Extreme learning machine for reduced order modeling of turbulent geophysical flows

Omer San and Romit Maulik

Phys. Rev. E **97**, 042322 — Published 30 April 2018

DOI: 10.1103/PhysRevE.97.042322

Extreme learning machine for reduced order modeling of turbulent geophysical flows

Omer San[1, *] and Romit Maulik[1]
[1]*School of Mechanical and Aerospace Engineering,*
Oklahoma State University, Stillwater, OK 74078, USA
(Dated: April 18, 2018)

We investigate the application of artificial neural networks to stabilize proper orthogonal decomposition based reduced order models for quasi-stationary geophysical turbulent flows. An extreme learning machine concept is introduced for computing an eddy-viscosity closure dynamically to take into account the effects of the truncated modes. We consider a four-gyre wind-driven ocean circulation problem as our prototype setting to assess the performance of the proposed data-driven approach. Our framework provides a significant reduction in computational time and effectively retains the dynamics of the full-order model during the forward simulation period beyond the training data set. Furthermore, we show that the method is robust for larger choices of time steps and can be used as an efficient and reliable tool for long time integration general circulation models.

I. INTRODUCTION

The spatiotemporal complexity of many applications in the computational sciences leads to very large-scale dynamical systems whose simulations make overwhelming and unmanageable demands on computational resources. Indeed, many problems remain intractable when multiple forward full-order numerical simulations are required. Since the computational cost of these high fidelity simulations is prohibitive, model order reduction approaches, also known as reduced order models (or ROMs), are commonly used to reduce this computational burden in many applications (e.g., see [1] for a review of closed-loop control applications in fluid turbulence, and [2–4] for a discussion of variational data assimilation applications in weather and climate modeling). A number of recent review articles have addressed the strengths of several modal analysis, reduced basis and model reduction techniques [5–7]. In their survey, dedicated primarily to the reduced order modeling for fluid analysis and control, Rowley and Dawson [6] have discussed several techniques including proper orthogonal decomposition (POD), balanced truncation and balanced POD, eigensystem realization algorithms (ERA), dynamic mode decomposition (DMD) and Koopman operator theory with attention devoted to the similarities and analogies between these methods. An excellent overview and introduction to such techniques may also be found in [7].

In this study, we consider the POD framework in combination with the Galerkin projection procedure [8], which is one of the prominent approaches for generating ROMs for nonlinear systems [9–11]. POD extracts the most energetic modes (usually from high-fidelity experimental or numerical data), which are expected to contain the dominant statistical characteristics of these systems. It is therefore possible to obtain good approximations to the high-fidelity data with a few POD modes in which fine scale details are embedded. The resulting systems are low dimensional (due to truncation) but dense and provide robust surrogate models for forward simulations. It has been widely used in various disciplines under a variety of different names (e.g., see [12] for an excellent historical discussion).

Although the standard Galerkin projection provides an efficient way to generate ROMs, its applicability to complex systems is limited primarily due to errors associated with the truncation of POD modes. The limitation is more prominent in turbulent flow systems where an intense scale separation leads to insufficient embedding of dynamics within a feasibly small number of modes. To model the effects of the discarded modes, several closure models are devised (see for instance [13–18]) which serve a dual purpose: that of numerical stabilization as well as statistical fidelity preservation. Following the large eddy simulation (LES) ideas, it has been shown that the eddy viscosity concept provides an efficient framework to account the effect of the truncated modes [19–21]. In this study, we put forth a robust dynamic procedure for computing the modal eddy viscosities in order to stabilize the ROMs. The novelty of our approach stems from the design of an artificial neural network (ANN) architecture to predict the magnitude of the mode dependent eddy viscosity dynamically, thus removing the need for an a-priori specification of an arbitrary value.

ANNs and other machine learning (ML) strategies have engendered a revolution in data-driven prediction applications and are seeing widespread investigation in the computational physics community. Previous studies into the feasibility of similar ML techniques for ROMs of various nonlinear systems may be found in [22–25]. In particular, we have recently illustrated the ANN concept for model order reduction of the one-dimensional Burgers equation and the performance of several training algorithms has been documented [25]. In the present study, however, we put forth a modified ANN architecture since it is more appropriate to turbulent flows. ML approaches have also been developed for use in feedback flow control where they generate a direct mapping of flow measurements to actuator control systems [26–29]. In our investigation, information from the high fidelity evolu-

[*] osan@okstate.edu

tion of governing laws is leveraged to provide a supervised learning framework for a single layer ANN to stabilize ROMs of the mesoscale forced-dissipative geophysical turbulence system. In brief, an ANN estimates a nonlinear relationship between a desired set of inputs and targets provided viable benchmark data for their underlying statistical relationship is available. This subset of the ML field has seen wide application in function approximation, data classification, pattern recognition and dynamic systems control applications [30, 31] and is generating great interest for its utility in the reproduction of systems with pronounced nonlinear interactions [32].

Before its deployment as a prediction or regression tool, an ANN is trained to accurately capture the nonlinear relationship between its inputs and outputs through some classical loss function (such as mean squared error). A *regularized* training ensures that the framework avoids overfitting any noise that may have been present in the training data. For our supervised learning framework, we utilize the extreme learning machine (ELM) [33] training procedure, which stands out from other learning methods with extremely fast training, good generalization and universal approximation capabilities. ELM is a kind of regularized neural network where the weights connecting inputs and hidden nodes are randomly assigned and never updated. The output weights of hidden nodes are then learned in a single step using a pseudoinverse approach, which provides an extremely fast learning mechanism, in the least squares sense, compared to the networks trained using traditional backpropagation approaches [34]. For our investigation it is seen that a single hidden layer feed-forward ELM algorithm satisfies generalized training requirements with extremely reduced computational cost yet substantially accurate reproductions of training statistics.

For assessing our proposed framework, we utilize the governing laws given by the barotropic vorticity equation (BVE) model. It is a commonly used simplified two-dimensional mathematical framework to study the forced-dissipative large scale ocean circulation problems, also known as the single-layer quasigeostrophic (QG) model [35]. While POD, along with other optimal bases choices, has been used to derive computationally efficient ROMs of the BVE (see, e.g., [36, 37]), the present work represents an attempt to model the unrepresented scales of the QG dynamics, mesoscale turbulence and their effect on mean circulation using an ANN based supervised machine learning framework. The novelty of our approach is therefore adding modal dissipation that is correlated to modal amplitude via a neural net. Our method can be considered as a hybrid approach (i.e., hybrid analytics) combining machine learning and physics-based modeling for QG dynamics. The decision to choice the ELM training approach is to ensure robust generalization for such noisy data.

