Data-driven low-dimensional dynamic model of Kolmogorov flow

Carlos E. Pérez De Jesús and Michael D. Graham*
Department of Chemical and Biological Engineering, University of Wisconsin–Madison, Madison, Wisconsin 53706, USA

(Received 1 November 2022; accepted 28 March 2023; published 10 April 2023)

Reduced order models (ROMs) that capture flow dynamics are of interest for decreasing computational costs for simulation as well as for model-based control approaches. This work presents a data-driven framework for minimal-dimensional models that effectively capture the dynamics and properties of the flow. We apply this to Kolmogorov flow in a regime consisting of chaotic and intermittent behavior, which is common in many flow processes and is challenging to model. The trajectory of the flow travels near relative periodic orbits (RPOs), interspersed with sporadic bursting events corresponding to excursions between the regions containing the RPOs. The first step in development of the models is use of an undercomplete autoencoder to map from the full state data down to a latent space of dramatically lower dimension. Then models of the discrete-time evolution of the dynamics in the latent space are developed. By analyzing the model performance as a function of latent space dimension, we can estimate the minimum number of dimensions required to capture the system dynamics. To further reduce the dimension of the dynamical model, we factor out a phase variable in the direction of translational invariance for the flow, leading to separate evolution equations for the pattern and phase dynamics. At a model dimension of five for the pattern dynamics, as opposed to the full state dimension of 1024 (i.e., a 32×32 grid), accurate predictions are found for individual trajectories out to about two Lyapunov times, as well as for long-time statistics. Further small improvements in the results occur as dimension is increased to nine, beyond which the statistics of the model and the true system are in very good agreement. The nearly heteroclinic connections between the different RPOs, including the quiescent and bursting timescales, are well captured. We also capture key features of the phase dynamics. Finally, we use the low-dimensional representation to predict future bursting events, finding good success.

DOI: 10.1103/PhysRevFluids.8.044402

I. INTRODUCTION

The development of reduced order dynamical models for complex flows is an issue of long-standing interest, with applications in improved understanding, as well as control, of flow phenomena. The classical approach for dimension reduction of these systems consists of extracting dominant modes from data via principal component analysis (PCA), also known as proper orthogonal decomposition (POD) and Karhunen-Loéve decomposition [1]. PCA determines a set of basis vectors ordered by their contribution to the total variance (fluctuating kinetic energy) of the flow. Given N_s data vectors ("snapshots") $x_i \in \mathbb{R}^N$, one can obtain these basis vectors by performing singular value decomposition (SVD) on the data matrix $X = [x_1, x_2, \dots] \in \mathbb{R}^{N \times N_s}$ such that $X = U \Sigma V^T$. Projecting the data onto the first d_h basis vectors (columns of U) then gives a low-dimensional representation—a projection onto a linear subspace of the full state space. To find a

*mdgraham@wisc.edu

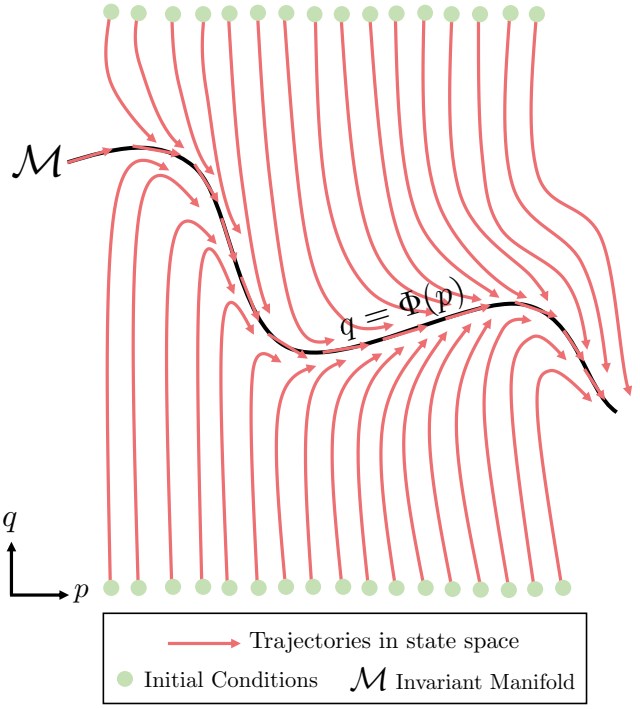

FIG. 1. Schematic of state space with initial conditions collapsing onto an invariant manifold where the long-time dynamics occur.

reduced order model (ROM), a Galerkin approximation of the Navier-Stokes equations (NSE) using this basis can be implemented; these have demonstrated some success in capturing the dynamics of coherent structures [2,3]. Previous research has also used POD as well as a filtered version thereof [4], which are linear reduction techniques, to reduce dimensions and learn a time evolution map from data with the use of neural networks (NNs) [5].

Although PCA provides the best linear representation of a data set in d_h dimensions, in general the long-time dynamics of a general nonlinear dynamical systems are not expected to lie on a linear subspace of the state space. For a primer and more details on data-driven dimension reduction methods for dynamical systems, refer to Linot and Graham [6]. For dissipative systems, such as the NSE, it is expected that the long-time dynamics will lie on an invariant manifold $\mathcal{M}$, which can be represented *locally* with Cartesian coordinates, but may have a complex global topology [7]. In fluid mechanics, this manifold is often called an *inertial manifold* [8–10]. Figure 1 schematically illustrates a simple example of this idea. Consider a dynamical system $\dot{x} = F(x)$ for state variable $x \in \mathbb{R}^N$. As time proceeds, general initial conditions in this space evolve toward an invariant manifold $\mathcal{M}$ of dimension $d_{\mathcal{M}}$, which in this example can be described by the equation $q = \Phi(p)$, where $x = [p, q]^T$, $p \in \mathbb{R}^{d_{\mathcal{M}}}$, $q \in \mathbb{R}^{N-d_{\mathcal{M}}}$. Furthermore, if we write the dynamics in terms of p and q as $\dot{p} = f(p, q)$, $\dot{q} = g(p, q)$, then trajectories on $\mathcal{M}$ evolve according to $\dot{p} = f(p, \Phi(p))$: i.e., the long-time dynamics are given by a set of ordinary differential equations in $d_{\mathcal{M}}$ dimensions, rather than the N dimensions of the original system. More generally, since $\mathcal{M}$ is invariant under the dynamics, the vector field on $\mathcal{M}$ is always tangent to $\mathcal{M}$, and the dynamics on $\mathcal{M}$ are determined by this vector field. In the present work, we do not require that the manifold be represented in this simple form, but rather a more general form $G(x) = 0$. In this example, $G(x) = q - \Phi(p)$.

In general one can think of breaking up $\mathcal{M}$ into overlapping regions that cover the domain, to find a local representation. These are called charts and they are equipped with a coordinate domain and

a coordinate map [11]. The strong Whitney's embedding theorem states that any smooth manifold of dimension $d_{\mathcal{M}}$ can be embedded into a Euclidean space of so-called *embedding* dimension $2d_{\mathcal{M}}$ [11,12]. This means that in the worst case we can expect in principle to be able to find a $2d_{\mathcal{M}}$-dimensional Euclidean space in which the dynamics lie. To find a $d_{\mathcal{M}}$-dimensional Euclidean space, one would in general need to develop overlapping local representations and evolution equations—this avenue is not pursued in the present work but has been done elsewhere [13]. In this work, we aim to find a high-fidelity low-dimensional dynamical model using data from simulations of two-dimensional Kolmogorov flow. In this work, the governing Navier-Stokes equations will only be used to generate the data—the models will only use these data, not the equations that generated them. NNs will be used to map between the full state space and the manifold, as well as for the dynamical system model on the manifold.

A number of previous studies have focused on finding data-driven models for fluid flow problems with the use of NNs. Srinivasan *et al.* [14] developed NN models to attempt to predict the time evolution of the Moehlis-Faisst-Eckhardt (MFE) model [15], which is a nine-dimensional model for turbulent shear flows. They used two approaches to finding discrete-time dynamical systems. The first is to simply use a neural network as a discrete-time map, yielding a Markovian representation of the time evolution. The second is to use a long short-term memory (LSTM) network, which yields a non-Markovian evolution equation. Despite the fact that the dynamics are in fact Markovian, the LSTM approach worked better, yielding reasonable agreement with the Reynolds stress profiles. Page *et al.* used deep convolutional autoencoders (CAEs) to learn low-dimensional representations for two-dimensional (in physical space) Kolmogorov flow, showing that these networks retain a wide spectrum of lengthscales and capture meaningful patterns related to the embedded invariant solutions [16]. They considered the case in which bursting dynamics is obtained at a Reynolds number of $\mathrm{Re} = 40$ and $n = 4$ wavelengths in the periodic domain. Nakamura *et al.* used CAEs for dimension reduction combined with LSTMs and applied it to minimal turbulent channel flow for $\mathrm{Re}_\tau = 110$, where they were shown to capture velocity and Reynolds stress statistics [17]. They studied various degrees of dimension reduction, showing good performance in terms of capturing the statistics; however, for drastic dimension reduction they showed how only large vortical structures were captured. Hence, the selection of the minimal dimension to accurately represent the state becomes a challenging task. Reservoir networks have also shown great potential in learning nonlinear models for time evolution. For example, Doan *et al.* trained what they call an autoencoded reservoir-computing (AE-RC) framework where the latent space is fed into an echo state network (ESN) to model evolution in discrete time [18]. By considering the two-dimensional Kolmogorov flow for $\mathrm{Re} = 30$ and $n = 4$, good performance was obtained when comparing the kinetic energy and dissipation evolution in time. They also showed how the model captures the velocity statistics. However, the nature of the reservoir in the ESN stores past history, making the model non-Markovian.

Although previous research has found data-driven ROMs for fluid flow problems, the focus on these has not been to find the minimal dimension required to capture the data manifold and dynamics. Linot and Graham have addressed this issue for the Kuramoto-Sivashinsky equation (KSE) [6,19]. They showed that the mean-squared error (MSE) of the reconstruction of the snapshots using an AE for the domain size of $L = 22$ exhibited an orders-of-magnitude drop when the dimension of the inertial manifold is reached. Furthermore, modeling the dynamics with a dense NN at this dimension either with a discrete-time map [19] or a system of ordinary differential equations (ODE) [6] yields excellent trajectory predictions and long-time statistics. Increasing domain size to $L = 44$ and 66, which makes the system more chaotic, affects the drops of MSE significantly. However a drop is still seen, and when obtaining the dynamics and calculating long-time statistics, good agreement with the true data is obtained. This work, denoted "Data-driven manifold dynamics" (DManD), has been extended to incorporate reinforcement learning control for reduction of dissipation in the KSE, yielding a very effective control policy [20].

