Learning ocean circulation models with reservoir computing

Cite as: Phys. Fluids **34**, 116604 (2022); https://doi.org/10.1063/5.0119061
Submitted: 07 August 2022 • Accepted: 20 October 2022 • Accepted Manuscript Online: 21 October 2022 • Published Online: 09 November 2022

Kevin Yao, Eric Forgoston and Philip Yecko

Physics of Fluids ARTICLE scitation.org/journal/phf

Learning ocean circulation models with reservoir computing

Cite as: Phys. Fluids **34**, 116604 (2022); doi: 10.1063/5.0119061
Submitted: 7 August 2022 · Accepted: 20 October 2022 ·
Published Online: 9 November 2022

Kevin Yao,[1] Eric Forgoston,[2] and Philip Yecko[1,a)]

AFFILIATIONS

[1]Albert Nerken School of Engineering, The Cooper Union for the Advancement of Science and Art, New York, New York 10003, USA
[2]Department of Applied Mathematics and Statistics, Montclair State University, Montclair, New Jersey 07043, USA

[a)]Author to whom correspondence should be addressed: philip.yecko@cooper.edu

ABSTRACT

Two elementary models of ocean circulation, the well-known double-gyre stream function model and a single-layer quasi-geostrophic (QG) basin model, are used to generate flow data that sample a range of possible dynamical behavior for particular flow parameters. A reservoir computing (RC) machine learning algorithm then learns these models from the stream function time series. In the case of the QG model, a system of partial differential equations with three physically relevant dimensionless parameters is solved, including Munk- and Stommel-type solutions. The effectiveness of a RC approach to learning these ocean circulation models is evident from its ability to capture the characteristics of these ocean circulation models with limited data including predictive forecasts. Further assessment of the accuracy and usefulness of the RC approach is conducted by evaluating the role of both physical and numerical parameters and by comparison with particle trajectories and with well-established quantitative assessments, including finite-time Lyapunov exponents and proper orthogonal decomposition. The results show the capability of the methods outlined in this article to be applied to key research problems on ocean transport, such as predictive modeling or control.

I. INTRODUCTION

The inherently high-dimensional and nonlinear nature of fluid dynamics presents formidable barriers to the analysis, prediction, and simulation of real-world and model flows.[1] Fluid-coupled control[2–5] of autonomous vehicles that can perform a variety of sensing tasks in the ocean and atmosphere[6,7] is just two examples of common tasks that generally lead to difficult optimization problems. Large scale multi-physics systems such as atmospheric or oceanic circulation tend to present additional complexities, placing efficient or detailed prediction in such systems far out of reach of current methods. Reduced models like quasi-geostrophy play an important role by identifying dynamical regimes and in facilitating studies that are not possible in more cumbersome models. More recently, quasi-geostrophic (QG) models have served as components within more complex models and as a framework to reconstruct and assimilate flows from satellite data.[8–10]

Data-based study of ocean flows is further challenged by the enormity and inaccessibility of the system, resulting in datasets that are both too much (raw volume) and too little (sparse or incomplete). Primitive equation-based ocean models are also complex and unwieldy and are, thus, shunned for certain tasks in favor of simpler reduced models. Quasi-geostrophy is an example of a reduced model which, despite being highly simplified, is still a partial differential equation (PDE)-based model that exhibits rich dynamics including low-frequency variability, multiple equilibria, and parametric bifurcation to aperiodic and chaotic solutions,[11–15] advancing the understanding of key dynamics underlying the behavior of complex primitive equation ocean models and ocean observations.

A dynamical systems approach to ocean flows has led to new tools and useful diagnostics, which has improved our understanding of transport. However, the determination of Lagrangian coherent structures (LCS)[2,16] through the finite-time Lyapunov exponent (FTLE) field and other geometric and probabilistic methods[17] is also computationally demanding and seems to offer limited capacity to make detailed flow predictions. Related approaches have exploited modal decomposition,[18] both as a modeling tool[19] and more broadly as a tool to analyze data for purposes of control, modeling, or prediction.[20,21] The double-gyre (DG) streamfunction (toy) model is the foundation of much of that work, becoming a workhorse and standard example within the dynamics community.[22] Furthermore, Aref's work[23] has highlighted the sensitive nature of particle trajectories and their chaotic nature even in very simple time-dependent flows.

Machine learning (ML) provides alternate, independent insights into dynamical systems and into fluid dynamics,[24–29] thanks to its ability to uncover patterns in large datasets. Machine learning derives models from data through optimization but often requires a large amount of data to achieve good results. Fortunately, many fluid dynamical problems, including large scale geophysical flows, offer a wealth of experimental or simulation data.

Reservoir computing (RC)[30] is an implementation of a recurrent neural network (RNN) that avoids several drawbacks of RNNs, such as their high computational cost and tendency to overfit. This is achieved in part by learning only on output weights,[31–33] allowing the hidden layer to act as a reservoir and capture features of the history of inputs. The echo state network (ESN) variety of RC that we use is described in more detail in Sec. II.

RC is noteworthy for its forecasting capability for complex systems.[34] Recently, an RC approach was trained on data from the intrinsically aperiodic Lorenz dynamical system,[35] and both reproduced characteristics of its attractor and made predictions of future states. The same authors[35,36] extended this approach to the Kuramoto–Sivashinsky (KS) equation, a one-dimensional, nonlinear PDE model of reaction–diffusion and fluid phenomena, and were similarly able to predict future spatiotemporal states. It follows naturally that RC can be applied to a range of geophysical fluid dynamics (GFD) problems. This study adopts as its focus a pair of reduced ocean models: the double-gyre and the barotropic QG models, which are closely related but with significant distinctions.

On the one hand, QG models and their solutions have provided researchers insights into the key dynamics of the global ocean circulation, the so-called Sverdrup transport, in terms of the vorticity balance between dissipation and wind stress driving (Ekman pumping) in a closed basin.[37] Using only three non-dimensional control parameters, the barotropic QG model can describe both the nature and the complex variability of the prominent Western boundary currents seen in the world's oceans; the history of this research has been reviewed in the text by Pedlosky[38] and in several well-known works.[39–41] QG, thus, allows direct examination of the role of nonlinearity, in terms of the so-called inertial or Fofonoff mode, within the Munk model, features that are generally obscured in complex physical equation simulations and the actual ocean. As a result, QG continues to be a powerful tool in emergent applications of ocean and climate research, ranging from theories for the thermocline,[42,43] to interannual variability,[44,45] to the separation of the western boundary current (i.e., gulf stream),[41,46,47] while more recently, QG models have been applied to the Atlantic meridional overturning circulation (AMOC)[48] and as a framework for data assimilation[49] and for turbulence model closure.[50]

Given their excellent dynamical capabilities, QG models have served as a foundation for the diagnosis and development of sensing strategies and control[2] and other applications based on or coupled to ocean transport, such as pollutant dispersal. In such applications, even the highly reduced barotropic QG model, a PDE, is sometimes too computationally expensive. As a result, many researchers have adopted the double-gyre stream function[51] model as a useful proxy for a full QG model, which, nevertheless, mimics its key *kinematic* transport features. Like QG, DG is typically driven by time-dependent forcing that can represent wind stress, but unlike QG, DG does not exhibit intrinsic variability; indeed, DG is a direct prescription of a flow stream function.

Even in DG-based studies of transport, researchers have faced challenges in predicting trajectories of tracer particles and in understanding key transport features. This is not surprising, as the pioneering work of Aref[23] has shown that tracer trajectories are often chaotic even in the simplest two-dimensional time-dependent flows, such as a Stokes flow. Tools such as modal analysis, while powerful and useful, were not ideally suited to the above challenges. With the work of Shadden *et al.*,[22] the application of Lagrangian coherent structures (LCS) and the finite-time Lyapunov exponent (FTLE) the LCS is based on were shown to give greater insight into transport features and trajectories of DG models. Indeed, in this study, we adopt and leverage the combination of modal analysis, FTLE, and particle trajectories to assess the efficacy of machine learning (ML) models to capture the complex nature of transport in both DG and QG flows, much as the attractor can be used to compare chaotic dynamical systems.

The approach presented in this study applies reservoir computing to data produced by: (i) a double-gyre stream function model with time-variable forcing and (ii) a single-layer (barotropic) QG basin model. Each model is formulated as a discrete, time-dependent dynamical system, which is used to produce stream function time-series data that provide the input to a reservoir computing model. For each system, we use RC to find a low-dimensional model of the system, and we compare the predictions of the RC model to the original system using several relevant testing frameworks, including global stream function and ideal particle trajectories.

The quality of the predictions is also examined as a function of RC model parameters and by comparing the FTLE and proper orthogonal decomposition (POD) found using the RC model with those found using the original model. Taken together, modal decomposition, FTLE, and trajectories characterize a particular model flow, and we use these to assess how well our RC models capture the generic dynamical and transport features of the QG and DG. We find that RC models trained with relatively little sparse data very effectively reproduce the above quantities and allow accurate forecasting over significant timescales.

It is worth noting that while this study relies entirely on synthetic data from various QG models, its results bear directly on the capability of RC models to work with any form of flow data in the quasi-geostrophic regime, whether from observations or more complex models.

II. RESERVOIR COMPUTING

As mentioned in Sec. I, echo state networks belong to a particular family of recurrent neural networks called reservoir computing, where the main idea is to drive a fixed, high-dimensional, random, sparse network, designated as the reservoir, with an input signal to produce a high-dimensional "echo" response, which is then used as a non-orthogonal signal basis to reconstruct the desired output, generally as a linear combination. In this work, we follow the standard implementation details.[30]

The main idea of reservoir computing is to map a set of sequential data $\mathbf{u}(t)$ of dimension n_{input} into a high-dimensional reservoir state vector $\mathbf{r}(t)$ of dimension n_{res} primarily by multiplying $\mathbf{u}(t)$ with a $n_{res} \times n_{input}$ matrix denoted as $\mathbf{W}^{\text{in}}$ [see Eq. (1a)]. A linear predictor is then applied on the two vectors to solve for an output $\mathbf{y}(t)$ of dimension n_{output}. Here, $t = 1, \ldots, \tau_{train}$ is the set of discrete time points with τ_{train} denoting the total number of data points in the training dataset.

In this case, the training procedure is formulated based on a dynamical system; the desired output $\mathbf{y}(t) = \mathbf{u}(t+1)$ and $n_{input} = n_{output}$. The training criterion depends solely on the hidden units, or state vector $\mathbf{r}(t)$, and the output targets $\mathbf{y}(t)$. As a result, this can be easily designed to be a convex optimization problem for linear predictors. The trade-off, however, is the problem of developing an effective representation of histories in the state vector $\mathbf{r}(t)$.

The state dynamics of the reservoir during the training phase is governed by the following system of equations:

$$\mathbf{r}(t+1) = f\big(\mathbf{W}\mathbf{r}(t) + \mathbf{W}^{\text{in}}\mathbf{u}(t+1) + \nu(t)\big), \tag{1a}$$

$$\hat{\mathbf{y}}(t+1) = \mathbf{W}^{\text{out}}[\mathbf{u}(t+1); \mathbf{r}(t+1)], \tag{1b}$$

where $\mathbf{r}(t)$ is the reservoir state vector, f is generally a sigmoid function (e.g., logistic, or tanh function), $\mathbf{W}$ is the $n_{res} \times n_{res}$ reservoir weight matrix, $\mathbf{W}^{\text{in}}$ is the $n_{res} \times n_{input}$ input weight matrix, $\mathbf{u}(t)$ is the input signal, $\hat{\mathbf{y}}(t)$ is the estimated value of the desired output $\mathbf{y}(t)$, which is nothing more than the observed data, $\mathbf{W}^{\text{out}}$ is a $n_{output} \times (n_{input} + n_{res})$-dimensional matrix of output weights to be applied as a final transformation on the extended system state, which consists of a concatenation of $\mathbf{u}(t+1)$ and $\mathbf{r}(t+1)$, and $\nu(t)$ is an n_{res}-dimensional noise vector. Note that the reservoir is driven by both the input sequence $\mathbf{u}(t+1)$ and the previous state $\mathbf{r}(t)$.

The goal of the training phase is to optimize the output layer weight matrix $\mathbf{W}^{\text{out}}$, such that the distance between $\hat{\mathbf{y}}(t)$ and the expected outputs $\mathbf{y}(t)$ is a minimum in the least squares sense, where $\hat{\mathbf{y}}(t)$ is the estimated value of $\mathbf{y}(t)$. We implement a linear regression with a regularization constraint β acting on the output weights; this scheme is known as ridge regression. For the set of training data, the optimal $\mathbf{W}^{\text{out}}_{\text{opt}}$ is given by

$$\mathbf{W}^{\text{out}}_{\text{opt}} = \underset{\mathbf{W}^{\text{out}}}{\operatorname{argmin}} \sum_{t=0}^{T} \left(\hat{\mathbf{y}}(t) - \mathbf{y}(t)\right)^2 + \beta ||\mathbf{W}^{\text{out}}||_2^2, \tag{2}$$

where $||\cdot||_2$ denotes the L_2 norm. This expression can be approximated using gradient descent or directly calculated as

$$\mathbf{W}^{\text{out}}_{\text{opt}} = (\mathbf{R} + \beta \mathbf{I})^{-1}\mathbf{P}, \tag{3}$$

where $\mathbf{R}$ is the correlation matrix of the extended system states, $\mathbf{P}$ is the cross correlation matrix of the extended system states and the desired outputs, and $\mathbf{I}$ is the identity matrix.