II. FULL ORDER MODELING

Oceanic and atmospheric flows display an enormous range of spatial and temporal scales, from seconds to decades and from centimeters to thousands of kilometers. Thus, a model incorporating all the relevant physics of the ocean and atmosphere would be impractical for numerical simulations. During the last decades, significant advancements were made in developing simplified models for geophysical fluid dynamics [38], which have been instrumental in providing relatively accurate numerical results at a reasonable computational price. Although these models have continued to produce increasingly accurate results and therefore improved weather forecasting, their use in long time integrations such as those required by climate modeling remains challenging [39, 40]. To illustrate our surrogote proposed framework, we consider the BVE model, which has been extensively used to study forced-dissipative QG dynamics [35]. The dimensionless BVE may be given by [41, 42]

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

where ω is the kinematic vorticity and ψ is the streamfunction. The nonlinear advection term is defined by the Jacobian

$$J(\omega, \psi) = \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}, \quad (2)$$

since we define the flow velocity components by

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \quad (3)$$

and the following kinematic relationship holds for satisfying the incompressibility constraint

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

where ∇^2 is the standard Laplacian operator. The dimensionless BVE given in Eq. (1), has two nondimensional parameters, the Reynolds and Rossby numbers, which are related to the characteristic length and velocity scales in the following way:

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

where ν is the horizontal eddy viscosity of the BVE model and β is the gradient of the Coriolis parameter. We note that Eq. (1) uses the β-plane approximation, valid for most oceanic basins, which accounts for the Earth's rotational effects by approximating the Coriolis parameter. For the purpose of nondimensionalization, L represents a characteristic horizontal length scale given by the basin dimension in the x direction, and V is a characteristic velocity scale (also known as the Sverdrup velocity) 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. Following [41–43], we consider a four-gyre circulation problem, a benchmark oceanic flow problem whose behavior is difficult to capture correctly in coarse grained models [42]. Indeed, as shown in [21], the standard model order reduction approaches without stabilization are incapable of resolving the correct physics. In our full order model (FOM) simulations we use a second-order accurate kinetic energy and enstrophy conserving Arakawa finite difference scheme [44]. The derivatives in the linear terms are also approximated using the standard second-order finite differences. Our time advancement scheme is given by the classical total variation diminishing third-order accurate Runge-Kutta scheme. Details of the Poisson solver, numerical schemes and boundary conditions used for this study may be found in [42].

III. REDUCED ORDER MODELING

We build our reduced order modeling framework based on a standard projection methodology using the method of snapshots [45]. Solving the FOM given by Eq. (1), the nth record of the prognostic variable (vorticity field) is denoted $\omega(x, y, t_n)$ for $n = 1, 2, ..., N$, where N is the number of snapshots recorded for basis construction. Then we decompose the solution field into a time invariant averaged $\bar{\omega}(x, y)$ and a fluctuating component $\omega'(x, y, t)$ through [8, 11],

$$\omega(x, y, t) = \bar{\omega}(x, y) + \omega'(x, y, t) \quad x, y \in \Omega\,, \tag{7}$$

where Ω is the two-dimensional domain and the mean of the snapshot data is

$$\bar{\omega}(x, y) = \frac{1}{N}\sum_{n=1}^{N} \omega(x, y, t_n). \tag{8}$$

In order to obtain the POD basis functions, a correlation matrix of the fluctuating part is constructed by

$$a_{ij} = \int_\Omega \omega'(x, y, t_i)\omega'(x, y, t_j)dxdy, \tag{9}$$

where the subscripts i and j refer to snapshot indexes. We must note that the data correlation matrix $\mathbf{A} = [a_{ij}]$ is a non-negative Hermitian matrix. We further define the inner product of two functions f and g as

$$\langle f, g\rangle = \int_\Omega fg dxdy, \tag{10}$$

such that Eq. (9) yields $a_{ij} = \langle \omega'(x, y, t_i), \omega'(x, y, t_j)\rangle$. The optimal POD basis functions may then be obtained by solving the following eigenvalue problem [46]

$$\mathbf{A}\mathbf{\Gamma} = \mathbf{\Gamma}\mathbf{\Lambda}, \tag{11}$$

where $\mathbf{\Lambda} = \text{diag}[\lambda_1, ..., \lambda_N]$ is the diagonal eigenvalue matrix and $\mathbf{\Gamma} = [\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_N]$ refers to right eigenvector matrix whose columns are eigenvectors of the correlation matrix $\mathbf{A}$. The eigenvalues are usually stored in descending order for practical purposes i.e., $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$. Then the orthogonal POD basis functions of the vorticity field can be obtained as

$$\phi_k(x, y) = \frac{1}{\sqrt{\lambda_k}}\sum_{n=1}^{N} \gamma_{nk}\omega'(x, y, t_n), \tag{12}$$

where λ_k is the kth eigenvalue, γ_{nk} is the nth component of the kth eigenvector, and $\phi_k(x, y)$ is the kth POD mode. The kinematic relationship between streamfunction and vorticity given by Eq. (4) may be utilized to obtain the kth basis function for the streamfunction, $\varphi_k(x, y)$, by solving a Poisson equation

$$\nabla^2\varphi_k = -\phi_k. \tag{13}$$

Now we can span our field variables into the POD modes as follows

$$\omega(x, y, t) = \bar{\omega}(x, y) + \sum_{k=1}^{M} \alpha_k(t)\phi_k(x, y), \tag{14}$$

$$\psi(x, y, t) = \bar{\psi}(x, y) + \sum_{k=1}^{M} \alpha_k(t)\varphi_k(x, y), \tag{15}$$

where we have decomposed $\omega'(x, y, t)$ using time dependent modal coefficient α_k and the POD modes $\phi_k(x, y)$. We note that the kinematic relationship given by Eq. (13) implies that the same α_k accounts for the streamfunction as well. A ROM can be generated by a truncation of the N total bases to only M retained modes where $M \ll N$. These largest energy containing modes correspond to the M largest eigenvalues, λ_1, λ_2, ..., λ_M. To obtain our standard ROM, an orthogonal Galerkin projection is performed by multiplying Eq. (1) with the POD basis functions and integrating over the domain Ω. The resulting dynamical system for α_k can be written as

$$\frac{d\alpha_k}{dt} = \mathfrak{B}_k + \sum_{i=1}^{M} \mathfrak{L}_k^i\alpha_i + \sum_{i=1}^{M}\sum_{j=1}^{M} \mathfrak{N}_k^{ij}\alpha_i\alpha_j, \tag{16}$$

where

$$\begin{aligned}
\mathfrak{B}_k &= \Big\langle \frac{1}{Re}\nabla^2\bar{\omega} + \frac{1}{Ro}\sin(\pi y) + \frac{1}{Ro}\frac{\partial\bar{\psi}}{\partial x} - J(\bar{\omega}, \bar{\psi}), \phi_k\Big\rangle,\\
\mathfrak{L}_k^i &= \Big\langle \frac{1}{Re}\nabla^2\phi_i + \frac{1}{Ro}\frac{\partial\varphi_i}{\partial x} - J(\bar{\omega}, \varphi_i) - J(\phi_i, \bar{\psi}), \phi_k\Big\rangle,\\
\mathfrak{N}_k^{ij} &= \Big\langle -J(\phi_i, \varphi_j), \phi_k\Big\rangle.
\end{aligned} \tag{17}$$

The ROM given by Eq. (16) consists of M coupled ordinary differential equations and can be solved by a standard time integrator (a third-order Runge-Kutta scheme is used in this study). We note that the resulting ROM is highly computationally efficient since all the POD basis functions and corresponding model coefficients given by Eq. (17) are precomputed from the data provided by snapshots. A complete specification of the dynamical system given by Eq. (16) may be obtained by the following projection of the initial condition:

$$\alpha_k(t_0) = \big\langle \omega(x,y,t_0) - \bar{\omega}(x,y), \phi_k \big\rangle, \tag{18}$$

where $\omega(x,y,t_0)$ is the vorticity field specified at time t_0.