We aim to extend this approach to the NSE, specifically to the two-dimensional Kolmogorov flow, where an external forcing drives the dynamics. As Re increases, the trivial state becomes

unstable, giving rise to periodic orbits (POs), relative periodic orbits (RPOs), and eventually chaos. Relative periodic orbits correspond to periodic orbits in a moving reference frame, such that in a fixed frame, the pattern at time $t + T$ is a phase-shifted replica of the pattern at time t. The nature of the weakly turbulent dynamics at a Reynolds number of $\mathrm{Re} = 14.4$, and connections with RPO solutions, are the focus of this study. Due to the symmetries of the system, the chaotic dynamics travels between unstable RPOs [21] through bursting events [22] that shadow heteroclinic orbits connecting the RPOs. A past study [23] shows that low-dimensional representations can be found with PCA for two-dimensional Kolmogorov flow, where in the case of weakly turbulent data, the first two PCA basis functions in the streamfunction formulation capture most of the energetic content when filtering out the bursting events before the analysis, and including a third basis function captures the bursting information. This point hints at the low-dimensional nature of this system, where a low number of PCA basis functions can energetically represent the data. However, even though the energy can be contained in a low number of basis functions, this does not imply that these will properly capture the dynamics [24]. In [23], development of a model of time-evolution was not considered.

Returning to the aims of the present work, our focus is twofold. We aim to learn a minimal-dimensional high-fidelity data-driven model for the long-time dynamics of two-dimensional Kolmogorov flow with the use of an autoencoder (AE), and a discrete-time map, in the form of a dense NN, of the dynamics on the invariant manifold. In this map, the future time prediction only depends on the present state (on the manifold), in keeping with the Markovian nature of the dynamics on the manifold. This approach contrasts with models that use an RNN such as an LSTM, which carry a memory of past states, thus they are not Markovian. It is important to note, however, that the dimension of the invariant manifold is not known *a priori*, and if we map the data onto a manifold of too low a dimension, then the dynamics on that manifold will not be Markovian. Accordingly, in this work we will carefully assess the performance of our Markovian models as a function of manifold dimension. For our results, the model predictions will be evaluated as a function of dimension, considering short-time trajectories, long-time statistics, quiescent and bursting time distributions, and predictions of bursting events. This paper is structured as follows: In Sec. II we present the governing equations together with the symmetries of the system. We also present the dynamics at the two values of Re considered and the connections of the RPOs with the chaotic regime. In Sec. III we show the methodology for data-driven dimension reduction and dynamic modeling, which includes the AE architecture and the time map NN. Section IV shows the results, and concluding remarks are given in Sec. V.

II. KOLMOGOROV FLOW FORMULATION AND DYNAMICS

The two-dimensional Navier-Stokes equations (NSE) with Kolmogorov forcing are

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla} p = \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{u} + \sin(ny)\hat{\boldsymbol{x}} \tag{1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{u} = [u, v]$ is the velocity vector, p is the pressure, n is the wave number of the forcing, and $\hat{\boldsymbol{x}}$ is the unit vector in the x direction. Here $\mathrm{Re} = \frac{\sqrt{\chi}}{\nu}(\frac{L_y}{2\pi})^{3/2}$, where χ is the dimensional forcing amplitude, ν is the kinematic viscosity, and L_y is the size of the domain in the y direction. We consider the periodic domain $[0, 2\pi/\alpha] \times [0, 2\pi]$ with $\alpha = 1$. Vorticity is defined as $\omega = \boldsymbol{\nabla}\times\boldsymbol{u}$. The equations are invariant under several symmetry operations [25], namely a shift (in y)-reflect (in x), a rotation through π, and a continuous translation in x:

$$\mathscr{S} : [u, v, \omega](x, y) \to [-u, v, -\omega]\left(-x, y + \frac{\pi}{n}\right), \tag{3}$$

$$\mathscr{R} : [u, v, \omega](x, y) \to [-u, -v, \omega](-x, -y), \tag{4}$$

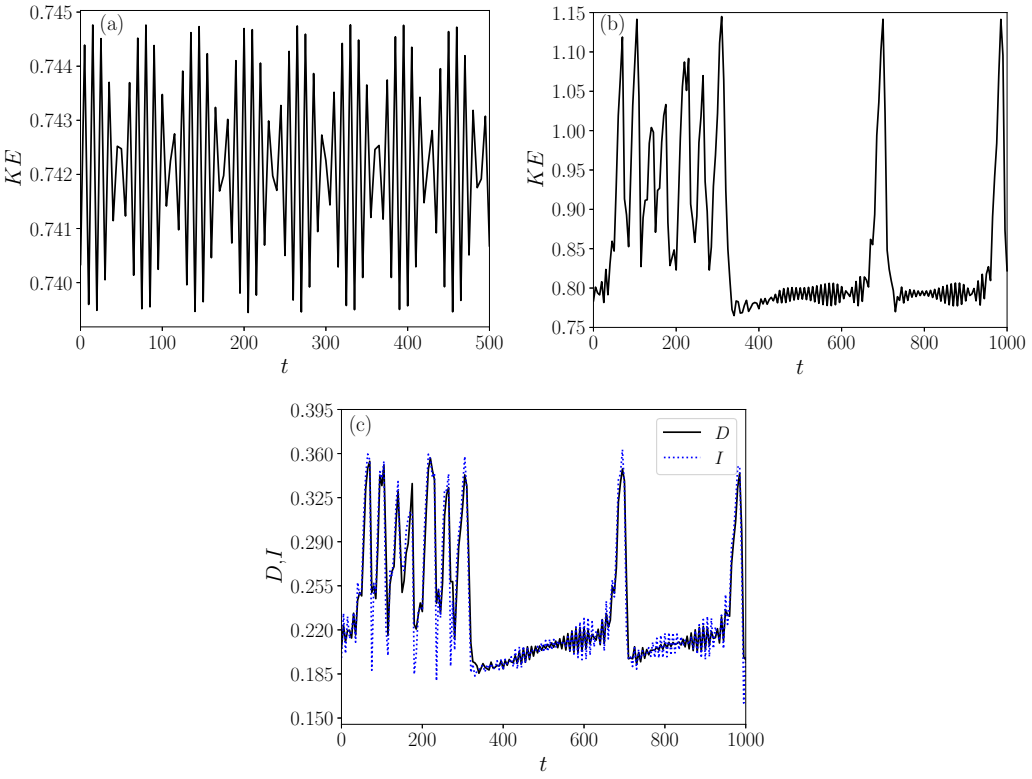

FIG. 2. (a) Time evolution of KE at Re $= 13.5$. (b) Time evolution of KE at Re $= 14.4$. (c) Time evolution of D and I at Re $= 14.4$.

$$\mathscr{T}_l : [u, v, \omega](x, y) \to [u, v, \omega](x + l, y) \quad \text{for } 0 \leqslant l < \frac{2\pi}{\alpha}. \tag{5}$$

The total kinetic energy for this system (KE), dissipation rate (D), and power input (I) are

$$KE = \frac{1}{2}\langle \boldsymbol{u}^2 \rangle_V, \quad D = \frac{1}{\mathrm{Re}}\langle |\nabla \boldsymbol{u}|^2 \rangle_V, \quad I = \langle u \sin(ny) \rangle_V, \tag{6}$$

where the subscript V corresponds to the average taken over the domain. For the case of $n = 1$, the trivial solution is linearly stable at all Re [26]. It is not until $n = 2$ that the laminar state becomes unstable, with a critical value of $\mathrm{Re}_c = n^{3/2}2^{1/4}$ [27–29].

The NSEs are evolved numerically in time in the vorticity representation on a $[d_x \times d_y] = [32 \times 32]$ grid following the pseudospectral scheme given by Chandler and Kerswell [25], which is based on the code by Bartello and Warn [30]. We show here time-series results for the two dynamical regimes considered in this work, an RPO regime at Re $= 13.5$ and a chaotic regime at Re $= 14.4$. Figure 2(a) shows the KE evolution for an RPO obtained at Re $= 13.5$. Due to the discrete symmetries of the system, there are several RPOs [22], as we discuss further below. Figure 2(b) shows the KE evolution for a trajectory at Re $= 14.4$. The dynamics are characterized by quiescent intervals where the trajectories are close to RPOs (which are now unstable), punctuated by heteroclinic-like excursions between the RPOs, which are indicated by the intermittent increases of the KE. The RPOs are all related by the symmetries $\mathscr{S}$ and $\mathscr{R}$ [22,31,32]. This behavior can also be seen in Fig. 2(c), where the black curve corresponds to the time evolution of D and the blue curve to the time evolution of I. Figure 3 shows a state-space projection of a trajectory onto the plane $\mathrm{Re}[a_{0,1}(t)] - \mathrm{Im}[a_{0,1}(t)]$, where $a(k_x, k_y, t) = a_{k_x,k_y}(t) = \mathcal{F}\{\omega(x, y, t)\}$ is the discrete Fourier

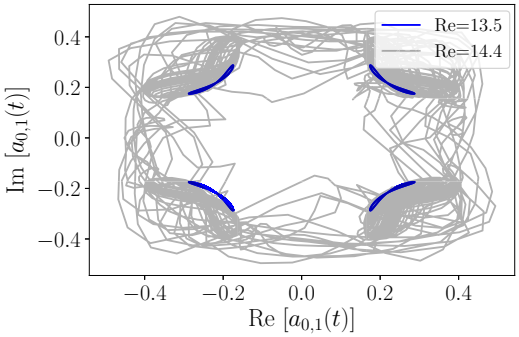

FIG. 3. Evolution of the real and imaginary components corresponding to the $a_{0,1}(t)$ Fourier mode for Re = 13.5 and 14.4.

transform in x and y. The gray curve corresponds to Re = 14.4 and the different blue curves show four different RPOs related by the shift-reflect symmetry $\mathscr{S}$ at Re = 13.5.