The main difference between this training procedure and traditional recurrent network approaches, as mentioned earlier, is that the weights on most of the network are randomly initialized and remain fixed throughout the procedure. The weight matrices $\mathbf{W}$ and $\mathbf{W}^{\text{in}}$ are both specified based on the recommendations of established research.[30] In particular, $\mathbf{W}^{\text{in}}$ is specified as a random matrix without any additional constraint, whereas $\mathbf{W}$ is specified as a random matrix with the additional constraint that its spectral radius (the maximum of the absolute value of the eigenvalues) is such that the system is brought to the edge of stability, where studied ESNs typically achieve maximum computational capability. To satisfy the constraint, it is usually sufficient to sample a range of spectral radii around unity, with preference given to higher values as the dynamics of the training data become more chaotic.[52] Note that the only weights that are changed are the matrix $\mathbf{W}^{\text{out}}$, which is ultimately determined through the training process to an optimal $\mathbf{W}^{\text{out}}_{\text{opt}}$ through Eqs. (2) or (3). This is one of the primary advantages of this type of a neural network structure and a training approach.

The prediction phase is different from the training phase in that the system is now autonomous and does not use $\mathbf{u}(t)$ as an input. Rather, the reservoir dynamics is governed by the single equation,

$$\mathbf{r}(t+1) = f\Big(\mathbf{W}\mathbf{r}(t) + \mathbf{W}^{\text{in}}\mathbf{W}^{\text{out}}_{\text{opt}}[\mathbf{u}(t); \mathbf{r}(t)] + \nu(t)\Big). \tag{4}$$

Equation (4) is the same as Eq. (1a), except that $\mathbf{u}(t+1)$ is replaced by the output feedback loop as $\mathbf{W}^{\text{out}}_{\text{opt}}[\mathbf{u}(t); \mathbf{r}(t)]$. The resultant reservoir states $\mathbf{r}(t)$ for all $t > \tau_{train}$ can always be projected to the input data space as

$$\hat{\mathbf{y}}(t) = \mathbf{W}^{\text{out}}_{\text{opt}}[\mathbf{u}(t); \mathbf{r}(t)]. \tag{5}$$

Figure 1 illustrates the training and prediction phases of the ESN framework. The resultant autonomous reservoir system has been found to successfully replicate different aspects of many chaotic dynamical systems, such as the Lorenz system.[35] Previous studies on these methods and their application to chaotic dynamical systems motivate the rest of the work in this study, where chaotic fluid dynamical systems are applied as inputs to the ESN framework.

III. STREAM FUNCTION DATA AND MODELING

For an open set $D \subset \mathbb{R}^2$, that is the domain of interest, we define the stream function $\boldsymbol{\psi}(\mathbf{x}, t)$ to be a time-dependent scalar field on D, where $\mathbf{x} \in D$. While analytic solutions of the stream function and velocity fields may be defined in some cases, many real world problems do not have a theoretical form. Instead, stream functions are usually presented as discrete snapshots over a finite-time period and, along with velocity fields, are interpolated from the discrete grid of points $\mathbf{X}_0 \subset D$ for all points $\mathbf{x} \notin \mathbf{X}_0$ to be used for particle trajectory integration. In this article, while the analytic stream function for the double-gyre flow is known, the methods are applied to the data as if the analytic solution is not known. In contrast, the analytic solutions of the more complicated QG flow are not known, and, thus, we have no choice but to apply our methodology to the data. Therefore, in this study, the stream function $\psi(\mathbf{x}, t)$ may refer to either the analytic form or the interpolated data. Although the two approaches are generally interchangeable, the interpolated version is of greater practical value.

We can formulate the stream function as a dynamical system and apply it as input to our reservoir computing framework; we frame the stream function as the feedback,

$$\boldsymbol{\psi}(\mathbf{x}, t_{i+1}) = \mathscr{F}(\boldsymbol{\psi}(\mathbf{x}, t_i)), \tag{6}$$

where $\mathscr{F} : D \mapsto D$ is a flow map of the stream function and t_i is the time index. The stream function can be used as a time series of τ snapshots, where each snapshot is in the domain D and $i = 1, \ldots, \tau$. By approximating the flow map $\mathscr{F}$ using machine learning, we can model the stream function and its temporal evolution and use the estimated model to calculate velocity fields and particle trajectories.

Stream function data are extracted for τ time steps at each grid point in $\mathbf{X}_0$. In our experiment, we are using a rectangular grid (though any grid scheme can be used); $\mathbf{X}_0$ is a $m \times n$ rectangular matrix, so each snapshot in time is an $m \times n$ matrix, and the entire data matrix containing all snapshots across τ time steps is of size $m \times n \times \tau$. Recall that the input to the ESN, i.e., $\mathbf{u}(t)$ in Sec. II, is a data vector of arbitrary size.

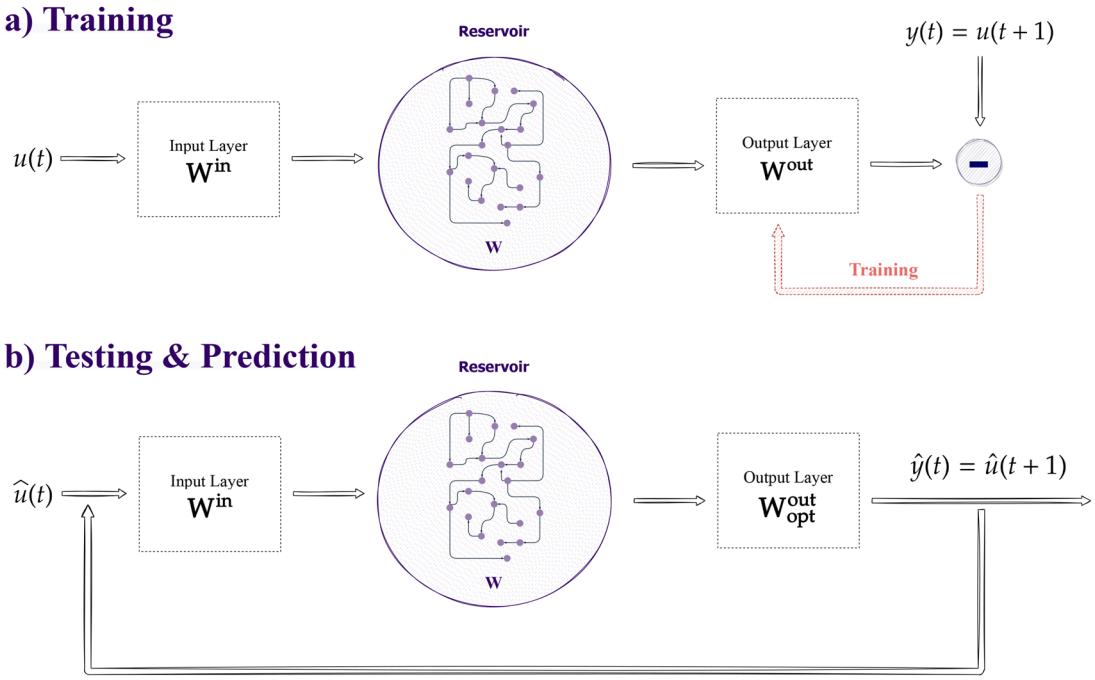

FIG. 1. Block diagrams representing the flow and transformation of data through (a) the training phase and (b) the testing and prediction phase of the ESN. The output layer weight matrix $\mathbf{W}^{\text{out}}$ changes between (a) and (b).

In order to use the data matrix in the ESN framework, we convert the $m \times n \times \tau$ data matrix to the (mn)-dimensional vector $\mathbf{u}(t)$ for each of the τ time steps.

In this work, we consider both the actual stream function $\psi(\mathbf{x}, t)$ (but sampled as though it was derived from an experiment) and the estimated stream function $\hat{\psi}(\mathbf{x}, t)$ modeled from training data as output from the ESN framework. A visualization of the data manipulation is shown in Fig. 2.

A. Training, testing, and prediction

The goal is to learn the flow map $\mathscr{F}$. Using the ESN framework, stream function time series data are input into the reservoir, which expands the dimensionality of the data via a series of matrix multiplications and nonlinear operations [recall Eq. (1a) and the description of reservoir computing in Sec. II]. The transformed data are referred to as the reservoir states.

A portion of the stream function data is designated as the training data. Given our complete data matrix is of dimension $(mn) \times \tau$, we can remove a portion of size $(mn) \times \tau_{train}$ where $\tau_{train} < \tau$. To actually use these data for training purposes, the stream function input data are paired to itself using a $1:1$ mapping through $t_i \rightarrow t_{i+1}$, where $t_i < \tau_{train}$. We will then have a mapping of $\psi(\mathbf{x}, t_i) \mapsto \psi(\mathbf{x}, t_{i+1})$. An unchanging dimension-reducing matrix is used to project the reservoir states to the input dimension, forming the output data. A model is fitted based on a linear regression of the output data on actual data. A visualization of the training is shown in Fig. 3.

Once the model is fitted on the designated training data, we allow the model to feedback its outputs, $\hat{\psi}(\mathbf{x}, t_{i+1})$ as inputs. We can compare these learned stream function fields against the actual stream function fields at every time index to evaluate the model.

IV. IMPLEMENTATION AND RESULTS

Large-scale ocean circulation is driven primarily by winds and the Earth's rotation, forming characteristic gyres spanning irregularly shaped basins. Dissipation together with the effective variation of planetary rotation with latitude lead to an intensification of these currents along a western boundary.[37]

To generate flow data, we adopt two well-known ocean models: a double-gyre flow and a quasi-geostrophic flow, both of which are used extensively as a foundation for the study of wind-driven ocean circulation. Stream function data, discretized on a spatial grid and in time, form the framework for modeling and forecasting. The same learning procedures are applied to DG and QG flows, and the results, predictions of stream function evolution and passive tracer transport, are evaluated by comparison to the actual solutions from the models. To further validate the predictions of our RC results, we compare both FTLE and modal decomposition (POD) of the actual and predicted systems.

A. Double-gyre model

We first consider the well-known double-gyre stream function model[2] to demonstrate our methodology. The DG flow is not the solution of any approximate ocean model; rather, it is a prescriptive formula for a flow's stream function with features that mimic ocean gyres, first used by Smith and Spiegel[53] to examine tracer transport and later widely adopted, and adapted, as a useful model of ocean basin flow. The DG stream function adopted here is prescribed as

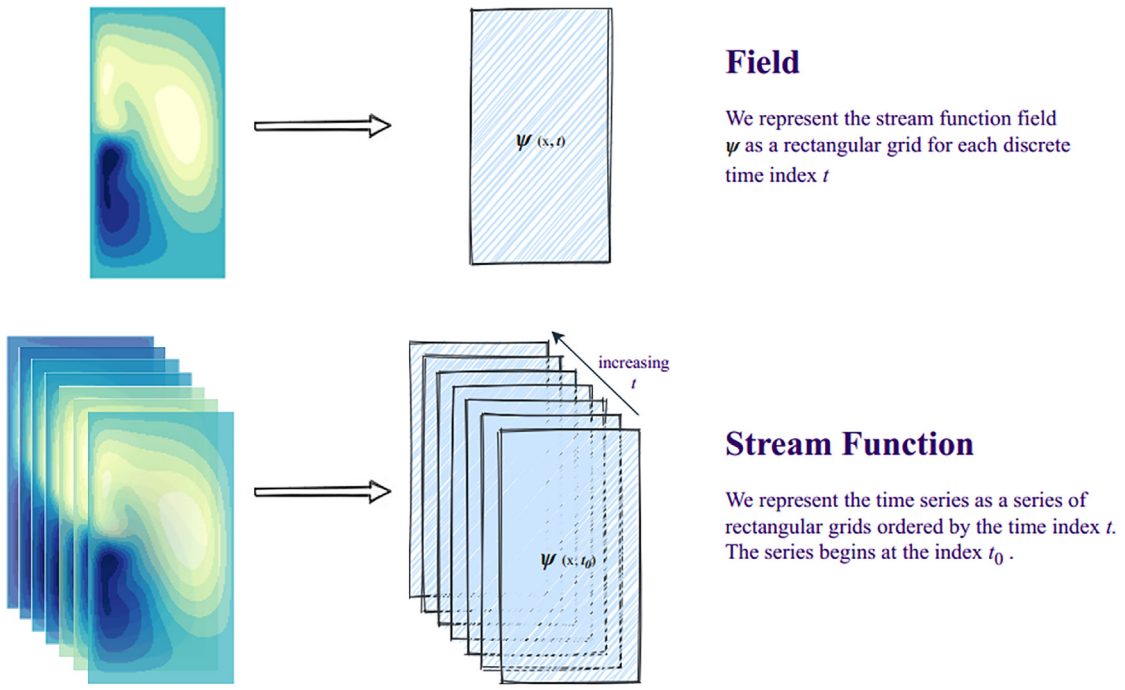

FIG. 2. Notation for a snapshot of the stream function field at a specific time t, and a time series of stream function fields that begins at the initial time t_0.

