but not explored in depth. The authors also note that the simulation is of two-dimensional flow over a smooth airfoil, which exhibits qualitative differences from three-dimensional simulations and experiments [15]. Thus, the present work should be considered as a study of a computational model of prototypical flow phenomena rather than experimental or application conditions. Nonetheless, the present work provides a basis for understanding the utility of various approaches towards predicting intermittent events in the flow around an airfoil.

Extension of the present work to higher Reynolds and three-dimensional cases will require addressing several further challenges. Simulation of flow at either higher Reynolds or in a three-dimensional setting will require a larger computational grid and exhibit higher intrinsic dimensionality. The output dimension for the FFNN-based flow reconstruction scales with grid size and therefore will be significantly larger. It is plausible that the resulting computational challenges may be mitigated by learning a coarsened solution, using multiple networks, or using a single network with a large memory footprint. While not found to be successful in this work, operator-type networks may be adapted using recent advances such as integration of physics-informed loss functions [43]. Reconstruction in a POD basis will also require a larger number of modes, challenging the task of pressure to POD network G. The authors note, however, that dynamic models used in this work ($H_{P/\psi/\xi}$) are non-Markovian. Such models have been shown to allow for data-driven forecasts even in the case of unresolved modes [44,45]. It is plausible to expect the current framework to generalize to more complicated dynamics, though lead time for accurate forecasts will likely diminish.

The present work considered the case where pressure measurements are taken at 50 positions around the perimeter of the airfoil. This is, of course, not practical, and future work could consider the use of various sensor placement techniques for determining optimal placement on the airfoil with respect to forecasting fluctuations [18,22].

The authors believe this work establishes compelling comparisons and baselines for the prediction of extreme events using measurements on the surface of an airfoil. Numerous techniques, including several adapted from other works on extreme events, were tested and compared. The results suggest that the use of data-driven dynamic models for quantities subsequently used to predict extreme events outperform those that ignore dynamics. This work also provides numerous opportunities for future research. In particular, sensor placement and uncertainty quantification are critical elements of practical engineering that may be explored in the context developed in this work.

Appendix A: Numerical Simulation Details

Numerical simulations of the airfoil used in this work were performed using the open-source spectral element solver Nek5000 [26] and run on Expanse at the San Diego Supercomputing Center using a grant through the National Science Foundation-funded Extreme Science and Engineering Discovery Environment (XSEDE) [46]. The spectral element method, proposed in [47], partitions the computational domain into nonoverlapping elements using polynomial interpolation within each element to represent the solution. Mesh was generated using gmsh [48].

Time integration was performed using a second-order semi-implicit scheme described.[‡] The operator-integrating-factor splitting method proposed in [49] was used to allow for stable time integration with a fixed timestep of 0.001. The filtering method proposed in [50] was also used to stabilize the simulation.

The simulation was initialized with a velocity of $\boldsymbol{u} = (1, 0)$. This led to a short transient, which was avoided when training neural networks and taking the proper orthogonal decomposition. The inflow boundary condition was held at a fixed value of $\boldsymbol{u} = (1, 0)$ throughout the simulation. Boundary conditions on the wall of the airfoil were no-slip (Dirichlet) and along the top and bottom on the domain were symmetric. The outflow boundary condition was of the convective type proposed in [27]. This allowed for the passage of strong vortices out of the domain while avoiding numerical issues.

Simulation for 1020 convective units took approximately 3.5 hours running on 128 cores or approximately 5 days running on 16 cores on the author's local computer. Aerodynamic quantities are saved every timestep, and pressure at discrete points on the surface of the airfoil is saved every 10 steps, or 0.01 time units. Velocity and pressure data on the full domain are saved every 0.25 time units. Codes for recreating datasets, as well as files for time series of

---

[‡] https://www.mcs.anl.gov/~fischer/nek5000/oifs.pdf [11 August 2021].

pressure and aerodynamic coefficients, are available online at https://github.com/snagcliffs/Airfoil_EE.

Appendix B: Further Details on Intermittent Events

In this appendix, we provide a closer look at the behavior of the flow around the airfoil and the nature of the intermittent bursts in the aerodynamic coefficients. Intermittency, characterized by short-term "bursts" that interrupt more regular behavior, is a common feature of dissipative dynamical systems in transitional regimes between regular and chaotic behavior [51]. In this work, we observe intermittency in the behavior of the vortex shedding behind the airfoil, resulting in characteristic bursts in the drag coefficient. To distinguish these events from periodic or near-periodic behavior, we include in Fig. B1 return times of both the drag coefficient C_d and smoothed drag coefficient q for various exceedance thresholds. Scattered values of return times are shown as red dots along the x axis of each subfigure. Kernel density estimation with tenfold cross validation for kernel bandwidth has been used to estimate densities for each distribution of return times, which have also been shown in blue. Return times for C_d less than $2/f_{\text{peak}}$ are not included, as these are considered to be within the same event.