The standard ROM given by Eq. (16) usually works well for a periodic or quasi-periodic system for which the largest M modes adequately capture the system's dynamics. However, one of the main sources of inaccuracy in a truncated ROM framework is the potential for instability due to neglecting the contributions of the higher POD modes. Therefore, many stabilization schemes are utilized in order to improve the performance of the ROMs [13–18]). Using an eddy viscosity approach, the stabilization of the ROM can be achieved by [20, 21]

$$\frac{d\alpha_k}{dt} = \mathfrak{B}_k + \tilde{\mathfrak{B}}_k + \sum_{i=1}^{M} (\mathfrak{L}_k^i + \tilde{\mathfrak{L}}_k^i)\alpha_i + \sum_{i=1}^{M}\sum_{j=1}^{M} \mathfrak{N}_k^{ij}\alpha_i\alpha_j, \tag{19}$$

where, using the Smagorinsky model and the analogy between ROM and LES, two additional terms can be written as

$$\begin{aligned} \tilde{\mathfrak{B}}_k &= \langle \nu_k^e \nabla^2 \bar{\omega}, \phi_k \rangle, \\ \tilde{\mathfrak{L}}_k^i &= \langle \nu_k^e \nabla^2 \phi_i, \phi_k \rangle, \end{aligned} \tag{20}$$

where ν_k^e is the modal eddy viscosity parameter. This free stabilization parameter may be simply considered as a global constant for all the modes [16, 47]. The global constant eddy viscosity idea may improved by supposing that the amount of dissipation is not identical for all the POD modes. It has been shown that finding an optimal value for this parameter significantly improves the predictive performance of ROMs [20, 21]. Therefore, the chief novelty of the present study is the utilization of a novel ML framework to estimate these modal eddy viscosity coefficients to stabilize and overcome errors due to the finite truncation in ROMs. We determine ν_k^e dynamically from our ML framework during the evolution of each temporal mode α_k at each time step.

IV. ARTIFICIAL NEURAL NETWORK ARCHITECTURE

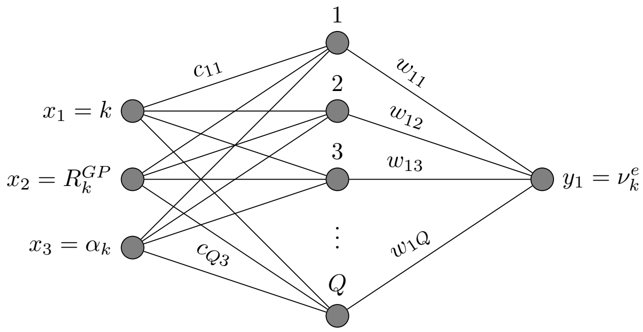

FIG. 1. A schematic of the single hidden layer feedforward ANN utilized for the stabilization framework in this study. Our inputs are resolved ROM variables ($P = 3$) and our prediction is a mode dependent eddy-viscosity ($J = 1$).

In this section, we introduce a single hidden layer feedforward ANN architecture for predicting modal eddy viscosity coefficients for stabilization of ROMs. Figure 1 illustrates our ANN architecture which consists of an input layer, a hidden layer and an output layer. Each layer possesses a predefined number of nodes called neurons. Except for the input neurons, each neuron has an associated bias and activation function. The main goal in any supervised learning framework is to find a mapping between input nodes and output nodes. Mathematically, we are looking for a mapping $\mathfrak{M}$ to establish a relationship between input nodes x_p and output nodes y_j as follows:

$$\mathfrak{M} : \{x_1, x_2, ..., x_P\} \in \mathbb{R}^P \rightarrow \{y_1, y_2, ..., y_J\} \in \mathbb{R}^J, \tag{21}$$

where P is the number of input neurons and J is the number of output neurons. If Q refers to the number of hidden layer neurons, the jth output node can be computed as

$$y_j = G\Big(d_j + \sum_{q=1}^{Q} w_{jq} F\big(b_q + \sum_{p=1}^{P} c_{qp} x_p\big)\Big) \tag{22}$$

where $c_{qp} \in \mathbb{R}^{Q\times P}$ are the connection weights between the neurons in input and hidden layers, and $w_{jq} \in \mathbb{R}^{J\times Q}$ are the weights between the neurons in hidden and output layers. Here, F and G are neurons' activation functions; and $b_q \in \mathbb{R}^Q$ and $d_j \in \mathbb{R}^J$ are called biases operating as thresholds for hidden and output layers, respectively. In this study, we have utilized the tan-sigmoid activation function for the hidden layer neurons, which can be expressed as

$$F(s) = \frac{2}{1+\exp(-2s)} - 1, \tag{23}$$

and a linear activation function for the output layer neurons given by

$$G(s) = s. \tag{24}$$

While it has been reported that sigmoidal activation functions saturate across a large portion of their domain [48], our reasoning behind the use of the classical tan-sigmoid activation was to leverage the benefit of the saturation behavior to obtain a bounded prediction from the network.

A. Extreme learning machine

Introducing N sample training data examples (i.e., input-output pairs), the weights and biases can be computed in a supervised learning framework using either well established iterative back propagation methods [49] or pseudoinverse approaches [34]. As mentioned previously, the ANN architecture is trained by utilizing an extreme learning machine (ELM) approach proposed in [33] for extremely fast training of a single hidden layer feedforward network. The ELM approach requires no biases in the output layer (i.e., $d_j = 0$). In the ELM method, the weights c_{qp} and biases b_q are initialized randomly from a uniform distribution (i.e., between -1 and 1 in our study) and no longer modified. Therefore the only unknowns to be determined are w_{jq} weights. Using the linear activation function for the output layer, Eq. (22) can be written for N sample examples

$$y_{jn} = \sum_{q=1}^{Q} w_{jq} F\big(b_q + \sum_{p=1}^{P} c_{qp} x_{pn}\big) \tag{25}$$

where $x_{pn} \in \mathbb{R}^{P\times N}$ and $y_{jn} \in \mathbb{R}^{J\times N}$ refer to the training input-output data pairs. Using a more convenient matrix notation (i.e., $\mathbf{X} = [x_{pn}]$, $\mathbf{Y} = [y_{jn}]$, $\mathbf{C} = [c_{qp}]$, $\mathbf{W} = [w_{jq}]$, and $\mathbf{b} = [b_q]$), our learning problem can be written as

$$\mathbf{Y} = \mathbf{W}\mathbf{H}^{\intercal} \tag{26}$$

where $\mathbf{H}^{\intercal} \in \mathbb{R}^{Q\times N}$ is given by

$$\mathbf{H}^{\intercal} = F(\mathbf{B} + \mathbf{C}\mathbf{X}) \tag{27}$$

where the vector $\mathbf{b}$ is repeated across N columns as shown in Eq. (25) (i.e., $\mathbf{B} = [\mathbf{b}, \mathbf{b}, ..., \mathbf{b}] \in \mathbb{R}^{Q\times N}$). By taking the transpose of both side of Eq. (26) we can write

$$\mathbf{H}\mathbf{W}^{\intercal} = \mathbf{Y}^{\intercal} \tag{28}$$

and the solution for the weights can be computed by

$$\mathbf{W}^{\intercal} = \mathbf{H}^{\dagger}\mathbf{Y}^{\intercal} \tag{29}$$

where $\mathbf{H}^{\dagger} \in \mathbb{R}^{Q\times N}$ is the pseudoinverse of $\mathbf{H} \in \mathbb{R}^{N\times Q}$. In order to compute the pseudoinverse, we apply the following singular value decomposition (SVD) to the matrix $\mathbf{H}$ since its number of rows is greater than its number of columns in typical ML applications (i.e., $N \geq Q$)