$$\psi(x, y, t) = A \sin(\pi f(x, t)) \sin(\pi y), \tag{7}$$

where

$$\begin{aligned} f(x, t) &= a(t)x^2 + b(t)x, \\ a(t) &= \varepsilon \sin(\omega t), \\ b(t) &= 1 - 2\varepsilon \sin(\omega t) \end{aligned} \tag{8}$$

and is defined over the rectangular domain $[0, 1] \times [0, 2]$. Although the DG model is a "toy" model, it has the advantage of providing direct flow data at any spatial or temporal resolution without the need to solve a PDE.[22]

An example of the DG stream function time evolution is shown in the bottom of Fig. 4. The quadratic function $f(x, t)$ causes periodic oscillations across the domain and is meant to capture the wind-stress driving effect. The term "double-gyre" comes from the visually apparent pair of vortices that oscillate across the domain. The terms ε and ω in Eq. (8) are parameters, which, respectively, determine the magnitude and frequency of the oscillations. Setting $\omega = 0$ would result in a double-gyre system with no oscillations and a pair of vortices centered

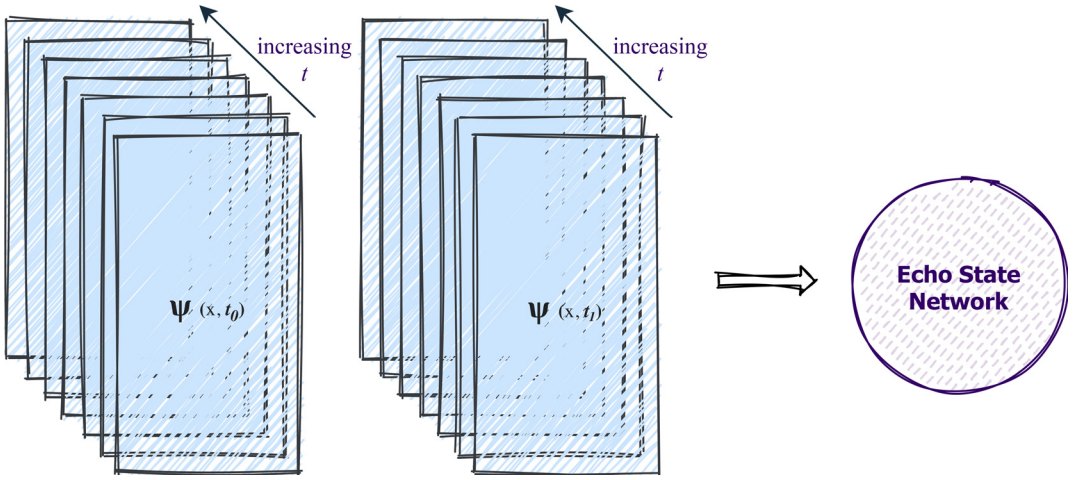

FIG. 3. The stream function input data for training is first paired through the time index $t_i \mapsto t_{i+1}$, creating a mapping for the training dataset, $\psi(\mathbf{x}, t_i) \mapsto \psi(\mathbf{x}, t_{i+1})$.

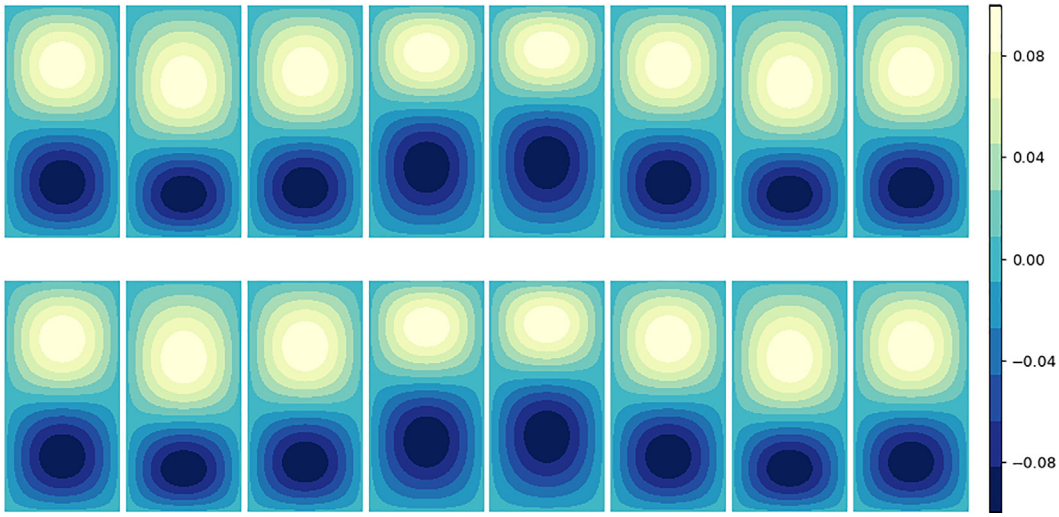

FIG. 4. The (top) actual and (bottom) estimated double-gyre stream function field, starting from some time step $t_i > \tau_{train} = 2000$ and increasing from left to right in intervals of ten time steps. The spectral radius is 2.3, the reservoir size is 5000 nodes, and the domain is $[0, 1] \times [0, 2]$.

at (0.5, 0.5) and (0.5, 1.5). In this study, we set $\varepsilon = 0.3$ and $\omega = \pi/5$. Importantly, the DG system has a complicated basin boundary structure in which the basins of attraction are intermingled, a signature of the existence of a fractal basin boundary. Because of this intermingling of the basin boundaries, one can expect a sensitive dependence on initial conditions in the particle trajectories.[2]

We can also consider the velocity field associated with this stream function. For any stream function ψ, the velocity field is given by its standard definition,

$$v = (u, v) = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}\right) \tag{9}$$

and for the double-gyre flow, it takes the following form:

$$\begin{aligned} u &= -\pi A \sin(\pi f(x)) \cos(\pi y), \\ v &= \pi A \cos(\pi f(x)) \sin(\pi y) \frac{df}{dx}. \end{aligned} \tag{10}$$

Although the DG flow does have a closed form solution for its stream function and velocity field, this is often not the case. In fact, as we shall soon see, more intricate and more realistic flow models of ocean dynamics generally do not have a closed form solution for the velocity field. As a result, while we do have the closed form solution to the velocity field for the double-gyre flow model, which provides a useful standard for evaluation, we will proceed as though we do not.

B. Quasi-geostrophic model

Quasi-geostrophy in its basic form represents a solution of the Navier–Stokes-based primitive equations of ocean circulation found using a series in powers of a small parameter, ε, identified as the Rossby number. Steady, geostrophic, flow is recovered as the leading order solution, representing dominant rotation (small ε). By expressing the next order solution in terms of leading order quantities, the QG model is found. Details of the derivation of QG and some of its variants can be found in Ref. 37.

The QG model is, thus, significantly more realistic but also more computationally demanding compared to the DG model. The QG model used in this study is a barotropic, single-layer QG model with a lateral (viscous, Munk) and bottom (drag, Stommel) friction that exhibits solutions that capture important geophysical fluid dynamics (GFD) features, including western intensification.[38] The QG model can be expressed as

$$\underbrace{\frac{\partial \nabla^2 \psi}{\partial t} + \overbrace{\varepsilon J(\psi, \nabla^2 \psi)}^{\text{Inertial}} + \frac{\partial \psi}{\partial x} = \mu \nabla^2 \psi}_{\text{Stommel}} + \overbrace{\lambda \nabla^4 \psi}^{\text{Munk}} + w_E, \tag{11}$$

where J is the Jacobian operator, μ is the Stommel friction, λ is the Munk friction, and w_E is the driving due to wind. Non-dimensional parameters, named Stommel (δ_S), Munk (δ_M), and Inertial (δ_I), are formulated with respect to the relative lengthscales of three important boundary layers.[38] The corresponding non-dimensional parameters, μ, λ, and ε, capture the relative importance of bottom friction, lateral diffusion, and the nonlinearity; these parameters and the various QG models are defined in Appendix A. In the following, we note that only the pure Stommel case, defined by $\delta_M = \delta_I = 0$, with dissipation in the form of frictional drag, has a linear closed-form solution. Nonlinear cases, having $\delta_I \neq 0$, and viscous dissipation, $\delta_S \neq 0$, are more commonly applied to specific problems[38] in ocean dynamics and, thus, of more interest. Such models are also expected to be more challenging to capture using machine learning. We focus primarily on modeling the Highly Nonlinear variation of the QG model since it encompasses all aspects of the QG model formulation. Of the four variations (Stommel, Munk, inertial, highly nonlinear, defined in Appendix A), the Highly Nonlinear version is the most complex in its behavior and might be expected to be the most difficult variation to model.[45,54,55] A sample of the QG stream function time evolution is shown in the bottom of Fig. 5. We note that all QG model solutions used in this study were verified, by monitoring kinetic energy, to be equilibrated in the sense that transients had effectively vanished.

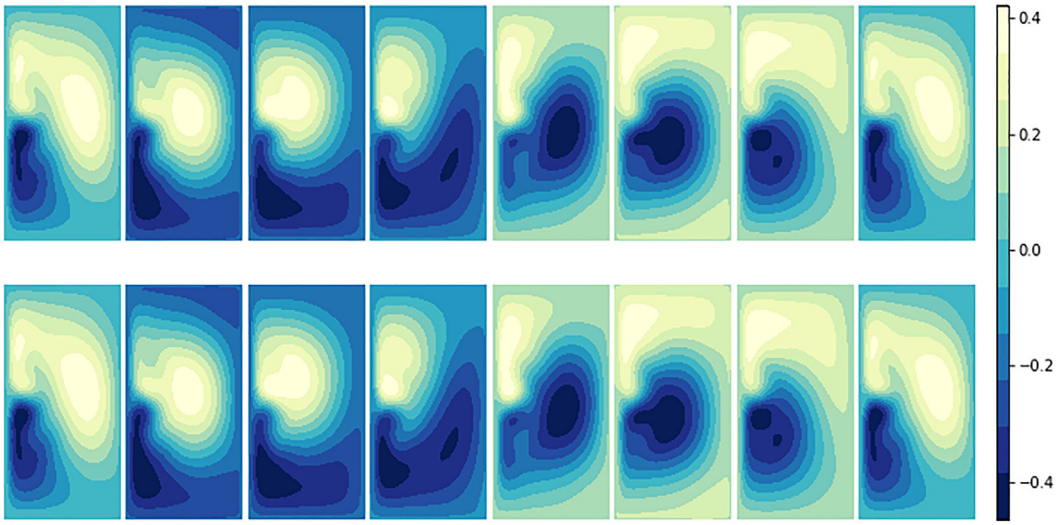

FIG. 5. The (top) actual and (bottom) estimated QG stream function field (highly nonlinear version), starting from some time step $t_i > \tau_{train} = 100$ and increasing from left to right in intervals of 10 time steps. The spectral radius is 1.0, the reservoir size is 4000 nodes, and the domain is $[0, 1] \times [0, 2]$.

C. Evaluation of the learned models

We begin evaluation of the learned models by analyzing the differences between the stream function fields. We then evaluate the differences in particle behavior within the actual and estimated stream function fields. Finite-time Lyapunov exponent (FTLE) fields are then the natural extension to the Lagrangian (particle trajectory) approach to understanding the difference in the flows, as we can essentially understand FTLE field analysis to be both an ensemble analysis of particle trajectories as well as a general structural analysis of the flow. Finally, modal structure decomposition comparisons between the actual and estimated stream function fields allow for some understanding of the dimensionality, or complexity, of the estimated field with respect to the actual field. In Secs. IV C 1–IV C 4, each of these techniques will be applied.

1. Stream function comparison

For both DG and QG models, we will analyze the difference between the estimated and actual stream function fields. In this section, we will only discuss the simpler DG model and the Highly Nonlinear variation of the QG model; however, in Appendix B, we include results for the other variations of the QG model. For our purposes, the Highly Nonlinear variation of the QG model is most relevant as it is the most difficult to model and solve numerically.