III. DATA-DRIVEN DIMENSION REDUCTION AND DYNAMIC MODELING

A. Dimension reduction with autoencoders

To learn a minimal-dimensional model for the two-dimensional Kolmogorov flow, we first have to find a low-dimensional nonlinear mapping from the full state to the reduced representation. For this purpose, we consider a common machine-learning architecture known as an undercomplete autoencoder (AE), whose purpose is to learn a reduced representation of the state such that the reconstruction error with respect to the true data is minimized. The AE consists of an encoder, $\mathcal{E}(\cdot)$, that maps from the full space $\mathbb{R}^N$ to the lower-dimensional latent space $h(t) \in \mathbb{R}^{d_h}$ (i.e., coordinates on the manifold $\mathcal{M}$), and a decoder, $\mathcal{D}(\cdot)$, that maps back to the full space. Flattened versions of $\omega(x, y, t)$ are used, which we refer to from this point on as $\omega(t)$, so $N = 32 \times 32 = 1024$. We shall see that the latent space dimension d_h will be much smaller than the dimension N of the full spatially resolved state. The encoder $\mathcal{E}(\omega(t))$ is a coordinate mapping from $\mathbb{R}^N$ to $\mathcal{M}$, and the decoder $\mathcal{D}(h(t))$ is the mapping back from $\mathcal{M}$ to $\mathbb{R}^N$.

We train the AEs with $\omega(t)$ obtained from the evolution of NSE for the original data as well as accounting for the discrete and continuous symmetries. By accounting for the symmetries, it is expected that the networks will perform better by not having to learn the symmetries in the latent space mapping. We account for the continuous symmetry in x, $\mathscr{T}_l$, with the method of slices [33,34]. The $k_x = 1, k_y = 0$ Fourier mode is used to find the spatial phase: $\phi_x(t) = \text{atan}\,2\{\text{Im}[a_{1,0}(t)], \text{Re}[a_{1,0}(t)]\}$. This can then be used to phase-align the vorticity snapshots such that this mode is a pure cosine: $\hat{\omega}(x, y, t) = \mathcal{F}^{-1}\{\mathcal{F}\{\omega(x, y, t)\}e^{-ik\phi_x(t)}\}$. Doing this ensures that the snapshots lie in a reference frame where no translation happens in the x direction. We will learn evolution equations for both $\hat{\omega}(t)$ and $\phi_x(t)$, which we will denote as the pattern dynamics and phase dynamics, respectively. We also consider the shift-reflect (SR) symmetry, $\mathscr{S}$, as well as the rotation through π, $\mathscr{R}$. To account for the SR symmetry, the goal is to collapse the phase-aligned snapshots to the same common state. We can define two indicator functions such that the SR subspace is specified. The first one, $I_{\text{even}} = \text{sgn}(\phi_y)$, where $\phi_y(t) = \text{atan}\,2\{\text{Im}[a_{0,1}(t)], \text{Re}[a_{0,1}(t)]\}$, is the spatial phase in y. The second indicator function is $I_{\text{odd}} = \text{sgn}\,(\text{Re}[a_{2,0}(t)])$, the sign of the real part of the second Fourier mode in x. We can then map the vorticity snapshots in such a way that $I_{\text{even}}, I_{\text{odd}} > 0$ by applying SR operations to the state. The rotation symmetry is accounted for, on top of the SR symmetry, by minimizing the l^2-norm of the data with respect to a template snapshot. This is done by applying the discrete operation that rotates and shift-reflects the vorticity snapshots and selecting

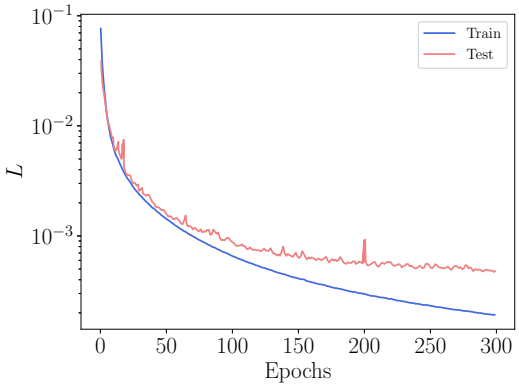

FIG. 4. Autoencoder loss vs epochs over training and test data sets corresponding to a trial from the case Re = 14.4, $d_h = 9$.

the snapshot that minimizes the norm. We note that we take a different approach for reducing the symmetries compared to previous research on symmetry-aware AEs [35].

Previous work [19] has shown that training a NN to learn the difference between the data and the projection onto the leading PCA basis vectors improved reconstruction performance compared to learning a latent space directly from the full data. To present the framework, we will use the phase-aligned and flattened vorticity $\hat{\omega}(t)$, since that is what we use for the time evolution. Below, however, we will present some results where other versions of the data are used—e.g., the data with phase-shifting. The autoencoder aspect of the analysis is identical.

We begin the process by computing the projection of the data onto the first d_h basis vectors, $P_{d_h}U^T\hat{\omega}(t)$. We then seek to learn a d_h-dimensional correction to that projection, $E(U^T\hat{\omega}(t))$—the sum of these is the latent-space representation $h(t)$. In other words, the encoding step learns the deviation from PCA,

$$E(U^T\hat{\omega}(t)) = h(t) - P_{d_h}U^T\hat{\omega}(t). \tag{7}$$

We emphasize that this step *is not* simply a projection onto a linear subspace defined by d_h PCA modes—rather, it is an approach that learns the deviation of the data from that projection. Similarly, the decoding section learns the difference

$$D(h(t)) = U^T\tilde{\hat{\omega}}(t) - \begin{bmatrix} h(t) \\ 0 \end{bmatrix}, \tag{8}$$

where $\tilde{\hat{\omega}}(t)$ corresponds to the reconstruction of $\hat{\omega}(t)$. Inserting Eq. (7) into Eq. (8) and noting that by definition $\tilde{\hat{\omega}}(t) = U[P_{d_h}U^T\hat{\omega}(t), P_{d-d_h}U^T\hat{\omega}(t)]^T$, we get that the exact solution satisfies $E(U^T\hat{\omega}(t)) + D_{d_h}((h(t)) = 0$. To satisfy this constraint, we add it to the loss function as a penalty to obtain

$$L = \|\hat{\omega}(t) - \tilde{\hat{\omega}}(t)\|^2 + \alpha_L\|E(U^T\hat{\omega}(t)) + D_{d_h}(h(t))\|^2, \tag{9}$$

where $\|\cdot\|$ is the l^2-norm, and we select $\alpha_L = 1$. We can now train the AEs by minimizing L via stochastic gradient descent. We train four AEs at each of several values of d_h to study the MSE of the reconstruction of $\hat{\omega}(t)$. All models were trained for 300 epochs with an Adam optimizer using Keras. After 300 epochs no further improvement over the test data was observed; see Fig. 4. The training data consist of long-time series from the direct simulations, with initial transients removed. We use a total of 10^5 snapshots separated by $\tau = 5$ time units for Re = 14.4, and 10^4 snapshots separated by $\tau = 5$ for Re = 13.5. We do an 80%/20% split for training and testing, respectively. Figure 5(a) shows a summary of the AE, and Table I gives information on the layer dimensions

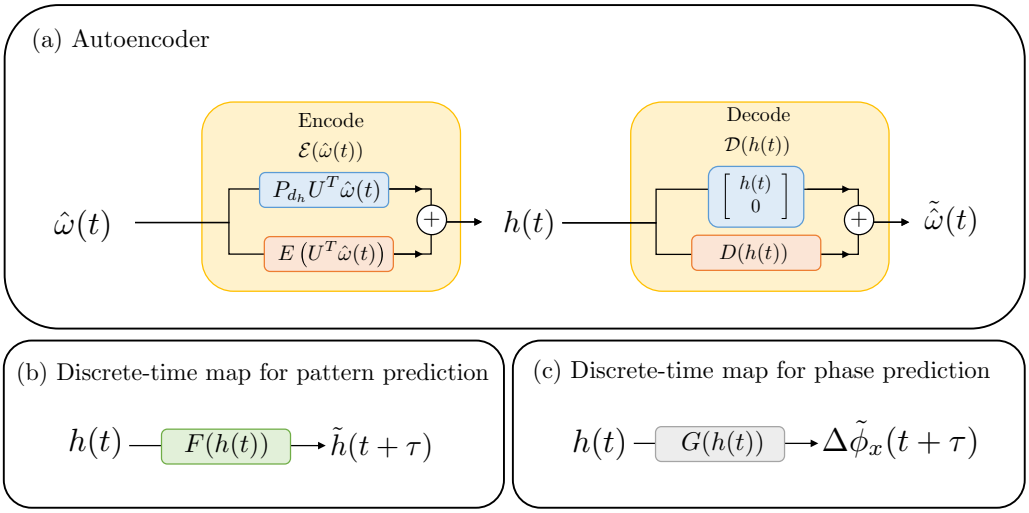

FIG. 5. Neural network frameworks for (a) autoencoder, (b) discrete-time map for pattern prediction, and (c) discrete-time for phase prediction.

and the activations used in each layer of the encoder and decoder. At each value of d_h, the model with the smallest MSE over a test data set from the phase-aligned data is then selected for the discrete-time map. We will show in Sec. IV A that factoring out the phase dramatically increases AE performance.

B. Time evolution via a dense NN

After finding $h(t)$ from the AEs, we seek a discrete-time map

$$h(t+\tau) = F(h(t)) \tag{10}$$

that evolves $h(t)$ from time t to $t+\tau$. We fix $\tau = 5$. The function F is also expressed as a dense NN. Here we train five NNs for the different d_h cases with the following loss:

$$L_t = \|\tilde{h}(t+\tau) - h(t+\tau)\|^2, \tag{11}$$

where $h(t+\tau)$ comes from true data and $\tilde{h}(t+\tau) = F(h(t))$ from the prediction, and we select the one with the best performance. For the discrete-time map, we trained for 600 epochs with the use of a learning rate scheduler. In this case, we noticed an increase in performance when dropping the learning rate hyperparameter by an order of magnitude after 300 epochs. Figure 5(b) shows a summary of the framework just described, and Table I gives information on the layer dimensions and activations used in each layer.

TABLE I. Neural network layer dimensions and activations used in each layer. Sigmoid function are denoted "S."

