Nonintrusive reduced order modeling framework for quasigeostrophic turbulence

Sk. M. Rahman,[1] S. Pawar,[1] O. San,[1,*] A. Rasheed,[2] and T. Iliescu[3]

[1]*School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA*

[2]*Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7465, Trondheim, Norway*

[3]*Department of Mathematics, Virginia Tech, Blacksburg, Virginia 24061, USA*

(Received 23 June 2019; revised manuscript received 14 September 2019; published 12 November 2019)

In this study, we present a nonintrusive reduced order modeling (ROM) framework for large-scale quasistationary systems. The framework proposed herein exploits the time series prediction capability of long short-term memory (LSTM) recurrent neural network architecture such that (1) in the training phase, the LSTM model is trained on the modal coefficients extracted from the high-resolution data snapshots using proper orthogonal decomposition (POD) transform, and (2) in the testing phase, the trained model predicts the modal coefficients for the total time recursively based on the initial time history. Hence, no prior information about the underlying governing equations is required to generate the ROM. To illustrate the predictive performance of the proposed framework, the mean flow fields and time series response of the field values are reconstructed from the predicted modal coefficients by using an inverse POD transform. As a representative benchmark test case, we consider a two-dimensional quasigeostrophic ocean circulation model which, in general, displays an enormous range of fluctuating spatial and temporal scales. We first demonstrate that the conventional Galerkin projection-based reduced order modeling of such systems requires a high number of POD modes to obtain a stable flow physics. In addition, ROM-Galerkin projection (ROM-GP) does not seem to capture the intermittent bursts appearing in the dynamics of the first few most energetic modes. However, the proposed nonintrusive ROM framework based on LSTM (ROM-LSTM) yields a stable solution even for a small number of POD modes. We also observe that the ROM-LSTM model is able to capture quasiperiodic intermittent bursts accurately, and yields a stable and accurate mean flow dynamics using the time history of a few previous time states, denoted as the lookback time window in this paper. We show several features of ROM-LSTM framework such as significantly higher accuracy than ROM-GP, and faster performance using larger time step size. Throughout the paper, we demonstrate our findings in terms of time series evolution of the field values and mean flow patterns, which suggest that the proposed fully nonintrusive ROM framework is robust and capable of predicting chaotic nonlinear fluid flows in an extremely efficient way compared to the conventional projection-based ROM framework.

DOI: 10.1103/PhysRevE.100.053306

I. INTRODUCTION

Large-scale turbulent flows, such as atmospheric and geophysical flows, are nonlinear dynamical systems which exhibit an enormous range of complex, coherent spatiotemporal scales. Over the past half century, computational approaches have made a significant contribution to characterize and understand the behavior of such flow phenomena. To resolve physical problems with high spatiotemporal variabilities through numerical simulation, one needs a high-fidelity modeling technique like direct numerical simulation. However, a huge amount of computational resources are required to capture the fine details of the flow dynamics which can become inefficient and unmanageable after some level of accuracy. Although there has been a continuous growth in computer power and performance following Moore's law during the past few decades [1], the progress has started to stagnate in the recent years [2–5]. As a result, one of the most active research areas in modeling of turbulent flow dynamics is the development of efficient and robust algorithms that aim at achieving the maximum attainable quality of numerical simulations with optimal computational costs. Indeed, computational costs can be reduced by using low-fidelity models such as large eddy simulation (LES) [6] and Reynolds-averaged Navier-Stokes [7] with additional approximations in the governing equations to neglect some of the physical aspects for closure modeling. Even so, these techniques require parameter calibration to approximate the true solution to any degree of confidence and may thus increase costs related to model validation, benchmark data generation, and efficient analysis of the generated data sets. As an alternative to the existing techniques, the reduced order modeling (ROM) approach has quickly become a promising approach to reduce the computational burden of high-fidelity simulations. In general, ROM works in such a way that the high-dimensional complex dynamical systems will be represented with much lower-dimensional (but dense) systems while keeping the solution quality within the acceptable range [8,9]. An introduction to ROM methodologies can be found in recent review articles [9–11].

There have been a significant number of strategies proposed over the years to obtain ROMs of nonlinear dynamical systems. These ROM techniques have been utilized for a wide variety of applications related to, e.g., flow control [12–14],

*osan@okstate.edu

data assimilation in weather and climate modeling [15,16], and uncertainty quantification [17–19]. Among the different variants of ROM strategies, the Galerkin projection combined with proper orthogonal decomposition (POD) -based ROMs (ROM-GP) have been utilized extensively in various areas [20–26]. POD, also known as principal component analysis, is a mathematical technique to extract the dominant statistical characteristics from turbulent flow fields by identifying the most energetic modes [10,11,27–29]. These few POD modes possess the fine-scale details of the system and have the capability of representing the true physics accurately. Over the years, considerable work has been done to improve the regular POD approaches [8–10,12,30–35].

In general, POD uses the data obtained from experiments or high-fidelity numerical simulations and generates an orthonormal set of spatial basis vectors describing the main directions (modes) by which the flow is represented optimally, in an L_2 sense [32]. The most energetic modes are kept to generate the reduced order system while the other modes are truncated. However, it has been observed that the discarded modes often contribute to the evolving dynamics of large-scale complex turbulent flow systems, like the geophysical flows [36], resulting in instabilities and modeling errors in the solution [25,37–39]. Thus, several research efforts have been devoted to improve the stability of ROM-GP frameworks by addressing the truncated modes contributions [40–44]. Noack *et al.* [45] proposed an extra "shift mode" for accurate representation of the unstable steady solution. Several closure modeling ideas are devised to resolve the weak dissipation associated with POD modes by introducing eddy-viscosity terms (similar to LES eddy-viscosity models) [46–48]. San and Iliescu [49] improved the ROM performance by finding an optimal value for eddy-viscosity parameter with the assumption that the amount of dissipation is not identical for all the POD modes. We also proposed an automated approach to find the eddy-viscosity parameter dynamically to stabilize the ROM-GP model [50,51]. An alternative approach to find the eddy-viscosity parameter dynamically has been proposed by using an extreme learning machine architecture [52]. With the growing interest in data-driven modeling of ROMs using machine learning (ML) architectures, there has been another dimension of research introduced to the community for the improvement in ROM performance, referred as hybrid ROM approach. Generally, the hybridization is done by combining an imperfect physics-based model with a data-driven technique to get a hybrid scheme, and it is observed that the hybrid model shows better predictive performance than the component models [53–56].

In this paper, we develop a fully nonintrusive ROM approach as a potential alternative to already existing ROM methodologies. Indeed, physics-based (intrusive) ROM frameworks require an approximation of stabilization or regularization parameters and depend on underlying governing equation to get the solution. On the other hand, the hybrid approaches require computation of both intrusive and nonintrusive contributions, which can make the overall computation expensive. However, it is well known that a nonintrusive approach can make the framework greatly efficient when it can be implemented successfully. With the abundance of massive amounts of data resources from high-fidelity simulations, field measurements, and experiments, the data-driven modeling approaches are currently considered some of the most promising methods across different scientific and research communities. In the past few years, artificial neural networks (ANNs) and other ML techniques have started a revolution in turbulence modeling community [57–66]. Interested readers are directed to Refs. [67–71] for more on the influence of ML on fluid mechanics, specifically turbulence modeling.

With a goal to develop an efficient and robust nonintrusive ROM framework for large-scale quasistationary systems like geophysical flows, we propose a methodology based on long short-term memory (LSTM) recurrent neural networks. Since reduced order modeling of such chaotic large-scale systems is comparatively difficult due to instabilities, which results in using a very large number of POD modes to capture the true physics, our main motivation in this study is to utilize the time series prediction capability of LSTM [71–75] to capture the flow physics with very few POD modes. As detailed in Ref. [74], LSTM is very robust in predicting a very chaotic sequential time series. In general, for this type of random time series, LSTM does the prediction using its own internal dynamics, which is found stable and close to the true solution [74,76]. For this reason, we choose to utilize LSTM architecture based on our problem of interest, which is the large-scale quasistationary turbulence. However, we emphasize that this nonintrusive model can be developed by using other relevant neural network architectures as well. We also mention that the development of ROM using POD and LSTM has been used only in a few other works and proven to be successful in capturing the temporal dynamics of fluid flows. Wang *et al.* [73] proposed a nonintrusive ROM based on LSTM and used it to predict laminar flows. In another recent work, Vlachas *et al.* [77] proposed a data-driven method based on LSTM to predict the state derivative of chaotic systems using the short-term history of the reduced order states. The predicted derivatives are then used for one-step forward prediction of the high-dimensional dynamics. The authors further developed a hybrid framework combining mean stochastic model and LSTM for data-driven to extend the forecasting capability of the proposed approach. To do the dimensionality reduction, the authors utilized discrete Fourier transform, singular value decomposition, and empirical orthogonal functions. Mohan and Gaitonde [76] developed a nonintrusive ROM using LSTM and POD for flow control applications through a detailed analysis on different ROM-LSTM training and testing hyperparameter tuning parameters. Even though the authors' idea of developing nonintrusive ROM-based LSTM by replacing Galerkin projection is similar to our present work, their work is mostly focused on exploring the capability of LSTM in modeling the flow in reduced order space for data sets with less randomness. Indeed, the data sets with less randomness have more "memory" in it, i.e., there are persistent or anti-persistent trends and thus, are more controllable through LSTM hyperparameters. On the other hand, our present work is focused on exploring the capability of ROM-LSTM framework in resolving large-scale geophysical flow problem where the data sets mostly do not follow any particular trend. To this end, we develop a modular ROM-LSTM approach in chaotic and quasistationary systems to see whether it can overcome the instability issues

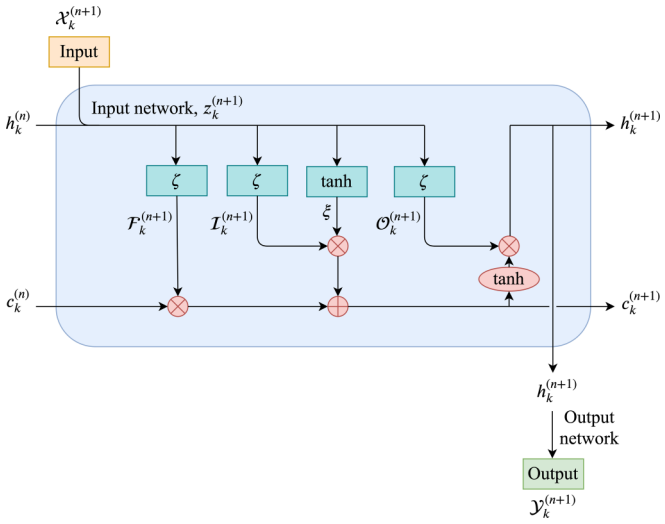

FIG. 1. Schematic representation of a typical LSTM network.

associated with conventional ROMs for chaotic dynamical systems. To assess our proposed framework, we consider the barotropic vorticity equation (BVE) representing the single-layer quasigeostrophic (QG) model as an example of the quasistationary system. We observe a remarkably efficient predictive performance by the proposed framework based on LSTM (ROM-LSTM) through a number of numerical experiments and analyses.

The layout of the paper is as follows: Sec. II provides an overview of the barotropic vorticity equation describing a single-layer QG ocean model. In Sec. III dimension reduction through Galerkin projection and proper orthogonal decomposition is illustrated briefly. Our proposed nonintrusive ROM-LSTM framework with a brief introduction to LSTM are presented in Sec. IV. In Sec. V we evaluate the predictive performance of the proposed ROM framework with respect to the standard ROM and full order model solutions. We demonstrate the the ROM-LSTM framework is a robust surrogate model and show boundedness of the present approach with larger time step size in Sec. VI. Finally, Sec. VII provides a summary of this study and the conclusions drawn from it.

II. SINGLE-LAYER QUASIGEOSTROPHIC OCEAN CIRCULATION MODEL

In the present study, we consider the simple single-layer QG ocean circulation model to develop and assess the performance of different ROM approaches. Following Refs. [78,79], we consider the single-layer QG problem as a benchmark for wind-driven, large-scale oceanic flow. Wind-driven flows of midlatitude ocean basins have been studied frequently by modelers using idealized single- and double-gyre wind forcing, which helps in understanding various aspects of ocean dynamics, including the role of mesoscale eddies and their effect on mean circulation. However, modeling the vast range of spatiotemporal scales of the oceanic flows with all the relevant physics has always been challenging. As a result, the numerical simulation of oceanic and atmospheric flows still requires approximations and simplifications of the full model. The BVE describing the single-layer QG

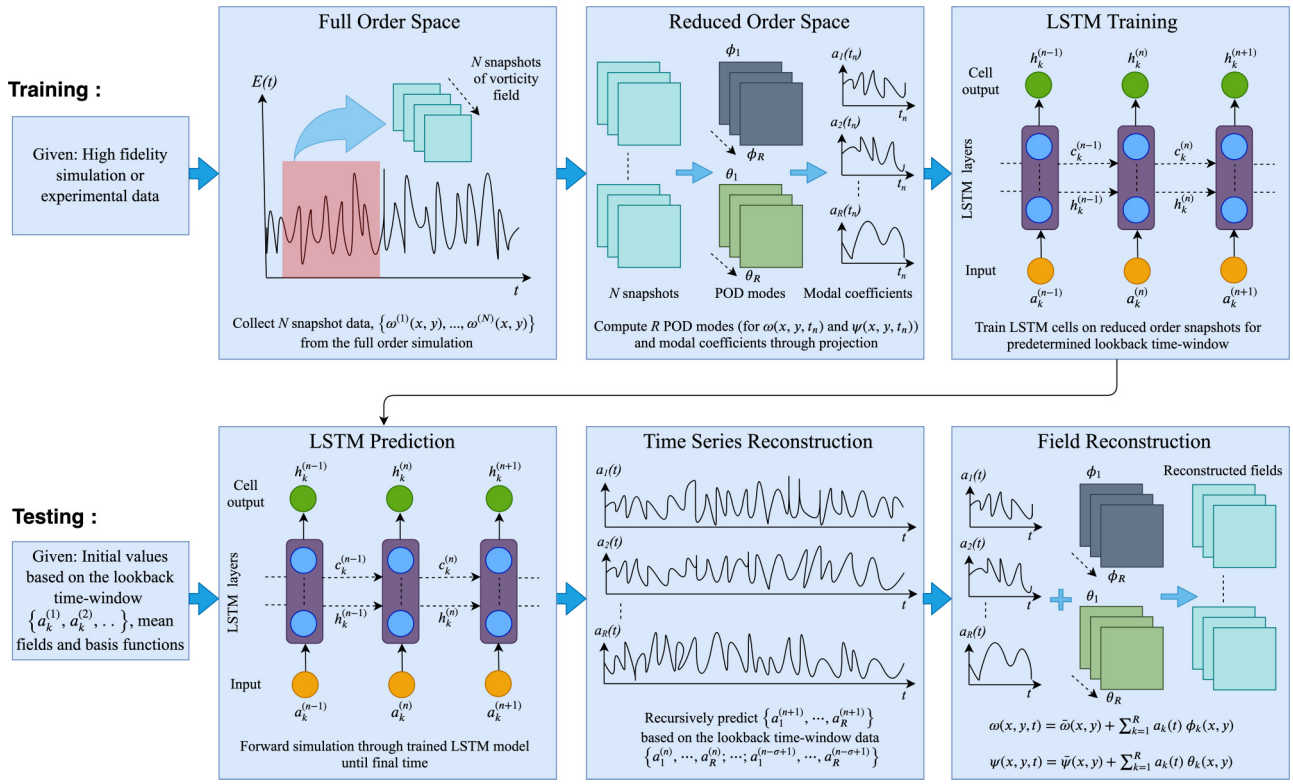

FIG. 2. Work-flow diagram of the ROM-LSTM framework. Note that the training phase (offline computation) is computationally heavier compared to the testing (online computation) phase.

equation with dissipative and forcing terms is one of the most commonly used models for the double-gyre wind-driven geophysical flows [80].