Qualitatively, Figs. 4 and 5 show that the estimated and actual stream functions exhibit almost identical behavior. The absolute error metrics between the actual and estimated stream function fields vary with the different hyperparameter tunings (training size, reservoir size, and spectral radius) used during the training of the ESN model. Figures 6 and 7 show the gradual increase in the average absolute error between the estimated and actual stream functions for various training sizes. While the QG model examined here represents the most dynamically complex case, a more complete picture of the QG stream

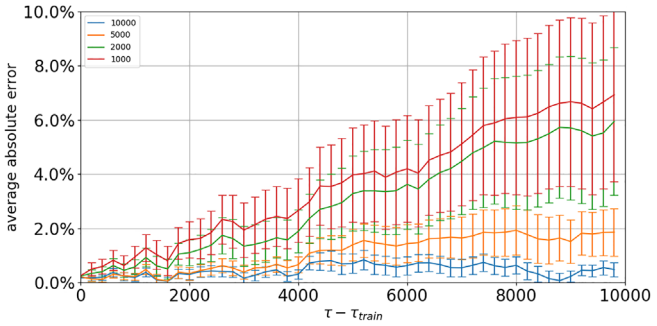

FIG. 6. Average error of estimated DG stream function against actual DG stream function evolved over 10 000 time steps starting from τ_{train}, with varying training lengths. The error bars represent the standard deviation of error.

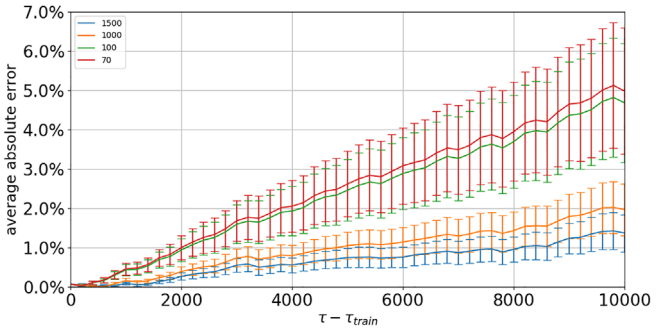

FIG. 7. Average error of estimated QG stream function against actual QG stream function (highly nonlinear version) evolved over 10 000 time steps starting from τ_{train}, with varying training lengths. The error bars represent the standard deviation of error.

function can be found in Appendix B, where additional model parameters are considered, including examples in the Stommel and Munk regimes. Other hyperparameter dependencies are explored in Appendix C.

Training size is specifically highlighted here because total training computation time has a quadratic complexity with respect to training size. Notably, the QG model only requires 1 "period" of training data, approximately 70 time steps, for reasonable results. A model with fewer than 70 time steps fails to replicate the actual stream function; this makes sense as the model never "sees" all the ways the stream function evolves within an entire period.

2. Particle trajectories

While a stream function is itself useful and is closely related to sea surface height (SSH) data, a major area of interest and application is the study of transport; in particular, the dynamics of particles placed within the field. Particle trajectories in time-dependent flows are complex and not generally predictable, even in simple flow fields, as has been appreciated since the early studies of chaotic advection.[23,53] In this study, we adopt trajectories and the related quantity of FTLE fields as a way to validate the fidelity of our RC model. Let the open set $D \subset \mathbb{R}^2$ be the domain of interest, and the stream function $\boldsymbol{\psi}(\mathbf{x}(t), t)$ be a time-dependent scalar field on D. Particle trajectories can be calculated for each point $\mathbf{x}(t_0) \in D$ initialized at t_0, as we consider a flow to consist only of incompressible fluid particles. Each particle's trajectory is governed by a flow map $\phi_{t_0}^{t_0+\Delta t} : D \mapsto D$, where

$$\mathbf{x} \mapsto \phi_{t_0}^{t_0+\Delta t}(\mathbf{x}) \tag{12}$$

for some time interval Δt. For the flows studied in this work, the flow map $\phi_{t_0}^{t_0+\Delta t}$ is directly dependent on the smooth velocity field $v(\mathbf{x}, t)$ on D, where $\mathbf{x}$ is a two-dimensional position vector (x, y) and where Eq. (12) is integrated forward in time using a fourth-order Runge-Kutta numerical method.[56]

Measuring the accuracy of the model can be approached in several ways. In practical fields of work and engineering, stream functions are useful for particle trajectory tracking. For this, it is important to remember that the underlying particle dynamics is chaotic and inherently unpredictable by nature, so any attempt to evaluate estimated models should be regarded with respect to the level of chaos in the system.

To account for the inherent chaotic nature of the particle dynamics, estimated particle trajectory deviations are compared to initially perturbed actual particle trajectories over time. Let an arbitrary particle trajectory has an initial condition $\mathbf{x}_0 \in D$, so that $\mathbf{x}(t, \mathbf{x}_0) \in \mathbb{R}^{2\times\tau}$. As the evolution of $\mathbf{x}(t, \mathbf{x}_0)$ depends on the governing stream function, there are two unique particle trajectories defined for each unique initial condition: one defined for an actual stream function and one defined for an estimated stream function. For each initial condition, the two defined particle trajectories are compared over time.

As previously mentioned, it is also important to acknowledge the inherent chaotic nature of the particle dynamics; the estimated particle trajectory's deviation is compared to an initially perturbed actual particle trajectory's deviation over time, which can be loosely thought of as an upper bound on the predictability of the system's particle trajectories. In Figs. 8 (DG) and 10 (QG), the deviation from the true particle trajectory governed by the actual stream function is compared to

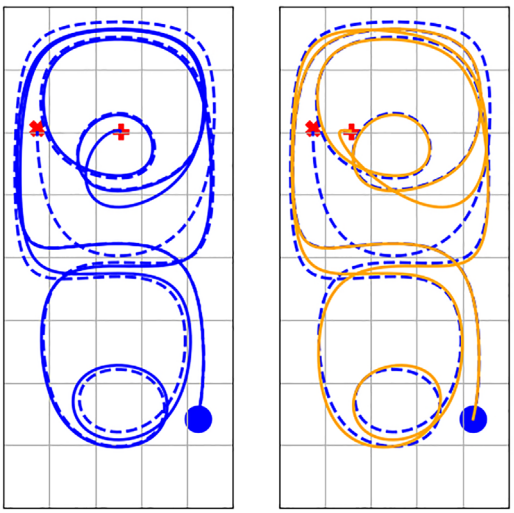

FIG. 8. Particle position plots for the first 1000 time steps after τ_{train} based on (left) actual and (right) estimated DG stream functions. The left image shows two particle trajectories governed by the same stream function but with initial conditions which differ by an order of magnitude 10^{-3} relative to the grid resolution of the stream function, which is 80×160. The right image shows two trajectories with the same initial conditions but which are governed by different stream functions, with the solid trajectory corresponding to the estimated stream function and the dashed trajectory corresponding to the actual stream function. The dashed trajectories in both the left and right images are the same.

(1) a particle trajectory governed by the actual stream function with slightly perturbed initial conditions, and
(2) a particle trajectory governed by the estimated stream function with the same initial conditions.

While these trajectories are not necessarily representative of all initial conditions, they do present some insight on how the particle trajectories based on the learned model track the actual particle trajectories with respect to the inherent chaotic nature of the dynamics.

Figures 8 and 10 show the trajectories found using perturbed initial conditions (left) and found using the actual and learned models (right) for the DG and QG models, respectively. The perturbed trajectory and the estimated particle trajectory both begin to deviate between 500 and 1000 time steps. In other words, trajectories governed by the estimated stream function are approximately equivalent to trajectories governed by the actual stream function with an initial deviation of 10^{-3} for DG (Fig. 9) and 10^{-2} for QG (Fig. 11).

It is worth noting in Fig. 10 that the dramatically different final position of the two trajectories is not caused by the actual and estimated streamfunctions being dramatically different, as, in fact, the two streamfunctions are quite close to one another as seen in Fig. 5. Rather, this particular choice of initial condition is such that the particles start essentially on top of an LCS, which serves as a division of the phase space dynamics. Even the small discrepancy between the actual and estimated streamfunction has led to the initial condition lying on different sides of the LCS (with respect to the actual and estimated model), and, thus, over time, the trajectories end up in different basins of attraction. If the initial condition was positioned on the same side of the LCS (for the actual and estimated models), their final

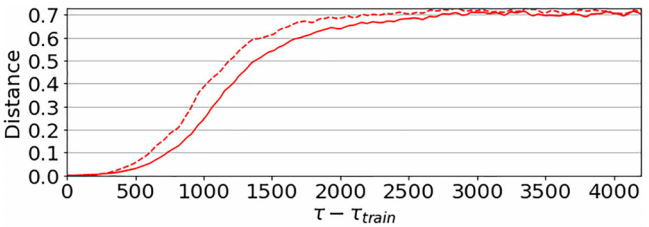

FIG. 9. Average deviation for the DG flow of 10 000 particle trajectories from the true particle trajectory for (solid) trajectories governed by the actual stream function with perturbed initial conditions, and (dashed) trajectories governed by the estimated stream function with the same initial conditions. The initial conditions differ by an order of magnitude 10^{-3} relative to the grid resolution of the stream function, which is 80×160.

positions would be close to one another. As with any deterministic flow, whether or not two trajectories diverge and end up in different basins is a matter of where they start in relation to the LCS.

To obtain a general sense of how particle trajectories deviate with respect to any initial condition in the domain, an average deviation is calculated over time using a large sample of initial conditions. For simplicity, deviation is quantified with the Euclidean norm between particle positions at corresponding time steps. As shown in Figs. 9 (DG) and 11 (QG), the particle trajectories associated with the estimated stream function deviate from the particle trajectories associated with the actual stream function at about the same time as trajectories associated with the actual stream function but with perturbed initial conditions. Note that different initial perturbation amplitudes, namely, 10^{-3} and 10^{-2} for the DG and QG models, respectively, were used in order to match the trajectory deviations for the two types of flow.

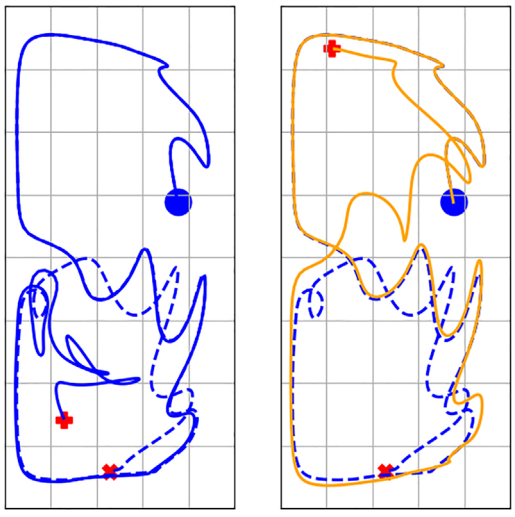

FIG. 10. Particle position plots based on (left) actual and (right) estimated QG stream functions (highly nonlinear version). The left image shows two particle trajectories governed by the same stream function but with initial conditions which differ by an order of magnitude 10^{-2} relative to the grid resolution of the stream function, which is 64×128. The right image shows two trajectories with the same initial conditions but which are governed by different stream functions, with the solid trajectory corresponding to the estimated stream function and the dashed trajectory corresponding to the actual stream function. The dashed trajectories in both the left and right images are the same.

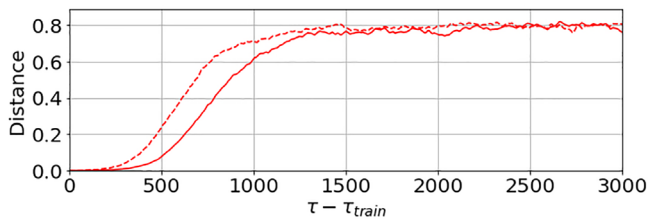

FIG. 11. Average deviation for the Highly Nonlinear QG flow of 10 000 particle trajectories from the true particle trajectory for (solid) trajectories governed by the actual stream function with perturbed initial conditions, and (dashed) trajectories governed by the estimated stream function with the same initial conditions. The initial conditions differ by an order of magnitude 10^{-2} relative to the grid resolution of the stream function, which is 80×160.

3. Finite-time Lyapunov exponent field

The computation of finite-time Lyapunov exponents is often used to find coherent structures in fluid flows.[22,57–59] The FTLE provides a measure of how sensitively the system's future behavior depends on its current state.

We consider a velocity field $\boldsymbol{v} : D \times I \to D$, with $D \subset \mathbb{R}^2$ that is defined over a time interval $I = [t_i, t_f]$, and the system of equations

$$\dot{\mathbf{x}}(t; t_i, \mathbf{x}_0) = \boldsymbol{v}(\mathbf{x}(t; t_i, \mathbf{x}_0), t), \tag{13a}$$

$$\mathbf{x}(t_i; t_i, \mathbf{x}_0) = \mathbf{x}_0, \tag{13b}$$

where $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^2$ and $t \in I$. This dynamical system has quantities known as Lyapunov exponents that measure the growth rates of the linearized dynamics about the trajectory of the system. To find the finite-time Lyapunov exponents (FTLE), the Lyapunov exponents are computed on a restricted finite time interval.