From Fig. B1, we see that there appear to be clusters of return times for lower exceedance threshold ($\mu + 1.5\sigma$). These may be seen on the figure for C_d at return times equal to approximately 2 and 16. However, many values remain scattered in a continuous range between clusters. A similar behavior is observed on return times of q, though clustering is less pronounced and not captured by the estimated density function.

Behavior of the drag coefficient q and flowfield in an interval of time spanning a burst are shown in greater detail in Fig. B2. Note that while q is defined using a smoothing operation on C_d, it is also centered and normalized. Snapshots of the vorticity field are shown at nine points within the interval that are marked on the time series of C_d as red dots. Note regular vortex shedding and periodic drag coefficient up to approximately $t = 488$, followed by irregular behavior and burst in C_d and vortex shedding up till approximately $t = 498$ and the subsequent return to regular behavior.

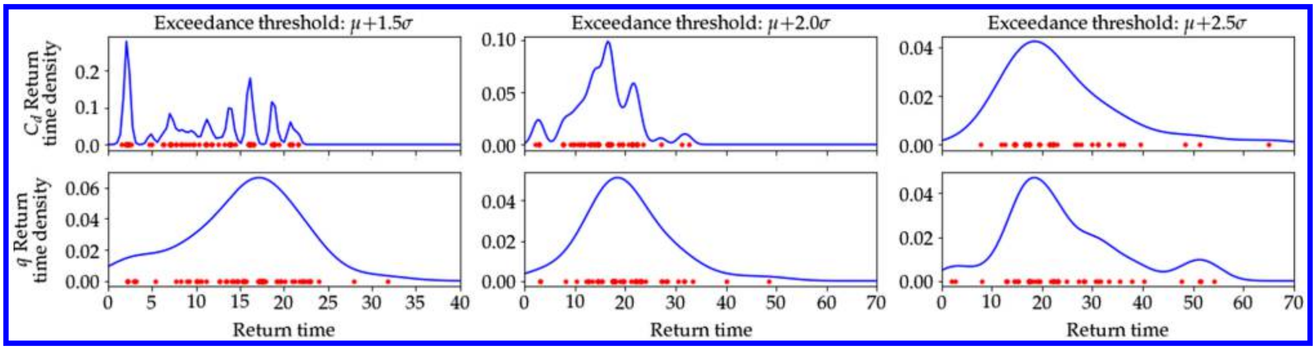

Fig. B1 Return times and estimated densities for C_d and q at various threshold values. Return time samples are shown as red dots, and kernel density estimate with tenfold cross validation to find Gaussian kernel bandwidth is shown in blue. For C_d, return times shorter than two characteristic shedding cycles have been discarded.