The BVE model shares many features with the two-dimensional (2D) Euler and Navier-Stokes equations and has been extensively used over the years to describe various aspects of the largest scales of turbulent geophysical fluid dynamics [81–86]. Using β-plane assumption reasonable for most oceanic flows, the dimensionless vorticity-stream-function formulation of the forced-dissipative BVE can be written as [53]

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) - \frac{1}{\mathrm{Ro}} \frac{\partial \psi}{\partial x} = \frac{1}{\mathrm{Re}} \nabla^2 \omega + \frac{1}{\mathrm{Ro}} \sin(\pi y), \quad (1)$$

where ∇^2 is the standard 2D Laplacian operator. ω and ψ are the kinematic vorticity and stream function, respectively, defined as

$$\omega = \nabla \times \mathbf{u}, \quad (2)$$

$$\mathbf{u} = \nabla \times \psi \hat{k}, \quad (3)$$

where $\mathbf{u}$ is the 2D velocity field and $\hat{k}$ refers to the unit vector perpendicular to the horizontal plane. The kinematic equation connecting the vorticity and stream function can be found by substituting the velocity components in terms of stream function in Eq. (2), which yields the following Poisson equation:

$$\nabla^2 \psi = -\omega. \quad (4)$$

Equation (1) contains two dimensionless parameters, Reynolds number (Re) and Rossby number (Ro), which are related to the physical parameters and nondimensional variables in the following way:

$$\mathrm{Re} = \frac{VL}{\nu}, \quad \mathrm{Ro} = \frac{V}{\beta L^2}, \quad (5)$$

where ν is the horizontal eddy viscosity of the BVE model and β is the gradient of the Coriolis parameter at the basin center ($y = 0$). L is the basin length scale and V is the velocity scale, also known as the Sverdrup velocity [87], and is given by

$$V = \frac{\tau_0}{\rho H} \frac{\pi}{\beta L}, \quad (6)$$

where τ_0 is the maximum amplitude of the double-gyre wind stress, ρ is the mean fluid density, and H is the mean depth of the ocean basin. Despite not being explicitly represented in Eq. (1), there are two important relevant physical parameters, the Rhines scale, δ_I, and the Munk scale, δ_M, which are the boundary layer thicknesses for the inertial and viscous layers of the basin geometry, respectively. As a physical interpretation of these parameters in BVE model, δ_I accounts for the strength of nonlinearity and δ_M is a measure of dissipation strength. Here δ_I and δ_M can be defined as

$$\frac{\delta_I}{L} = \left(\frac{V}{\beta L^2}\right)^{\frac{1}{2}}, \quad \frac{\delta_M}{L} = \left(\frac{\nu}{\beta L^3}\right)^{\frac{1}{3}} \quad (7)$$

and are related to Ro and Re by the following relations:

$$\frac{\delta_I}{L} = (\mathrm{Ro})^{\frac{1}{2}}, \quad \frac{\delta_M}{L} = \left(\frac{\mathrm{Ro}}{\mathrm{Re}}\right)^{\frac{1}{3}}. \quad (8)$$

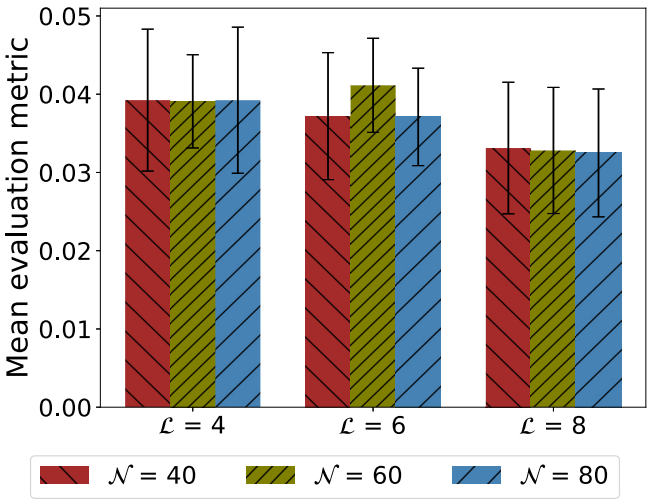

FIG. 3. Evaluation results for hyperparameters search for LSTM architecture. We perform fivefold cross-validation for different number of layers $\mathcal{L} = 4, 6, 8$ and number of neurons $\mathcal{N} = 40, 60, 80$. The mean of the evaluation metric for five data samples is used to select the hyperparameters. The error bars shows the standard deviation of evaluation metric score for five samples.

Finally, the nonlinear advection term in Eq. (1) is given by the Jacobian

$$J(\omega, \psi) = \frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y}. \quad (9)$$

Since ocean circulation models where the Munk and Rhines scales are close to each other, like the QG model, remain time dependent instead of reaching a steady state as time approaches to infinity [88], numerical computations of these models are conducted in a statistically steady state, also known as the quasistationary state. Hence, in our study, we utilize numerical schemes suited for simulation of such type of ocean models and for long-time integration. For the full order model (FOM) simulations, we utilize the second-order central finite difference schemes for the derivatives in the linear terms. The nonlinear Jacobian term in Eq. (1) is modeled by the second-order skew-symmetric, energy- and enstrophy-conserving Arakawa scheme [89] to avoid computational instabilities arising from nonlinear interactions. For the time integration, we use the optimal total variation diminishing third-order accurate Runge-Kutta scheme [90]. Following previous theoretical studies of large-scale ocean circulation models in simplified Cartesian oceanic basins [79,84,85,91,92], we utilize slip boundary condition for the velocity, which implies homogeneous Dirichlet boundary condition for the vorticity and stream function:

$$\omega|_\Gamma = \psi|_\Gamma = 0, \quad (10)$$

where Γ refers to all boundary coordinates. We start our computations from a quiescent state, i.e., $\omega_{t=0} = \psi|_{t=0} = 0$ initial conditions, and integrate the model until a statistically steady state is obtained. Details of the relevant numerical discretization schemes for this study can be found in related work [53,79,92].

III. INTRUSIVE ROM-GP METHODOLOGY

The intrusive ROM framework is developed based on a standard Galerkin projection methodology using the method of snapshots, an efficient method for computing the POD basis functions [93]. In this section, we briefly describe the ROM-GP framework utilized in our work. We obtain N snapshots for vorticity field, $\omega(x, y, t_n)$ for $n = 1, 2, \ldots, N$ at pseudotime $t = t_n$ from FOM simulation. Algorithm 1 describes the POD basis construction procedure from the stored snapshots.

Algorithm 1. POD basis construction

1: Compute the time-invariant mean fields and the fluctuation fields (mean-subtracted snapshots) for the given number of snapshots of the 2D vorticity field as

$$\bar{\omega}(x, y) = \frac{1}{N}\sum_{n=1}^{N}\omega(x, y, t_n), \tag{11}$$

$$\omega'(x, y, t_n) = \omega(x, y, t_n) - \bar{\omega}(x, y). \tag{12}$$

2: An $N \times N$ snapshot data matrix $\mathbf{A} = [a_{ij}]$ is computed from the inner product of mean-subtracted snapshots

$$a_{ij} = \langle \omega'(x, y, t_i); \omega'(x, y, t_j)\rangle, \tag{13}$$

where i and j refer to the snapshot indices.

3: Compute the optimal POD basis functions by performing an eigendecomposition of $\mathbf{A}$ as $\mathbf{AV} = \mathbf{V\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix whose entries are the eigenvalues λ_k of $\mathbf{A}$, and $\mathbf{V}$ is a matrix whose columns $\mathbf{v}_k$ are the corresponding eigenvectors. In our computations, we use the eigensystem solver based on the Jacobi transformations since $\mathbf{A}$ is a symmetric positive definite matrix [94].

4: Using the eigenvalues stored in a descending order (i.e., $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_N$), for proper selection of the POD modes in $\mathbf{\Lambda}$, compute the orthogonal POD basis functions for the vorticity field ϕ_k as

$$\phi_k(x, y) = \frac{1}{\sqrt{\lambda_k}}\sum_{n=1}^{N} v_k^n \omega'(x, y, t_n), \tag{14}$$

where v_k^n is the nth component of the eigenvector $\mathbf{v}_k$. The scaling factor, $1/\sqrt{\lambda_k}$, guarantees the orthonormality of POD modes, i.e., $\langle \phi_i; \phi_j\rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta.

5: Obtain the kth mode for the stream function, $\theta_k(x, y)$ utilizing the linear dependence between stream function and vorticity given by Eq. (4):

$$\nabla^2\theta_k = -\phi_k, \tag{15}$$

for each $k = 1, 2, \ldots, R$. To be able to use the same $a_k(t)$ coefficients for both stream-function and vorticity fields, the following elliptic equation holds true for the mean variables:

$$\nabla^2\bar{\psi} = -\bar{\omega}. \tag{16}$$

6: Construct kth time-dependent modal coefficients $a_k(t_n)$ for N snapshots by using POD modes and forward transformation:

$$a_k(t_n) = \langle \omega(x, y, t_n) - \bar{\omega}(x, y); \phi_k\rangle. \tag{17}$$

We can approximate the field variables, i.e., kinematic vorticity and stream function using the most energetic R POD modes, where $R \ll N$, such that these R largest energy containing modes correspond to the largest eigenvalues $(\lambda_1, \ldots, \lambda_R)$. The resulting full expression for the field variables can be written as

$$\omega(x, y, t) = \bar{\omega}(x, y) + \sum_{k=1}^{R} a_k(t)\phi_k(x, y), \tag{18}$$

$$\psi(x, y, t) = \bar{\psi}(x, y) + \sum_{k=1}^{R} a_k(t)\theta_k(x, y), \tag{19}$$

where $a_k(t)$ accounts for both stream function and vorticity based on the kinematic relation given by Eq. (4). It should be mentioned that in our ROM formulations, we use the following angle-parenthesis definition for the inner product of two arbitrary functions f and g:

$$\int_\Omega f(x, y)g(x, y)\,dx\,dy = \langle f; g\rangle. \tag{20}$$

We refer to Ref. [50] for the details of the integration technique utilized in this study. In conventional projection-based intrusive ROM framework, we apply Galerkin projection to the governing equation, which yields R coupled ordinary differential equations (ODEs) for the time evolution of the temporal modes of the system while the spatial modes are kept constant [34,40,95]. Any standard time integration technique can be utilized to solve the coupled ODE system, since the basis functions and corresponding modal coefficients will be precomputed in the offline computation stage. The Galerkin projection approach is summarized in Algorithm 2.

Algorithm 2. Galerkin projection to obtain ROM

1: Given an initial condition $\omega(x, y, t_0)$ at time t_0, compute the initial modal coefficients $a_k(t_0)$ using the following relation:

$$a_k(t_0) = \langle \omega(x, y, t_0) - \bar{\omega}(x, y); \phi_k\rangle. \tag{21}$$

2: Perform an orthogonal Galerkin projection by multiplying the governing equation with the POD basis functions and integrating over the domain [96], which will yield the following dynamical system for a_k:

$$\frac{da_k}{dt} = \mathfrak{B}_k + \sum_{i=1}^{R}\mathfrak{L}_k^i a_i + \sum_{i=1}^{R}\sum_{j=1}^{R}\mathfrak{N}_k^{ij} a_i a_j, \tag{22}$$

where $k = 1, 2, \ldots, R$ and the predetermined model coefficients can be computed by the following numerical integration (offline computing):

$$\mathfrak{B}_k = \left\langle -J(\bar{\omega}, \bar{\psi}) + \frac{1}{\mathrm{Ro}}\left[\sin(\pi y) + \frac{\partial\bar{\psi}}{\partial x}\right] + \frac{1}{\mathrm{Re}}\nabla^2\bar{\omega}; \phi_k\right\rangle,$$

$$\mathfrak{L}_k^i = \left\langle -J(\bar{\omega}, \theta_i) - J(\phi_i, \bar{\psi}) + \frac{1}{\mathrm{Ro}}\frac{\partial\theta_i}{\partial x} + \frac{1}{\mathrm{Re}}\nabla^2\phi_i; \phi_k\right\rangle,$$

$$\mathfrak{N}_k^{ij} = \langle -J(\phi_i, \theta_j); \phi_k\rangle. \tag{23}$$

TABLE I. A list of hyperparameters utilized to train the LSTM network for all numerical experiments.

Parameters	Values
Number of hidden layers	6
Number of neurons in each hidden layer	40
Batch size	16
Epochs	500
Activation functions in the LSTM layers	tanh
Validation data set	20%
Training-testing ratio	4:9
Loss function	MSE
Optimizer	ADAM
Learning rate (α)	0.001
First moment decay rate (β_1)	0.9
Second moment decay rate (β_2)	0.999

TABLE II. Hurst exponents of modal coefficients.

Modal coefficient	Hurst exponent
$a_1(t)$	0.52
$a_2(t)$	0.35
$a_3(t)$	0.63
$a_4(t)$	0.59
$a_5(t)$	0.49
$a_6(t)$	0.59
$a_7(t)$	0.59
$a_8(t)$	0.46
$a_9(t)$	0.58
$a_{10}(t)$	0.53

IV. NONINTRUSIVE ROM-LSTM METHODOLOGY

In this section, we discuss our proposed ROM-LSTM methodology. As outlined in Algorithm 1, we obtain the time-dependent modal coefficients a_k by performing a POD transform on stored snapshot data. The modal coefficients are a sequence of data points with respect to time, i.e., a time series representing the underlying dynamical system. In intrusive or physics-based ROM, we do Galerkin projection using governing equation to obtain a coupled system of ODEs for a_k, and then solve the ODE system on the given time interval. However, the limitations of projection-based ROMs, such as susceptibility to instability for chaotic data set, numerical constraints for solving ODE system, or inefficient reduced order modeling, encourage us to replace the physics-based Galerkin projection phase of ROM-GP methodology with a data-driven approach. Among the variety of ideas to resolve the issues associated with projection-based ROM, a number of published works related to ROM based on POD and neural networks have shown signs of future success. The recurrent neural network (RNN) is one of the widely used neural network architectures in ROMs which is designed to operate on input information as well as the previously stored observations to predict the dependencies among the temporal data sequences [97,98]. LSTM is a special variant of RNN which is capable of tracking long-term dependencies among the input data sequences. Hence, we consider LSTM recurrent neural network to develop our nonintrusive ROM-LSTM framework. Before describing the ROM-LSTM procedure, we first briefly review the LSTM architecture.

As the name suggests, RNNs contain recurrent or cyclic connections that enable them to model complex time-varying data sequences with a wide range of temporal dependencies or correlations between them. In general, RNNs map a sequence of data to another sequence through time using cyclic connections, and constrain some of the connections to hold the same weights using back-propagation algorithm [99]. However, the standard RNN architecture suffers from vanishing gradient problem when the gradient of some weights starts to become too small or too large [100]. This leads to the development of improved RNN architectures which overcome the modeling issues of standard RNNs. There are several variants of RNN which are found to work well in tackling long-term dependencies [101–104]. One of the most successful forms of improved RNN architectures is the LSTM network, which solves the limitation of vanishing gradients [105]. In contrast to most of the ANN architectures, LSTM operates by cell states and gating mechanisms to actively control the dynamics of cyclic connections and thus, resolves the vanishing gradient issues. Similar to the standard RNNs, LSTM can learn and predict the temporal dependencies based on the input information and previously acquired information, i.e., the internal memory of LSTM allows the network to find the relationship between the current input and stored information to make a prediction. There has been a number of variants of LSTM developed later as well which showed promise for better performance, for example, grid LSTM [106].

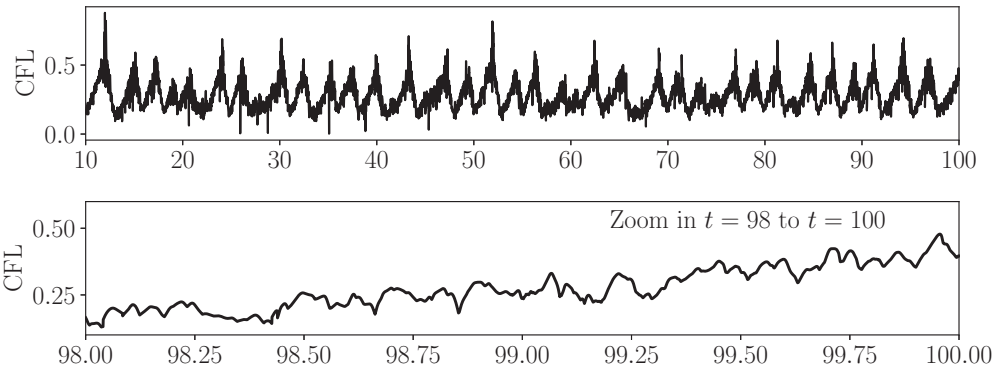

FIG. 4. Variation of CFL number from time $t = 10$ to $t = 100$ for the FOM simulation.