To compute FTLE values, we define the domain of interest D as an evenly spaced grid of two-dimensional points with initial position $\mathbf{x_0}$ defined at the grid points. Then, all points are numerically integrated using Eqs. (13a) and (13b). The flow map ϕ determines the advection of the initial points as follows:[22,57–59]

$$\phi_{t_i}^{t_i+T} : \mathbf{x}_0 \to \phi_{t_i}^{t_i+T}(\mathbf{x}_0) = \mathbf{x}(t_i + T; t_i, \mathbf{x}_0). \tag{14}$$

Then, the FTLE can be defined as

$$\sigma(\mathbf{x}, t_i + T, T) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\Delta)}, \tag{15}$$

where $\lambda_{\max}(\Delta)$ is the maximum eigenvalue of the right Cauchy–Green deformation tensor Δ, which is given as

$$\Delta(\mathbf{x}, t_i + T, T) = \left(\frac{d\phi_{t_i}^{t_i+T}(\mathbf{x}(t))}{d\mathbf{x}(t)}\right)^* \left(\frac{d\phi_{t_i}^{t_i+T}(\mathbf{x}(t))}{d\mathbf{x}(t)}\right) \tag{16}$$

with $*$ denoting the adjoint.

For a given $\mathbf{x} \in \mathbb{R}^2$ at initial time t_i, Eq. (15) gives the maximum finite-time Lyapunov exponent for some finite integration time T (forward or backward) and provides a measure of the sensitivity of a

trajectory to small perturbations. The FTLE field given by $\sigma(\mathbf{x}, t_i, T)$ can be shown to exhibit ridges of local maxima in phase space. The ridges of the field indicate the location of attracting (backward time FTLE field) and repelling (forward time FTLE field) structures. In two-dimensional space, the ridge is a curve which locally maximizes the FTLE field, so that transverse to the ridge one finds the FTLE to be a local maximum.[2,22]

Given the two-dimensional grid of points spanning the domain of the flow field, one can compute $[x_{ij}(t), y_{ij}(t)]$ and $[x_{ij}(t+T), y_{ij}(t+T)]$, where $(x_{ij}(t), y_{ij}(t))$ denotes the (i,j)th points in the computational grid, and $[x_{ij}(t+T), y_{ij}(t+T)]$ denotes the corresponding set of points after they have been advected by the flow for integration time T. The spatial gradient of the flow map is approximated as

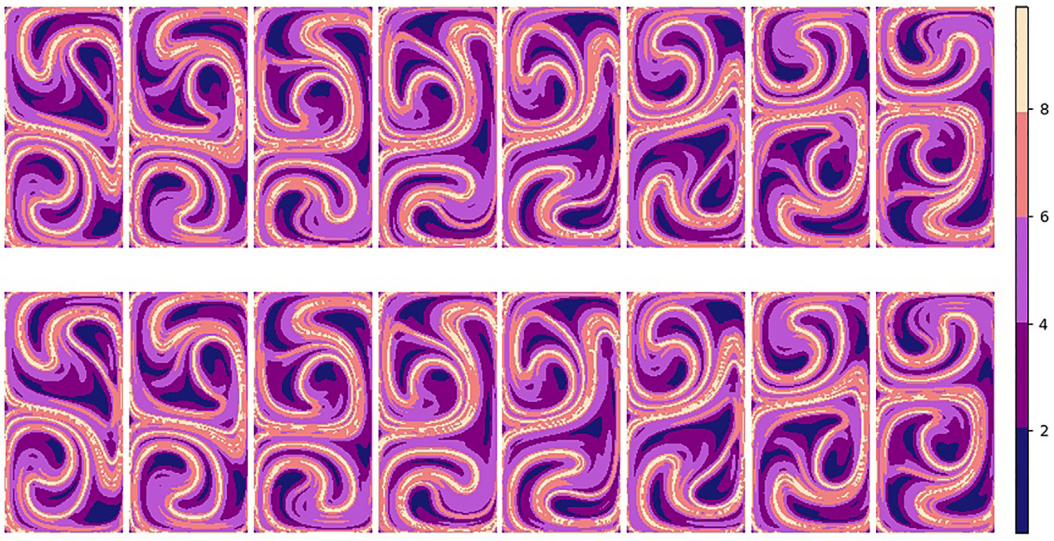

FIG. 12. FTLE fields calculated using particle trajectories governed by (top) actual and (bottom) estimated double-gyre stream functions within 100 time steps after the start of the prediction phase.

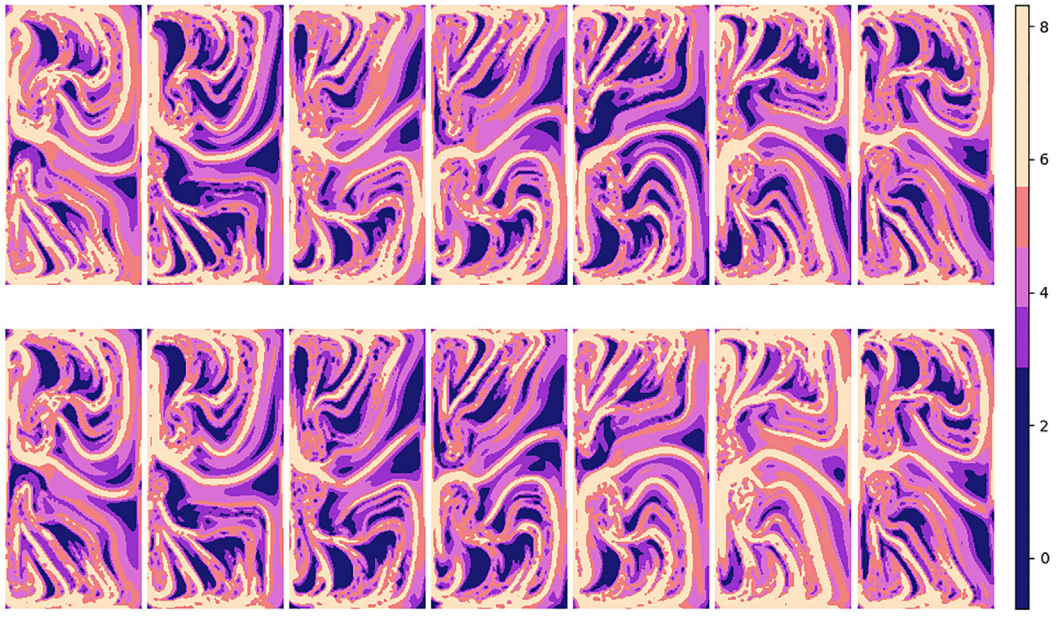

FIG. 13. FTLE fields calculated using particle trajectories governed by (top) actual and (bottom) estimated QG stream functions (highly nonlinear version) within 100 time steps after the start of the prediction phase.

$$\frac{\mathrm{d}\phi_t^{t+T}}{\mathrm{d}\mathbf{x}} \approx \begin{pmatrix} \frac{x_{i+1,j}(t+T)-x_{i-1,j}(t+T)}{x_{i+1,j}(t)-x_{i-1,j}(t)} & \frac{x_{i,j+1}(t+T)-x_{i,j-1}(t+T)}{y_{i,j+1}(t)-y_{i,j-1}(t)} \\ \frac{y_{i+1,j}(t+T)-y_{i-1,j}(t+T)}{x_{i+1,j}(t)-x_{i-1,j}(t)} & \frac{y_{i,j+1}(t+T)-y_{i,j-1}(t+T)}{y_{i,j+1}(t)-y_{i,j-1}(t)} \end{pmatrix}$$

using central finite differences.

The computations of the FTLE field for the DG and QG flows were performed using a fourth-order Runge-Kutta algorithm. However, the final integration time T differed between the two problems since particles in the DG flow never leave the domain D, while particles in the QG flow will exit the domain D after some period of time. To resolve this issue, the integration time must be limited by some upper bound, which was chosen based on observation of the FTLE and particle fields at various integration times.

Figures 12 and 13 show the forward-time FTLE fields computed using the actual and estimated stream functions fields for the DG and QG models, respectively. The FTLE field based on the estimated, interpolated velocity field qualitatively agrees well with the actual FTLE field. The FTLE fields of the estimated and actual velocity fields for the variations of the QG flows agree qualitatively as well. As the flows become more nonlinear (e.g., Fig. 13), it becomes more difficult to pinpoint the coherent structures. Nevertheless, the same phenomenon is seen in the estimated FTLE field as compared with the actual FTLE field. FTLE fields for Stommel, Munk, and Inertial QG models are shown in Appendix B and can be seen to have more structure than DG FTLE fields, yet have fewer features than the Highly Nonlinear QG case.

4. Modal structure decomposition

We use the proper orthogonal decomposition (POD) method[60] to generate prominent modal structures (basis functions) and quantify the amount of information extracted from both the estimated and actual stream function data.[61,62] Again, we are observing the scalar stream function field $\psi(\mathbf{X}_0, t)$, where $\mathbf{X}_0 \subset D$ are the grid points uniformly sampled in the domain of interest $D \subset \mathbb{R}^2$. The POD method serves to decompose $\psi(\mathbf{X}_0, t)$ as a number of optimal (in the L^2 sense) orthogonal basis functions $\phi_i(\mathbf{X}_0)$ where $i \leq n$, so that the original stream function field can be reconstructed as the linear combination,

$$\psi(\mathbf{X}_0, t) - \bar{\psi}(\mathbf{X}_0) = \sum_{i=1}^{n} \alpha_i(t)\phi_i(\mathbf{X}_0), \tag{17}$$

where $\bar{\psi}(\mathbf{X}_0)$ is the temporal mean of the stream function field, α_i is the temporal coefficient, and the left hand side of Eq. (17) provides the fluctuations from the mean field.

Given the $m \times n$-dimensional $\mathbf{X}_0$, then one can construct the data matrix $\mathbf{X} \in \mathbb{R}^{mn \times \hat{\tau}}$, where $\hat{\tau}$ is the total number of time steps in the prediction phase, by concatenating the flattened snapshot vectors $\psi(\mathbf{X}_0, t)$. The optimal orthogonal basis functions ϕ_i are determined by solving the eigenproblem,

$$\mathbf{R}\phi_i = \lambda_i \phi_i, \quad \phi_i \in \mathbb{R}^n, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0, \tag{18}$$

where $\mathbf{R}$ is the covariance matrix $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{n \times n}$ (where T denotes transpose). The largest eigenvalues λ_i correspond to the eigenvectors ϕ_i with the most contribution to reconstructing the original stream function field. The results of this decomposition can be used as a visual evaluation of the estimated stream function fields. By decomposing both the actual and estimated stream function fields to their respective optimal basis functions, one can analyze the differences between the different modal structures.

As shown in Fig. 14, the modal structures for the DG model are ordered by variance contribution (i.e., eigenvalue magnitude). The agreement between the actual and estimated modal structures is excellent. Moreover, the first two modes contribute almost all of the variance in both cases. Similar observations can be made for the highly nonlinear variant of the QG model, as shown in Fig. 15. The modal structure of the Stommel, Munk, and inertial cases exhibits a similar close agreement and need not be shown. While all the QG cases include some degree of nonlinearity, two or three modes are dominant, as we expect from this single-layer QG model.[37] These POD results reflect the fact that the models used here, despite nonlinearity, involve a basic two attractor basin with spatially and temporally periodic driving. The efficacy of POD to capture the dynamics for such systems is expected and now well-known. The trajectories and the FTLE portraits that can be thought of as capturing families of

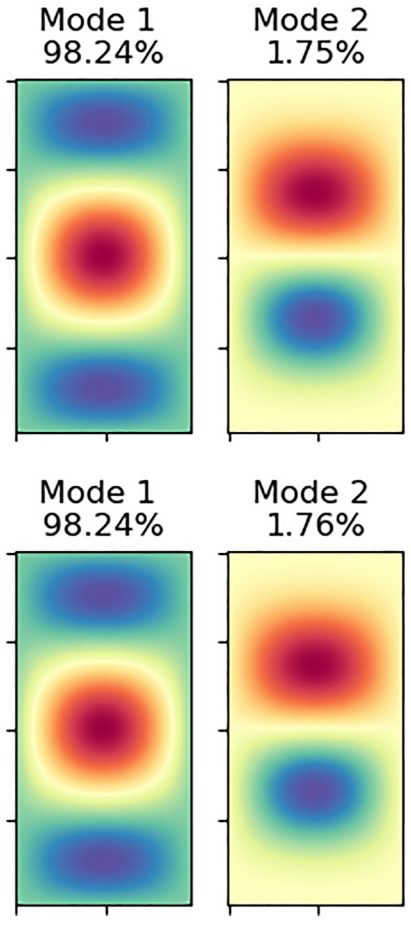

FIG. 14. POD modes for the (top) actual and (bottom) estimated DG stream function. The percentage above each modal structure is the proportion of variance contributed by each mode.