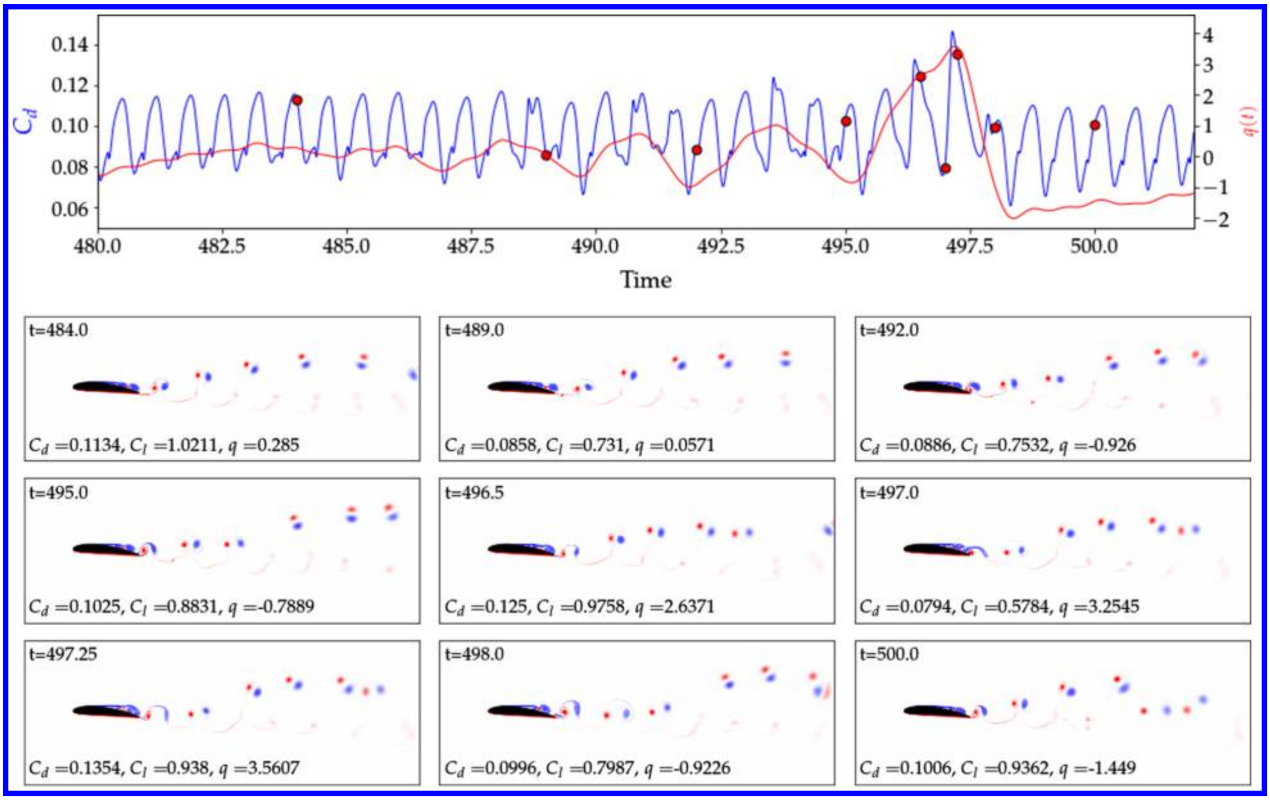

Fig. B2 Behavior of drag coefficient C_d, smoothed drag coefficient q, and flow preceding and during burst. Top: time series for $C_d(t)$ (blue) and $q(t)$ (red) preceding and during burst. Note that q has been centered and normalized. Scattered points indicate time where snapshots of vorticity are presented. Bottom: Snapshots of vorticity at times indicated in time series.

Appendix C: Neural Network Structures and Implementation

All neural networks used in this work were implemented in Python using the TensorFlow library [52]. The swish activation function [53] was used between fully connected nonrecurrent layers, and the sigmoid function was used in LSTM layers. In Sec. III.B.3 we note that prediction of q following LSTM dynamic models is performed using nonrecurrent networks. This step, which does not forecast any quantity, is a fairly trivial problem, and details of networks have been omitted here but are available in the online code repository.

C.1. Structure

Structures for neural networks used in this work are given in Eq. (C1). We use the notation (input shape) → (layer type, layer sizes) with multiple integer layer sizes indicating repeated layers. Hence, the first line tells us F_p uses 140 history points of dimension 50 as input, then a 32-dimensional fully connected layer, two LSTM layers of size 32, and finally a series of fully connected layers having sizes 32, 16, 8, 4, and 1. The final layer of size 1 is the output layer. For networks including a latent representation of the dynamics, the quantity r has been used as a stand-in for rank. This work includes examples having $r \in \{8, 16, 32, 64\}$. The output size of E is the number of interpolation points in the spectral element grid $n = 279{,}552$, multiplied by the dimension $d = 2$, for a total size of $n \cdot d = 559{,}104$. In all cases using history terms, a stride of three was used between inputs. This was found to yield a slight performance advantage over using a stride of one. Connections in the decoder network, excluding those to the final large layer we modeled after residual type networks, having a linear connection added to those feeding into nonlinear activation functions. A larger number of history points were used for F_p to account for the history terms used to compute $\hat{\psi}$ and ξ, though this was not found to significantly affect results:

$$\begin{aligned}
&F_p\colon (50\times 140) \to (FC, 32) \to (\mathrm{LSTM}, 32, 32) \to (FC, 32, 16, 8, 4, 1)\\
&F_\psi\colon (r\times 70) \to (FC, 32) \to (\mathrm{LSTM}, 32, 32) \to (FC, 32, 16, 8, 4, 1)\\
&F_\xi\colon (r\times 70) \to (FC, 32) \to (\mathrm{LSTM}, 32, 32) \to (FC, 32, 16, 8, 4, 1)\\
&G\colon (50\times 70) \to (FC, 64) \to (\mathrm{LSTM}, 128) \to (FC, 64, r)\\
&E\colon (50\times 70) \to (FC, 64) \to (\mathrm{LSTM}, 64) \to (FC, 64, r)\\
&D\colon (r\times 1) \to (FC, 64, 128, 256, n\cdot d)\\
&H_p\colon (50\times 70) \to (FC, 50) \to (\mathrm{LSTM}, 100, 100) \to (FC, 100, 50, 50)\\
&H_{\psi/\xi}\colon (r\times 70) \to (FC, 32) \to (\mathrm{LSTM}, 64, 64) \to (FC, 64, 32, 32)
\end{aligned} \tag{C1}$$

As noted in Sec. V, designing neural networks allows tremendous freedom on seemingly arbitrary choices such as layer sizes and activation functions. When testing different networks for this work, we found significant differences between network architectures (LSTM, fully connected, branch-trunk, etc.). Small changes such as slightly altering layer sizes or activation functions did not, in general, significantly affect results.