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\intercal} \tag{30}$$

where $\mathbf{U} \in \mathbb{R}^{N\times Q}$ and $\mathbf{V} \in \mathbb{R}^{Q\times Q}$ are column-orthogonal and orthogonal matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{Q\times Q}$ is a diagonal matrix whose elements (i.e., $\sigma_{qq} = \sigma_q$) are non-negative and called singular values. Using the SVD, the pseudoinverse of $\mathbf{H}$ becomes

$$\mathbf{H}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{\dagger}\mathbf{U}^{\intercal} \tag{31}$$

where $\mathbf{\Sigma}^{\dagger}$ can be computed from $\mathbf{\Sigma}$ by taking the reciprocal of each non-zero element (i.e., $\sigma_q^{\dagger} = 1/\sigma_q$). However, the presence of tiny singular values can cause numerical instability. Therefore, a well-known Tikhonov type regularization is often introduced by

$$\sigma_q^{\dagger} = \frac{\sigma_q}{\sigma_q^2 + \epsilon} \tag{32}$$

where λ controls the trade off between the least-squares error and the penalty term for regularization (e.g., see [34]). In the present study we set $\epsilon = 10^{-12}$. Finally, using Eq. (29), unknown weights can be computed by

$$\mathbf{W} = \mathbf{Y}\mathbf{U}\mathbf{\Sigma}^{\dagger}\mathbf{V}^{\intercal}. \tag{33}$$

B. Training data

Our architecture is devised to take inputs accessible to us during the time integration of the ROM and estimate the modal eddy viscosity coefficient. Our high fidelity snapshot data (from which POD bases are constructed) are also used to train our architecture. First, we denote the right hand side of Eq. (16) as

$$R_k^{GP} = \mathfrak{B}_k + \sum_{i=1}^{M} \mathfrak{L}_k^i \alpha_i + \sum_{i=1}^{M}\sum_{j=1}^{M} \mathfrak{N}_k^{ij} \alpha_i \alpha_j, \tag{34}$$

and then apply the Galerkin projection to FOM given by Eq. (1), which yields the true solution

$$R_k^{FOM} = \Big\langle \frac{1}{Re}\nabla^2\omega + \frac{1}{Ro}(\sin(\pi y) + \frac{\partial\psi}{\partial x}) - J(\omega,\psi), \phi_k \Big\rangle. \tag{35}$$

The ideal stabilization would thus conform to the differences between these quantities i.e.,

$$\tilde{R}_k = R_k^{FOM} - R_k^{GP}. \tag{36}$$

We know from Eq. (20) that

$$\tilde{R}_k = \nu_k^e \Big(\langle \nabla^2 \bar{\omega}, \phi_k \rangle + \sum_{i=1}^{M} \langle \nabla^2 \phi_i, \phi_k \rangle \alpha_i \Big), \tag{37}$$

where we redefine

$$R_k^{STAB} = \langle \nabla^2 \bar{\omega}, \phi_k \rangle + \sum_{i=1}^{M} \langle \nabla^2 \phi_i, \phi_k \rangle \alpha_i, \tag{38}$$

and therefore we compare Eq. (36) and Eq. (37) to obtain the modal eddy viscosity coefficients

$$\nu_k^e = \frac{R_k^{FOM} - R_k^{GP}}{R_k^{STAB}}, \tag{39}$$

as the eddy viscosity stabilization for each mode within the training data set. Although Eq. (39) is an exact relationship, we use a clipping procedure for numerical stability by discarding negative entries of ν_k^e in our training data set. Therefore, our training data is generated by considering the following bounds

$$\epsilon = 10^{-12} \leq \nu_k^e \leq \frac{c}{Re} = c\frac{\nu}{VL}, \tag{40}$$

where c is the upper bound of the relative ratio between the stabilized viscosity and physical model viscosity. In the present study, we set $c = 6$, which provides six times larger stabilization viscosity than the specified ν of the original model. We have also verified that the proposed ROM-ANN approach is robust to the selection of c (i.e., similar statistical results have been observed for $c = 4$ and $c = 10$ sets). The clipping approach presented by Eq. (40) can be considered as a physical realizability bounds of ROM training data. With this realizable calculation of the stabilization viscosity, we hypothesize that a mode dependent nonlinear (but unknown) relationship exists between the resolved modes in the ROM that estimates ν_k^e dynamically. To conclude, our ANN framework is trained between inputs given by the modal index k, R_k^{GP}, and α_k (i.e., they are all available during the ROM time stepping) and to predict an approximation for ν_k^e. We thus have 3 inputs to our network with Q hidden layer neurons to obtain 1 output (which is the modal eddy viscosity coefficient). The architecture of our ANN is shown in Figure 1. We emphasize that this simplified ANN is basically a non-linear regression or a curve fitting between input and target states. As we will show in next section, however, its generalization is quite remarkable for both in-sample data and out-of-sample data predictions.

V. RESULTS

To validate our proposed ANN framework, we consider the four-gyre barotropic circulation problem [41–43]). This test problem yields four gyres circulation patterns in the time mean in a shallow ocean basin and represents an ideal test for the viability of the proposed ROM. Indeed it was shown that ROMs without stabilization are incapable of resolving the mean dynamics [21].

The dimensionless form of the BVE describing the QG problem is evolved from $t = 0$ to $t = 100$ using a fixed time step $\Delta t = 2.5 \times 10^{-5}$ on a Munk layer resolving 256×512 computational grid resolution. The dimensionless parameters of the BVE system are chosen as $Re = 450$ and $Ro = 3.6\times10^{-3}$. We must note that $t = 10$ to $t = 100$ is our data collection window (for the purpose of POD basis generation as well as ANN training) due to a statistically steady state reached after the initial transient period. 900 snapshots are collected during this period which are equally distributed in time. The ideal eddy viscosity is also computed at these snapshots for the use of training our machine learning framework. We note here that our ROM (whether purely truncated or stabilized by ANN) is utilized for predictions upto $t = 200$ utilizing the POD modes obtained from our previously mentioned data collection window. This may be considered to be a challenging validation of our dual data-driven methodology for the QG problem. For both our standard ROM and stabilized ROM-ANN computations, the same time step with $\Delta t = 2.5 \times 10^{-5}$ is used for time integration of the dynamical system. Sensitivity studies for varying time steps will be also presented later.

TABLE I. L$_2$-norm errors of the reduced order models (with respect to FOM) for the mean vorticity and streamfunction fields. Note that the ROM-ANN retains only $M = 10$ modes.