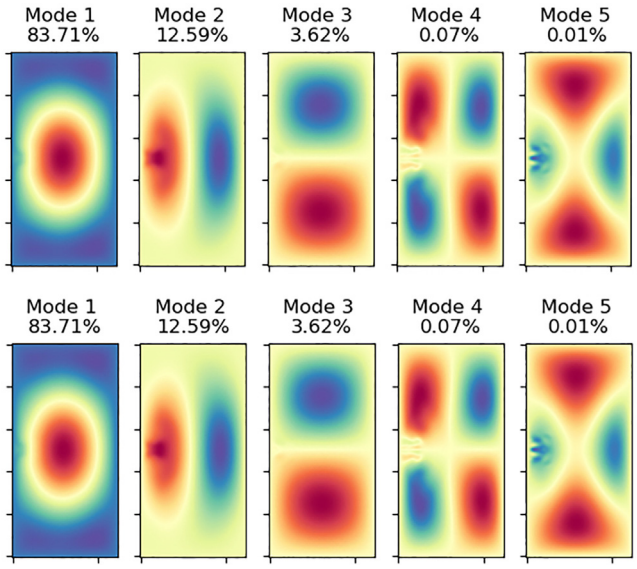

FIG. 15. POD modes for the (top) actual and (bottom) estimated highly nonlinear QG stream function. The percentage above each modal structure is the proportion of variance contributed by each mode.

trajectories are more challenging to model because they sample the fractal basin boundaries created by the time-dependent forcing and are, thus, themselves, chaotic. This dichotomy between the simplicity of the flow and the complexity of its trajectories is the central idea first described by Aref[23] for a time-dependent, two-dimensional Stokes flow.

V. CONCLUSION

This work explains the methodology, development, and evaluation of a machine learning approach to modeling and prediction of geophysical flows. Reservoir computing's relevance to dynamical systems is introduced, and RC is applied to two well-known ocean circulation models: the simple prescriptive double-gyre stream function model (DG) and the more complicated quasi-geostrophic PDE model (QG).

The effectiveness of this reservoir computing approach to modeling and characterizing ocean circulation models through the stream function is evaluated using a range of quantitative and qualitative measures, including mean stream function error, the predictability of particle trajectories, and comparisons of FTLE and modal signatures. Reservoir computing models formed from QG model data exhibited good predictive power even in these chaotic dynamical systems described by a nonlinear PDE, and over a range of values of its control parameters. Addressing the problems caused by limited datasets, our reservoir computing approach showed that estimated flows derived from a small amount of data are shown to agree well with the actual flows.

Of particular relevance to the work presented here is a number of recent studies that involve machine learning in a quasi-geostrophic context. Bolton and Zanna[8] demonstrated that ML can deduce unresolved physical processes from sparse ocean data, which correlates with our findings that RC models capture the dynamics of QG models. While Bolton and Zanna used both QG (albeit 3D layered models) and ocean general circulation models to produce data, their ML approach was limited to very computationally expensive convolutional neural networks. The effectiveness of an RC framework over other ML options was validated recently by Chattopadhyay *et al.*[63] for the Lorenz96 system, both in terms of its short-term predictability and long-term statistics. Also, for highly realistic ocean models (e.g., CESM2), a RC framework has been shown to perform better forecasting than other approaches[64] including those involving modal analysis. Models of complex physical systems that have predictive capability without demanding perfect data or vast computational resources, such as the RC models explored here, have countless applications. Ocean flows and QG models of them can serve as a framework for climate sub-models, contaminant tracking,[65] or search and recovery.[66]

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation (Award Nos. DMS 1418956, CMMI 1462823, CNS 1625636, CMMI 2121919, and CMMI 2121923). K.Y. and P.Y. are grateful to L. Smith for his valuable feedback, and P.Y. acknowledges K. Helfrich for providing the core QG model code.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Kevin Yao: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Eric Forgoston:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Philip Yecko:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: QUASI-GEOSTROPHIC MODEL

The quasi-geostrophic (QG) ocean circulation model used here is derived in many standard texts. Our formulation is based on that of Pedlosky,[37,38] which can be consulted for more details.

The single layer (or barotropic) QG model can be formulated using stream function ψ, or vorticity, ξ, where $\xi = \nabla^2\psi$, viz

$$\frac{\partial \nabla^2\psi}{\partial t} + J(\psi, \nabla^2\psi) + \beta\frac{\partial \psi}{\partial x} = \frac{f_0}{D}w_E - \frac{f_0\delta_E}{2D}\nabla^2\psi + A_H\nabla^4\psi, \quad \text{(A1)}$$

where f_0 is the Coriolis parameter at the central latitude of the gyre, δ_E is the bottom layer thickness, β is the northward spatial derivative, D is the major layer, A_H is the turbulent viscosity coefficient, and J is the Jacobian,

$$J(f,g) := \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial g}{\partial x}\frac{\partial f}{\partial y}, \quad \text{(A2)}$$

describing the advection of relative vorticity by the motion field. Vertical frictional forces, though weak, contribute to the flow by stretching the planetary vorticity. We let the upper Ekman pumping velocity be

$$w_E = w_0 \sin\left(\frac{2\pi y}{L} - w_P \frac{2\pi y}{L} \sin(\omega t)\right) \tag{A3}$$

representing a time-dependent pumping velocity, where w_0 is the amplitude of the Ekman pumping, w_P is the amplitude of the periodic driving perturbations, and L is the characteristic latitudinal length scale. Applying dimensional analysis, we assume the flow is contained in a basin of this characteristic length scale L, has characteristic horizontal velocity scale U, and has characteristic Ekman pumping scale w_E, so that

$$U = w_E \frac{f_0}{\beta D}. \tag{A4}$$

We can non-dimensionalize the stream function ψ, the coordinate distances x and y, and we can scale t with $1/\beta L$, to arrive at the form

$$\underbrace{\frac{\partial \nabla^2 \psi}{\partial t} + \overbrace{\varepsilon J(\psi, \nabla^2 \psi)}^{\text{Inertial}} + \frac{\partial \psi}{\partial x} = \mu \nabla^2 \psi}_{\text{Stommel}} + \overbrace{\lambda \nabla^4 \psi}^{\text{Munk}} + w_E \tag{A5}$$

TABLE I. Numerical values of the distinguishing parameters, including the length scales, for each of the four QG model variations: Stommel, Munk, inertial, and highly nonlinear; quantities described in text.

QG model variations					
	δ_S	δ_M	δ_I	w_p	Re
Stommel	0.03	⋯	0.02	0.3	>1
Munk	0.04	0.03	0.02	0.2	<1
Inertial	0.04	0.03	0.05	0.2	>1
Highly nonlinear	0.01	0.03	0.05	0.5	>1

as given in the main text, Eq. (11), where the non-dimensional parameters

$$\mu = \frac{\kappa}{\beta L} = \left(\frac{\delta_S}{L}\right), \quad \lambda = \frac{A_H}{\beta L^3} = \left(\frac{\delta_M}{L}\right)^3, \quad \varepsilon = \frac{U}{\beta L^2} = \left(\frac{\delta_I}{L}\right)^2 \tag{A6}$$

are formulated with respect to the relative length scales of three important boundary layers: (δ_S), (δ_M), and (δ_I), which are, respectively, named Stommel, Munk, and inertial. Note that in the Stommel formulation, κ is the coefficient associated with bottom drag. The corresponding non-dimensional parameters, μ, λ, and ε, capture the relative importance of bottom friction, lateral diffusion, and nonlinearity.

The flow is contained in a basin of characteristic scale L; hence, each of the non-dimensional parameters is calculated based on ratios of the boundary length scales to the characteristic length L. In general, we associate L with the scale of the motion in the flow, though we expect small-scale motions specifically in the western boundary current.

The Reynolds number, Re, may be defined as the ratio of the inertial advection of relative vorticity to the diffusion of vorticity, i.e.

$$\mathrm{Re} = \frac{\varepsilon}{\lambda} = \left(\frac{\delta_I}{\delta_M}\right)^2, \tag{A7}$$

where the second equality holds when the boundary layer is viscous and has the scale of the Munk layer.[38] When $\mathrm{Re} > 1$, nonlinearity influences the flow around the boundary layer, and when $\mathrm{Re} < 1$, linear viscous physics dominates the boundary layer flow.

There are no closed form solutions for QG models with the exception of a pure Stommel case where $\lambda = \varepsilon = 0$. The Stommel, Munk, and inertial monikers are common, and Pedlosky's review[38] can be consulted for more details. What we refer to as Highly Nonlinear is a special case of Inertial with a relatively large amplitude driving; gyres of this model are, thus, highly spatially asymmetric and aperiodic. The main text contains results of our machine learning framework for this model because it exhibits the most nonlinear and

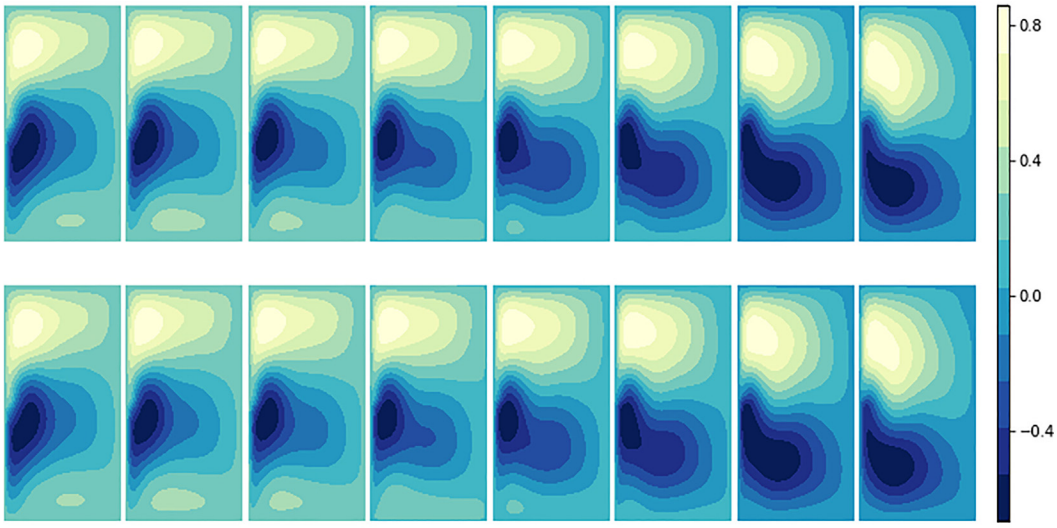

FIG. 16. The (top) actual and (bottom) estimated QG stream function field (Stommel version), with time increasing from left to right.

irregular behavior. Results for the other variants can be found in Appendix B. The four models and their length scale values used for computation are outlined in Table I.

APPENDIX B: RESULTS FOR QUASI-GEOSTROPHIC VARIATIONS

This appendix highlights the various results for the Stommel, Munk, and inertial variations of the QG model. Figures 16, 17, and 18 compare the estimated and actual stream functions for the Stommel, Munk, and Inertial variants, respectively. Figures 19, 20, and 21 compare the estimated and actual FTLE fields for the Stommel, Munk, and inertial variants, respectively.

APPENDIX C: ECHO STATE NETWORK HYPERPARAMETER STUDIES

This appendix presents the impact of the values of the principal hyperparameters, namely, reservoir size and spectral radius, on the error of the estimated stream function relative to the actual stream function over time.

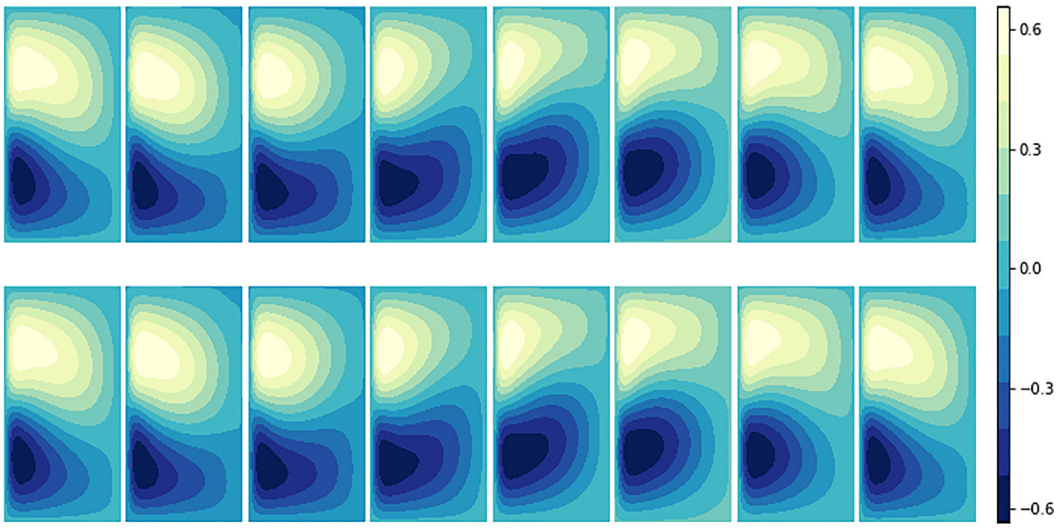

FIG. 17. The (top) actual and (bottom) estimated QG stream function field (Munk version), with time increasing from left to right.