Table C1 Training parameters for neural networks

Network	Restarts	Decay	Patience	Batch
F_P	3	0.95,2	10	1000
F_ψ	3	0.95,2	3	1000
F_ξ	3	0.95,2	3	1000
H_P	3	0.98,2	5	250
H_ψ	3	0.98,2	5	250
H_ξ	3	0.98,2	5	250
G	3	none	20	100
E/D	1	0.98,1	5	10

C.2. Data Partitions

In many machine-learning tasks, and particularly those with a high dimensional parameter space such as neural networks, a dataset is split into distinct sets for training, validation, and testing. The training dataset is used to update model parameters according to the gradient of the cost function. The validation set is used to prevent overfitting through the use of early stopping when performance metrics on validation data have stalled. Finally, the testing set is reserved for a comparison between models. We use a (70/15/15%) split, meaning 70 percent of the available data in each case are used for training, 15 for validation, and 15 for testing. Training and validation datasets are mixed randomly from within an interval of time spanning the initial 85% of the available data, and testing data are the remaining contiguous 15%. The length of data available for training models exhibits slight variability due to input sequence lengths and lead times. Comparison between models in Sec. IV is therefore performed on the final 15,000 data points, or equivalently 150 convective units of the simulation.

C.3. Training Procedure

Parameters for the training procedures used for each neural network are given in Table C1. We used the Adam optimizer [29] with batch size as given in the last column. Restarts, set to 3 for all networks except E/D, indicates the number of optimizations from random initial weights that were performed. Each was run until validation error failed to decrease for a specified number of epochs, called the patience. The encoder–decoder pair was only trained once due to computational expense. A pair (0.98,2) for decay indicates that learning rate was multiplied by 0.98 every two epochs. An ℓ^2 regularizer with weight 10^{-5} was used for F_ξ. LSTM layers as well

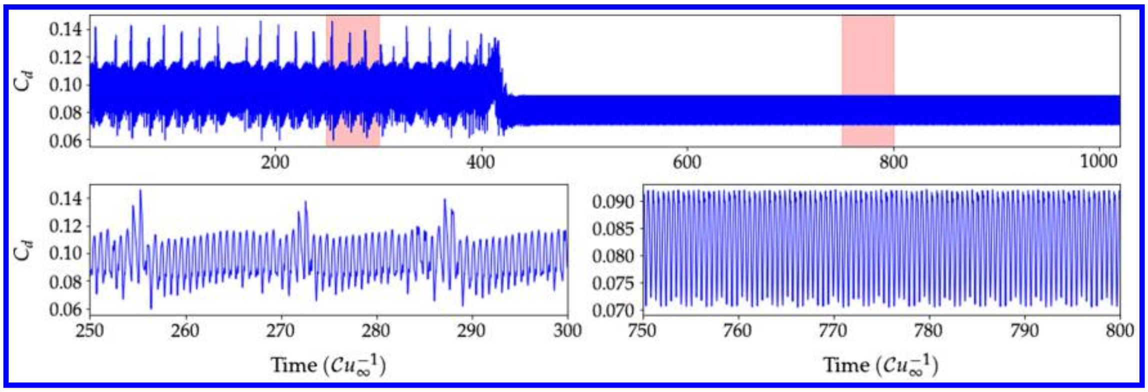

Fig. D1 Time series for the drag coefficient showing transition for intermittent to more regular behavior. Lower row shows close-up of time series from two shaded regions.

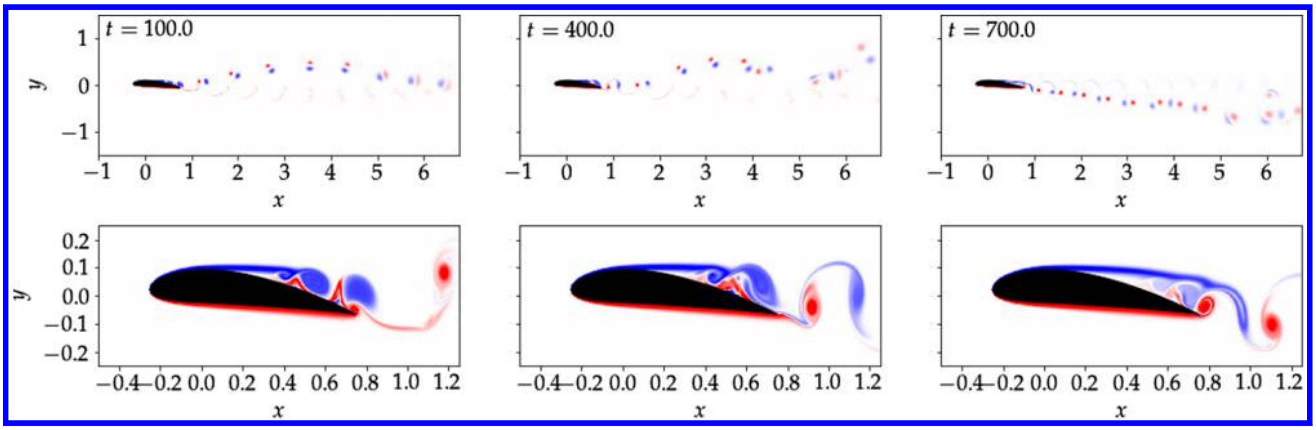

Fig. D2 Vorticity snapshots of the simulation exhibiting mode switching at time $t = 100$ before switching, $t = 400$ soon before switching, and $t = 700$ after switching.