	Function	Shape	Activation
Encoder	E	$1024:5000:1000:d_h$	S:S:S
Decoder	D	$d_h:1000:5000:1024$	S:S:linear
Evolution	F	$d_h:500:500:d_h$	S:S:linear
Phase prediction	G	$d_h:500:500:500:1$	S:S:S:linear

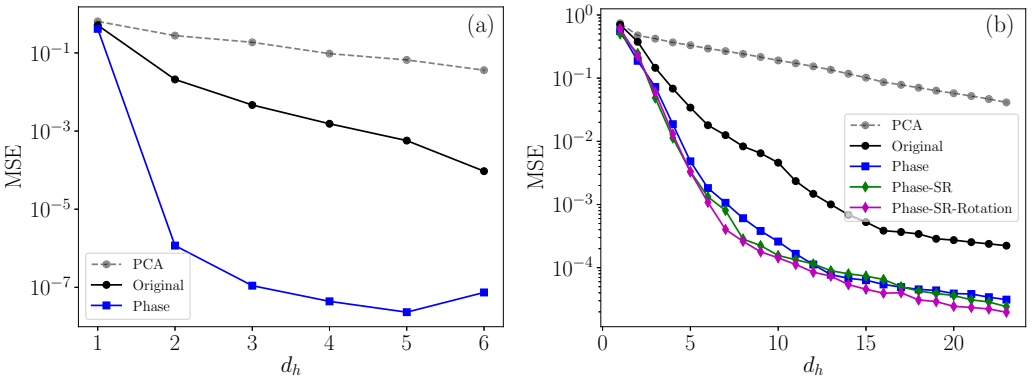

FIG. 6. MSE vs dimension d_h over the test data corresponding to (a) Re = 13.5 and (b) Re = 14.4. The PCA curve corresponds to the MSE of the reconstruction for the test data set with respect to the true data $\omega(t)$, with no symmetries factored out, using the truncated U into d_h dimensions such that $\tilde{\omega}(t) = U_{d_h} U_{d_h}^T \omega(t)$; the "Original," "Phase," "Phase-SR," and "Phase-SR-Rotation" curves correspond to the MSEs of the reconstruction for the test data set with respect to the true data using AEs. In the curve labeled "Original," no symmetries are factored out, and in the other curves the corresponding symmetries in the labels are factored out.

As discussed previously, the time evolution is done in the phase-aligned space. To complete the dynamical picture, we seek a discrete-time map for the phase evolution,

$$\Delta\tilde{\phi}_x(t+\tau) = G(h(t)), \tag{12}$$

where $\Delta\phi_x(t+\tau) = \phi_x(t+\tau) - \phi_x(t)$. Because of translation equivariance, the actual phase is only unique to within a constant. We train five NNs for the different d_h cases with the following loss:

$$L_p = \|\Delta\tilde{\phi}_x(t+\tau) - \Delta\phi_x(t+\tau)\|^2, \tag{13}$$

such that $\Delta\tilde{\phi}_x(t+\tau) = G(h(t))$. Figure 5(c) shows a summary of the framework we have described, and Table I gives information on the layer dimensions and activations used in each layer.

IV. RESULTS

We present results as follows. First we will show the AE performance for the various d_h and symmetries considered. We then report results for time evolution models, again studying performance as a function of the number of dimensions. Both the evolution of the pattern and the phase dynamics are considered. We wrap up the results by predicting bursting events based on the low-dimensional representation.

A. Dimension reduction with autoencoders

We begin by showing results for Re = 13.5. In Fig. 6(a), we see the MSE versus d_h trend, where the gray curve corresponds to the PCA reconstruction for the original data $[\tilde{\omega}(t) = U_{d_h} U_{d_h}^T \omega(t)]$, the black curve to the AE with the original data, and the blue curve to the AE with the phase factored out before training. The MSE is calculated over the test data set. Notice that, as expected, the AEs perform better than PCA. This is because of the nonlinearities that are added to the linear optimal latent space found in PCA in combination with the nonlinear decoder. The blue curve exhibits a sharp drop in the MSE at a dimension of $d_h = 2$, which is the correct embedding dimension for a limit cycle. This happens because the phase is accounted for; the dynamics of the system in the phase-aligned reference frame corresponds to a PO, and the autoencoder does not have to learn all the possible phases due to the continuous translation in x. The overall embedding dimension is

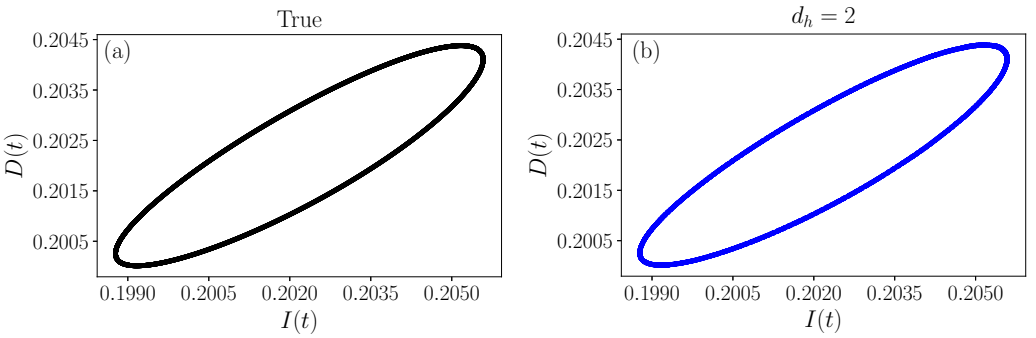

FIG. 7. Trajectory of $I(t)$ vs $D(t)$ corresponding to Re = 13.5 for (a) true and (b) predicted data corresponding to dimensions $d_h = 2$.

$d_h + 1 = 3$, where 1 corresponds to the phase. Hence we are able to estimate the dimension for this system by looking at the drop in the MSE curve.

We now consider the Re = 14.4 case, where the dynamics are chaotic, moving between the regions near the now unstable RPOs. In Fig. 6(b), we show the same curves as in Fig. 6(a) but we also include the green and magenta curves, which in addition factor out the SR and the SR-Rotation symmetries, respectively, before training the AEs. These are included due to the added complexity of Re = 14.4, where the chaotic trajectory travels in the vicinity of the RPOs related by the symmetry groups previously discussed. A monotonic decrease in MSE can be seen for the different symmetries considered in the blue, green, and magenta curves, but no sharp drop is apparent. Instead we notice that the MSE drops at different rates in different regions. For example, in the blue curve corresponding to the phase-aligned data, we see a sharp drop from $d_h = 1$–6 followed by a more gradual drop from $d_h = 6$–13. In the following sections, we couple the dimension-reduction analysis with models for prediction of time evolution for the phase-aligned data. We expect that this combination will help us determine how many dimensions are needed to correctly represent the state.

B. Time evolution as a function of dimension—Short-time predictions

The focus of this work is the chaotic dynamics at Re = 14.4. Before considering that case, for completeness we briefly present results for Re = 13.5. In Fig. 7 we see $D(t)$ versus $I(t)$ for the true and predicted dynamics at $d_h = 2$; they are indistinguishable. At $d_h = 1$, which is not shown, the model fails and the dynamics cannot be captured. The reason for this is simple—the embedding dimension for a limit cycle is two.

Now we return to the case of Re = 14.4, focusing first on short-time trajectory predictions. The Lyapunov time t_L for this system is approximately $t_L \approx 20$ [36], hence $t_L \approx 4\tau$. We take initial conditions $h(t) \in \mathbb{R}^{d_h}$ to evolve recurrently with the discrete-time map $F(\cdot)$, such that $\tilde{h}(t+\tau) = F(h(t))$, $\tilde{h}(t+2\tau) = F(\tilde{h}(t+\tau))$, $\tilde{h}(t+3\tau) = F(\tilde{h}(t+2\tau))$ and so on. After evolving in time, the data are then decoded to get $\tilde{\hat{\omega}}_h(t)$ and compared with $\hat{\omega}(t)$. We consider trajectories with ICs starting in the quiescent as well as in the bursting regions. The nature of the intermittency of the data makes it challenging to assign either bursting or quiescent labels. We consider a window of past and future snapshots and a criterion on $\|\hat{\omega}(t)\|$ to make this decision, using the algorithm described in Algorithm 1.

Doing this we ensure that snapshots that are contained in the bursting events and have a value of $\|\hat{\omega}(t)\|$ similar to quiescent snapshots are correctly classified. We use a threshold on $\|\hat{\omega}(t)\|$ to determine if a check is needed. For the classification strategy, any snapshot above a threshold of 60 is classified as bursting with a label of 1. Below 60 we enter a loop as shown in Algorithm 1 to determine if it should be classified as bursting or quiescent, where quiescent corresponds to a label

Algorithm 1. Quiescent/Bursting labeling of vorticity snapshots.

$W \leftarrow [\hat{\omega}(t_1), \hat{\omega}(t_2) \cdots]$ ▷ Matrix with N_s vorticity snapshots, $W \in \mathbb{R}^{N \times N_s}$
S ▷ Initialize label array S
$W_{l2} \leftarrow \|W\|$ ▷ Calculate l^2-norm of snapshots, $W_{l2} \in \mathbb{R}^{N_s}$
$b \leftarrow 10$ ▷ Number of past snapshots in time to consider
$f \leftarrow 10$ ▷ Number of future snapshots in time to consider
for $i = b, b+1, \ldots, N_s - f$ **do** ▷ i is snapshot I.D.
  if $W_{l2}[i] < 60$ **then**
    $d_p \leftarrow \text{abs}(W_{l2}[i-b:i] - W_{l2}[i])$ ▷ Difference between current and past snapshots
    $b_p \leftarrow \text{sum}(d_p > 5)$ ▷ Sums values that exceed a threshold of 5 (user-defined)
    $d_f \leftarrow \text{abs}(W_{l2}[i:i+f] - W_{l2}[i])$ ▷ Difference between current and future snapshots
    $b_f \leftarrow \text{sum}(d_f > 5)$ ▷ Sums values that exceed a threshold of 5 (user-defined)
    if $b_p = 0$ or $b_f = 0$ **then**
      $S[i-b] \leftarrow 0$
    else
      $S[i-b] \leftarrow 1$
    end if
  else
    $S[i-b] \leftarrow 1$
  end if
end for

of 0. This check is needed to correctly label snapshots that have comparable $\|\hat{\omega}(t)\|$ but are still in the bursting regime. Figure 8 shows a short-time trajectory where the black line corresponds to $\|\hat{\omega}(t)\|$ and the red to the 0/1 labels. Notice that, as shown in Algorithm 1, some of the data at the beginning and at the end of the time series will not be labeled, there are no past or future snapshots to compare to, and they can be removed.