The conventional LSTM architecture contains memory blocks in the recurrent hidden layers, which have memory cells to store the cell states and gates to control the flow of information. Each memory block has an input gate controlling the flow of input activations into the cell, a forget gate to adaptively forgetting and resetting the cell's memory (to prevent overfitting by processing continuous inflow of input streams), and the output gate controlling the output flow of cell activations into the next cell. We have shown a schematic of a conventional LSTM cell in Fig. 1 to illustrate how the data stream flows through LSTM cells. In our LSTM architecture, we consider an input sequential data matrix $\mathcal{X}_k$ and the output sequential data matrix $\mathcal{Y}_k$. Each sample of the input training matrix $\mathcal{X}_k$ is constructed as $\{a_1^{(n)}, \ldots, a_R^{(n)}; \ \ldots \ ; a_1^{(n-\sigma+1)}, \ldots, a_R^{(n-\sigma+1)}\}$ and the corresponding output sample in output sequential data matrix $\mathcal{Y}_k$ is $\{a_1^{(n+1)}, \ldots, a_R^{(n+1)}\}$. In our study, we call σ as the lookback time window. The lookback time window, in our definition, means the time history over which the LSTM model does the training and prediction recursively. Indeed, increasing the value of σ increases the quality of training the model but makes the model dependent on an increased number of initial states during prediction. Considering input gate as $\mathcal{I}$, the forget gate as $\mathcal{F}$, the output gate as $\mathcal{O}$, the cell activation vectors as c, and the LSTM cell output activation vector or the hidden state vector as h, the LSTM model does the mapping from the input sequence to an output sequence by using the following set of equations iteratively [74,75,105,107]. The equations for input network and gate functions, respectively, are

$$z_k^{(n+1)} = \lambda\left(W_h h_k^{(n)} + W_{\mathcal{X}} \mathcal{X}_k^{(n+1)}\right), \tag{24}$$

$$m_k^{(n+1)} = \zeta\left(W_m z_k^{(n+1)} + b_m\right), \tag{25}$$

where $m \in (\mathcal{I}, \mathcal{F}, \mathcal{O})$. The internal cell state equation is given by

$$c_k^{(n+1)} = \mathcal{F}_k^{(n+1)} \odot c_k^{(n)} + \mathcal{I}_k^{(n+1)} \odot \xi, \tag{26}$$

where $\xi = \tanh(W_c z_k^{(n+1)} + b_c)$. The output state of the LSTM cell and the output network can be expressed by the following

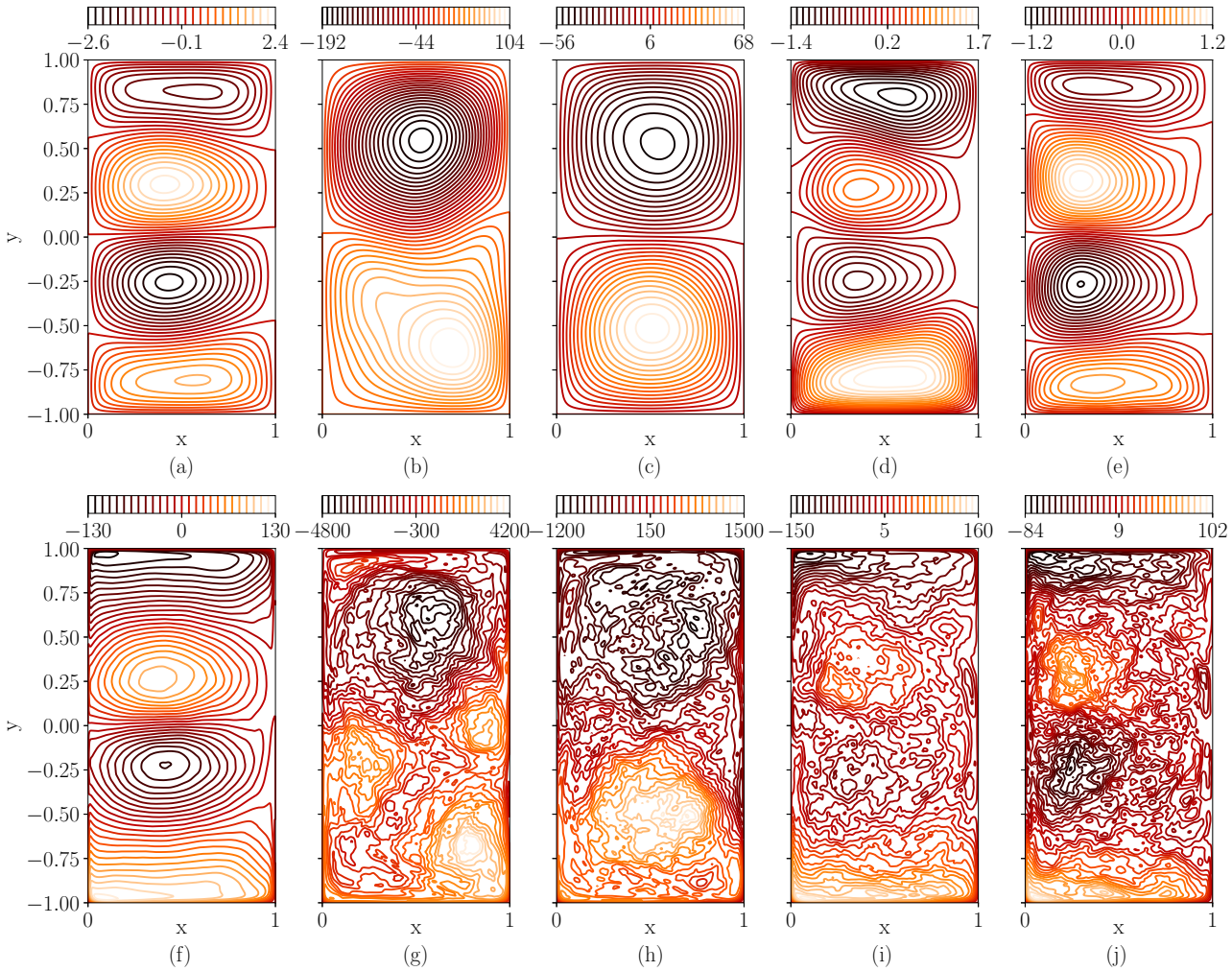

FIG. 5. Mean stream-function and vorticity fields obtained by the FOM simulation and the standard ROM-GP simulation at Re = 450 and Ro = 3.6×10^{-3} flow condition. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-GP}}$ with $R = 10$ modes, (c) $\psi_{\text{ROM-GP}}$ with $R = 20$ modes, (d) $\psi_{\text{ROM-GP}}$ with $R = 40$ modes, (e) $\psi_{\text{ROM-GP}}$ with $R = 80$ modes, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-GP}}$ with $R = 10$ modes, (h) $\omega_{\text{ROM-GP}}$ with $R = 20$ modes, (i) $\omega_{\text{ROM-GP}}$ with $R = 40$ modes, and (j) $\omega_{\text{ROM-GP}}$ with $R = 80$ modes.

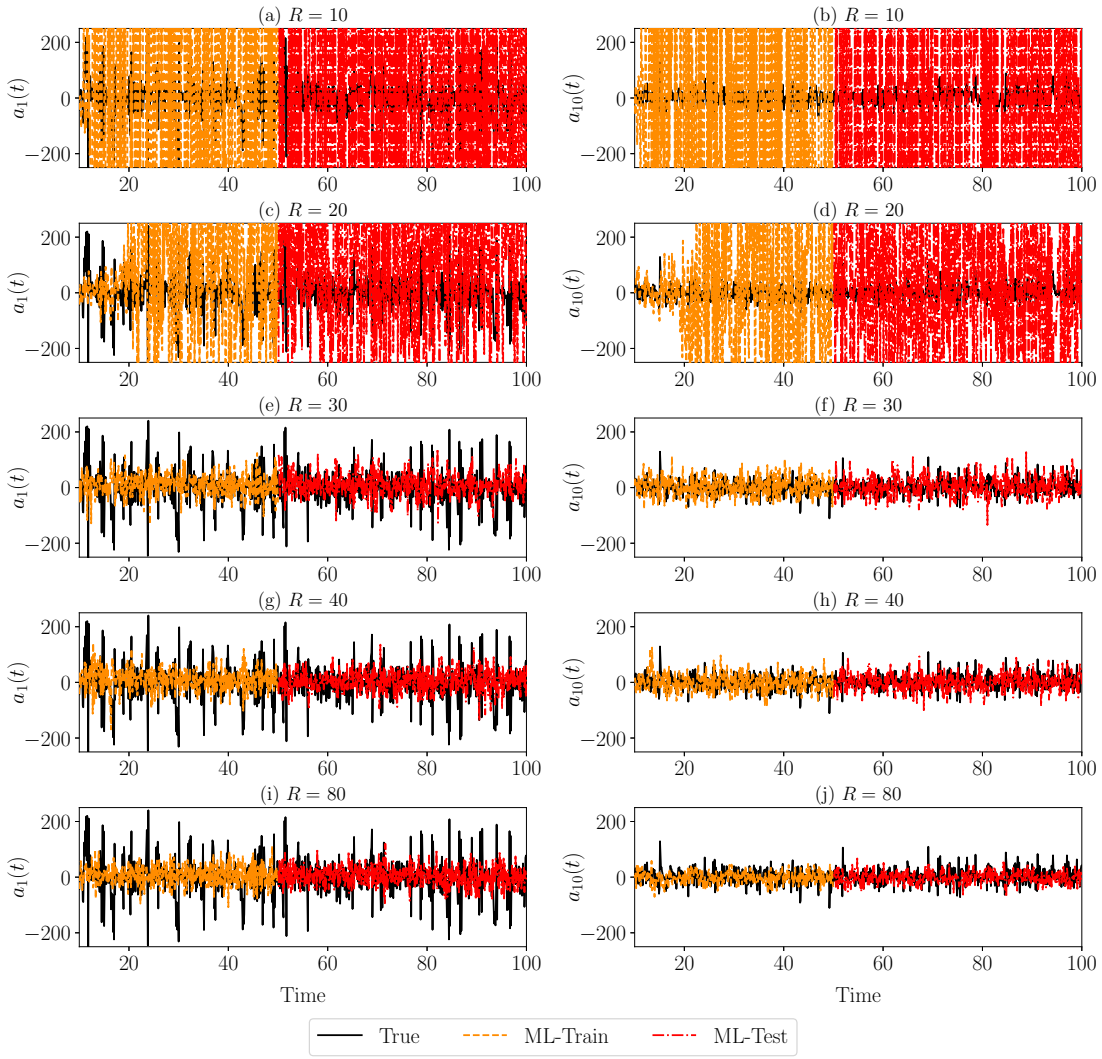

FIG. 6. Time series evolution of the first and tenth modal coefficients, $a_1(t)$ and $a_{10}(t)$, respectively, between $t = 10$ and $t = 100$ for standard ROM-GP simulation at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. (a) $a_1(t)$ for ROM-GP with $R = 10$ modes, (b) $a_{10}(t)$ for ROM-GP with $R = 10$ modes, (c) $a_1(t)$ for ROM-GP with $R = 20$ modes, (d) $a_{10}(t)$ for ROM-GP with $R = 20$ modes, (e) $a_1(t)$ for ROM-GP with $R = 30$ modes, (f) $a_{10}(t)$ for ROM-GP with $R = 30$ modes, (g) $a_1(t)$ for ROM-GP with $R = 40$ modes, (h) $a_{10}(t)$ for ROM-GP with $R = 40$ modes, (i) $a_1(t)$ for ROM-GP with $R = 80$ modes, and (j) $a_{10}(t)$ for ROM-GP with $R = 80$ modes. True projection series is underlined in each figure with black straight line. The training zone is shown with orange dashed line (from $t = 10$ to $t = 50$) and the out-of-sample testing zone is shown with red dashed line (from $t = 51$ to $t = 100$) in ROM-LSTM solution series in each figure.

equations, respectively:

$$h_k^{(n+1)} = \mathcal{O}_k^{(n+1)} \odot \tanh\left(c_k^{(n+1)}\right), \tag{27}$$

$$\mathcal{Y}_k^{(n+1)} = \lambda\left(W_{\mathcal{Y}} h_k^{(n+1)} + b_{\mathcal{Y}}\right), \tag{28}$$

where z vector is given by the input vector and hidden state vector of previous time step, W represents the weight matrices for each gates, b denotes the bias vectors for each gates, $\odot$ is the elementwise product or Hadamard product of two vectors, λ is the network input and output activation function which is tanh in our paper, and ζ is the logistic sigmoid function.

Similar to the ROM-GP methodology, the workflow of the ROM-LSTM framework consists of two phases as displayed in Fig. 2. In the offline training phase, we first obtain POD basis functions and modal coefficients using Algorithm 1. The known time series of modal coefficients from training snapshots are used to train the LSTM model. Based on the values of σ, the input of the LSTM model $\mathcal{M}$ will be the previous time states of the input modal coefficients for R retained modes and the output of the model will be the next time state recursively for R modes. Training LSTM model is the computationally heavier part of the ROM-LSTM framework, but this is done offline. In online testing phase, we recursively predict the modal coefficients for the total time using the trained model $\mathcal{M}$. When the model is deployed, the input of the trained model $\mathcal{M}$ will be the initial states $\{a_1^{(1)}, \ldots, a_R^{(1)}; \ldots; a_1^{(\sigma)}, \ldots, a_R^{(\sigma)}\}$ based on the preselected value of σ and the output will be recursive prediction of corresponding future time states. Thus, we bypass the physics-based Galerkin projection part with

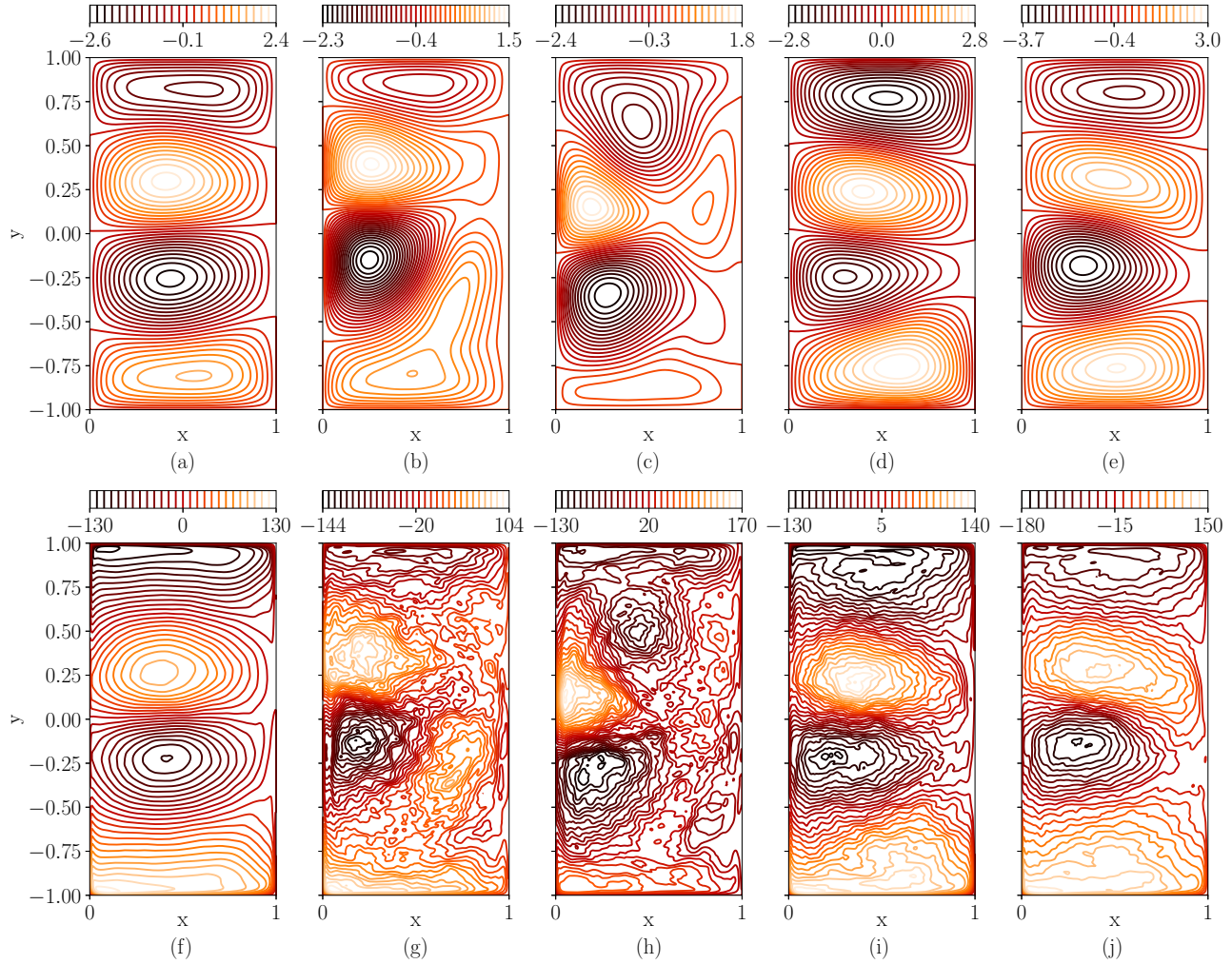

FIG. 7. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on different lookback time window, σ at $\mathrm{Re} = 450$ and $\mathrm{Ro} = 3.6 \times 10^{-3}$ flow condition. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ with $\sigma = 1$, (c) $\psi_{\text{ROM-LSTM}}$ with $\sigma = 2$, (d) $\psi_{\text{ROM-LSTM}}$ with $\sigma = 4$, (e) $\psi_{\text{ROM-LSTM}}$ with $\sigma = 5$, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ with $\sigma = 1$, (h) $\omega_{\text{ROM-LSTM}}$ with $\sigma = 2$, (i) $\omega_{\text{ROM-LSTM}}$ with $\sigma = 4$, and (j) $\omega_{\text{ROM-LSTM}}$ with $\sigma = 5$. Note that the LSTM model is trained with $R = 10$ modes.

completely data-driven neural network approach to predict the modal coefficients. Also, the computational cost of prediction through trained LSTM network is significantly lower than the physics-based approach. Finally, we reconstruct the mean vorticity and stream-function fields using inverse transform to analyze the behavior of the quasistationary flow. The key steps of the ROM-LSTM framework are outlined below in Algorithm 3.

Algorithm 3. ROM-LSTM framework

Training (offline)

1: Collect N snapshots data for the vorticity field, $\omega(x, y, t_n) = \{\omega^{(1)}(x, y), \omega^{(2)}(x, y), \ldots, \omega^{(N)}(x, y)\}$ from the FOM simulation.

2: Compute R POD modes for kinematic vorticity, ϕ_k and stream function, θ_k using Eq. (14) and Eq. (15), respectively, for $k = 1, 2, \ldots, R$.

3: Construct modal coefficients by a forward transform through projection

$$a_k(t_n) = \langle \omega(x, y, t_n) - \bar{\omega}(x, y); \phi_k \rangle, \tag{29}$$

where $a_k(t_n) = \{a_k^{(1)}, a_k^{(2)}, \ldots, a_k^{(N)}\}$.

4: Train LSTM model on reduced order snapshots for selected lookback time window σ:

$$\mathcal{M} : \left\{a_1^{(n)}, \ldots, a_R^{(n)}; \ \ldots \ ; a_1^{(n-\sigma+1)}, \ldots, a_R^{(n-\sigma+1)}\right\}$$
$$\Rightarrow \left\{a_1^{(n+1)}, \ldots, a_R^{(n+1)}\right\}. \tag{30}$$

Testing (Online)

5: Given initial values $\{a_k^{(1)}, a_k^{(2)}, \ldots, a_k^{(\sigma)}\}$ based on σ, precomputed mean values and basis functions.

6: Use the trained LSTM model $\mathcal{M}$ to recursively predict $a_k(t)$ until final time reached.

7: Reconstruct the mean fields by inverse transform using Eq. (18) and Eq. (19).