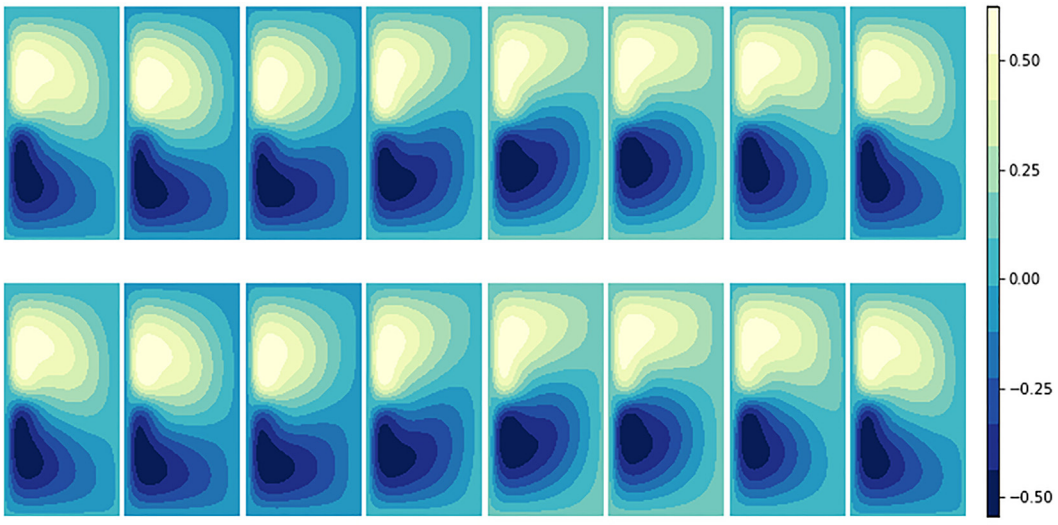

FIG. 18. The (top) actual and (bottom) estimated QG stream function field (inertial version), with time increasing from left to right.

The DG model performs reasonably well for reservoir sizes of 4000, 5000, or 6000 nodes for short training times. For longer training times, a reservoir size of 5000 nodes gives less error, but we were not able to identify this reservoir size dependence. Nevertheless, the stream function error remains small, 1%–2%, at all reservoir sizes (Fig. 22). The QG model is found to perform well around 4000 nodes, as shown by the comparisons of models with 4000, 5000, and 6000 node reservoirs shown in Fig. 23.

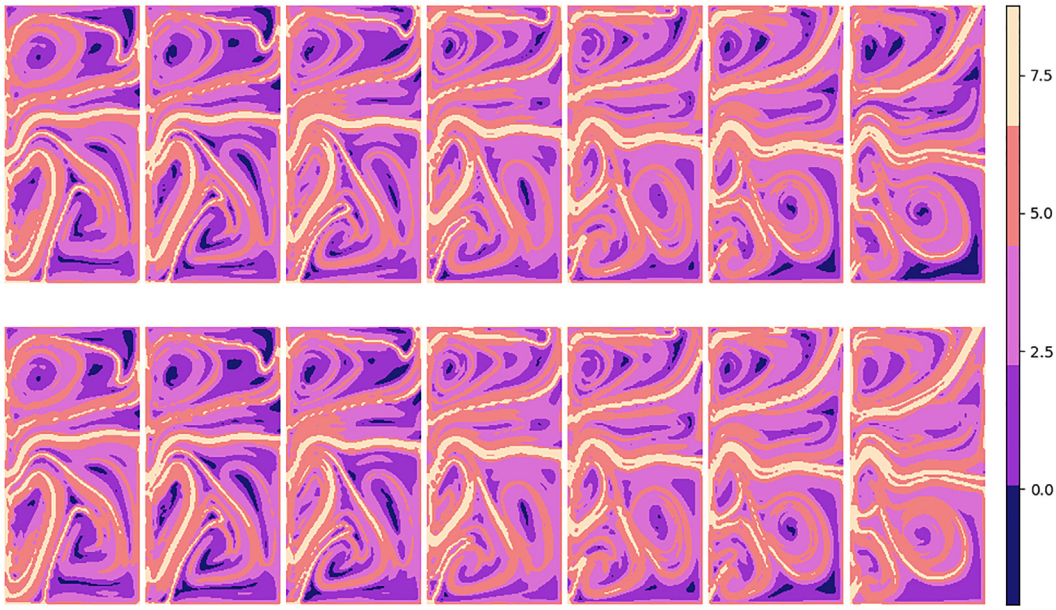

FIG. 19. FTLE fields calculated from the (top) actual and (bottom) estimated Stommel version of the QG stream functions, with time increasing from left to right.

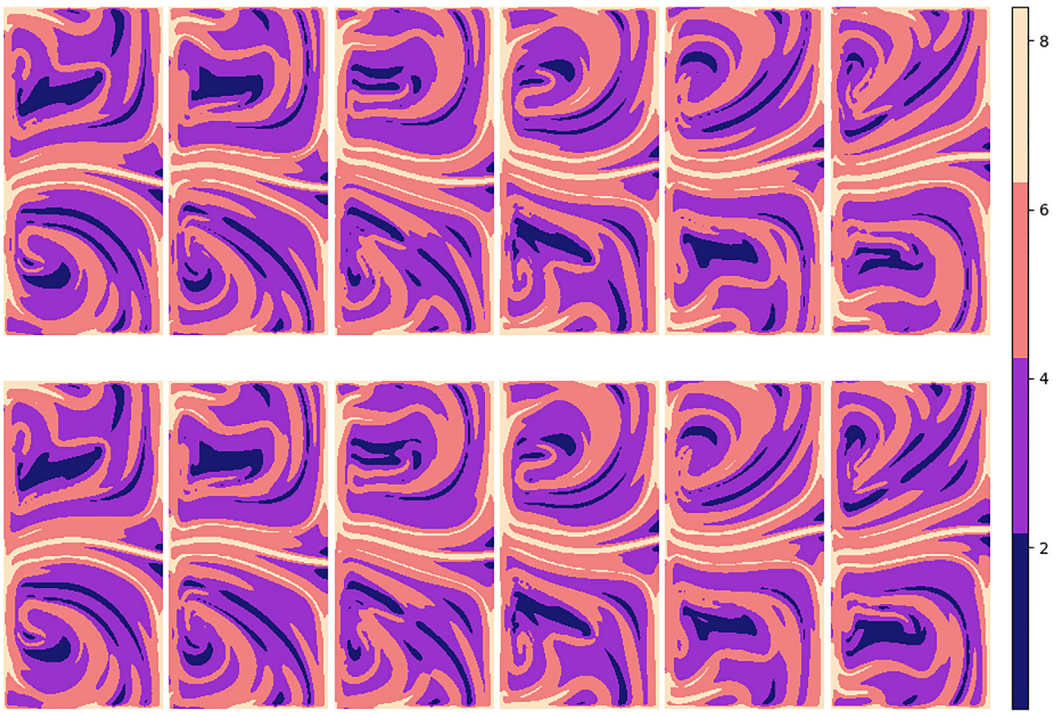

FIG. 20. FTLE fields calculated from the (top) actual and (bottom) estimated Munk version of the QG stream functions, with time increasing from left to right.

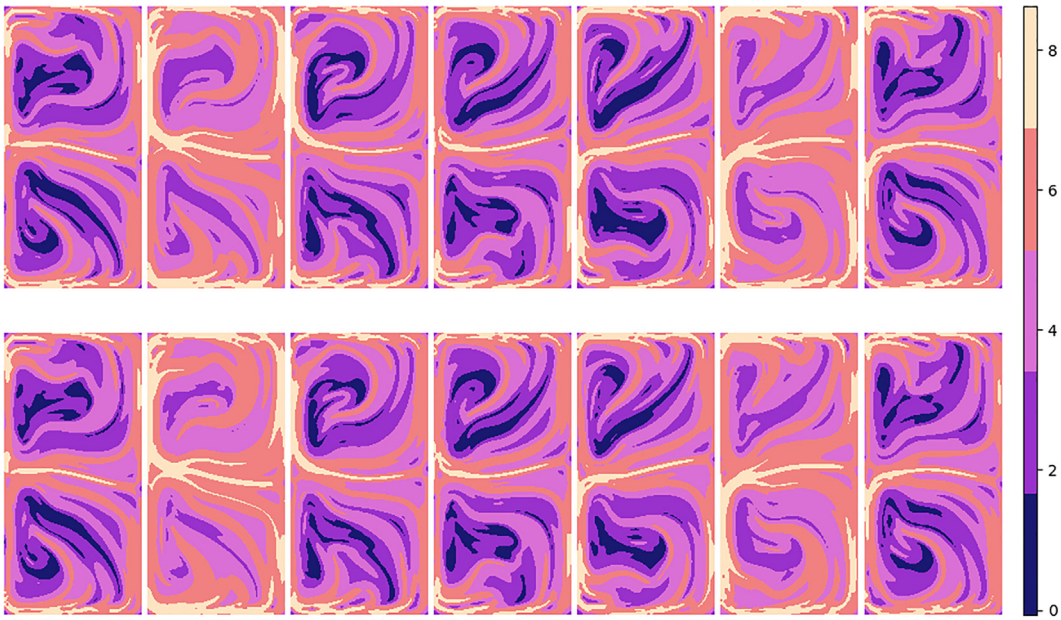

FIG. 21. FTLE fields calculated from the (top) actual and (bottom) estimated Inertial version of the QG stream functions, with time increasing from left to right.

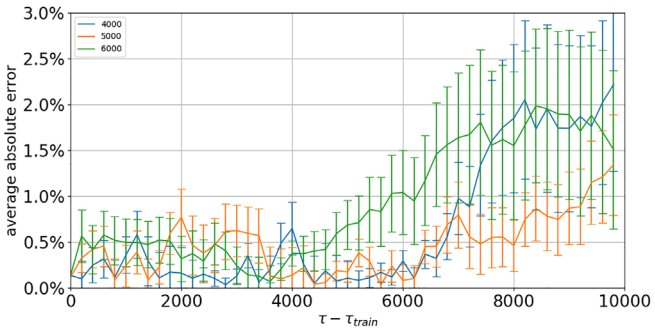

FIG. 22. Error of estimated DG stream function against actual DG stream function over time with varying reservoir sizes.

Different hyperparameter tunings are model-dependent, as expected. It is important to note that the adjacency matrix representing the node connections in the reservoir is sparse, so although there is a high number of nodes, there are not necessarily as many connections. On the other hand, the complexity of the reservoir still increases with the additional node count, which could contribute positively to modeling capability.

With regard to spectral radius, the DG model performs well with a spectral radius of approximately 2.3 or less (Fig. 24), while the QG model performs well with a value in the range $0.4 - 0.6$ (Fig. 25). Spectral radius is a vague proxy to long-term memory required by the model to learn, so this result suggests that the QG model evolves based on shorter-term history.

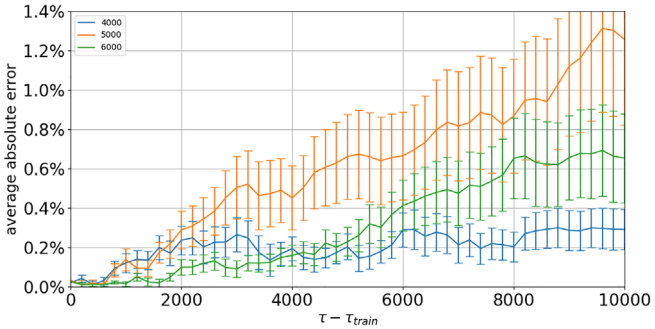

FIG. 23. Error of estimated QG stream function against actual QG stream function (Highly Nonlinear version) over time with varying reservoir sizes.