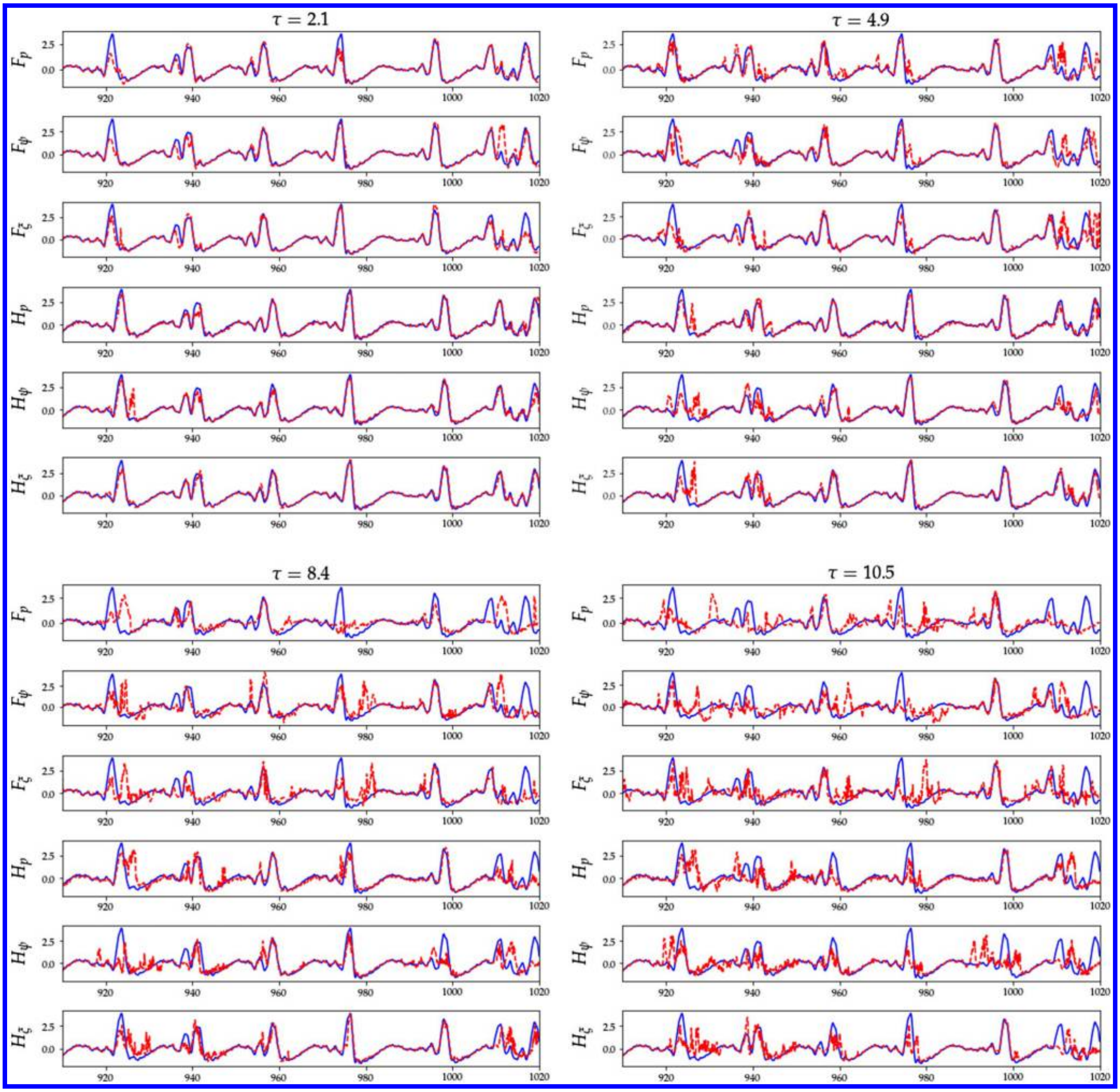

Fig. E1 Time series for $q(t)$ and $\hat{q}(t)$ using each of the neural network-based methods on the test set for lead times $\tau = 2.1$, 4.9, 8.4, and 10.5.

as connections to the output layer were left unregularized. The initial learning rate in each case was set to 0.001.

Parameters for training procedures were largely set based on empirical evidence and considerations for computational resources. While we claim to have put forward due diligence in tuning all networks for the sake of a valid comparison, we make no claim that these values represent the optimal set for the problem at hand. The authors are not aware of any convincing methods for optimizing such hyperparameters.