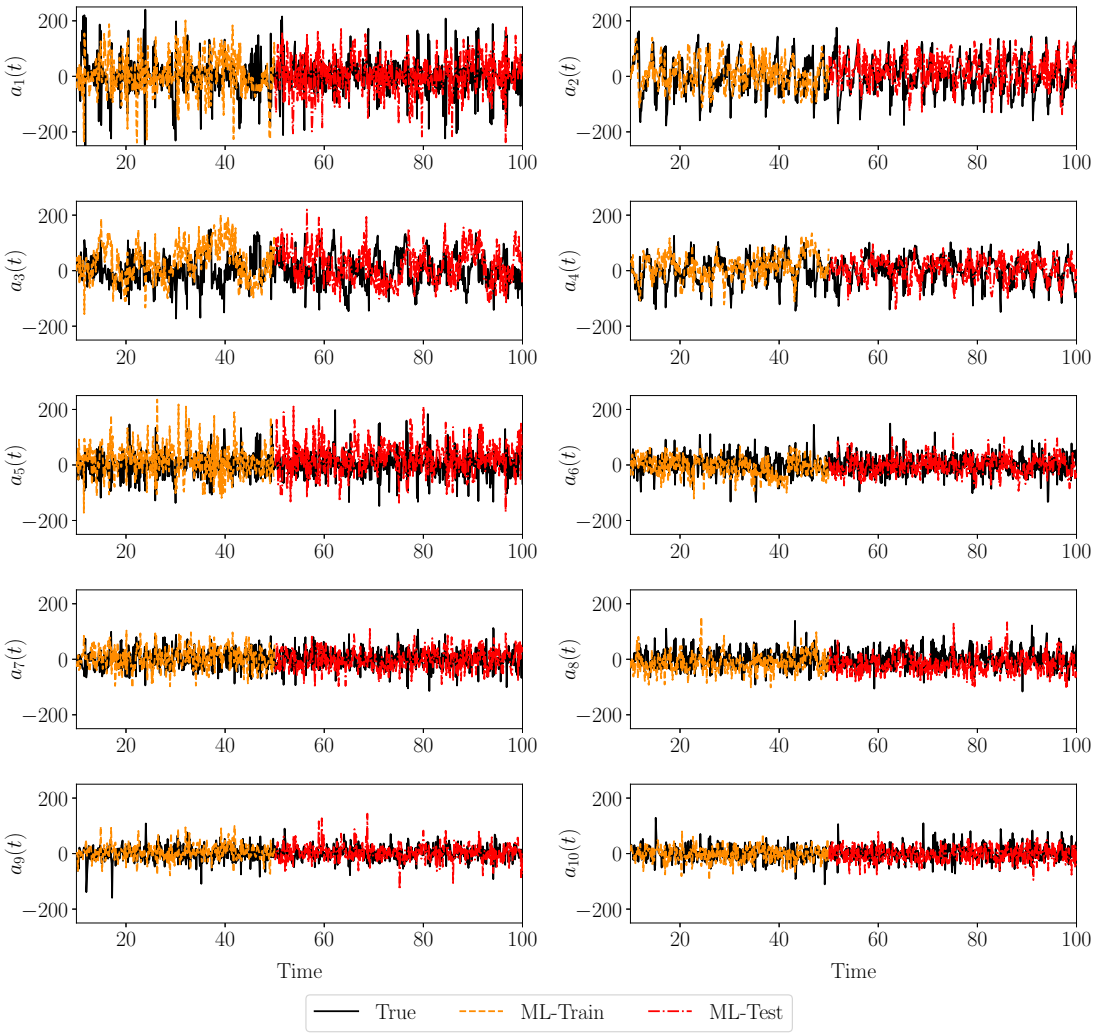

FIG. 8. Time series evolution of the modal coefficients between $t = 10$ and $t = 100$ for ROM-LSTM simulation at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. Note that the LSTM model is trained with $R = 10$ modes and $\sigma = 5$. True projection series is underlined in each figure with black straight line. The training zone is shown with orange dashed line (from $t = 10$ to $t = 50$) and the out-of-sample testing zone is shown with red dashed line (from $t = 51$ to $t = 100$) in ROM-LSTM solution series in each figure.

To design our LSTM architecture for ROM-LSTM framework, we utilize Keras [108], a high-level API designed for deep learning, combined with standard Python libraries. The FOM simulation for data snapshots generation and POD basis construction codes are written in FORTRAN programming language. We use grid-search selection procedure coupled with k-fold cross-validation to study the performance of LSTM network for different sets of hyperparameters. Cross-validation procedure is used to measure the performance of the model on unseen data. In this procedure, the training data are first divided into k groups. For each group, the remaining groups are used for training the LSTM model and the model's performance is evaluated for that group. Once the performance is estimated for all groups, the mean and standard deviation of the evaluation metric is used to select the hyperparameters. Therefore, if we use fivefold cross-validation, the model is trained five times and the process can become computationally expensive as number of folds increases. For the single-layer QG ocean circulation model, we are interested in the ability of LSTM network to predict the mean field and hence we compare mean modal coefficients for hyperparameters selection. We define the evaluation metric as

$$\text{Evaluation metric} = \frac{1}{R}\sum_{k=1}^{R}\left|\frac{1}{N}\sum_{i=1}^{N}\left(a_k^{(i)} - \hat{a}_k^{(i)}\right)\right|, \qquad (31)$$

where N is number of snapshots in validation data set, a_k is true modal coefficients, $\hat{a}_k$ are modal coefficients predicted by LSTM model, and R is the number of modes.

We use three different numbers of layers $\mathcal{L} = 4, 6, 8$ and three different numbers of neurons $\mathcal{N} = 40, 60, 80$ for each of these layers. We report the mean evaluation metric for these hyperparameters along with standard deviation in Fig. 3. It can be observed that the mean evaluation metric is almost the same for all hyperparameters. We utilize six LSTM layers with 40 neurons in each layer. These hyperparameters were found to be suitable for all our numerical experiments in which we test the LSTM model for different numbers of

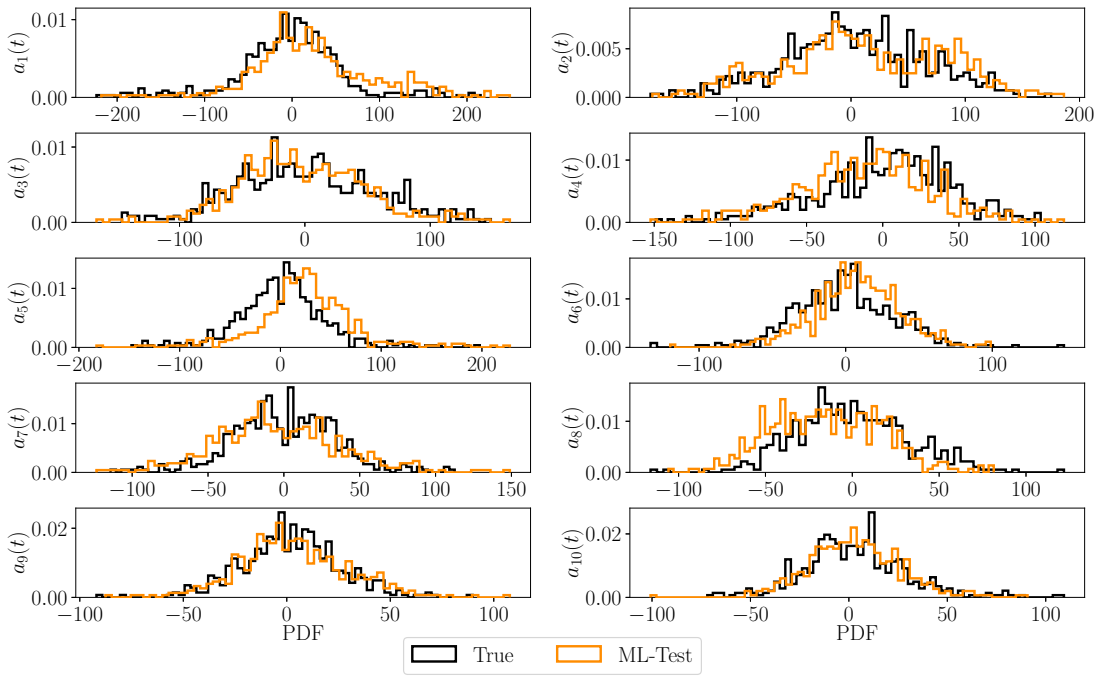

FIG. 9. PDF for true and predicted modal coefficients between $t = 50$ and $t = 100$ for ROM-LSTM simulation at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. Note that the LSTM model is trained with $R = 10$ modes and $\sigma = 5$. The training is done using true modal coefficients between $t = 10$ and $t = 50$.

modes and different numbers of lookback time windows. Also, the computational cost is found manageable in this deep architecture setup, which encourages us to perform all the numerical experiments with this same setup. The mean-squared error (MSE) is chosen as the loss function for weight optimization, and a variant of stochastic gradient descent method, called ADAM [109], is used to optimize the mean-squared loss. The other relevant hyperparameters utilized in our LSTM architecture are documented in Table I. The hyperparameters are kept constant for all simulations to obtain a fair comparison between the results in different numerical experiment runs. It should be noted that the training data is normalized by the minimum and maximum of each time series to be in between the range $[-1, +1]$.

V. NUMERICAL RESULTS

The predictive performance of the ROM-LSTM framework is thoroughly examined in this section in terms of time series evolution of the modal coefficients and mean flow fields. It is well documented in literature that the ROM-GP framework is incapable of capturing mean flow dynamics for quasistationary flows using lower number of POD modes and susceptible to instability [49,110]. There have been a number of approaches proposed in previous literature to improve the ROM performance. One way to stabilize the ROM is by adding an empirical stabilization parameter based on the analogy between large eddy simulation and truncated modal projection [40,51]. Later, it is found that the ROM performance further improves taking the optimal value for the stabilization parameter rather than selecting it arbitrarily [49,110–112]. In our previous work, we have shown that computing the stabilization parameter dynamically at each time step improve the ROM performance significantly [50]. However, the proposed ROM-LSTM methodology has several advantages over the physics-based approaches, such as no dependence on the underlying governing dynamical system to obtain the solution, i.e., the process is free of numerical constraints, no burden of adding stabilization parameter to account for instability issues and so on. To reach a conclusion about the performance of the ROM-LSTM framework, we compare ROM-LSTM predictions with the FOM simulation and the standard ROM-GP results. Moreover, we present the performance of the ROMs based on lookback time window σ and LSTM training for different number of POD modes to show the robustness and capability of the proposed framework. Furthermore, we present the L_2-norm errors to perform a quantitative assessment on the accuracy of the ROM-LSTM solutions with respect to ROM-GP solutions.

We choose the single-layer QG problem as our test bed to evaluate the performance of ROMs. Because of the complex flow behavior with wide range of scales, QG problem has been utilized as test problem in many references [78,79,84,85,113]. To make the analyses simple and easily understandable, we present simulation results only for Re $= 450$ and Ro $= 3.6 \times 10^{-3}$ flow condition, which can be considered turbulent enough and suitable for reduced order modeling. The FOM simulation is done from $t = 0$ to $t = 100$ using a constant time step of $\Delta t = 2.5 \times 10^{-5}$ on a Munk layer resolving 256×512 grid resolution (i.e., consisting of about four grid points in the Munk scale, i.e., $\delta_M/L = 0.02$). The computational domain of our test problem is $(x, y) \in [0, 1] \times [-1, 1]$. The time step size $\Delta t = 2.5 \times 10^{-5}$ ensures that Courant-Friedrichs-Lewy (CFL) condition for stability is met at all

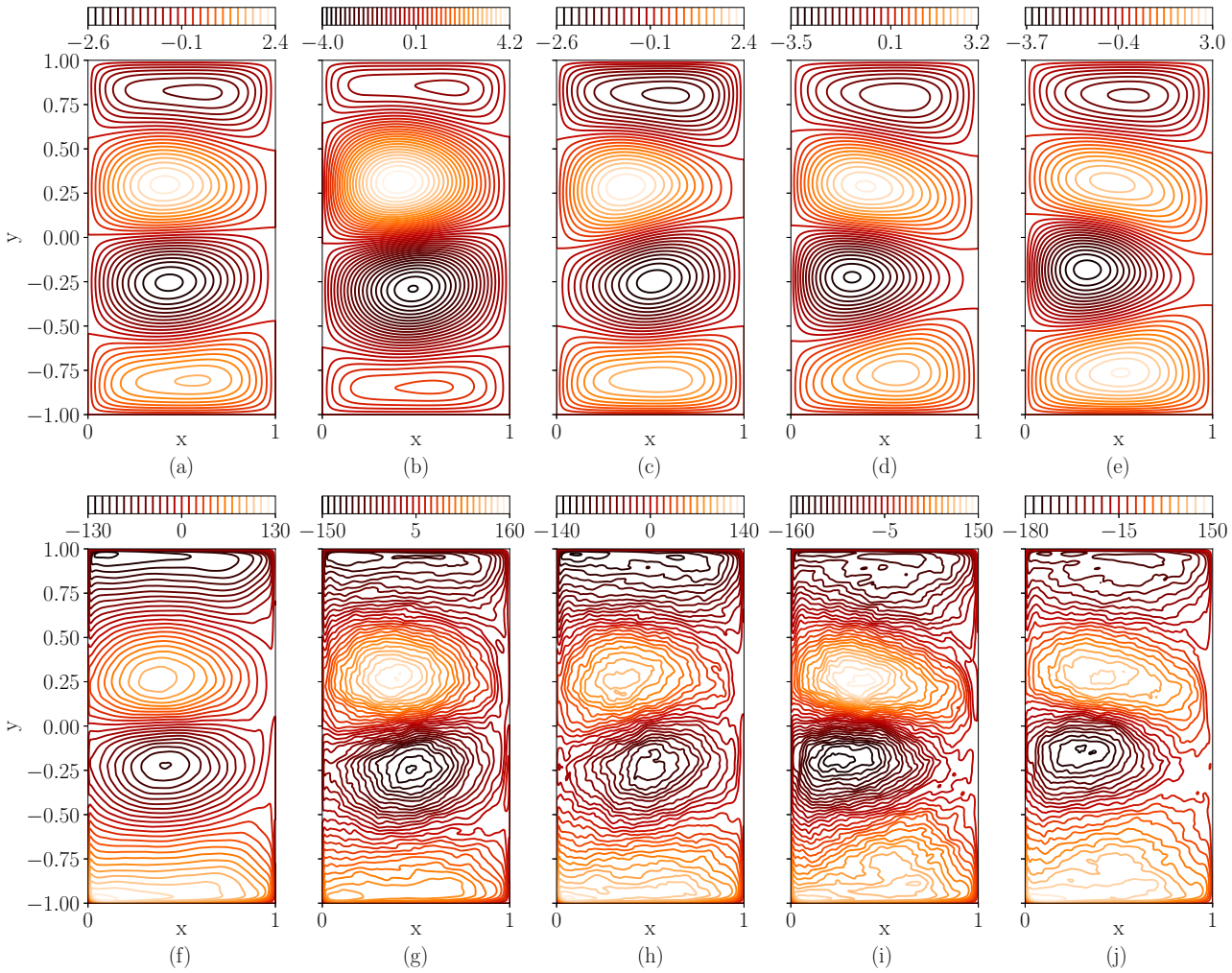

FIG. 10. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on the number of modes to train the LSTM model at Re = 450 and Ro = 3.6×10^{-3} flow condition. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 2$ modes, (c) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 4$ modes, (d) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 8$ modes, (e) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 10$ modes, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 2$ modes, (h) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 4$ modes, (i) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 8$ modes, and (j) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 10$ modes. Note that the LSTM model is trained with $\sigma = 5$.

times as shown in Fig. 4 in which the CFL number is below 1 at all times. In our computations, CFL number is computed by

$$\text{CFL} = U_{\max} \frac{\Delta t}{\Delta x}, \tag{32}$$

where $U_{\max}$ is the maximum speed $|\mathbf{u}|$ at time t within the entire ocean basin. Therefore, for constant Δt and Δx, the time evolution of the CFL number also relates to the variation of the maximum flow speed in the QG dynamics. Although we have utilized dimensionless numbers in our analysis, one can easily compute corresponding dimensional quantities using typical midlatitude ocean basin dimensions. For example, CFL = 0.3 refers to a physical maximum speed of approximately 2.53 m/s with the help of Eq. (6) if we set $L = 1000$ km, $H = 1$ km, $\tau_0 = 0.266$ N/m^2, $\rho = 1030$ kg/m^3, $\beta = 1.5 \times 10^{-11}$ m^{-1} s^{-1}, Sverdrup velocity scale $V = 0.054$ m/s, and reference timescale $L/V = 214$ days. Further details of nondimensionalization can be found in Ref. [92]. We note that the dimensionless time unit in our computations becomes approximately 7 months using the above reference values. To avoid the initial transient time interval, we store 400 data snapshots from $t = 10$ to $t = 50$ to generate the POD bases and modal coefficients to train the LSTM model. We refer to Ref. [50] to get an idea on the POD analysis as well as the instantaneous vorticity field plots for the same flow condition. To understand the nature of the QG data set, we compute the Hurst exponent, H, for the modal coefficients. The Hurst exponent is a statistical measure of the presence of long-term trends in a nonstationary time series [114]. Thus, the Hurst exponent can help in selecting the appropriate model for a given time series prediction. We also note that the Hurst exponent has been utilized in many research fields, e.g., hydrology, finance, and health-care industry [115–119]. H can be statistically defined as [120]

$$\text{E}\left[\frac{\text{Range}(n)}{\text{SD}(n)}\right] = kn^H, \quad \text{as } n \to \infty. \tag{33}$$