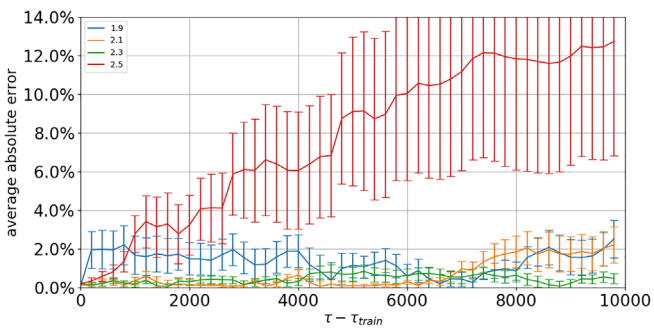

FIG. 24. Error of estimated DG stream function against actual DG stream function over time with varying spectral radii.

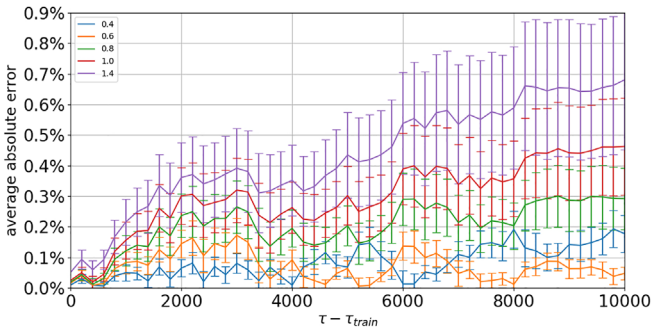

FIG. 25. Error of estimated QG stream function against actual QG stream function (Highly Nonlinear version) over time with varying spectral radii.

REFERENCES

[1]O. Regev, O. Umurhan, and P. Yecko, *Modern Fluid Dynamics for Physics and Astrophysics* (Springer Science and Business Media, 2016).

[2]E. Forgoston, L. Billings, P. Yecko, and I. B. Schwartz, "Set-based corral control in stochastic dynamical systems: Making almost invariant sets more invariant," Chaos Interdiscip. J. Nonlinear Sci. **21**, 013116 (2011).

[3]M. Michini, M. A. Hsieh, E. Forgoston, and I. B. Schwartz, "Robotic tracking of coherent structures in flows," IEEE Trans. Rob. **30**, 593–603 (2014).

[4]M. A. Hsieh, H. Hajieghrary, D. Kularatne, C. R. Heckman, E. Forgoston, I. B. Schwartz, and P. A. Yecko, "Small and adrift with self-control: Using the environment to improve autonomy," in *Robotics Research* (Springer, 2018), pp. 387–402.

[5]D. Kularatne, E. Forgoston, and M. A. Hsieh, "Using control to shape stochastic escape and switching dynamics," Chaos Interdiscip. J. Nonlinear Sci. **29**, 053128 (2019).

[6]K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," IEEE Trans. Rob. **24**, 710–724 (2008).

[7]K. Mallory, M. A. Hsieh, E. Forgoston, and I. B. Schwartz, "Distributed allocation of mobile sensing swarms in gyre flows," Nonlinear Processes Geophys. **20**, 657–668 (2013).

[8]T. Bolton and L. Zanna, "Applications of deep learning to ocean data inference and subgrid parameterization," J. Adv. Model. Earth Syst. **11**, 376–399 (2019).

[9]G. E. Manucharyan, L. Siegelman, and P. Klein, "A deep learning approach to spatiotemporal sea surface height interpolation and estimation of deep currents in geostrophic ocean turbulence," J. Adv. Model. Earth Syst. **13**, e2019MS001965 (2021).

[10]A. Sinha and R. Abernathey, "Estimating ocean surface currents with machine learning," Front. Mar. Sci. **8**, 672477 (2021).

[11]S. Jiang, F.-F. Jin, and M. Ghil, "Multiple equilibria, periodic, and aperiodic solutions in a wind-driven, double-gyre, shallow-water model," J. Phys. Oceanogr. **25**, 764–786 (1995).

[12]P. Cessi and G. R. Ierley, "Symmetry-breaking multiple equilibria in quasigeostrophic, wind-driven flows," J. Phys. Oceanogr. **25**, 1196–1205 (1995).

[13]S. Meacham, "Low-frequency variability in the wind-driven circulation," J. Phys. Oceanogr. **30**, 269–293 (2000).

[14]H. A. Dijkstra and M. Ghil, "Low-frequency variability of the large-scale ocean circulation: A dynamical systems approach," Rev. Geophys. **43**, RG3002, https://doi.org/10.1029/2002RG000122 (2005).

[15]M. Ghil, "The wind-driven ocean circulation: Applying dynamical systems theory to a climate problem," Discrete Contin. Dyn. Syst. **37**, 189–228 (2017).

[16]S. C. Shadden, "Lagrangian coherent structures," *Transport and Mixing in Laminar Flows: From Microfluidics to Oceanic Currents*, edited by R. Grigoriev (Wiley, 2011), pp. 59–89.

[17]A. Hadjighasem, M. Farazmand, D. Blazevski, G. Froyland, and G. Haller, "A critical comparison of Lagrangian methods for coherent structure detection," Chaos Interdiscip. J. Nonlinear Sci. **27**, 053104 (2017).

[18]A. Chatterjee, "An introduction to the proper orthogonal decomposition," Curr. Sci. **78**, 808–817 (2000).

[19]F. Fang, C. Pain, I. Navon, M. Piggott, G. Gorman, P. Allison, and A. Goddard, "Reduced-order modelling of an adaptive mesh ocean model," Int. J. Numer. Methods Fluids **59**, 827–851 (2009).

[20]P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, "Applications of the dynamic mode decomposition," Theor. Comput. Fluid Dyn. **25**, 249–259 (2011).

[21]C. W. Rowley and S. T. Dawson, "Model reduction for flow analysis and control," Annu. Rev. Fluid Mech. **49**, 387–417 (2017).

[22]S. C. Shadden, F. Lekien, and J. E. Marsden, "Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows," Physica D **212**, 271–304 (2005).

[23]H. Aref, "Stirring by chaotic advection," J. Fluid Mech. **143**, 1–21 (1984).

[24]R. Lguensat, J. L. Sommer, S. Metref, E. Cosme, and R. Fablet, "Learning generalized quasi-geostrophic models using deep neural numerical models," in Proceedings of the NeurIPS 2019: 33rd Conference on Neural Information Processing Systems, Vancouver, Canada (2019).

[25]S. M. Rahman, S. Pawar, O. San, A. Rasheed, and T. Iliescu, "Nonintrusive reduced order modeling framework for quasigeostrophic turbulence," Phys. Rev. E **100**, 053306 (2019).

[26]J. Vidal and N. Schaeffer, "Quasi-geostrophic modes in the earth's fluid core with an outer stably stratified layer," Geophys. J. Int. **202**, 2182–2193 (2015).

[27]M. Qraitem, D. Kularatne, E. Forgoston, and M. A. Hsieh, "Bridging the gap: Machine learning to resolve improperly modeled dynamics," Physica D **414**, 132736 (2020).

[28]T. Z. Jiahao, M. A. Hsieh, and E. Forgoston, "Knowledge-based learning of nonlinear dynamics and chaos," Chaos Interdiscip. J. Nonlinear Sci. **31**, 111101 (2021).

[29]M. Sonnewald, R. Lguensat, D. C. Jones, P. Dueben, J. Brajard, and V. Balaji, "Bridging observations, theory and numerical simulation of the ocean using machine learning," Environ. Res. Lett. **16**, 073008 (2021).

[30]M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Comput. Sci. Rev. **3**, 127–149 (2009).

[31]H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Proceedings of the Advances in Neural Information Processing Systems* (M.I.T. Press, Cambridge, MA, 2002), Vol. 15, pp. 609–616.

[32]W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," Neural Comput. **14**, 2531–2560 (2002).

[33]H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**, 78–80 (2004).

[34]E. Bollt, "On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD," Chaos Interdiscip. J. Nonlinear Sci. **31**, 013108 (2021).

[35]J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," Chaos Interdiscip. J. Nonlinear Sci. **27**, 121102 (2017).

[36]J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," Phys. Rev. Lett. **120**, 024102 (2018).

[37]J. Pedlosky, *Geophysical Fluid Dynamics* (Springer, 1987).

[38]J. Pedlosky, *Ocean Circulation Theory* (Springer Science and Business Media, 1996).

[39]R. R. Blandford, "Boundary conditions in homogeneous ocean models," in *Deep Sea Research and Oceanographic Abstracts* (Elsevier, 1971), Vol. 18, No. 7, pp. 739–751.

[40]G. R. Ierley, "On the onset of inertial recirculation in barotropic general circulation models," J. Phys. Oceanogr. **17**, 2366–2374 (1987).

[41]P. Cessi, "Laminar separation of colliding western boundary currents," J. Mar. Res. **49**, 697–717 (1991).

[42]W. K. Dewar, "A nonlinear, time-dependent thermocline theory," J. Mar. Res. **47**, 1–31 (1989).

[43]J. Pedlosky, "A history of thermocline theory," in *Physical Oceanography* (Springer, 2006), pp. 139–152.

[44]H. A. Dijkstra and C. A. Katsman, "Temporal variability of the wind-driven quasi-geostrophic double gyre ocean circulation: Basic bifurcation diagrams," Geophys. Astrophys. Fluid Dyn. **85**, 195–232 (1997).

[45]E. Simonnet, M. Ghil, and H. Dijkstra, "Homoclinic bifurcations in the quasi-geostrophic double-gyre circulation," J. Mar. Res. **63**, 931–956 (2005).
[46]J. Dengo, "The problem of gulf stream separation: A barotropic approach," J. Phys. Oceanogr. **23**, 2182–2200 (1993).
[47]D. B. Haidvogel, J. C. McWilliams, and P. R. Gent, "Boundary current separation in a quasigeostrophic, eddy-resolving ocean circulation model," J. Phys. Oceanogr. **22**, 882–902 (1992).
[48]M. A. Spall, "An idealized modeling study of the mid-latitude variability of the wind-driven meridional overturning circulation," J. Phys. Oceanogr. **51**, 2425–2441 (2021).
[49]A. Farchi, P. Laloyaux, M. Bonavita, and M. Bocquet, "Using machine learning to correct model error in data assimilation and forecast applications," Q. J. R. Meteorol. Soc. **147**, 3067–3084 (2021).
[50]A. Charalampopoulos and T. Sapsis, "Uncertainty quantification of turbulent systems via physically consistent and data-informed reduced-order models," Phys. Fluids **34**, 075120 (2022).
[51]Note that there is relatively simple direct relation between stream function and vorticity, the central quantity of QG models.
[52]F. M. Bianchi, L. Livi, and C. Alippi, "Investigating echo-state networks dynamics by means of recurrence analysis," IEEE Trans. Neural Networks Learn. Syst. **29**, 427–439 (2018).
[53]L. A. Smith and E. Spiegel, "Pattern formation by particles settling in viscous flows," in *Macroscopic Modelling of Turbulent Flows* (Springer, 1985), pp. 306–318.
[54]K.-I. Chang, M. Ghil, K. Ide, and C.-C. A. Lai, "Transition to aperiodic variability in a wind-driven double-gyre circulation model," J. Phys. Oceanogr. **31**, 1260–1286 (2001).
[55]B. T. Nadiga and B. P. Luce, "Global bifurcation of Shilnikov type in a double-gyre ocean model," J. Phys. Oceanogr. **31**, 2669–2690 (2001).
[56]G. Dahlquist and A. Björck, *Numerical Methods* (Prentice Hall, 1974).
[57]G. Haller, "Finding finite-time invariant manifolds in two-dimensional velocity fields," Chaos **10**, 99–108 (2000).
[58]G. Haller, "Distinguished material surfaces and coherent structures in three-dimensional fluid flows," Physica D **149**, 248–277 (2001).
[59]G. Haller, "Lagrangian coherent structures from approximate velocity data," Phys. Fluids **14**, 1851–1861 (2002).
[60]G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," Annu. Rev. Fluid Mech. **25**, 539–575 (1993).
[61]K. Taira, S. L. Brunton, S. T. 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**, 4013–4041 (2017).
[62]K. Taira, "Proper orthogonal decomposition in fluid flow analysis. I. Introduction," J. Jpn. Soc. Fluid Mech. (Nagare) **30**, 115–123 (2011).
[63]A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network," Nonlinear Processes Geophys. **27**, 373–389 (2020).
[64]B. T. Nadiga, "Reservoir computing as a tool for climate predictability studies," J. Adv. Model. Earth Syst. **13**, e2020MS002290 (2021).
[65]Y. Liu, R. H. Weisberg, C. Hu, and L. Zheng, "Tracking the deepwater horizon oil spill: A modeling perspective," EOS Trans. Am. Geophys. Union **92**, 45–46 (2011).
[66]V. J. García-Garrido, A. M. Mancho, S. Wiggins, and C. Mendoza, "A dynamical systems approach to the surface search for debris associated with the disappearance of flight MH370," Nonlinear Processes Geophys. **22**, 701–712 (2015).