After labeling the data as quiescent or bursting, we then consider the time evolution from ICs of $h(t)$ using the models of various dimensions. We will first show sample trajectories from ICs starting in the two regions, then show the ensemble-averaged prediction error as a function of time. Figure 9(a) shows the KE evolution for an IC starting in the quiescent region. The black curve corresponds to the true data and the colored curves to the different d_h models. At a dimension of $d_h = 3$ the predicted KE diverges quickly with respect to the true KE. In the case of $d_h = 5$ we see that the bursting event is correctly captured, but with a slight lag. However, $d_h = 7$ does not capture

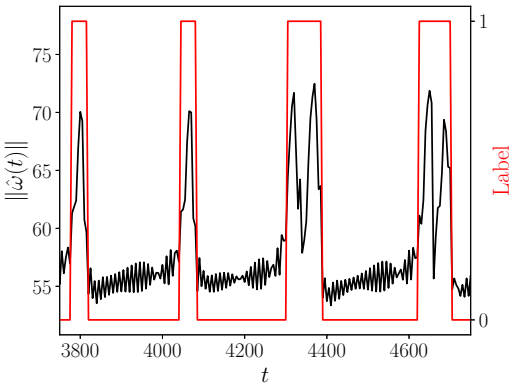

FIG. 8. Labeling of $\hat{\omega}(t)$ snapshots in a short-time series where 1 corresponds to bursting and 0 to quiescent.

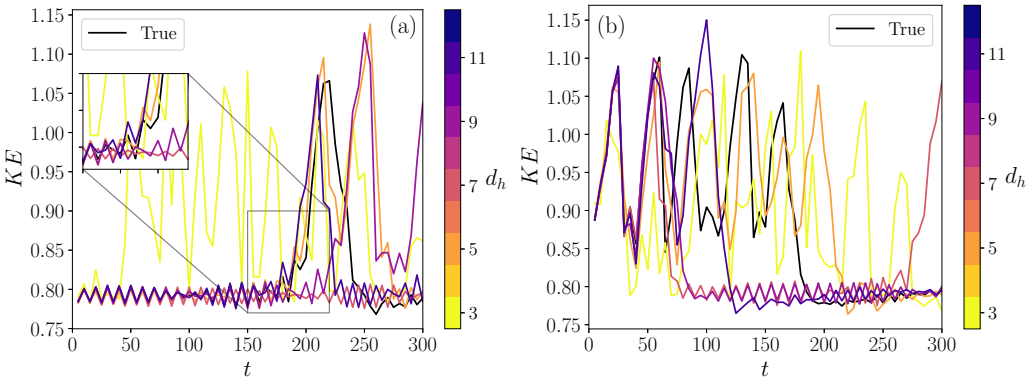

FIG. 9. Example trajectories of KE at different d_h for (a) a quiescent initial condition and (b) a bursting initial condition, for dimensions $d_h = 3$, 5, 7, 9, and 11.

the bursting in this time frame considered. For $d_h = 9$ the bursting event happens with a significant lag with respect to the true data and $d_h = 11$ captures the event similar to $d_h = 5$. Figure 9(b) shows the KE evolution for an IC starting in the bursting region. The black curve corresponds to the true data and the colored curves to the different d_h models. At a dimension of $d_h = 3$ the KE stays bursting and does not show agreement with the true KE. However, $d_h = 5$ shows better agreement and is also capable of closely predicting the end of the bursting event. In the case of $d_h = 7$, 9, and 11 these agree closely with the KE evolution before traveling to the quiescent region.

Turning from examples of individual trajectories to ensemble averages, Fig. 10(a) shows ensemble averages of the difference between the true and predicted trajectories, separately considering ICs in the bursting and quiescent regions. Solid curves correspond to quiescent ICs and dashed curves to bursting ICs. Starting from $d_h = 3$ we increase up to $d_h = 12$. We selected 10^4 ICs in total where approximately $1/3$ of the ICs correspond to bursting. As expected, predictions at $d_h = 3$ diverge quickly from the true dynamics in both quiescent and bursting IC scenarios. With increasing d_h, trajectories track better for both types of ICs. We can also notice that the two darkest curves, corresponding to $d_h = 11$, 12, fall on top of each other in the case of quiescent ICs, and the trajectories for the quiescent ICs track almost perfectly for approximately two Lyapunov times for dimensions $d_h = 5$ and higher. In Fig. 10(b) we show ensemble averages of the difference between the true and predicted dynamics based on all ICs. The same trend is obtained as discussed for Fig. 10(a) with dimensions of $d_h = 9$ and higher in similar agreement, and as expected the errors increase for all of the curves due to the divergence of the bursting ICs. We can conclude that models of dimensions $d_h = 5$ and higher are very good at capturing trajectories in the quiescent regions, which happens through the accurate prediction of the oscillatory behavior of the unstable RPO right before a bursting occurs. Prediction from bursting ICs is harder, due to the complex dynamics involved in this region. We also consider, in Fig. 10(c), the ensemble averages of the difference between the true and predicted trajectories versus d_h for all ICs at time instants $t = 0, t_L, 2t_L, 3t_L$. As expected, with increasing t the trajectories deviate from the true data. However, we notice that for all of the curves the error decreases with increasing d_h, and after $d_h = 9$ little to no improved performance is observed.

C. Time evolution as a function of dimension—Long-time predictions

In this section, we present long-time statistics for the models and true data at $\mathrm{Re} = 14.4$. From ICs on the attractor, we evolve for 2×10^5 time units, yielding to get 4×10^4 snapshots of data. This duration is sufficient to densely sample the quiescent and bursting regions. We note that long-time statistics did not change if the IC was in a bursting or quiescent region.

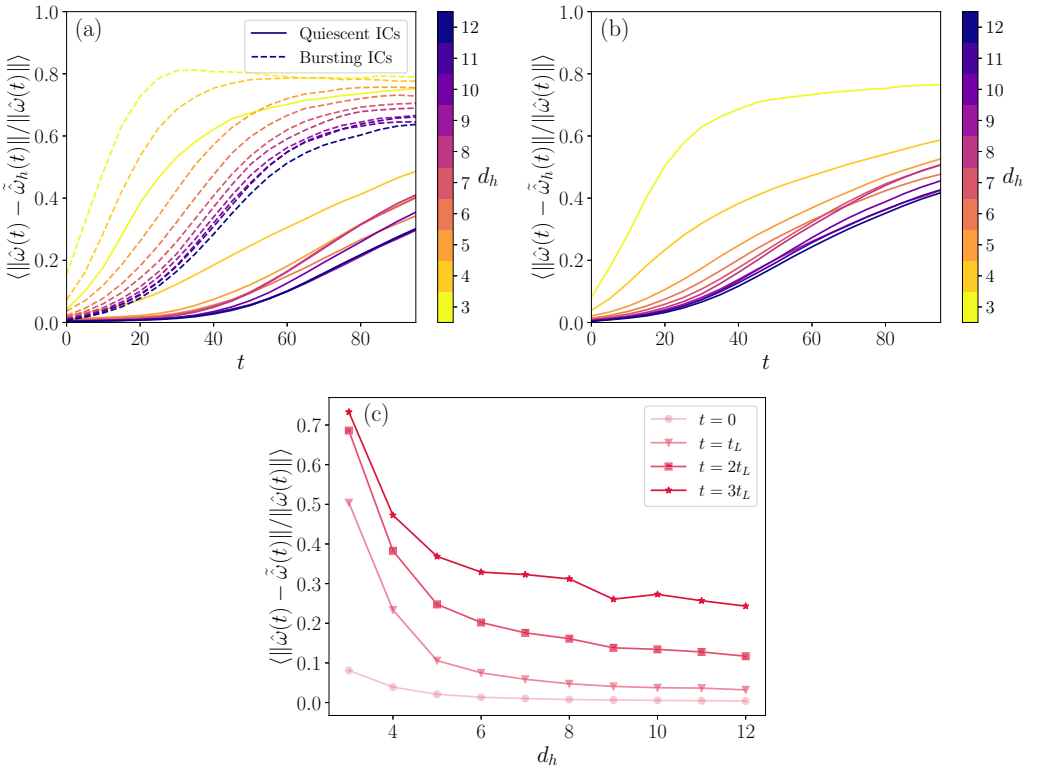

FIG. 10. Difference between true vorticity evolution and vorticity evolution obtained from the time map F from $h(t)$, where (a) corresponds to averages taken over bursting and quiescent ICs, and (b) averages over all the data. (c) Difference between true vorticity evolution and vorticity evolution obtained from the time map F from $h(t)$ with varying d_h for increasing t_L. This corresponds to averages over all the data.

Figure 11 shows the joint probability density function (PDF) of I and D for true and predicted data from models with $d_h = 3, 5, 7, 9$, and 11—note the logarithmic scale, here and below. We notice that at $d_h = 3$ the different areas corresponding to quiescent and bursting regions are populated similarly in terms of the probability intensity compared with the true PDF shown, but the shape of the predicted PDF takes a curved form that is not seen in the true PDF. When we get to $d_h = 5$, the D and I events are captured better, and similarly for increasing dimensions. We also compute the joint PDF of $\mathrm{Re}[a_{0,1}]$ and $\mathrm{Im}[a_{0,1}]$, shown in Fig. 12. From this quantity we can observe the heteroclinic-like connections between the unstable RPOs, which correspond to the four ribbonlike regions of high probability. Here we see similar trends as in the joint PDF for I-D: $d_h = 3$ shows poor qualitative reconstruction compared with higher dimensions, and once $d_h \geqslant 5$, the joint PDFs from the model prediction are virtually indistinguishable from the true PDFs. To further quantify the relationship of the PDFs from the models to the true data, we calculate the Kullback-Leibler (KL) divergence,

$$D_{\mathrm{KL}}(\tilde{P}||P) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{P}\{a,b\} \ln \frac{\tilde{P}\{a,b\}}{P\{a,b\}} da\, db, \tag{14}$$

where $\tilde{P}$ corresponds to the predicted PDF and P to the true PDF. Due to the approximation of the integral to discrete data, we ignore areas where either the true or predicted PDFs are zero. Let us first consider the case $a = I$ and $b = D$. Figure 13(a) shows D_{KL} calculated with varying d_h. The dashed gray line corresponds to D_{KL} calculated over different true data sets. This serves as a baseline