	Vorticity	Streamfunction
<u>*No stabilization*</u>		
ROM ($M = 10$)	6.89×10^7	1.50×10^5
ROM ($M = 20$)	2.85×10^5	5.69×10^2
ROM ($M = 30$)	1.07×10^3	1.14×10^0
<u>*With stabilization*</u>		
ROM-ANN ($Q = 20$)	1.20×10^3	9.45×10^{-1}
ROM-ANN ($Q = 40$)	6.54×10^2	1.22×10^{-1}
ROM-ANN ($Q = 60$)	4.31×10^2	3.09×10^{-1}

Figure 2 shows the accumulation of energies in the form of eigenvalue magnitudes where it can be seen that a large majority (close to 75%) of the energies are accumulated in the first 30 modes of the transformed space. Figure 3 shows the gradual convergence of the ROM (i.e., without stabilization) to the four-gyre circulation pattern with increasing M. Indeed, non-physical two-gyre pattern is observed for the case of $M = 10$ and $M = 20$.

Figure 4 shows the performance of the proposed framework (i.e., ROM-ANN) against the standard Galerkin projection based ROM with $M = 10$. Full order model (FOM) projections to reduced space are also shown for the purpose of comparison. It can easily be seen that the ELM stabilization reproduces the four-gyre pattern accurately as against the standard implementation of the ROM which fails to capture the pattern. This is observed for both streamfunction and vorticity contours. Figure 5 shows a qualitative comparison of the effect of the number of neurons Q where similar performance improvements are obtained for our choice of $Q = 20, 40, 60$ neurons. Table I shows a quantitative comparison of the improvement obtained by the proposed stabilization (for different neurons as well) against the standard ROM implementations with different number of modes. It is easily observed that the stabilization acts adequately in reproducing excellent agreement with full-order statistics at a very low number of retained modes. Note that these plots and tabulated statistics are all for the statistically steady state behavior of the QG problem in our assessment window (i.e., $t = 10$ to $t = 200$), which is beyond the training data window.

Figure 6 shows a comparison for the evolution of α_1 through nondimensional time for both ROM and ROM-ANN implementations in comparison to the FOM projection. The ROM-ANN has a default $Q = 40$ neurons

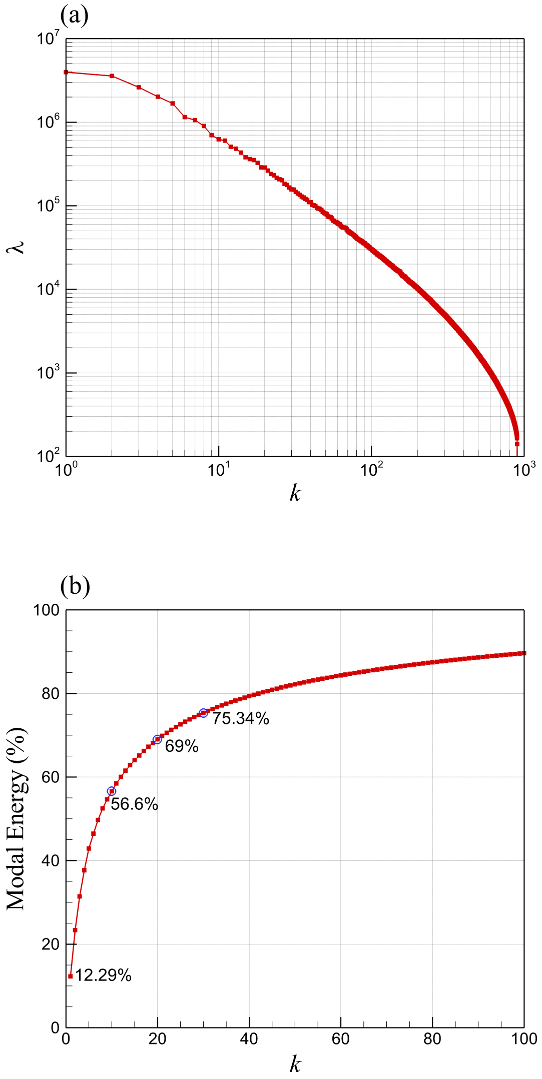

FIG. 2. POD analysis by the snapshot data for $Re = 450$ and $Ro = 3.6 \times 10^{-3}$. (a) The distribution of eigenvalues; and (b) their energy levels with k denoting the modal index.

in this example. It can clearly be seen that the use of the stabilization prevents the explosion of numerical instability in the coarse truncated ROMs with $M = 10$ and $M = 20$. At $M = 30$ however, the first modal evolution shows a stable statistical steady state for the ROM.

Another benefit of the ROM-ANN mechanism over the standard ROM implementation is the possibility of using large time steps in the ordinary differential equation integrator. In the present study, a time step of $\Delta t = 2.5\times10^{-5}$ was chosen for the FOM simulation to ensure a CFL criterion of less than 1.0 was always respected (as observed in the time series plot in Figure 7) due to the numerical stability of the numerical schemes. Figure 8 shows the vorticity and streamfunction contours when our stabilized method (i.e., the ROM-ANN with $M = 10$

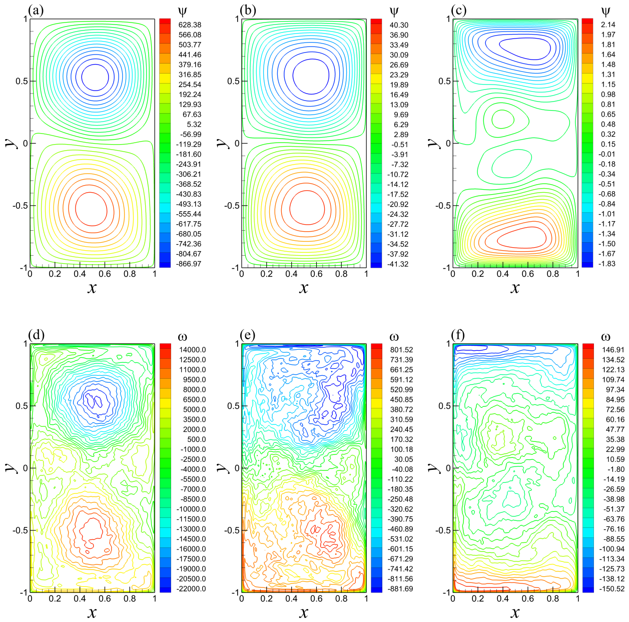

FIG. 3. Mean streamfunction and vorticity fields retrieved from the standard Galerkin projection method. (a) ψ with $M = 10$ modes; (b) ψ with $M = 20$ modes; (c) ψ with $M = 30$ modes; (d) ω with $M = 10$ modes; (e) ω with $M = 20$ modes; and (f) ω with $M = 30$ modes.