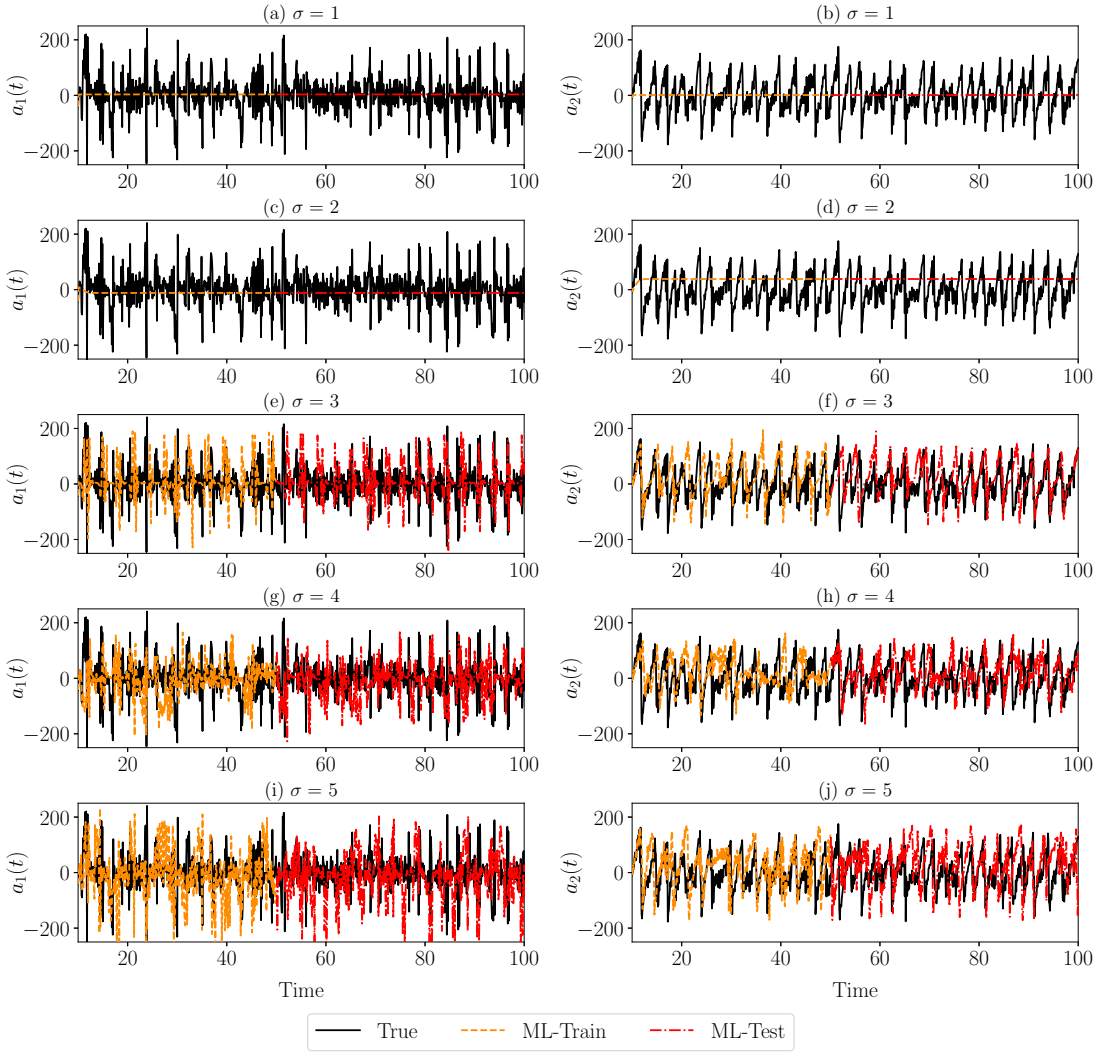

FIG. 11. Time series evolution of the modal coefficients between $t = 10$ and $t = 100$ for ROM-LSTM simulation based on different lookback time windows, σ and LSTM training with $R = 2$ modes at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. (a) $a_1(t)$ with $\sigma = 1$, (b) $a_2(t)$ with $\sigma = 1$, (c) $a_1(t)$ with $\sigma = 2$, (d) $a_2(t)$ with $\sigma = 2$, (e) $a_1(t)$ with $\sigma = 3$, (f) $a_2(t)$ with $\sigma = 3$, (g) $a_1(t)$ with $\sigma = 4$, (h) $a_2(t)$ with $\sigma = 4$, (i) $a_1(t)$ with $\sigma = 5$, and (j) $a_2(t)$ with $\sigma = 5$. True projection series is underlined in each figure with black straight line. The training zone is shown with orange dashed line (from $t = 10$ to $t = 50$) and the out-of-sample testing zone is shown with red dashed line (from $t = 51$ to $t = 100$) in ROM-LSTM solution series in each figure.

Here E is the expected value of the ratio between the range of the first n cumulative deviations from the mean and their corresponding standard deviations (SD), n is the time span of the observations, and k is constant. The range of H is in between 0 and 1. $H \to 1$ means a persistent series (a strong trend in the time series at hand), $H \to 0$ means an antipersistent series (a time series with long-term switching between high and low values) and $H \approx 0.5$ indicates a random series (fewer correlations between current and future observations). Interested readers are directed to Ref. [76] for a detailed description of suitability of LSTM as a predictive modeling approach for different time series data using the measurement of H. We calculate the H for modal coefficients of QG data set for given flow conditions using the so called rescaled range (R/S) analysis, popularized by Mandelbrot and Wallis [121,122]. The details of (R/S) analysis can be found in Ref. [120]. The Hurst exponents for the modal coefficients of QG case are tabulated in Table II, where we can see that the values of H are around 0.5. This indicates the randomness of the QG problem, which can be a good representative of large-scale quasistationary geophysical turbulent flow systems.

Figure 5 shows the mean stream-function and vorticity field contours obtained by the ROM-GP model. To compare the predictive performance of the ROM-GP model with respect to the true solution, we include the mean contour plots of FOM simulation on the left column as well. We can see the full order solution displays a four-gyre circulation pattern for both mean stream-function and vorticity fields. Since the instantaneous fields for the QG flow are always fluctuating in time, it becomes difficult to compare solutions of different models at the same time state. However, the mean fields always exhibit the four-gyre circulation for higher Re (highly turbulent regime, i.e., turbulence with weak dissipation), which implies a state of turbulent equilibrium between

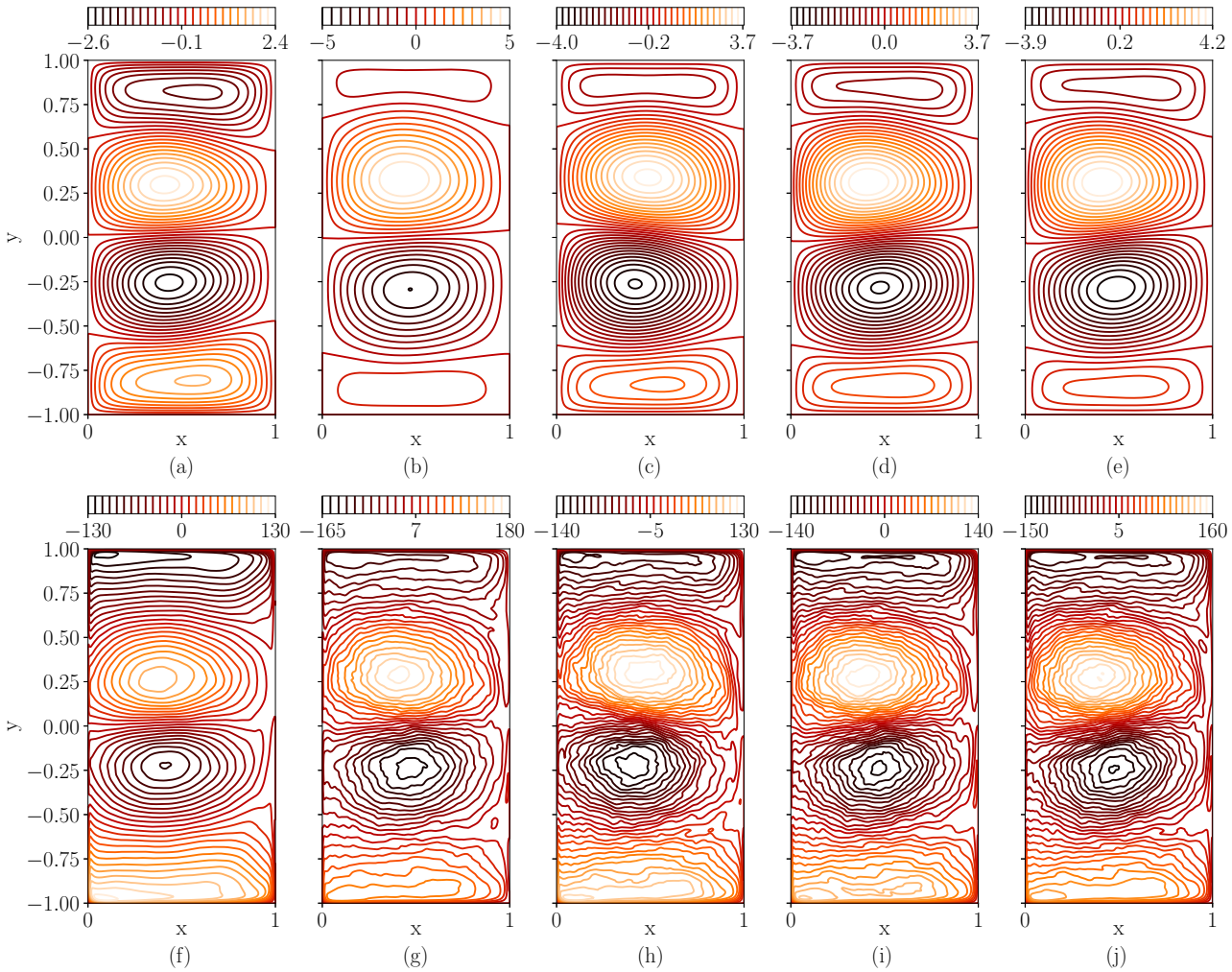

FIG. 12. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on different lookback time windows, σ and LSTM training with $R = 2$ modes at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (c) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (d) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, (e) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (h) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (i) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, and (j) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$.

two inner gyres circulation representing the wind stress curl forcing and the outer gyres representing the eddy flux of potential vorticity (the northern and southern gyres found in geostrophic turbulence) [85]. In our study, the time-averaged (mean) field data is obtained by averaging between $t = 50$ and $t = 100$ (since the model has already seen the training data between $t = 10$ and $t = 50$ during training). Another point to be noted in FOM field plots is that the bright orange circulations in the four-gyres (top circulation of the inner gyres and bottom circulation of the outer gyres) indicate the circulation in counterclockwise or positive direction and the other two circulations represent the circulation in clockwise direction. We can observe in Fig. 5 that the ROM-GP simulations with $R = 10$ and $R = 20$ modes display a nonphysical two-gyre circulation for stream function whereas the vorticity field does not capture almost any conclusive physical pattern. However, the results improve with increasing the number of modes as we can see the stream-function contour is showing clear four-gyre patterns even though the vorticity plots are very chaotic compared to the true solution. These observations are supported by the time series evolution of first and tenth modal coefficient plots in Fig. 6. It is apparent that increasing the number of modes stabilizes the system to yield a physical solution for both modal coefficients.

We note that the timescale in our formulation is normalized by L/V to obtain dimensionless time unit. Following Ref. [123], typical oceanic values (e.g., $L = 2000$ km and $\beta = 1.75 \times 10^{-11}$ m^{-1} s^{-1}) yield approximately $L/V = 0.25$ year for Ro $= 0.0036$. Therefore, a numerical simulation over 100 computational time units refers to the evolution of flow dynamics over 25 years in physical time. Therefore, the intermittent bursts in the true projection of the most energetic mode, i.e., $a_1(t)$, indicate the seasonal variations in QG dynamics. Although ROM-GP yields nonphysical solution for $R = 10$ and $R = 20$ cases, $a_k(t)$ series reaches more meaningful levels for $R = 30$ and beyond. However, it is hard to claim from Fig. 6 that the ROM-GP yields an accurate prediction of these seasonal bursts even for higher R values.

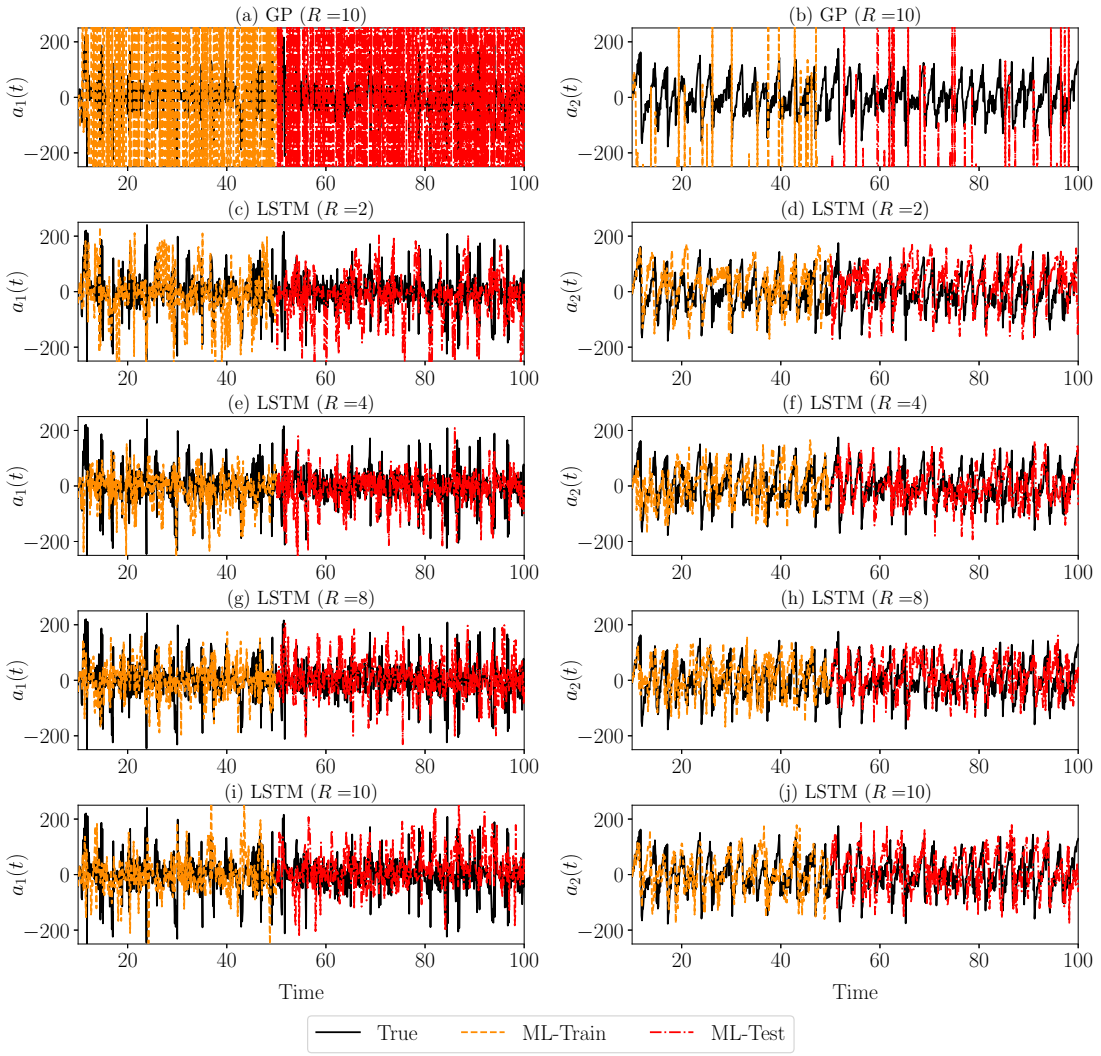

FIG. 13. Time series evolution of first two modal coefficients, $a_1(t)$ and $a_2(t)$, respectively, between $t = 10$ and $t = 100$ for different ROMs at Re $= 450$ and Ro $= 3.6 \times 10^{-3}$. (a) $a_1(t)$ for ROM-GP with $R = 10$ modes, (b) $a_2(t)$ for ROM-GP with $R = 10$ modes, (c) $a_1(t)$ for ROM-LSTM trained with $R = 2$ modes, (d) $a_2(t)$ for ROM-LSTM trained with $R = 2$ modes, (e) $a_1(t)$ for ROM-LSTM trained with $R = 4$ modes, (f) $a_2(t)$ for ROM-LSTM trained with $R = 4$ modes, (g) $a_1(t)$ for ROM-LSTM trained with $R = 8$ modes, (h) $a_2(t)$ for ROM-LSTM trained with $R = 8$ modes, (i) $a_1(t)$ for ROM-LSTM trained with $R = 10$ modes, and (j) $a_2(t)$ for ROM-LSTM trained with $R = 10$ modes. True projection series is underlined in each figure with black straight line. The training zone is shown with orange dashed line (from $t = 10$ to $t = 50$) and the out-of-sample testing zone is shown with red dashed line (from $t = 51$ to $t = 100$) in ROM-LSTM solution series in each figure.