Appendix D: Wake Instability

In Sec. II we noted that the intermittent behavior examined in this work is quasi-stable and that in some cases a change in the wake pattern occurs, resulting in less chaotic behavior. To illustrate this, we include here a trajectory similar to the one considered in this work where this mode switching does occur. Figure D1 shows $C_d(t)$ for a simulation exhibiting a shift in wake behavior around $t = 400$. Plots of the vorticity that clearly illustrate the transition are shown in Fig. 19. Wake patterns before and after transition resemble the P and $2S$ behaviors discussed in [11] and also shown in [10] at lower Reynolds numbers and higher angle of attack.

Appendix E: Time Series of Predicted Quantity of Interest Using All Methods

Time series predictions of $\hat{q}$ compared to the true value for test set examples at various values of lead time τ are shown in Fig. E1. Networks used to generate these time series are described in further detail in Appendix C. The reduced rank representations of $\boldsymbol{u}$ using both POD and FFNN use $r = 32$. Figures showing time series result for τ sampled between 0 and 10.5 in increments of 0.7 may be found online in the code repository for this work.

Acknowledgments

This material is based upon work supported by the Army Research Office (grant no. W911NF-17-1-0306) and used the Extreme Science and Engineering Discovery Environment (XSEDE) [46], which is supported by the National Science Foundation under grant no. ACI-1548562. TPS has also been supported by the Air Force Office of Scientific Research (grant no. MURI FA9550-21-1-0058). Simulations were performed on Expanse at the San Diego Supercomputing Center through allocation TG-MTH210003.

References

[1] Sapsis, T. P., "Statistics of Extreme Events in Fluid Flows and Waves," *Annual Review of Fluid Mechanics*, Vol. 53, Jan. 2020. https://doi.org/10.1146/annurev-fluid-030420-032810

[2] Mohamad, M. A., and Sapsis, T. P., "Sequential Sampling Strategy for Extreme Event Statistics in Nonlinear Dynamical Systems," *Proceedings of the National Academy of Sciences*, Vol. 115, No. 44, 2018, pp. 11138–11143. https://doi.org/10.1073/pnas.1813263115

[3] Sapsis, T. P., "Output-Weighted Optimal Sampling for Bayesian Regression and Rare Event Statistics Using Few Samples," *Proceedings of the Royal Society A*, Vol. 476, No. 2234, 2020, Paper 20190834. https://doi.org/10.1098/rspa.2019.0834

[4] Blanchard, A., and Sapsis, T., "Bayesian Optimization with Output-Weighted Optimal Sampling," *Journal of Computational Physics*, Vol. 425, Jan. 2021, Paper 109901. https://doi.org/10.1016/j.jcp.2020.109901

[5] Blonigan, P. J., Farazmand, M., and Sapsis, T. P., "Are Extreme Dissipation Events Predictable in Turbulent Fluid Flows?" *Physical Review Fluids*, Vol. 4, No. 4, 2019, Paper 044606. https://doi.org/10.1103/PhysRevFluids.4.044606

[6] Guth, S., and Sapsis, T. P., "Machine Learning Predictors of Extreme Events Occurring in Complex Dynamical Systems," *Entropy*, Vol. 21, No. 10, 2019, p. 925. https://doi.org/10.3390/e21100925

[7] Qi, D., and Majda, A. J., "Using Machine Learning to Predict Extreme Events in Complex Systems," *Proceedings of the National Academy of Sciences*, Vol. 117, No. 1, 2020, pp. 52–59. https://doi.org/10.1073/pnas.1917285117

[8] Lissaman, P., "Low-Reynolds-Number Airfoils," *Annual Review of Fluid Mechanics*, Vol. 15, No. 1, 1983, pp. 223–239. https://doi.org/10.1146/annurev.fl.15.010183.001255

[9] Kim, D.-H., Chang, J.-W., and Chung, J., "Low-Reynolds-Number Effect on Aerodynamic Characteristics of a NACA 0012 Airfoil," *Journal of Aircraft*, Vol. 48, No. 4, 2011, pp. 1212–1215. https://doi.org/10.1016/j.ast.2013.08.018

[10] Menon, K., and Mittal, R., "Aerodynamic Characteristics of Canonical Airfoils at Low Reynolds Numbers," *AIAA Journal*, Vol. 58, No. 2, 2020, pp. 977–980. https://doi.org/10.2514/1.J058969

[11] Gopalakrishnan Meena, M., Taira, K., and Asai, K., "Airfoil-Wake Modification with Gurney Flap at Low Reynolds Number," *AIAA Journal*, Vol. 56, No. 4, 2017, pp. 1348–1359. https://doi.org/10.2514/1.J056260

[12] Wang, S., Zhou, Y., Alam, M. M., and Yang, H., "Turbulent Intensity and Reynolds Number Effects on an Airfoil at Low Reynolds Numbers," *Physics of Fluids*, Vol. 26, No. 11, 2014, Paper 115107. https://doi.org/10.1063/1.4901969

[13] Williamson, C. H., "Vortex Dynamics in the Cylinder Wake," *Annual Review of Fluid Mechanics*, Vol. 28, No. 1, 1996, pp. 477–539. https://doi.org/10.1146/annurev.fl.28.010196.002401