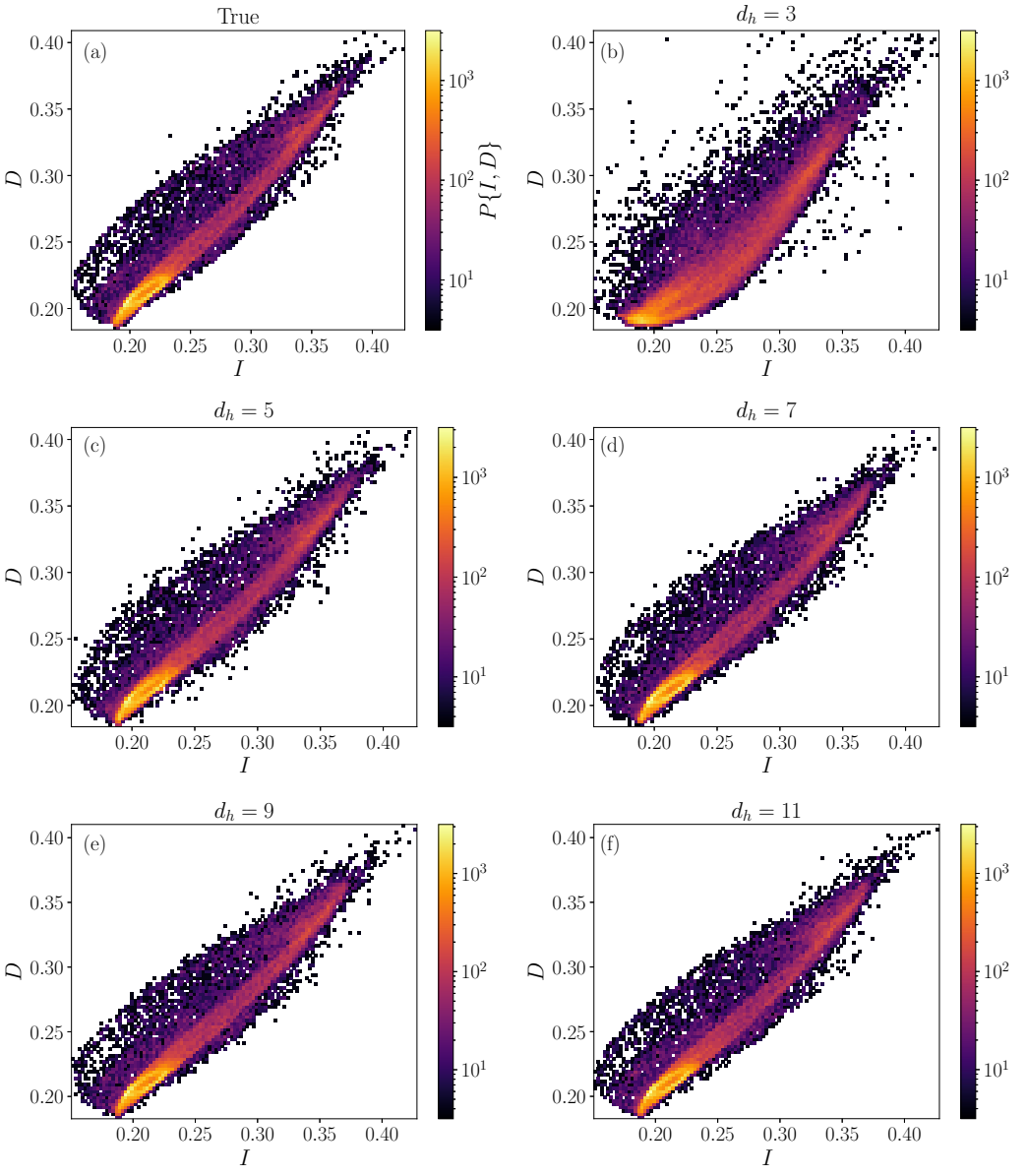

FIG. 11. Re = 14.4: Joint PDFs of I-D corresponding to Re = 14.4 for (a) true and (b)–(f) predicted data corresponding to dimensions $d_h = 3$, 5, 7, 9, and 11.

for comparison to the predicted PDFs. A significant decrease happens at $d_h = 4$ followed by small decreases at higher dimensions. We see that after $d_h = 5$ no significant information is gained, with errors plateauing at approximately $d_h \geqslant 7$. We can also look at the case in which $a =$ Re $[a_{0,1}]$ and $b =$ Im $[a_{0,1}]$ in Fig. 13(b). We notice that errors of the joint PDF in Fig. 13(b) show a similar trend as Fig. 13(a) with errors plateauing at approximately $d_h \geqslant 9$. We can infer from these results that the embedding dimension of this system lies in the range $d_h = 5$–9, and furthermore that the data-driven model can reproduce the long-time statistics with very high fidelity.

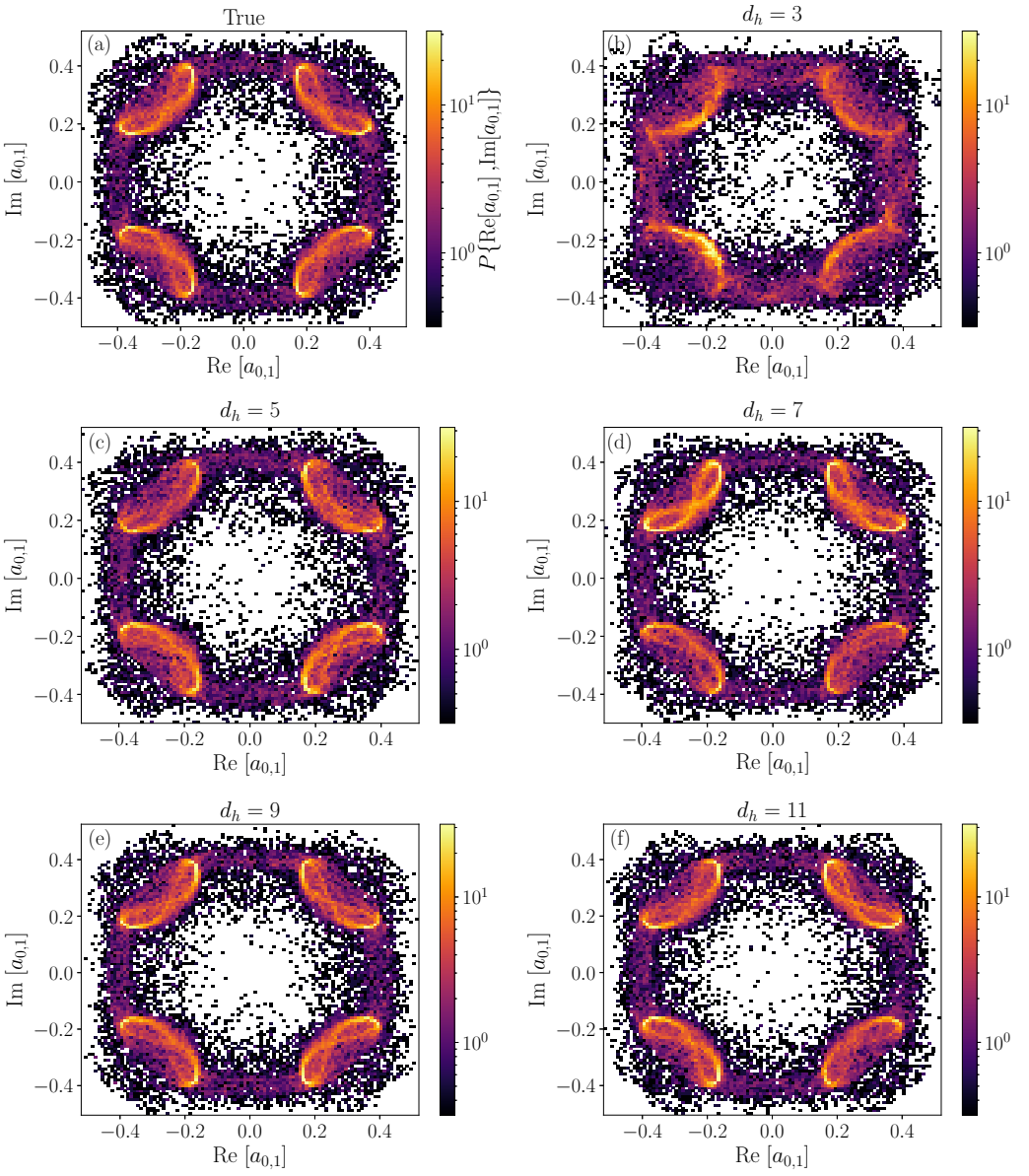

FIG. 12. Re = 14.4: Joint PDFs of $\mathrm{Re}[a_{0,1}(t)] - \mathrm{Im}[a_{0,1}(t)]$ corresponding to Re = 14.4 for (a) true and (b)–(f) predicted data corresponding to dimensions $d_h = 3$, 5, 7, 9, and 11.