We present the field contours obtained by ROM-LSTM based on different σ values in Fig. 7. It can be seen that $\sigma = 1$ and $\sigma = 2$ do not provide very accurate results as the patterns get distorted to some extent even though they are being able to capture the four-gyre. However, both stream-function and vorticity contours show a stable and accurate prediction of the true mean fields for $\sigma = 4$ and $\sigma = 5$. Though the vorticity field contour is not displaying as smooth contour lines as the true solution due to the reduction of dimension order, it is showing a better performance compared to the ROM-GP solutions. As shown in the recent work of Yeo [107], the LSTM network trained on chaotic data learns to reduce the contributions of randomness of input data by developing its own dynamics and thus, the prediction remains close to the truth rather than being unstable. Hence, the LSTM prediction is expected to yield a stable and physical solution for a fluctuating quasistationary system. It should be noted that these results are obtained for LSTM training with $R = 10$ modes. The time series evolution plots for the modal coefficients based on $\sigma = 5$ and $R = 10$ modes in Fig. 8 show that ROM-LSTM time series predictions are almost on top of the true projection of modal coefficients. Even though the model is trained for $t = 10$ to $t = 50$ only, the ROM-LSTM model is able to obtain a stable and accurate prediction up to the final time $t = 100$. To further evaluate whether the ROM-LSTM can predict the statistical features of the temporal variation of POD coefficients accurately or not, we compare the probability density function (PDF) of true and predicted modal coefficients in Fig. 9. The PDF is plotted for modal coefficients only between $t = 50$ and $t = 100$. Fig. 9

TABLE III. L_2-norm errors of the reduced order models (with respect to FOM) for the mean vorticity and stream-function fields. Note that the ROM-LSTM model trained with $R = 10$ modes results are presented here.

	Vorticity	Stream function
	Intrusive ROM	
ROM-GP ($R = 10$)	3.19×10^6	5.59×10^3
ROM-GP ($R = 20$)	4.46×10^5	9.87×10^2
ROM-GP ($R = 30$)	9.35×10^2	9.99×10^{-1}
ROM-GP ($R = 40$)	6.60×10^2	4.33×10^{-1}
ROM-GP ($R = 80$)	1.16×10^3	3.84×10^{-1}
	Nonintrusive ROM	
ROM-LSTM ($\sigma = 1$)	1.90×10^3	6.12×10^{-1}
ROM-LSTM ($\sigma = 2$)	2.65×10^3	7.44×10^{-1}
ROM-LSTM ($\sigma = 3$)	8.31×10^2	4.34×10^{-1}
ROM-LSTM ($\sigma = 4$)	5.17×10^2	4.68×10^{-1}
ROM-LSTM ($\sigma = 5$)	8.78×10^2	3.87×10^{-1}

shows that the PDF of predicted modal coefficients matches accurately the true PDF for all modal coefficients.

Another impressive observation on the predictive capability of the ROM-LSTM framework is presented in Fig. 10 where we show the mean field plots based on the number of modes retained to train the LSTM model. We keep $\sigma = 5$ for this numerical experiment. As we can see the ROM-LSTM model is being able to capture the four-gyre circulation even with only two modes. Indeed, the first few modes contain most of the dynamics in the system and we can also see reduction of some smaller scales for lower mode predictions. Nevertheless, this finding indicates the prediction capability of the ROM-LSTM framework to produce a stable solution of a chaotic system. However, we have seen the ROM-GP model becomes unstable to predict chaotic data set with lower number of modes which makes it very inefficient. In contrast, the proposed nonintrusive framework can be very efficient to produce stable solution with very few modes. Since we observe promising predictive performance for training with two modes only, we present a couple more analyses on results obtained by the ROM-LSTM framework retaining two modes for LSTM training. We can see in Fig. 11 that lower σ values simulations are unable to capture the fluctuations along the mean and go almost straight along the line after a few time states. The model starts to capture the fluctuating flow fields with the increase of σ values. The field plots in Fig. 12 also yield similar conclusions. Since the lower σ value solutions stay along the line around the mean (unlike rapid oscillations in ROM-GP solutions), the field plots still show the mean physics to some extent. It is obvious that the model with lower σ ignores most of the scales of the system. However, the prediction improves with higher σ as displayed in the Fig. 12.

Finally, we include a comparison plot in Fig. 13 where we present the first two modal coefficients prediction obtained by different ROM set up. The σ value is kept 5 for all the ROM-LSTM simulations. As expected, the ROM-GP solutions for 10 modes become totally nonphysical and unstable. On the contrary, the ROM-LSTM predictions for $R = 2$, $R = 4$, $R = 8$, and $R = 10$ modes show a good match between the true solution and the prediction. For the quantitative assessment on the accuracy of both ROM-GP and ROM-LSTM frameworks, L_2-norm errors of the reduced order models (with respect to FOM) for the mean vorticity and stream-function fields are tabulated in Table III. The root mean-square error or Euclidean L_2-norm error is computed by

$$\mathrm{L}_2 = ||e||^2 = \sqrt{\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} e_{i,j}^2}, \tag{34}$$

where N_x and N_y are the grid resolutions in x and y directions. For the vorticity field, the error, i.e., the difference between the predicted mean and FOM solution mean is

$$e_{i,j} = \left| \bar{\omega}_{i,j}^{\mathrm{ROM}} - \bar{\omega}_{i,j}^{\mathrm{FOM}} \right|. \tag{35}$$

TABLE IV. Computational overhead for the ROM-LSTM model trained with $R = 10$ modes. For training, CPU time is presented as per epoch for 400 samples and for testing, CPU time is presented as per time step. Note that the time step for testing is set 1×10^{-1} since the nonintrusive set up is free of numerical stability constraints.

ROM-LSTM	Training time (s)	Testing time (s)
$\sigma = 1$	8.10×10^{-2}	1.15×10^{-3}
$\sigma = 2$	1.07×10^{-1}	1.38×10^{-3}
$\sigma = 3$	1.30×10^{-1}	1.56×10^{-3}
$\sigma = 4$	1.59×10^{-1}	1.79×10^{-3}
$\sigma = 5$	1.80×10^{-1}	2.00×10^{-3}

For ROM-LSTM framework, the results are presented for $R = 10$ modes. We can observe that the prediction accuracy increases with the increase in lookback time window σ, and we can obtain a more accurate result than the ROM-GP simulation with $R = 80$ using only 10 modes in ROM-LSTM framework. We present the CPU time per time step (between $t = 10$ and $t = 100$) for ROM-LSTM framework simulations based on $R = 10$ modes and different σ in Table IV. We can observe a gradual reduction of computational time (for both training and testing) with lower values of σ. All the simulations of ROM-LSTM frameworks are done in Python programming environment and CPU time is computed as per time step. The computational time step is set to 1×10^{-1} for online testing. In our FOM simulation in FORTRAN, the CPU time per time step is about 1.17×10^{-1} s, where computational time step is set $\Delta t = 2.5 \times 10^{-5}$ due to the CFL restriction of numerical stability for our explicit forward model, a third-order Runge-Kutta integrator, on the resolution of 256×512. With the same time integration tool, the computational CPU time required by the ROM-GP approach using similar flow conditions can be found elsewhere [50]. It should be noted that the ROM-GP computations are also computed using FORTRAN programming platform, where as LSTM-ROM computations are performed using PYTHON. Even so, we observe our ROM-LSTM CPU times are on the same order as ROM-GP simulations with $R = 80$ modes (i.e., 4.84×10^{-3} s per time step). Since the ROM-GP requires a small time step to get a converged solution (i.e., see Fig. 17 in Sec. VI), the total CPU time required for the time integration of the ROM-GP model often becomes large (i.e., 1.7×10^3 s). Instead, ROM-LSTM

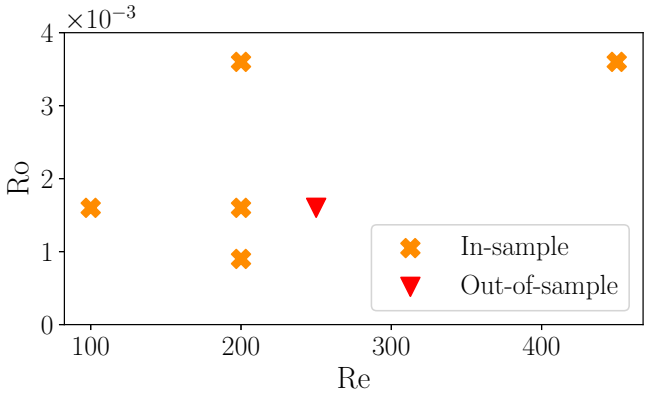

FIG. 14. In-sample and out-of-sample flow parameters used for evaluating interpolatory performance of ROM-LSTM model.

can be used with much bigger time step, which constitutes one of the strengths of this nonintrusive modeling framework.

VI. OUT-OF-SAMPLE PERFORMANCE

The numerical results in Sec. V show that the data-driven ROM-LSTM framework can predict the mean field with sufficient accuracy. On the other hand, the conventional ROM-GP framework cannot predict the mean field accurately for fewer POD modes, since the instability of ROM-GP causes errors in calculating POD coefficients. It can be argued that the ROM-LSTM has learned from the history of POD coefficients, and it is not surprising that the ROM-LSTM was able to predict the mean field correctly. The fundamental question then is, can the ROM-LSTM learn something more than the solution field obtained by simply averaging the training data set? To address this question, we test the performance of ROM-LSTM model for operating conditions which are not included in the training data set.

We generate the FOM solution for different operating conditions (Re, Ro) for the single-layer QG ocean circulation model. We construct the global basis functions using Algorithm 1. The only difference is that instead of using snapshots for only one operating condition, we select snapshots for different operating conditions. This ensures that the global basis represents dynamics for different physical conditions. The different operating conditions used for POD bases reconstruction is shown in Fig. 14. We select 180 snapshots randomly between $t = 10$ and $t = 100$ for each operating condition, and hence, the total number of snapshots utilized is 900. The first 10 modes capture around 58% of the energy. We test the performance of ROM-LSTM model for Re $= 250$ and Ro $= 1.6 \times 10^{-3}$ which lies within the design space and hence the POD bases represent the dynamics of test condition too.

We construct modal coefficients using a forward transform through projection as described in Algorithm 3 to generate the training data set for ROM-LSTM model. Since, there are five operating conditions in the in-sample design space, we get 2000 training samples (400 samples from $t = 10$ to $t = 50$ for five different operating conditions). The input to the LSTM network is modified slightly with Re and Ro included in input features. Therefore, the LSTM model can be given as

$$\mathcal{M} : \{\mathrm{Re}, \mathrm{Ro}, a_1^{(n)}, \ldots, a_R^{(n)}; \ldots; a_1^{(n-\sigma+1)}, \ldots, a_R^{(n-\sigma+1)}\} \Rightarrow \{a_1^{(n+1)}, \ldots, a_R^{(n+1)}\}. \tag{36}$$

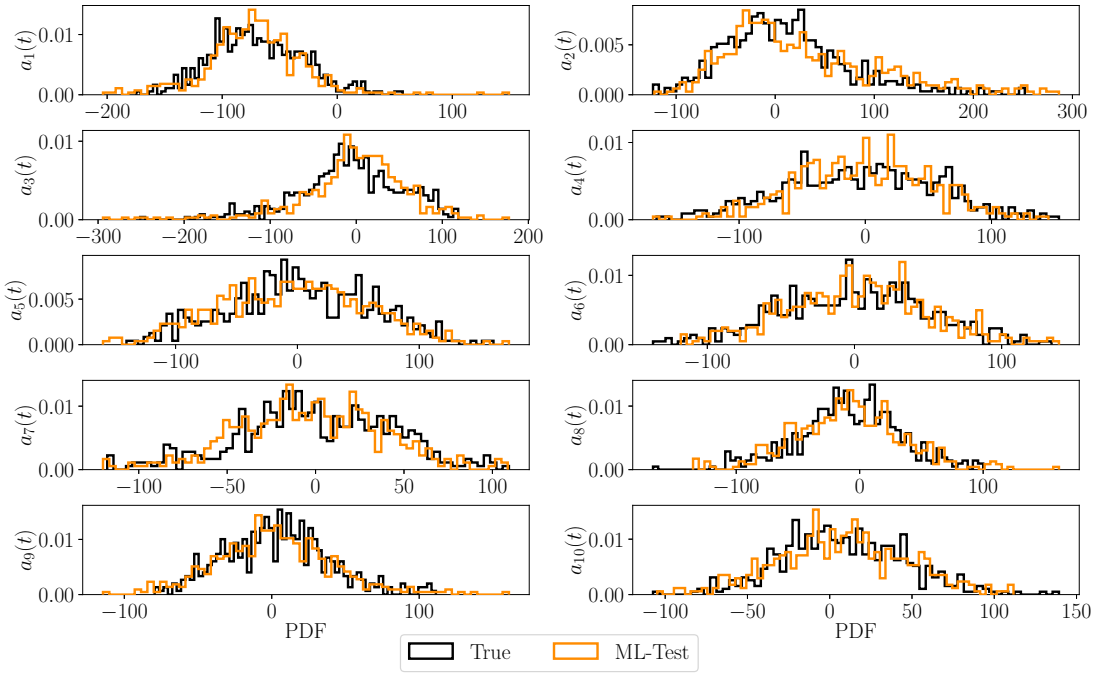

FIG. 15. PDF for true and predicted modal coefficients between $t = 50$ and $t = 100$ for ROM-LSTM simulation at Re $= 250$ and Ro $= 1.6 \times 10^{-3}$. Note that the LSTM model is trained with $R = 10$ modes and $\sigma = 5$. The training is done using true modal coefficients between $t = 10$ and $t = 50$ for (Re, Ro) $= (100, 1.6 \times 10^{-3})$, $(200, 0.9 \times 10^{-3})$, $(200, 1.6 \times 10^{-3})$, $(200, 3.6 \times 10^{-3})$, and $(450, 3.6 \times 10^{-3})$.

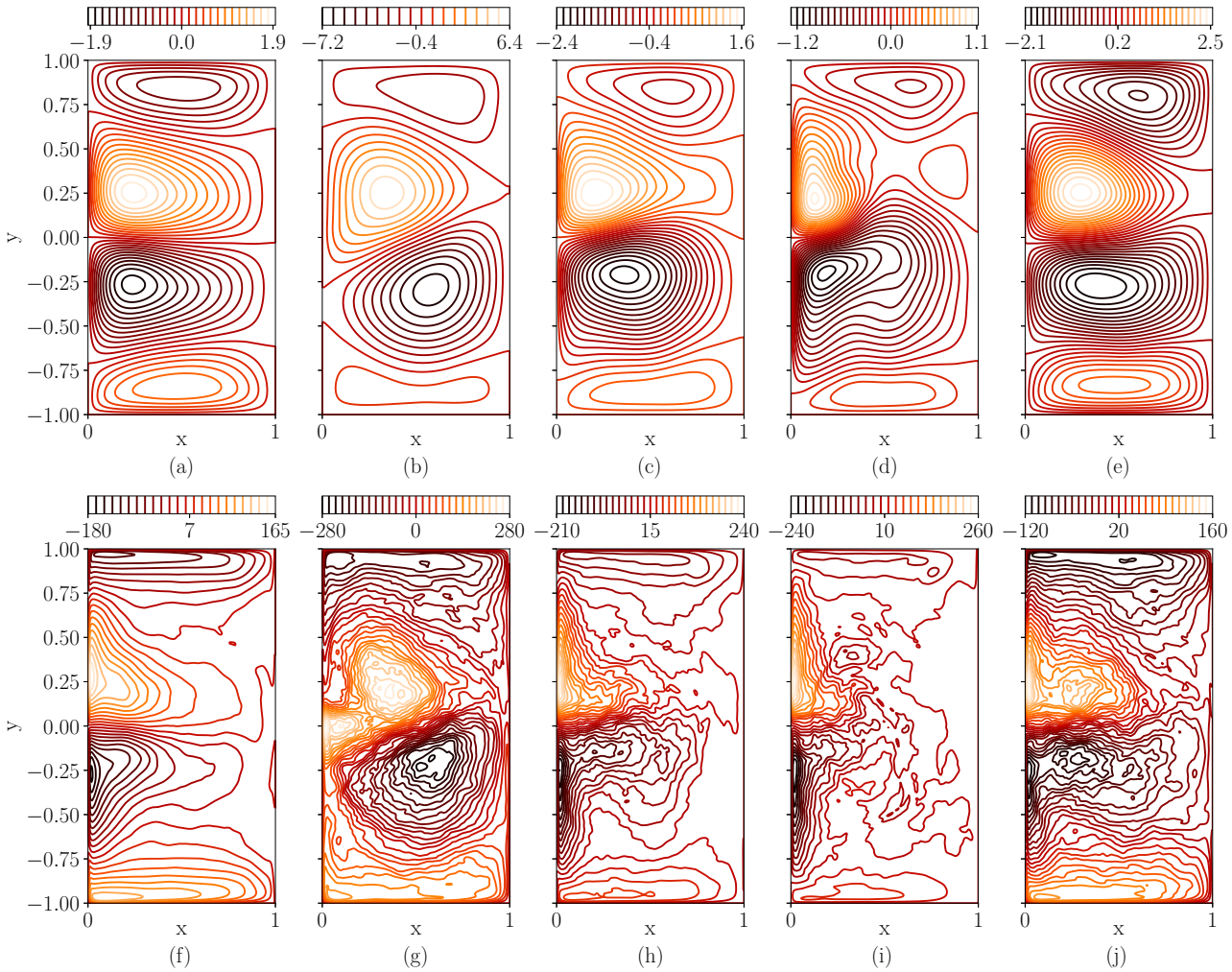

FIG. 16. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on the number of modes to train the LSTM model at Re = 250 and Ro = 1.6×10^{-3} flow condition. The model is trained for five different operating conditions and is evaluated for out-of-sample parameters. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 2$ modes, (c) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 4$ modes, (d) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 8$ modes, (e) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $R = 10$ modes, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 2$ modes, (h) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 4$ modes, (i) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 8$ modes, and (j) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $R = 10$ modes. Note that the LSTM models are trained with $\sigma = 5$.