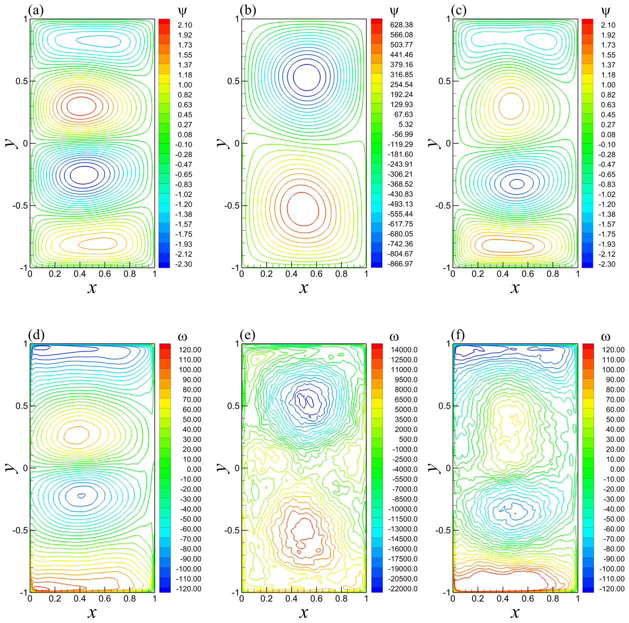

FIG. 4. A comparison of the standard Galerkin approach (ROM) and the proposed ANN based stabilized approach (ROM-ANN) for $M = 10$ modes. (a) ψ by FOM; (b) ψ by ROM; (c) ψ by ROM-ANN; (d) ω by FOM; (e) ω by ROM; and (f) ω by ROM-ANN.

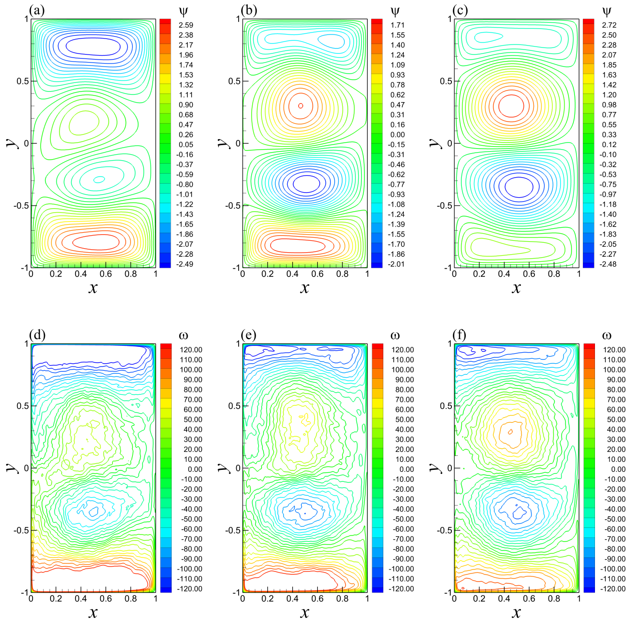

FIG. 5. A sensitivity analysis with respect to the number of neurons in ELM using $M = 10$ modes. (a) ψ with $Q = 20$ nodes; (b) ψ with $Q = 40$ nodes; (c) ψ with $Q = 60$ nodes; (d) ω with $Q = 20$ nodes; (e) ω with $Q = 40$ nodes; and (f) ω with $Q = 60$ nodes.

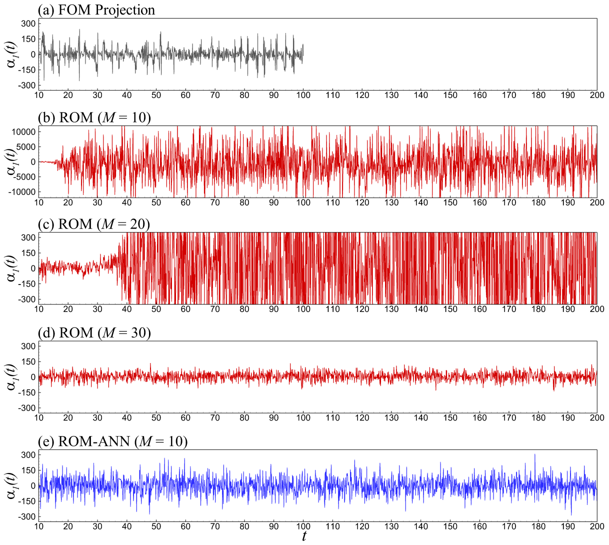

FIG. 6. Time series for the first temporal coefficient $\alpha_1(t)$.

and $Q = 40$) is used with different time steps. It can be seen that a much larger time step of $\Delta t = 10^{-2}$ can be effectively used to obtain statistically accurate results without any divergence. Thus our proposed ANN based eddy viscosity stabilization is ideally suited to a fast prediction of the underlying dynamics. Figure 9 shows the evolution of the first temporal coefficient $\alpha_1(t)$ for the aforementioned ROM-ANN framework where it is seen that very high values of the time step do not affect the statistical viability of the stabilized ROM and leads to an excellent reduction in computational expense (the largest time step provides excellent results at a CPU time of 1.61 seconds in comparison to approximately 700 seconds for the default time step which is required for the FOM). We also note that the FOM required 195.4 hours CPU time to complete the forward simulation between $t = 10$ and

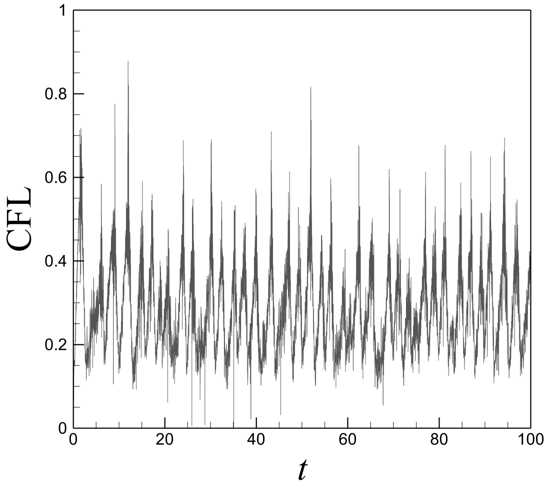

FIG. 7. Time series for the CFL criterion assuming a fixed time step of $\Delta t = 2.5 \times 10^{-5}$ for the FOM simulation.

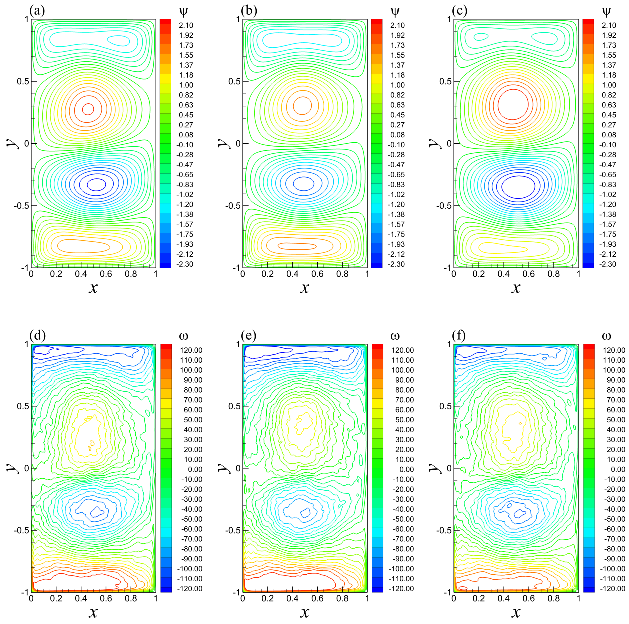

FIG. 8. Mean streamfunction and vorticity contours obtained by the proposed ROM-ANN with various time steps. (a) ψ with $\Delta t = 5 \times 10^{-4}$; (b) ψ with $\Delta t = 2.5 \times 10^{-3}$; (c) ψ with $\Delta t = 10^{-2}$; (d) ω with $\Delta t = 5 \times 10^{-4}$; (e) ω with $\Delta t = 2.5 \times 10^{-3}$; and (f) ω with $\Delta t = 10^{-2}$. Note that our ROM-ANN implementation uses $M = 10$ and $Q = 40$.