The above PDFs yield no information about the temporal behavior of the system. One temporal feature of significant interest in problems with intermittency is the probability density of the durations of time intervals with different behavior. To address this, we consider the PDFs of time spent in bursting (t_b) and in quiescent (t_q) regions. The labeling method discussed in the previous section is used. For this calculation we take a trajectory of 10^5 snapshots from an arbitrary IC. The PDF for the true data is shown in Fig. 14(a) followed by the PDFs that come from the $d_h = 3$, 5, 7, 9, and 11 models in Figs. 14(b)–14(f). The true data show that t_q is mostly concentrated between $t \approx 200$ and 300 with a high intensity peak shown at $t = 5$. We attribute this peak to a

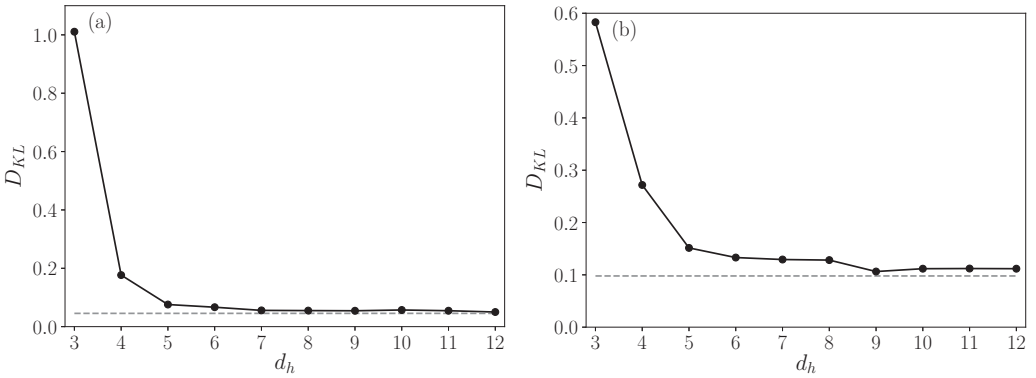

FIG. 13. Re = 14.4: D_{KL} vs dimension d_h for (a) I-D and (b) $\mathrm{Re}[a_{0,1}] - \mathrm{Im}[a_{0,1}]$ predicted vs true joint PDFs. Dashed gray line corresponds to D_{KL} calculated over true data sets.

small fraction of snapshots in the bursting region that get mislabeled as quiescent due to the weakly chaotic nature of the data. We do not expect for this to drastically change our conclusions because the same labeling system is used for the true data and the models. In the case of t_b we notice that these are mostly concentrated between $t \approx 0$ and 200. Looking at both the PDFs and averages of the times we see that $d_h = 3$ fails to correctly capture the shape of the PDF and also underpredicts $\langle t_q \rangle$ and $\langle t_b \rangle$. At $d_h = 5$ we start getting better agreement where we see that the PDFs clearly show the two regions where t_b and t_q are concentrated. In the case of $d_h = 7$ we can see that the quiescent PDF spreads into regions with higher t_q and for $d_h = 9$, 11 these seem to agree better with the true PDF. Figure 15 shows D_{KL} with varying d_h for these PDFs. As expected from observing the PDFs, we see that D_{KL} decreases up until $d_h = 5$ for both cases. In the case of t_q we see an increase in the error after $d_h = 5$, which agrees with the above observation of the PDF at $d_h = 7$. For t_b, D_{KL} seems to keep slightly decreasing after $d_h = 5$. We also notice that for t_q, D_{KL} reaches a minimum at $d_h = 9$ and for t_b no significant decrease is observed at $d_h \geqslant 9$. In short, these duration statistics achieve similar agreement at $d_h = 9$, and for the case of t_b errors keep decreasing with increasing d_h. We also calculate the mean of t_q and t_b for the case of $d_h = 9$, and we obtain values of $\langle t_q \rangle = 174$ and $\langle t_b \rangle = 97$, which agree closely with the true values of $\langle t_q \rangle = 176$ and $\langle t_b \rangle = 97$.

D. Phase prediction

Recall that we gain substantial accuracy in dimension reduction by factoring out the spatial phase $\phi_x(t)$ of the data. Here we complete the dynamical picture of the model predictions at Re = 14.4 by illustrating the predictions of phase evolution, as given by the learned phase evolution Eq. (12). Figure 16(a) shows a short-time evolution of $\phi_x(t)$ corresponding to the true and predicted data for the $d_h = 3$, 5, 7, 9, and 11 models. The smooth increases and decreases in Fig. 16(a) correspond to trajectories during time intervals where they are near an RPO and thus are traveling in the x-direction. The intervals where the phase fluctuates rapidly are the bursts during which the trajectories are moving between the RPO regions. This behavior is well-captured for all of the dimensions shown except for $d_h = 3$. Notice that although the trajectories diverge, for short times we get around two t_L of prediction horizon where the models still capture the correct dynamics, and Fig. 16(a) provides a clear visual indication that the loss of predictability occurs during the bursts.

We now take an approach to quantify how well the model performs with respect to the true data. Taking a look at the drops and increases for $\phi_x(t)$ we can observe that after every burst the trajectory will travel, essentially randomly, in either the positive (increasing ϕ_x) or negative (decreasing ϕ_x) x direction. This behavior is essentially a run and tumble or random-walk behavior in the sense that the long periods of positive or negative phase drift correspond to "runs" that are separated by

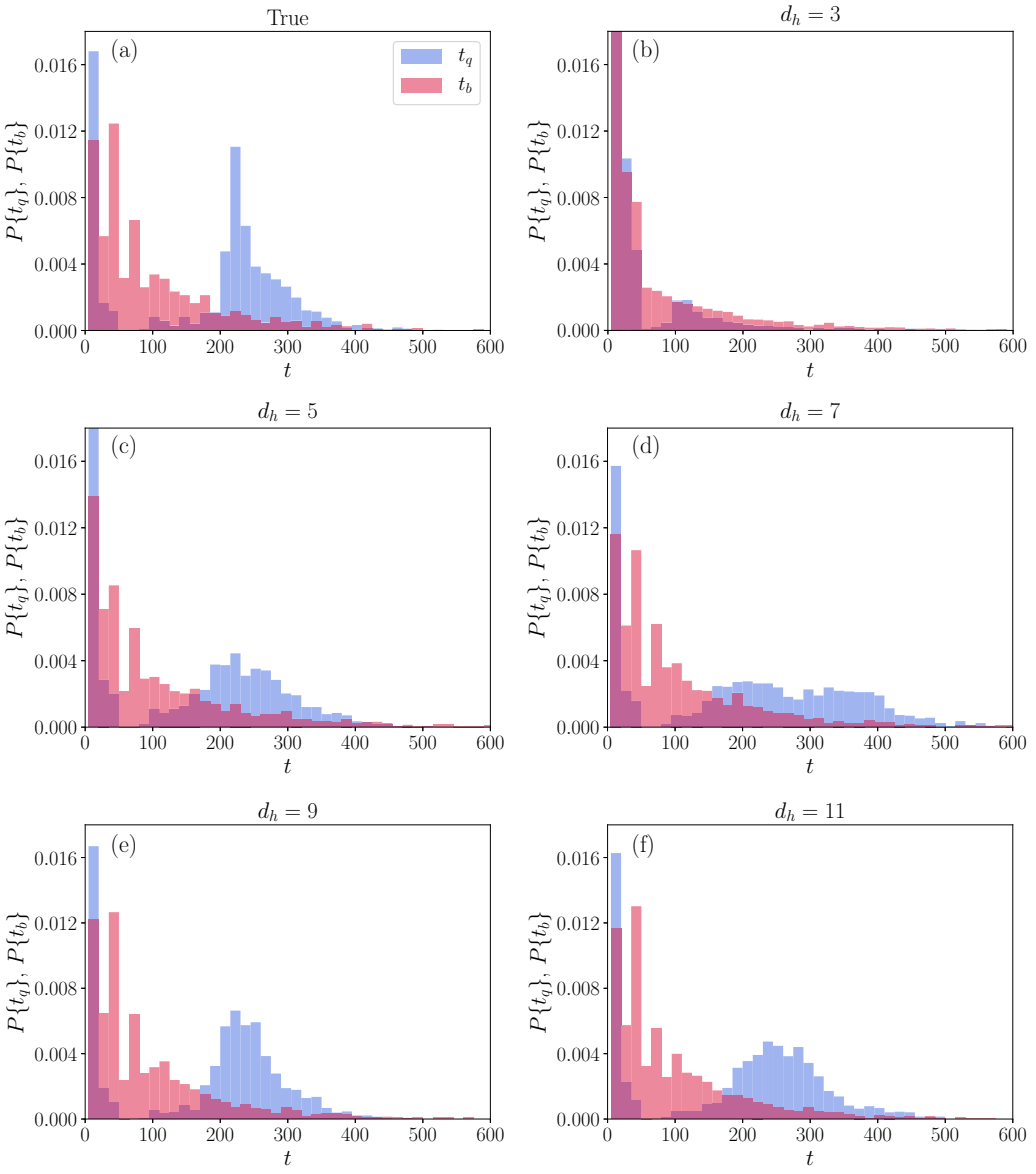

FIG. 14. PDFs of t_q and t_b at Re $= 14.4$ for (a) true and (b)–(f) predicted data for dimensions $d_h = 3$, 5, 7, 9, and 11.

"tumbles" that correspond to the bursts, in which the direction of phase motion is reset. Hence, a natural analysis of quantification for this type of dynamics consists of calculating the mean-squared displacement (MSD) of the phase,

$$\mathrm{MSD}(t) = \langle [\phi_x(t) - \phi_x(0)]^2 \rangle. \tag{15}$$

Figure 16(b) shows the time evolution of MSD of true and predicted data. The black line corresponds to the true data, and the black and green dashed lines serve as references with slopes of 1 and 1.5, respectively. The colored lines correspond to models with various dimensions. Looking at the true curve, we notice a change from superdiffusive (slope $= 1.5$) to diffusive (slope $= 1$) scaling that

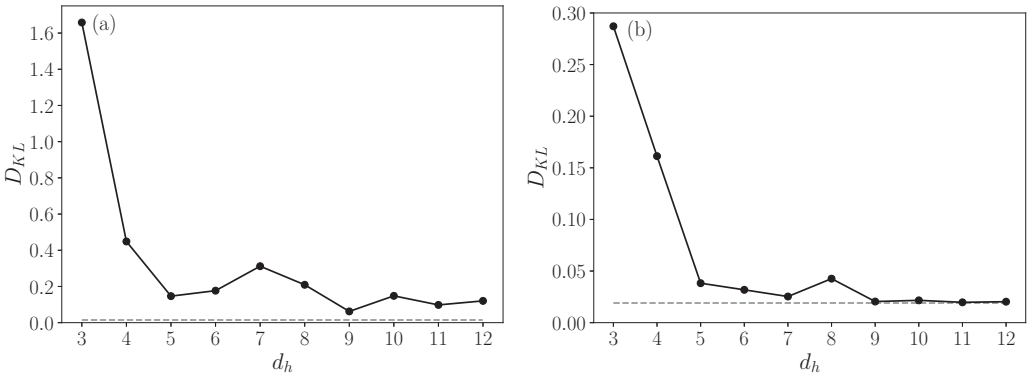

FIG. 15. Re = 14.4: D_{KL} vs dimension d_h corresponding to PDFs for (a) t_q and (b) t_b. Dashed gray line corresponds to D_{KL} calculated over different true data sets.

happens around $t \approx 200$, which corresponds to the mean duration of the quiescent intervals, as discussed above: i.e., to the average time the trajectories travel along the RPOs before bursting. The trajectory then bursts and reorients, which is captured by the long-time diffusive trend. Looking at the performance of the models, we observe that $d_h = 3$ does a good job at capturing the short-time scaling, however it is not able to capture the change in slope that is observed in the true data. It is not until $d_h \geqslant 5$ that the correct behavior at long times is observed—indeed, the predictions agree very well with the data, with a slight upward shift at long times corresponding to the slight overprediction of the mean duration of the quiescent periods.