[14] Modarres-Sadeghi, Y., Chasparis, F., Triantafyllou, M., Tognarelli, M., and Beynet, P., "Chaotic Response Is a Generic Feature of Vortex-Induced Vibrations of Flexible Risers," *Journal of Sound and Vibration*, Vol. 330, No. 11, 2011, pp. 2565–2579. https://doi.org/10.1016/j.jsv.2010.12.007

[15] Tank, J., Smith, L., and Spedding, G., "On the Possibility (or Lack Thereof) of Agreement Between Experiment and Computation of Flows over Wings at Moderate Reynolds Number," *Interface Focus*, Vol. 7, No. 1, 2017, Paper 20160076. https://doi.org/10.1098/rsfs.2016.0076

[16] Maulik, R., Fukami, K., Ramachandra, N., Fukagata, K., and Taira, K., "Probabilistic Neural Networks for Fluid Flow Surrogate Modeling and Data Recovery," *Physical Review Fluids*, Vol. 5, No. 10, 2020, Paper 104401. https://doi.org/10.1103/PhysRevFluids.5.104401

[17] Gomez, D. F., Lagor, F., Kirk, P. B., Lind, A., Jones, A. R., and Paley, D. A., "Unsteady DMD-Based Flow Field Estimation from Embedded Pressure Sensors in an Actuated Airfoil," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-0346, 2019. https://doi.org/10.2514/6.2019-0346

[18] de Castro da Silva, A. F., and Colonius, T., "An Enkf-Based Flow State Estimator for Aerodynamic Flows," *8th AIAA Theoretical Fluid Mechanics Conference*, AIAA Paper 2017-3483, 2017. https://doi.org/10.2514/6.2017-3483

[19] da Silva, A. F., and Colonius, T., "Flow State Estimation in the Presence of Discretization Errors," *Journal of Fluid Mechanics*, Vol. 890, March 2020. https://doi.org/10.1017/jfm.2020.103

[20] Dawson, S. T., Schiavone, N. K., Rowley, C. W., and Williams, D. R., "A Data-Driven Modeling Framework for Predicting Forces and Pressures on a Rapidly Pitching Airfoil," *45th AIAA Fluid Dynamics Conference*, AIAA Paper 2015-2767, 2015. https://doi.org/10.2514/6.2015-2767

[21] Hou, W., Darakananda, D., and Eldredge, J. D., "Machine-Learning-Based Detection of Aerodynamic Disturbances Using Surface Pressure Measurements," *AIAA Journal*, Vol. 57, No. 12, 2019, pp. 5079–5093. https://doi.org/10.2514/1.J058486

[22] Le Provost, M., Hou, W., and Eldredge, J., "Deep Learning and Data Assimilation Approaches to Sensor Reduction in Estimation of Disturbed Separated Flows," *AIAA Scitech 2020 Forum*, AIAA Paper 2020-0799, 2020. https://doi.org/10.2514/6.2020-0799

[23] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine Learning for Fluid Mechanics," *Annual Review of Fluid Mechanics*, Vol. 52, Jan. 2020, pp. 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214

[24] Fukami, K., Fukagata, K., and Taira, K., "Assessment of Supervised Machine Learning Methods for Fluid Flows," *Theoretical and Computational Fluid Dynamics*, Vol. 34, No. 4, 2020, pp. 497–519. https://doi.org/10.1007/s00162-020-00518-y

[25] Brenner, M., Eldredge, J., and Freund, J., "Perspective on Machine Learning for Advancing Fluid Mechanics," *Physical Review Fluids*, Vol. 4, No. 10, 2019, Paper 100501. https://doi.org/10.1103/PhysRevFluids.4.100501

[26] Paul, F., Fischer, J. W. L., and Kerkemeier, S. G., "nek5000 Web Page," 2008, http://nek5000.mcs.anl.gov [retrieved 11 Aug. 2021].

[27] Dong, S., "A Convective-Like Energy-Stable Open Boundary Condition for Simulations of Incompressible Flows," *Journal of Computational Physics*, Vol. 302, Dec. 2015, pp. 300–328. https://doi.org/10.1016/j.jcp.2015.09.017

[28] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[29] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.

[30] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, Cambridge, MA, 2016, http://www.deeplearningbook.org [retrieved 11 Aug. 2021].