This approach was found to be robust and accurate for nonlinear transient flows [124].

We run a similar numerical experiment with a different number of modes and $\sigma = 5$ as in Sec. V to examine the effectiveness of ROM-LSTM framework for prediction of the mean field for out-of-design parameters. Figure 15 shows the PDF for true and predicted modal coefficients for Re = 250 and Ro = 1.6×10^{-3} test case. The statistics of all modal coefficient are accurately captured by the ROM-LSTM framework which will lead to accurate prediction of mean field. Figure 16 shows the performance of interpolatory ROM-LSTM model in predicting mean field for different numbers of modes. We use the same lookback time window $\sigma = 5$ for all these cases. The ROM-LSTM model can capture the four-gyre even when only two POD modes are used since the first two modes are the most dominant modes. As we increase the number of modes, we see an improvement in the prediction of mean field. The results in Fig. 16 demonstrate that ROM-LSTM model can also be used for predicting the mean field for parameters that are different from the one included in the training data set. If we have a high-fidelity data set for different physical conditions (different dimensionless numbers characterizing the flow, in the present case Re, Ro), a robust surrogate model can be obtained using LSTM network which can go beyond the training data.

In addition to acting as a robust surrogate model, we highlight one more advantage of nonintrusive ROM framework concerning large time step that can be used. The ROM-GP results shown in Sec. V were obtained by using third-order Runge-Kutta numerical scheme with the time step $\Delta t = 2.5 \times 10^{-5}$. The same time step is used for calculating the FOM solution. In the case of the FOM, the time step that can be used is restricted by the CFL condition. In the case of ROM-GP, we can also use a higher time step to speed up the time integration. It is difficult to estimate the stability region for ordinary differential equations used in Galerkin

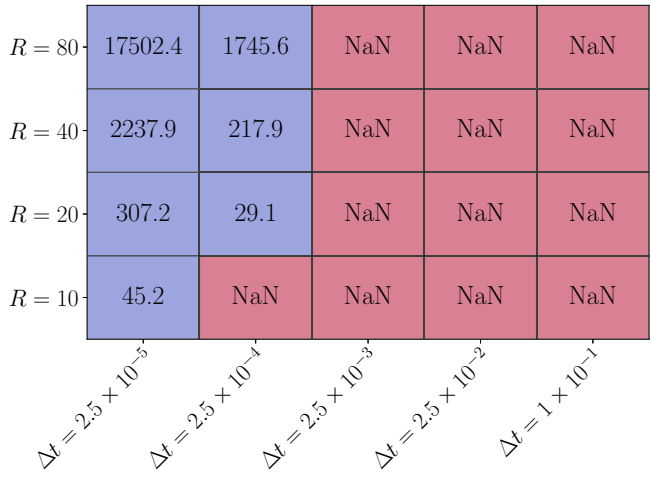

FIG. 17. Computational overhead for the ROM-GP framework for different number of modes and different time step size Δt for $\mathrm{Re} = 250$ and $\mathrm{Ro} = 1.6 \times 10^{-3}$ test condition. The number in each box presents the CPU time required for integration of Galerkin projection ODEs from time $t = 10$ to $t = 100$. NaN means the solution diverges after few time steps.

projection. We perform numerical experiments with a different time step sizes and different numbers of modes for $\mathrm{Re} = 250$ and $\mathrm{Ro} = 1.6 \times 10^{-3}$. We find that for time step size larger than $\Delta t = 2.5 \times 10^{-3}$, the solution diverges for all cases as shown in Fig. 17. As we increase the number of modes, the computational time of time integration also increases for ROM-GP framework. Even if the solution does not diverge for fewer modes and smaller time step size, the predicted modal coefficients are much larger than the true modal coefficients (as seen in Fig. 5).

The ROM-LSTM framework is purely data-driven, and hence it is not restricted by time step size that can be used between two snapshots. For all our previous numerical experiments, the data snapshots were separated by time step size $\Delta t = 1 \times 10^{-1}$. We can also train the LSTM network using data snapshots that are separated by some other time step size. We illustrate the performance of ROM-LSTM framework using the same data snapshots separated by time step size $\Delta t = 2 \times 10^{-1}$ (i.e., we use every other data snapshots between $t = 10$ and $t = 50$, and therefore, we have 200 training examples in this case). Figure 18 shows field plots for vorticity and stream function for $\mathrm{Re} = 250$ and $\mathrm{Ro} = 1.6 \times 10^{-3}$ test case using data snapshots separated by $\Delta t = 1 \times 10^{-1}$ for different σ values with 10 POD modes. This is similar to what we saw in Sec. V for $\mathrm{Re} = 450$ and $\mathrm{Ro} = 3.6 \times 10^{-2}$ test case. Furthermore, we perform the same numerical experiment with time step size $\Delta t = 2 \times 10^{-1}$ for $\mathrm{Re} = 250$ and $\mathrm{Ro} = 1.6 \times 10^{-3}$ test case, and the field plots are given in Fig. 19. We observe that the mean field predicted by ROM-LSTM framework is not as accurate as the mean field predicted with $\Delta t = 1 \times 10^{-1}$. We see that the mean field prediction improves with larger lookback time window σ. However, the modal coefficients and the solution field predicted with larger time step size are bounded and stable. This can be considered as one of the main accomplishments of this approach in using nonintrusive ROM framework for flow-control applications and surrogate models where computational performance is one of the main bottlenecks.

VII. SUMMARY AND CONCLUSIONS

In this paper, we propose an efficient and robust fully nonintrusive ROM framework to capture the large spatiotemporal scale of fluctuating quasistationary systems. Due to the robustness and stability of LSTM recurrent neural network in predicting chaotic dynamical systems, we consider LSTM architecture to develop our data-driven ROM, denoted as ROM-LSTM in this paper. As an example of large-scale turbulent flows exhibiting a wide range of spatiotemporal scales, we investigate the reduced order modeling of a simple general ocean circulation model, single-layer QG turbulence, to assess the predictive performance of our proposed ROM-LSTM framework. It was previously observed that the conventional physics-based (or intrusive) ROM of QG model requires a large number of POD modes to yield stable and physical flow dynamics. However, the proposed ROM-LSTM framework shows a very promising improvement in reduced order modeling since only a few modes are able to capture a physical solution without any prior knowledge about the underlying governing equations. We first demonstrate that the conventional Galerkin projection ROM approach yields nonphysical predictions when we use a small number of representative modes. Although ROM-GP converges to a more physical solution when increasing the number of modes, it does not seem to capture the intermittent bursts appearing in the dynamics of the first few most energetic modes. However, the proposed ROM-LSTM approach is able to capture these bursts and yields remarkably accurate results even when using a small number of modes.

The proposed methodology consists of two phases: offline training and online testing or prediction phase. Initially, we collect the high-fidelity simulation or experimental data snapshots for a certain flow condition. The data snapshots are collected up to a certain time of the FOM simulation for training. Then we do a mapping of the high-resolution instantaneous data snapshots into a reduced order, i.e., low-dimensional space through POD transform. In this process, we generate POD basis functions of the field variables and time dependent modal coefficients for training the LSTM architecture. The LSTM architecture is trained for the modal coefficients based on a preselected lookback time window, σ. In the online phase, the trained model is used to predict the modal coefficients recursively for the total time based on initial time history and σ. Finally, we reconstruct the mean fields for analyses using the predicted coefficients, precomputed basis functions, and mean field values.

We demonstrate the performance of the ROM-LSTM through time series evolution of modal coefficients and mean vorticity and stream-function fields. To assess the performance of the proposed model, the ROM-LSTM predictions are compared with the high-dimensional solutions as well as with the conventional Galerkin projection-based ROM (ROM-GP) solutions. We also compare the PDF of true and predicted modal coefficients to estimate the ability of ROM-LSTM framework to capture mean flow dynamics. We find that the

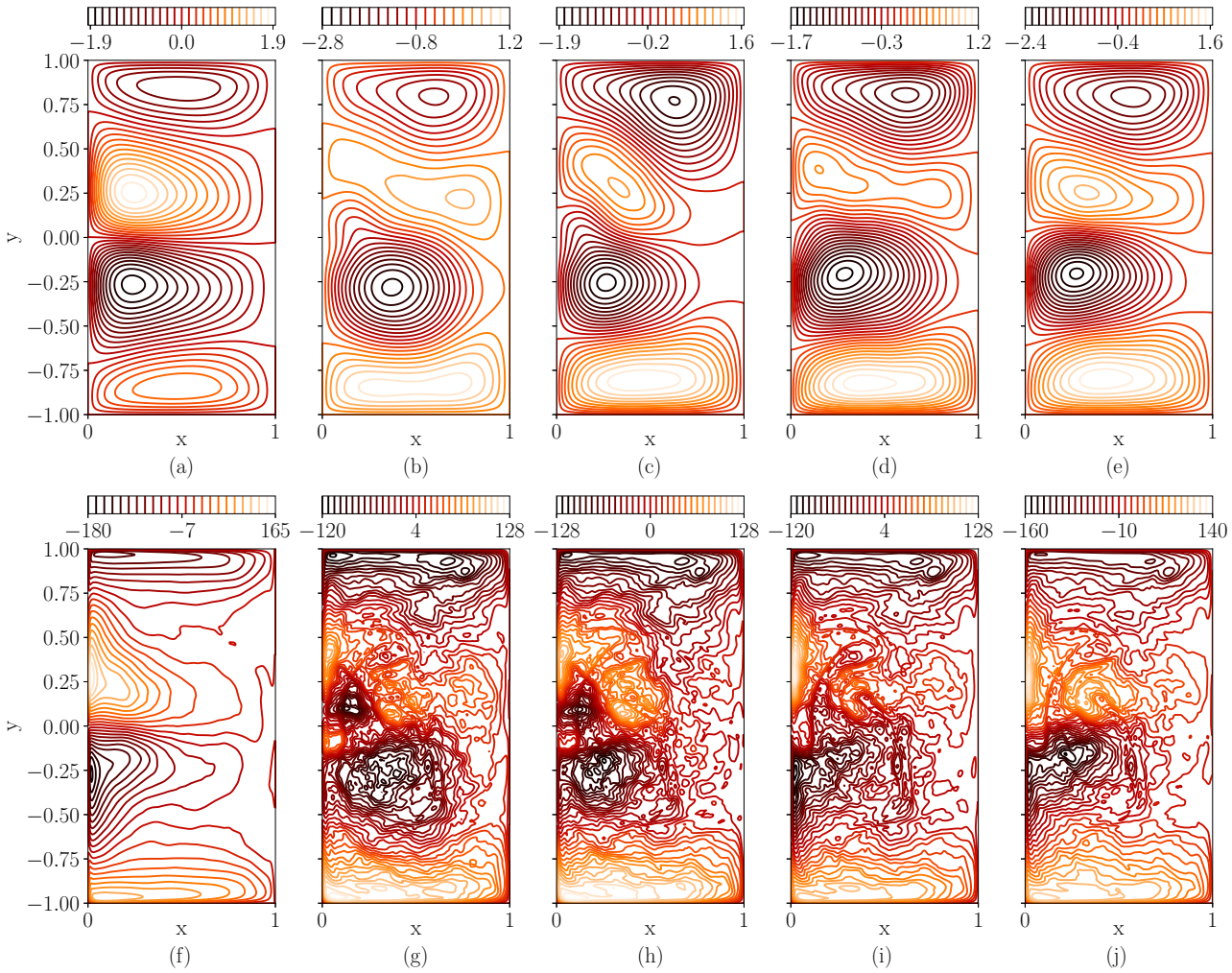

FIG. 18. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on different lookback time windows, σ and LSTM training with $R = 10$ modes at Re $= 250$ and Ro $= 1.6 \times 10^{-3}$. The model is trained using 400 snapshots stored at time interval $\Delta t = 1 \times 10^{-1}$ from $t = 10$ to $t = 50$. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (c) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (d) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, (e) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (h) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (i) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, and (j) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$.

ROM-LSTM predictions are stable and accurate even with only a couple of POD modes. On the other hand, the ROM-GP framework, as expected, requires a very large number of modes to obtain a physically stable solution, since the ROM-GP framework is susceptible to numerical instability in quasistationary flow fields. We further observe that the ROM-LSTM framework gives accurate and physical predictions based on a few time history data points. Indeed, if we increase the value of σ, the prediction accuracy will increase, but the computational cost of offline training and online prediction will also go up. To quantify the accuracy of the prediction of ROM-LSTM framework, we present the L_2-norm errors for ROM-GP and ROM-LSTM frameworks, which show that the proposed framework trained with 10 modes and $\sigma = 5$ gets a better accuracy than the ROM-GP predictions with 40 or 80 modes.

We extend the ROM-LSTM framework as a surrogate model and demonstrate the performance of the present approach for unseen training data. It is seen that the ROM-LSTM trained using a set of parameters can produce a bounded mean solution field for parameters which are not included in the training. This shows that if the high-fidelity data set is available for different parameters characterizing the flow, then these parameters can also be included during the training process to predict the solution field for out-of-sample parameters with good accuracy. This opens up the application of ROM-LSTM framework in surrogate modeling, and flow control areas. We analyze the predictive capability of ROM-LSTM framework for different time step sizes between data snapshots and show that the mean field prediction remains bounded even with a large time step size. This can be considered as one of the major advantages of nonintrusive ROM over the conventional ROM-GP framework, which is restricted by certain stability conditions.

Based on our findings, we conclude that the ROM-LSTM framework provides a stable emulator for large-scale

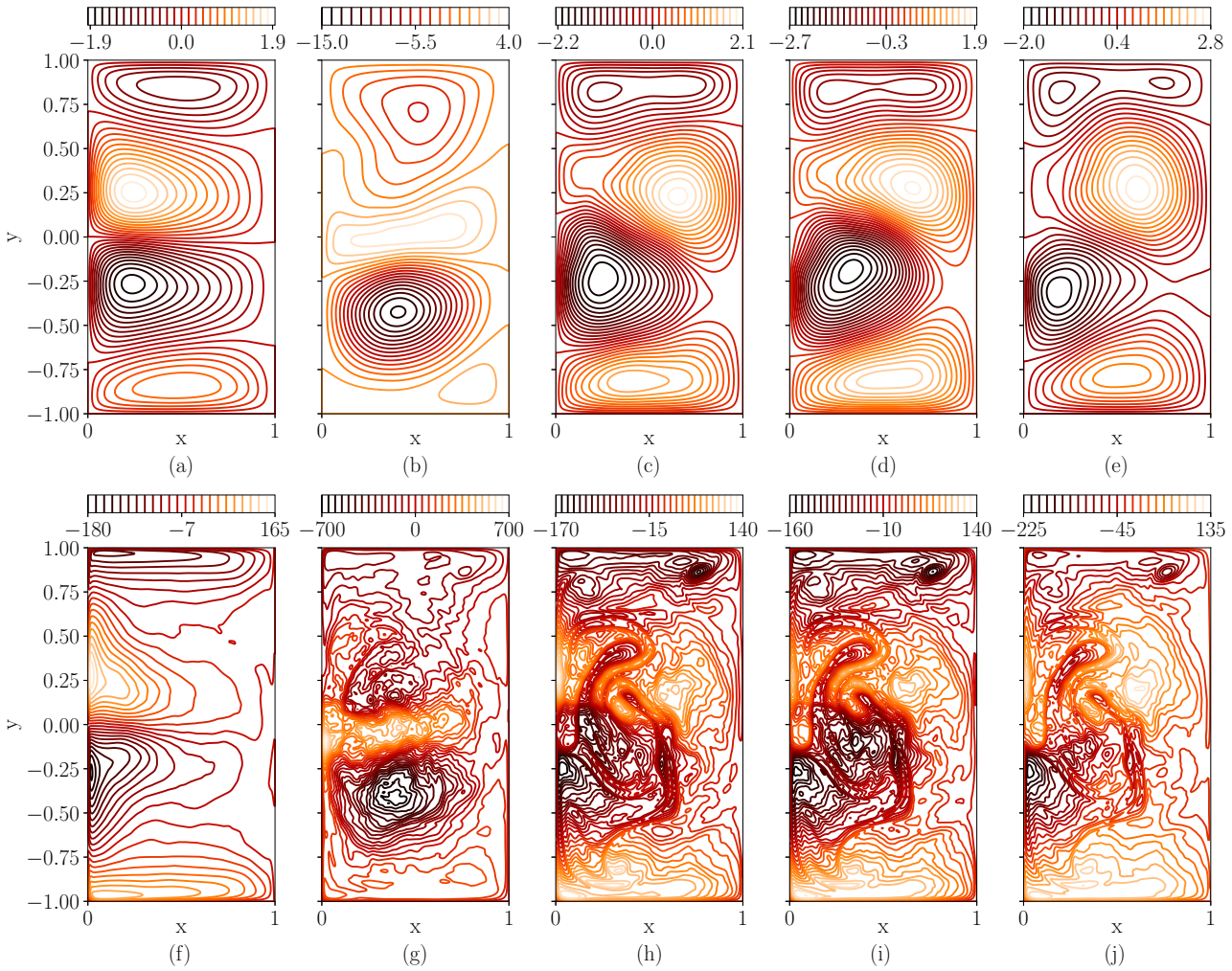

FIG. 19. Mean stream-function and vorticity fields obtained by the ROM-LSTM simulation based on different lookback time windows, σ and LSTM training with $R = 10$ modes at Re $= 250$ and Ro $= 1.6 \times 10^{-3}$. The model is trained using 200 snapshots sampled at time interval $\Delta t = 2 \times 10^{-1}$ from $t = 10$ to $t = 50$. (a) ψ_{FOM} at a resolution of 256×512, (b) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (c) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (d) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, (e) $\psi_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$, (f) ω_{FOM} at a resolution of 256×512, (g) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 2$, (h) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 3$, (i) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 4$, and (j) $\omega_{\text{ROM-LSTM}}$ for LSTM training with $\sigma = 5$.

quasistationary flows in terms of prediction and reduced order modeling. Since the ROM-LSTM framework is fully nonintrusive, it does not rely on the governing equations to obtain the solution, which means that there are no numerical constraints while predicting the solutions. Additionally, it is computationally more efficient to predict the solution using a trained model rather than the physics-based approach of solving ODEs. Hence, the proposed ROM-LSTM framework can be considered a very promising approach in developing a robust and efficient ROMs for large-scale flows with chaotic spatiotemporal behavior. In our future studies, we will investigate how physics can be incorporated into ML-based emulators to enforce certain symmetries and physical considerations to build more robust and interpretable ML methods [125]. We will also focus on testing the ROM-LSTM framework in more complex three-dimensional turbulent flow problems. Furthermore, we plan to improve extrapolation capabilities of the existing framework based on our findings, and implement the proposed approaches in several ROM-in-the-loop applications, such as flow control, parameter estimation, uncertainty quantification, and data assimilation.