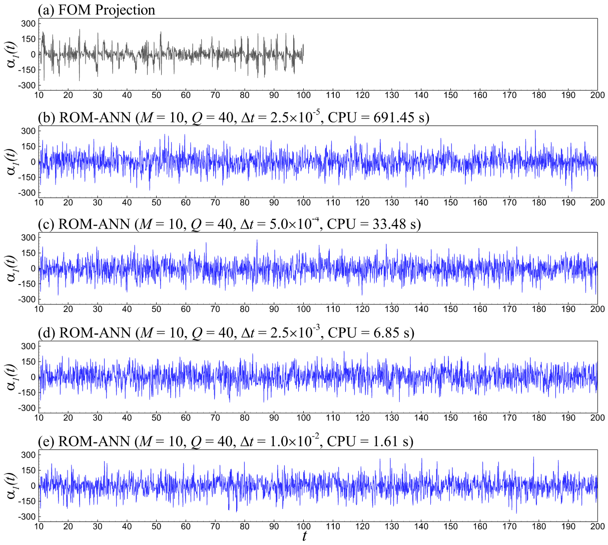

FIG. 9. Temporal mode evolution of α_1 for various time steps.

$t = 100$.

Finally, we perform an out-of-sample a-posteriori analysis considering different physical parameters than those used in the training data. As explained earlier the physical model parameters are $Re = 450$ and $Ro = 3.6 \times 10^{-3}$ to generate our data snapshots (therefore POD basis functions) and the supervised training data set for ELM. Using the same ELM network and POD basis functions, Figure 10 compares the predictive performance of the models at $Re = 200$ and $Ro = 1.6 \times 10^{-3}$, a distinct test set-up from the training data. A new FOM simulation is performed for our assessments (i.e., required about 200.6 hours CPU time). It is clear that the ROM-ANN captures the main dynamics requiring only the order of seconds CPU time for the simulation. Time series of α_1 are also illustrated in Figure 11. Similar to the in-sample case, the standard Galerkin ROM approach cannot capture the correct dynamics for $M = 10$ or $M = 20$ modes. On the other hand, the proposed stabilized ROM-ANN approach captures the underlying four-gyre dynamics and provides significantly accurate results for using $M = 10$ modes. Our assessments conclude that the proposed architecture is robust to provide a reliable mode dependant damping coefficient for the out-of-sample forecasting.

VI. SUMMARY AND CONCLUSIONS

In this paper, we have studied the feasibility of using a machine learning framework to stabilize projection based ROMs for solving a forced-dissipative general circulation problem. We construct a single hidden layer feedforward ANN to predict modal eddy viscosity coefficients dynamically. Our approach can be considered semi non-intrusive (without the need for an online access to the FOM for the ROM prediction), since the ANN architecture only

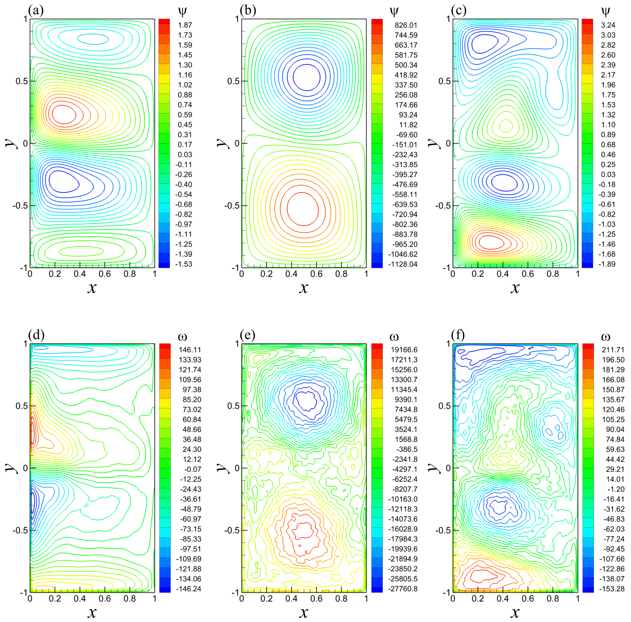

FIG. 10. A comparison of the standard Galerkin approach (ROM) and the proposed ANN based stabilized approach (ROM-ANN) for $M = 10$ modes at $Re = 200$ and $Ro = 1.6 \times 10^{-3}$ where the training data has been extracted from the snapshots at $Re = 450$ and $Ro = 3.6 \times 10^{-3}$. (a) ψ by FOM; (b) ψ by ROM; (c) ψ by ROM-ANN; (d) ω by FOM; (e) ω by ROM; and (f) ω by ROM-ANN. Note that the ROM-ANN uses $Q = 40$ hidden nodes.

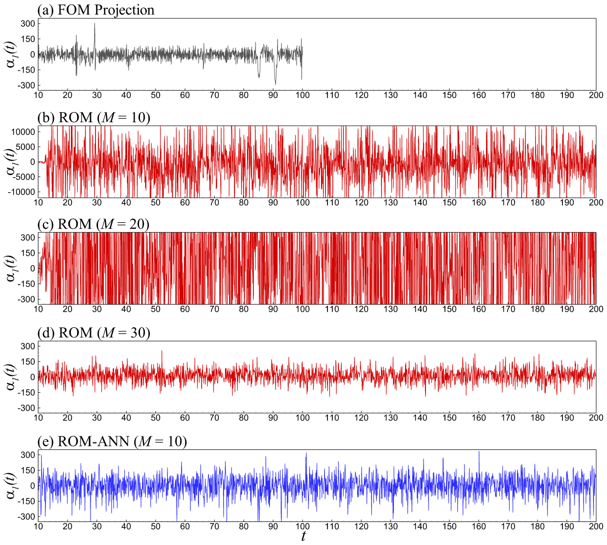

FIG. 11. Time series for of first temporal coefficient $\alpha_1(t)$ for the out-of-sample forecast. Predictive performance is shown for $Re = 200$ and $Ro = 1.6 \times 10^{-3}$ while the training has been performed using the data generated at $Re = 450$ and $Ro = 3.6 \times 10^{-3}$.

requires reduced order space quantities to predict the stabilization term. A regularized ELM approach is used for training where we use the same data snapshots as we used for generating the POD basis functions. In that sense, there are two data-driven components to this research: high fidelity snapshots of data from DNS are utilized not just for POD basis synthesis but also for training our machine learning framework utilized for a-posteriori stabilization of the ROM-ANN. Both in-sample and out-of-sample simulation results indicate that the utilization of the proposed framework lets the user deploy an extremely truncated system without losing any statistical fidelity. Also, time steps much larger than those necessary for FOM forward simulations may be utilized in the ROM-ANN thus leading to exceptional computational performance. We conclude that the method presented in this paper is robust enough to stabilize ROMs dynamically and satisfies the dual demands of statistical accuracy as well as low computational expense in surrogate forward predictions in long-term evolution of geophysical turbulent flows.