[31] Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Univ. Press, Cambridge, England, U.K., 2012, pp. 86–128. https://doi.org/10.1017/CBO9780511622700

[32] Taira, K., Brunton, S. L., Dawson, S. T., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., "Modal Analysis of Fluid Flows: An Overview," *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4013–4041. https://doi.org/10.2514/1.J056060

[33] Venturi, D., and Karniadakis, G. E., "Gappy Data and Reconstruction Procedures for Flow Past a Cylinder," *Journal of Fluid Mechanics*, Vol. 519, Oct. 2004, pp. 315–336. https://doi.org/10.1017/S0022112004001338

[34] Milano, M., and Koumoutsakos, P., "Neural Network Modeling for Near Wall Turbulent Flow," *Journal of Computational Physics*, Vol. 182, No. 1, 2002, pp. 1–26. https://doi.org/10.1006/jcph.2002.7146

[35] Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E., "Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators," *Nature Machine Intelligence*, Vol. 3, No. 3, 2021, pp. 218–229. https://doi.org/10.1038/s42256-021-00302-5

[36] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 378, Feb. 2019, pp. 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[37] Erichson, N. B., Mathelin, L., Yao, Z., Brunton, S. L., Mahoney, M. W., and Kutz, J. N., "Shallow Neural Networks for Fluid Flow Reconstruction with Limited Sensors," *Proceedings of the Royal Society A*, Vol. 476, No. 2238, 2020, Paper 20200097. https://doi.org/10.1098/rspa.2020.0097

[38] Levine, M. E., and Stuart, A. M., "A Framework for Machine Learning of Model Error in Dynamical Systems," arXiv preprint arXiv:2107.06658, July 2021.

[39] Lee, K., and Carlberg, K. T., "Model Reduction of Dynamical Systems on Nonlinear Manifolds Using Deep Convolutional Autoencoders," *Journal of Computational Physics*, Vol. 404, March 2020, Paper 108973. https://doi.org/10.1016/j.jcp.2019.108973

[40] Yarusevych, S., and Boutilier, M. S., "Vortex Shedding of an Airfoil at Low Reynolds Numbers," *AIAA Journal*, Vol. 49, No. 10, 2011, pp. 2221–2227. https://doi.org/10.1017/S0022112009007058

[41] West, G., and Apelt, C., "The Effects of Tunnel Blockage and Aspect Ratio on the Mean Flow Past a Circular Cylinder with Reynolds Numbers Between 104 and 105," *Journal of Fluid Mechanics*, Vol. 114, Jan. 1982, pp. 361–377. https://doi.org/10.1017/S0022112082000202

[42] Turki, S., Abbassi, H., and Nasrallah, S. B., "Effect of the Blockage Ratio on the Flow in a Channel with a Built-In Square Cylinder," *Computational Mechanics*, Vol. 33, No. 1, 2003, pp. 22–29. https://doi.org/10.1007/s00466-003-0496-2

[43] Wang, S., Wang, H., and Perdikaris, P., "Learning the Solution Operator of Parametric Partial Differential Equations with Physics-Informed DeepONets," *Science Advances*, Vol. 7, No. 40, 2021, Paper eabi8605. https://doi.org/10.1126/sciadv.abi8605

[44] Gouasmi, A., Parish, E. J., and Duraisamy, K., "A Priori Estimation of Memory Effects in Reduced-Order Models of Nonlinear Systems Using the Mori–Zwanzig Formalism," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 473, No. 2205, 2017, Paper 20170385. https://doi.org/10.1098/rspa.2017.0385

[45] Gupta, A., and Lermusiaux, P. F., "Neural Closure Models for Dynamical Systems," *Proceedings of the Royal Society A*, Vol. 477, No. 2252, 2021, Paper 20201004. https://doi.org/10.1098/rspa.2020.1004

[46] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J., and Wilkins-Diehr, N., "XSEDE: Accelerating Scientific Discovery," *Computing in Science & Engineering*, Vol. 16, No. 05, 2014, pp. 62–74. https://doi.org/10.1109/MCSE.2014.80

[47] Patera, A. T., "A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion," *Journal of Computational Physics*, Vol. 54, No. 3, 1984, pp. 468–488. https://doi.org/10.1016/0021-9991(84)90128-1

[48] Geuzaine, C., and Remacle, J.-F., "Gmsh: A 3-D Finite Element Mesh Generator with Built-In Pre- and Post-Processing Facilities," *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, 2009, pp. 1309–1331. https://doi.org/10.1002/nme.2579

[49] Maday, Y., Patera, A. T., and Rønquist, E. M., "An Operator-Integration-Factor Splitting Method for Time-Dependent Problems: Application to Incompressible Fluid Flow," *Journal of Scientific Computing*, Vol. 5, No. 4, 1990, pp. 263–292. https://doi.org/10.1007/BF01063118

[50] Fischer, P., and Mullen, J., "Filter-Based Stabilization of Spectral Element Methods," *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, Vol. 332, No. 3, 2001, pp. 265–270. https://doi.org/10.1016/S0764-4442(00)01763-8

[51] Pomeau, Y., and Manneville, P., "Intermittent Transition to Turbulence in Dissipative Dynamical Systems," *Communications in Mathematical Physics*, Vol. 74, No. 2, 1980, pp. 189–197. https://doi.org/10.1007/BF01197757

[52] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, software available from tensorflow.org, https://www.tensorflow.org/ [retrieved 11 Aug. 2021].

[53] Ramachandran, P., Zoph, B., and Le, Q. V., "Searching for Activation Functions," arXiv preprint arXiv:1710.05941, 2017.

K. Taira
Associate Editor