ACKNOWLEDGEMENT

This material is based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award No. DE-SC0019290. O.S. gratefully acknowledges their support.

[1] C. A. Mack, Fifty years of Moore's law, IEEE Trans. Semiconductor Manuf. **24**, 202 (2011).

[2] J. R. Powell, The quantum limit to Moore's law, Proc. IEEE **96**, 1247 (2008).

[3] M. M. Waldrop, The chips are down for Moore's law, Nature News **530**, 144 (2016).

[4] T. N. Theis and H.-S. P. Wong, The end of Moore's law: A new beginning for information technology, Comput. Sci. Eng. **19**, 41 (2017).

[5] S. Kumar, in *High-Speed and Lower Power Technologies: Electronics and Photonics*, edited by J. H. Choi (CRC Press, Boca Raton, FL, 2018) pp. 144–147.

[6] P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction* (Springer Science & Business Media, Berlin, 2006).

[7] H. Tennekes *et al.*, *A First Course in Turbulence* (MIT Press, Cambridge, MA, 1972).

[8] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Unitext (Springer, New York, 2015), Vol. 92.

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

[10] P. Benner, S. Gugercin, and K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. **57**, 483 (2015).

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

[12] B. R. Noack, M. Morzynski, and G. Tadmor, *Reduced-Order Modelling for Flow Control*, International Centre for Mechanical Sciences Vol. 528 (Springer Science & Business Media, Berlin, 2011).

[13] S. L. Brunton and B. R. Noack, Closed-loop turbulence control: Progress and challenges, Appl. Mech. Rev. **67**, 050801 (2015).

[14] K. Ito and S. Ravindran, A reduced-order method for simulation and control of fluid flows, J. Comput. Phys. **143**, 403 (1998).

[15] D. Daescu and I. Navon, A dual-weighted approach to order reduction in 4DVAR data assimilation, Mon. Weather Rev. **136**, 1026 (2008).

[16] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo, A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition, Int. J. Numer. Methods Fluids **53**, 1571 (2007).

[17] S. Ullmann and J. Lang, in *Sparse Grids and Applications–Munich 2012*, edited by J. Garcke and D. Pflüger (Springer International Publishing, Cham, Switzerland, 2014), pp. 295–315.

[18] P. Chen and A. Quarteroni, A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods, J. Comput. Phys. **298**, 176 (2015).

[19] B. Haasdonk, K. Urban, and B. Wieland, Reduced basis methods for parameterized partial differential equations with stochastic influences using the Karhunen–Loeve expansion, SIAM/ASA J. Uncertainty Quantif. **1**, 79 (2013).

[20] K. Carlberg, C. Bou-Mosleh, and C. Farhat, Efficient nonlinear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Intl. J. Numer. Methods Eng. **86**, 155 (2011).

[21] D. J. Lucia, P. S. Beran, and W. A. Silva, Reduced-order modeling: New approaches for computational physics, Prog. Sci. **40**, 51 (2004).

[22] D. Bistrian and I. Navon, An improved algorithm for the shallow water equations model reduction: Dynamic Mode Decomposition vs POD, Int. J. Numer. Methods Fluids **78**, 552 (2015).

[23] T. A. Brenner, R. L. Fontenot, P. G. Cizmas, T. J. O'Brien, and R. W. Breault, A reduced-order model for heat transfer in multiphase flow and practical aspects of the proper orthogonal decomposition, Comput. Chem. Eng. **43**, 68 (2012).

[24] J. Burkardt, M. Gunzburger, and H.-C. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows, Comput. Methods Appl. Mech. Eng. **196**, 337 (2006).

[25] A. Iollo, S. Lanteri, and J.-A. Désidéri, Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations, Theor. Comput. Fluid Dyn. **13**, 377 (2000).

[26] B. A. Freno and P. G. Cizmas, A proper orthogonal decomposition method for nonlinear flows with deforming meshes, Int. J. Heat Fluid Flow **50**, 145 (2014).

[27] J. L. Lumley, The structure of inhomogeneous turbulent flows, in *Atmospheric Turbulence and Radio Wave Propagation*, edited by A. M. Yaglom and V. I. Tatarski (Nauka, Moscow, 1967).

[28] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, A survey of model reduction methods for large-scale systems, Contemp. Math. **280**, 193 (2001).

[29] F. Chinesta, P. Ladeveze, and E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. **18**, 395 (2011).

[30] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations* (Springer, New York, 2015).

[31] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza, in *Reduced Order Methods for Modeling and Computational Reduction*, edited by A. Quarteroni and G. Rozza (Springer, Milan, 2013).

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

[33] N. Aubry, R. Guyonnet, and R. Lima, Spatio-temporal symmetries and bifurcations via bi-orthogonal decompositions, J. Nonlinear Sci. **2**, 183 (1992).

[34] C. W. Rowley, T. Colonius, and R. M. Murray, Model reduction for compressible flows using POD and Galerkin projection, Physica D **189**, 115 (2004).

[35] M. Sieber, C. O. Paschereit, and K. Oberleithner, Spectral proper orthogonal decomposition, J. Fluid Mech. **792**, 798 (2016).

[36] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza, *Reduced Order Methods for Modeling and Computational Reduction* (Springer, 2014), pp. 235–273.

[37] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows, Theor. Comput. Fluid Dyn. **23**, 213 (2009).

[38] D. Rempfer, On low-dimensional Galerkin models for fluid flow, Theor. Comput. Fluid Dyn. **14**, 75 (2000).

[39] C. W. Rowley and D. R. Williams, Dynamics and control of high-Reynolds-number flow over open cavities, Annu. Rev. Fluid Mech. **38**, 251 (2006).

[40] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, J. Fluid Mech. **192**, 115 (1988).

[41] D. Rempfer and H. F. Fasel, Evolution of three-dimensional coherent structures in a flat-plate boundary layer, J. Fluid Mech. **260**, 351 (1994).

[42] D. Rempfer and H. F. Fasel, Dynamics of three-dimensional coherent structures in a flat-plate boundary layer, J. Fluid Mech. **275**, 257 (1994).

[43] W. Cazemier, R. Verstappen, and A. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids **10**, 1685 (1998).

[44] X. Xie, D. Wells, Z. Wang, and T. Iliescu, Approximate deconvolution reduced order modeling, Comput. Methods Appl. Mech. Eng. **313**, 512 (2017).

[45] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, J. Fluid Mech. **497**, 335 (2003).

[46] S. Sirisup and G. E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, J. Comput. Phys. **194**, 92 (2004).

[47] L. Cordier, B. R. Noack, G. Tissot, G. Lehnasch, J. Delville, M. Balajewicz, G. Daviller, and R. K. Niven, Identification strategies for model-based control, Exp. Fluids **54**, 1580 (2013).

[48] J. Östh, B. R. Noack, S. Krajnović, D. Barros, and J. Borée, On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body, J. Fluid Mech. **747**, 518 (2014).

[49] O. San and T. Iliescu, Proper orthogonal decomposition closure models for fluid flows: Burgers equation, Intl. J. Numer. Anal. Model. B **5**, 217 (2014).

[50] Sk. M. Rahman, S. E. Ahmed and O. San, A dynamic closure modeling framework for model order reduction of geophysical flows, Phys. Fluids **31**, 046602 (2019).

[51] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison, Comput. Methods Appl. Mech. Eng. **237–240**, 10 (2012).

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

[53] S. Rahman, O. San, and A. Rasheed, A hybrid approach for model order reduction of barotropic quasi-geostrophic turbulence, Fluids **3**, 86 (2018).

[54] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis, Data-assisted reduced-order modeling of extreme events in complex dynamical systems, PloS ONE **13**, e0197704 (2018).

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

[56] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model, Chaos **28**, 041101 (2018).

[57] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. **378**, 686 (2019).

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

[59] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, arXiv:1801.01236 (2018).

[60] C. Lee, J. Kim, D. Babcock, and R. Goodman, Application of neural networks to turbulence control for drag reduction, Phys. Fluids **9**, 1740 (1997).

[61] R. Maulik, O. San, A. Rasheed, and P. Vedula, Data-driven deconvolution for large eddy simulations of Kraichnan turbulence, Phys. Fluids **30**, 125109 (2018).

[62] W. E. Faller and S. J. Schreck, Unsteady fluid mechanics applications of neural networks, J. Aircr. **34**, 48 (1997).

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

[64] Q. Wang, J. S. Hesthaven, and D. Ray, Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem, J. Comput. Phys. **384**, 289 (2019).

[65] A. Moosavi, R. Stefanescu, and A. Sandu, Efficient construction of local parametric reduced order models using machine learning techniques, arXiv:1511.02909 (2015).

[66] J. N. Kani and A. H. Elsheikh, DR-RNN: A deep residual recurrent neural network for model reduction, arXiv:1709.00939 (2017).

[67] S. Brunton, B. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. **52** (2019), doi: 10.1146/annurev-fluid-010719-060214.

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

[69] P. A. Durbin, Some recent developments in turbulence closure modeling, Annu. Rev. Fluid Mech. **50**, 77 (2018).

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

[71] J. C. B. Gamboa, Deep learning for time-series analysis, arXiv:1701.01887 (2017).

[72] F. A. Gers, D. Eck, and J. Schmidhuber, Applying LSTM to time series predictable through time-window approaches, in *Neural Nets WIRN Vietri-01*, edited by R. Tagliaferri, and M. Marinaro, (Springer, London, 2002), pp. 193–200.

[73] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Intl. J. Numer. Methods Fluids **86**, 255 (2018).

[74] K. Yeo and I. Melnyk, Deep learning algorithm for data-driven simulation of noisy dynamical system, J. Comput. Phys. **376**, 1212 (2019).

[75] H. Sak, A. Senior, and F. Beaufays, in *Fifteenth Annual Conference of the International Speech Communication Association* (2014).

[76] A. T. Mohan and D. V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, arXiv:1804.09269 (2018).

[77] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, Proc. R. Soc. A **474**, 20170844 (2018).
[78] D. D. Holm and B. T. Nadiga, Modeling mesoscale turbulence in the barotropic double-gyre circulation, J. Phys. Oceanogr. **33**, 2355 (2003).
[79] O. San, A. E. Staples, Z. Wang, and T. Iliescu, Approximate deconvolution large eddy simulation of a barotropic ocean circulation model, Ocean Model. **40**, 120 (2011).
[80] A. Majda and X. Wang, *Nonlinear Dynamics and Statistical Theories for Basic Geophysical Flows* (Cambridge University Press, Cambridge, 2006).
[81] W. R. Holland and P. B. Rhines, An example of eddy-induced ocean circulation, J. Phys. Oceanogr. **10**, 1010 (1980).
[82] W. H. Munk and C. I. Wunsch, Observing the ocean in the 1990s, Philos. Trans. R. Soc. London A **307**, 439 (1982).
[83] A. Griffa and R. Salmon, Wind-driven ocean circulation and equilibrium statistical mechanics, J. Mar. Res. **47**, 457 (1989).
[84] P. F. Cummins, Inertial gyres in decaying and forced geostrophic turbulence, J. Mar. Res. **50**, 545 (1992).
[85] R. J. Greatbatch and B. Nadiga, Four-gyre circulation in a barotropic model with double-gyre wind forcing, J. Phys. Oceanogr. **30**, 1461 (2000).
[86] B. T. Nadiga and L. G. Margolin, Dispersive-dissipative eddy parameterization in a barotropic model, J. Phys. Oceanogr. **31**, 2525 (2001).
[87] H. U. Sverdrup, Wind-driven currents in a baroclinic ocean; with application to the equatorial currents of the eastern Pacific, Proc. Nat. Acad. Sci. USA **33**, 318 (1947).
[88] T. T. Medjo, Numerical simulations of a two-layer quasi-geostrophic equation of the ocean, SIAM J. Numer. Anal. **37**, 2005 (2000).
[89] A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I, J. Comput. Phys. **1**, 119 (1966).
[90] S. Gottlieb and C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Math. Comput. **67**, 73 (1998).
[91] T. M. Özgökmen and E. P. Chassignet, Emergence of inertial gyres in a two-layer quasigeostrophic ocean model, J. Phys. Oceanogr. **28**, 461 (1998).
[92] O. San, Numerical assessments of ocean energy extraction from western boundary currents using a quasi-geostrophic ocean circulation model, Intl. J. Marine Energy **16**, 12 (2016).
[93] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, Q. Appl. Math. **45**, 561 (1987).
[94] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in FORTRAN* (Cambridge University Press, New York, 1992).
[95] K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. **40**, 492 (2002).
[96] M.-L. Rapún and J. M. Vega, Reduced order models based on local POD plus Galerkin projection, J. Comput. Phys. **229**, 3046 (2010).
[97] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).
[98] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).
[99] D. E. Rumelhart *et al.*, Learning representations by back-propagating errors, Cognitive Model. **5**, 1 (1988).
[100] Y. Bengio *et al.*, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. **5**, 157 (1994).
[101] K. Yao, T. Cohn, K. Vylomova, K. Duh, and C. Dyer, Depth-gated LSTM, arXiv:1508.03790 (2015).
[102] R. Jozefowicz, W. Zaremba, and I. Sutskever, in *International Conference on Machine Learning* (2015), pp. 2342–2350.
[103] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. **28**, 2222 (2016).
[104] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, A clockwork RNN, in *Proceedings of the 31st International Conference on Machine Learning* edited by E. P. Xing and T. Jebara, (PMLR, Bejing, China, 2014), Vol. 32.
[105] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Comput. **9**, 1735 (1997).
[106] N. Kalchbrenner, I. Danihelka, and A. Graves, Grid long short-term memory, arXiv:1507.01526 (2015).
[107] K. Yeo, Short note on the behavior of recurrent neural network for noisy dynamical system, arXiv:1904.05158 (2019).
[108] F. Chollet *et al.*, Keras, https://keras.io (2015).
[109] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2014).
[110] O. San and T. Iliescu, A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation, Adv. Comput. Math. **41**, 1289 (2015).
[111] D. Rempfer, Ph.D. thesis, University of Stuttgart, 1991.
[112] W. Cazemier, Ph.D. thesis, Rijksuniversiteit Groningen, 1997.
[113] B. Cushman-Roisin and J.-M. Beckers, *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects* (Academic Press, Waltham, MA, 2011).
[114] H. E. Hurst, Long-term storage capacity of reservoirs, Trans. Am. Soc. Civ. Eng. **116**, 770 (1951).
[115] B. S. Chandra, C. S. Sastry, and S. Jana, in *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)* (IEEE, Piscataway, 2013), pp. 350–354.
[116] D. Koutsoyiannis, Climate change, the Hurst phenomenon, and hydrological statistics, Hydrol. Sci. J. **48**, 3 (2003).
[117] S. Lahmiri and M. Boukadoum, New approach for automatic classification of Alzheimer's disease, mild cognitive impairment and healthy brain magnetic resonance images, Healthcare Tech. Lett. **1**, 32 (2014).
[118] K. E. Bassler, G. H. Gunaratne, and J. L. McCauley, Markov processes, Hurst exponents, and nonlinear diffusion equations: With application to finance, Physica A **369**, 343 (2006).
[119] A. Carbone, G. Castelli, and H. E. Stanley, Time-dependent Hurst exponent in financial time series, Physica A **344**, 267 (2004).
[120] B. Qian and K. Rasheed, in *IASTED Conference on Financial Engineering and Applications* (2004), pp. 203–209.
[121] B. B. Mandelbrot and J. R. Wallis, Noah, Joseph, and operational hydrology, Water Resour. Res. **4**, 909 (1968).

[122] B. B. Mandelbrot and J. R. Wallis, Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence, Water Resour. Res. **5**, 967 (1969).

[123] O. San and A. E. Staples, An efficient coarse grid projection method for quasigeostrophic models of large-scale ocean circulation, Int. J. Multiscale Comput. Eng. **11** (2013).

[124] O. San, R. Maulik, and M. Ahmed, An artificial neural network framework for reduced order modeling of transient flows, Commun. Nonlinear Sci. Numer. Sim. **77**, 271 (2019).

[125] M. G. Bergomi, P. Frosini, D. Giorgi, and N. Quercioli, Towards a topological-geometrical theory of group equivariant non-expansive operators for data analysis and machine learning, Nature Machine Intell. **1**, 423 (2019).