ACKNOWLEDGMENT

The computing for this project was performed by using resources from the High Performance Computing Center (HPCC) at Oklahoma State University.

[1] S. L. Brunton and B. R. Noack, “Closed-loop turbulence control: Progress and challenges,” Applied Mechanics Reviews **67**, 050801 (2015).

[2] D. N. Daescu and I. M. Navon, “Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation,” International Journal for Numerical Methods in Fluids **53**, 985 (2007).

[3] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo, “A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition,” International Journal for Numerical Methods in Fluids **53**, 1571 (2007).

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

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

[6] C. W. Rowley and S. T. Dawson, “Model reduction for flow analysis and control,” Annual Review of Fluid Mechanics **49**, 387 (2017).

[7] 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 Journal **55**, 4013 (2017).

[8] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry* (Cambridge University Press, 1998).

[9] K. Ito and S. Ravindran, “A reduced-order method for simulation and control of fluid flows,” Journal of Computational Physics **143**, 403 (1998).

[10] A. Iollo, S. Lanteri, and J.-A. Désidéri, “Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations,” Theoretical and Computational Fluid Dynamics **13**, 377 (2000).

[11] 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,” Journal of Fluid Mechanics **497**, 335 (2003).

[12] R. Narasimha, “Kosambi and proper orthogonal decomposition,” Resonance **16**, 574 (2011).

[13] M. Couplet, P. Sagaut, and C. Basdevant, “Intermodal energy transfers in a proper orthogonal decomposition-Galerkin representation of a turbulent separated flow,” Journal of Fluid Mechanics **491**, 275 (2003).

[14] V. L. Kalb and A. E. Deane, “An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models,” Physics of fluids **19**, 054106 (2007).

[15] M. Bergmann, C.-H. Bruneau, and A. Iollo, “Enablers for robust POD models,” Journal of Computational Physics **228**, 516 (2009).

[16] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, “Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison,” Computer Methods in Applied Mechanics and Engineering **237**–**240**, 10 (2012).

[17] J. Baiges, R. Codina, and S. Idelsohn, “Reduced-order subscales for POD models,” Computer Methods in Applied Mechanics and Engineering **291**, 173 (2015).

[18] X. Xie, D. Wells, Z. Wang, and T. Iliescu, “Approximate deconvolution reduced order modeling,” Computer Methods in Applied Mechanics and Engineering **313**, 512 (2017).

[19] I. Akhtar, Z. Wang, J. Borggaard, and T. Iliescu, “A new closure strategy for proper orthogonal decomposition reduced-order models,” Journal of Computational and Nonlinear Dynamics **7**, 034503 (2012).

[20] O. San and T. Iliescu, “Proper orthogonal decomposition closure models for fluid flows: Burgers equation,” International Journal of Numerical Analysis and Modeling, Series B **5**, 217 (2014).

[21] O. San and T. Iliescu, “A stabilized proper orthogonal decomposition reduced-order model for large scale quasi-geostrophic ocean circulation,” Advances in Computational Mathematics **41**, 1289 (2015).

[22] S. Narayanan, A. Khibnik, C. Jacobson, Y. Kevrekedis, R. Rico-Martinez, and K. Lust, in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on* (IEEE, 1999), vol. 2, pp. 1151–1156.

[23] R. Sahan, N. Koc-Sahan, D. Albin, and A. Liakopoulos, in *Proceedings of the 1997 IEEE International Conference on Control Applications,* (IEEE, 1997), pp. 359–364.

[24] A. Moosavi, R. Stefanescu, and A. Sandu, “Efficient Construction of Local Parametric Reduced Order Models Using Machine Learning Techniques,” arXiv preprint arXiv:1511.02909 (2015).

[25] O. San and R. Maulik, “Neural network closures for nonlinear model order reduction,” Advances in Computational Mathematics, DOI:10.1007/s10444-018-9590-

z (2018).

[26] E. Gillies, “Low-dimensional control of the circular cylinder wake,” Journal of Fluid Mechanics **371**, 157 (1998).

[27] W. E. Faller and S. J. Schreck, “Unsteady fluid mechanics applications of neural networks,” Journal of Aircraft **34**, 48 (1997).

[28] M. O. Efe, M. Debiasi, H. Ozbay, and M. Samimy, in *Proceedings of the International Conference on Mechatronics, 2004.* (IEEE, 2004), pp. 560–565.

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

[30] B. Widrow, D. E. Rumelhart, and M. A. Lehr, “Neural networks: applications in industry, business and science,” Communications of the ACM **37**, 93 (1994).

[31] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design* (Martin Hagan, 2014).

[32] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations,” arXiv preprint arXiv:1711.10561 (2017).

[33] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” Neurocomputing **70**, 489 (2006).

[34] R. Cancelliere, R. Deluca, M. Gai, P. Gallinari, and L. Rubini, “An analysis of numerical issues in neural training by pseudoinversion,” Computational and Applied Mathematics **36**, 599 (2017).

[35] A. Majda and X. Wang, *Nonlinear dynamics and statistical theories for basic geophysical flows* (Cambridge University Press, 2006).

[36] F. M. Selten, “An efficient description of the dynamics of barotropic flow,” Journal of the Atmospheric Sciences **52**, 915 (1995).

[37] D. T. Crommelin and A. J. Majda, “Strategies for model reduction: comparing different optimal bases,” Journal of the Atmospheric Sciences **61**, 2206 (2004).

[38] J. C. McWilliams, *Fundamentals of geophysical fluid dynamics* (Cambridge University Press, 2006).

[39] M. Ghil, M. D. Chekroun, and E. Simonnet, “Climate dynamics and fluid mechanics: Natural variability and related uncertainties,” Physica D: Nonlinear Phenomena **237**, 2111 (2008).

[40] P. Lynch, “The origins of computer weather prediction and climate modeling,” Journal of Computational Physics **227**, 3431 (2008).

[41] R. J. Greatbatch and B. Nadiga, “Four-gyre circulation in a barotropic model with double-gyre wind forcing,” Journal of Physical Oceanography **30**, 1461 (2000).

[42] O. San, A. E. Staples, Z. Wang, and T. Iliescu, “Approximate deconvolution large eddy simulation of a barotropic ocean circulation model,” Ocean Modelling **40**, 120 (2011).

[43] B. T. Nadiga and L. G. Margolin, “Dispersive-dissipative eddy parameterization in a barotropic model,” Journal of Physical Oceanography **31**, 2525 (2001).

[44] A. Arakawa, “Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I,” Journal of Computational Physics **1**, 119 (1966).

[45] L. Sirovich, “Turbulence and the dynamics of coherent structures. I. Coherent structures,” Quarterly of Applied Mathematics **45**, 561 (1987).

[46] S. Ravindran, “A reduced-order approach for optimal control of fluids using proper orthogonal decomposition,” International Journal for Numerical Methods in Fluids **34**, 425 (2000).

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

[48] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1 (MIT press Cambridge, 2016).

[49] M. Carrillo, U. Que, and J. A. González, “Estimation of Reynolds number for flows around cylinders with lattice Boltzmann methods and artificial neural networks,” Physical Review E **94**, 063304 (2016).