E. Bursting prediction

Previous research has focused on finding indicators that guide predictions of when a burst will occur. It has been shown for the Kolmogorov flow that before a burst there is a depletion of the content in the (1,0) Fourier mode, which then feeds into the forcing mode $(0, n)$ [32]. Figure 17 shows how this looks for Re = 14.4, $n = 2$. By considering a variational framework and finding solutions to a constrained optimization problem, it was also found that examination of these modes can lead to predictions of when a burst will occur [37].

With our framework, natural indicators are the latent variables h, which we will consider here along with some variations, including the indicators used in previous work. To predict bursting

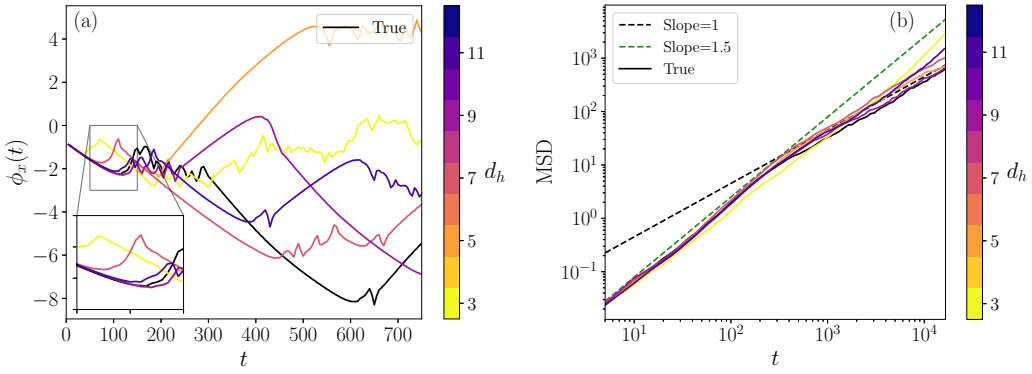

FIG. 16. (a) Time evolution of ϕ_x corresponding to the true data and models with dimensions $d_h = 3$, 5, 7, 9, and 11. (b) MSD of $\phi_x(t)$ corresponding to true data and models with dimensions $d_h = 3$, 5, 7, 9, and 11.

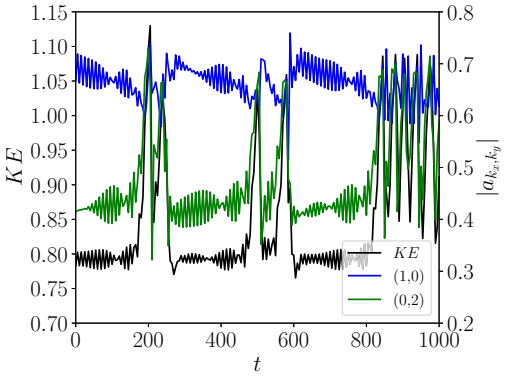

FIG. 17. Time evolution of KE and amplitudes corresponding to (1,0) and (0,2) Fourier mode for Re = 14.4.

events based on a given indicator, we will use a simple binary classifier in the form of a support vector machine (SVM) with a radial basis function kernel [38]. These have shown success in predicting extreme events for problems such as extreme rainfall [39]. With this approach, data at time t are used to learn a function that outputs a binary label of bursting/not bursting at time $t + \tau_b$. For all of the cases considered, we use the $d_h = 5, 9$ models, taking a dataset of 5×10^4 snapshots to train the SVM and another 5×10^4 as a test set.

Figure 18(a) shows the percent correct classification of bursting events with varying time τ_b in the future. The gray and black curves corresponds to predicting the events based on the PCA projection of the data, $P_{d_h}U^T\omega$, into the first $d_h = 5$ and 9 coefficients, respectively. The cyan and red curves correspond to h of dimensions $d_h = 5$ and 9, respectively. We notice that the PCA and h curves fall on top of one another and have a high probability of correct classification when considering prediction horizons less than one t_L. For this purpose, we see that PCA is enough to predict bursting events. Figure 18(b) shows the percent correct classification of bursting at time τ_b in the future for the previously discussed indicators. None of these work nearly as well as $P_{d_h}U^T\omega$ or h. The blue curve corresponds to (1,0) amplitude of the original true data, the green curve to the forcing (0,2) amplitude, and we also consider $\Delta\phi$ in the purple curve. In the case of $\Delta\phi$ we see some predictability at times longer than one t_L and less than two. This also happens for the case of (1,0), however there seems to be no decrease or increase in the probability of correct classification. We

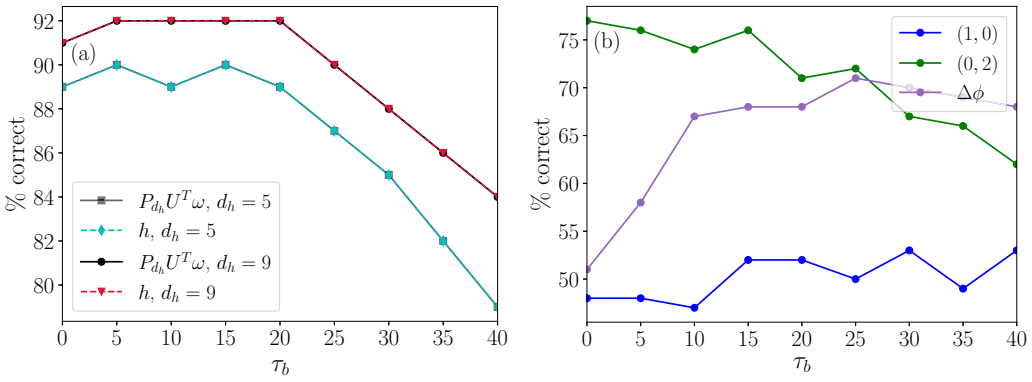

FIG. 18. Percent of correctly classified bursting events at τ_b forward in time for (a) $P_{d_h}U^T\omega$ and h at $d_h =$ 5, 9, (b) and indicators $\Delta\phi$, (1,0), and (0,2). Note that the vertical scales in (a) and (b) are very different.

can see from Fig. 17 that even though there is a depletion in the (1,0) mode preceding bursts, its amplitude does not change dramatically between quiescent and bursting intervals, which may be a reason that it does not provide much predictive power. The amplitude (0,2), which changes more strongly between quiescent and bursting regions, is seen to be the better predictor for bursting events. At small τ_b its predictions outperform (1,0) and $\Delta\phi$, however at times larger than one t_L, $\Delta\phi$ performs better.

V. CONCLUSION

The nonlinearity of the NSE poses challenges when using ROMs, where the dynamics are expected to evolve on an invariant manifold that will not lie in a linear subspace. Neural networks have proven to be powerful tools for learning efficient ROMs solely from data, however finding and exploiting a *minimal*-dimensional model has not been emphasized. We present a data-driven methodology to learn an estimate of the embedding dimension of the manifold for chaotic Kolmogorov flow and the time evolution on it. An autoencoder is used to find a nonlinear low-dimensional subspace and a dense neural network to evolve it in time.

Our autoencoders are trained on vorticity data from two cases: a case in which the dynamics show a relative periodic orbit solution ($\mathrm{Re} = 13.5$), and a case with chaotic dynamics ($\mathrm{Re} = 14.4$). The chaotic regime we consider comes with challenges due to the intermittent behavior observed where the trajectory travels in between quiescent intervals and bursting events. We factor out the rich symmetries of Kolmogorov flow before training of the autoencoders, which dramatically improves reconstruction error of the snapshots. This improves training efficiency by not having to learn a compression of the full state. Specifically, factoring out the translation symmetry decreases the mean-squared reconstruction error by an order of magnitude compared to the case in which phase is not factored out, and several orders of magnitude compared to PCA. The phase-aligned low-dimensional subspace is then used for time evolution where the RPO dynamics is learned essentially perfectly at $d_h = 2$ for $\mathrm{Re} = 13.5$, and very good agreement for short- and long-time statistics is obtained at $d_h = 5$ for $\mathrm{Re} = 14.4$. Further small improvements in the results occur as dimension is increased to nine, beyond which the statistics of the model and true system are in very good agreement. For comparison, the full state space of the numerical simulation data is $N = 1024$.

We also show phase prediction evolution results based on the low-dimensional subspace learned. The time evolution of the true phase exhibits a superdiffusive scaling at short times and a diffusive scaling at long times, which we attribute to the traveling near an RPO and the reorientation due to bursting. Finally, using the low-dimensional representation enables accurate prediction of bursting events based on conditions about a Lyapunov time ahead of the event. This work opens new avenues for data-driven ROMs with applications such as control for drag reduction, an example of which is presented for turbulent Couette flow in [40]. One important challenge that remains is more effective treatment of systems with intermittent dynamics like those described here. A recent study [13] has introduced a method that uses the differential topology formalism of charts and atlases to develop *local* manifold representations and dynamical models that can be stitched together to form a global dynamical model. One attractive feature of that formalism is that it enables use of separate representations for regions of state space with very different dynamics, and it has already been shown in specific cases to provide dramatically improved results for dynamics with intermittency.

ACKNOWLEDGMENTS

This work was supported by AFOSR FA9550-18-1-0174 and ONR N00014-18-1-2865 (Vannevar Bush Faculty Fellowship). We also want to thank the Graduate Engineering Research Scholars (GERS) program and funding through the Advanced Opportunity Fellowship (AOF) as well as the PPG Fellowship.

[1] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge University Press, Cambridge, 2012).
[2] B. R. Noack and H. Eckelmann, A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder, Phys. Fluids **6**, 124 (1994).
[3] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, J. Fluid Mech. **192**, 115 (1988).
[4] M. Sieber, C. O. Paschereit, and K. Oberleithner, Spectral proper orthogonal decomposition, J. Fluid Mech. **792**, 798 (2016).
[5] H. F. S. Lui and W. R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, J. Fluid Mech. **872**, 963 (2019).
[6] A. J. Linot and M. D. Graham, Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations, Chaos **32**, 073110 (2022).
[7] E. Hopf, A mathematical example displaying features of turbulence, Commun. Pure Appl. Math. **1**, 303 (1948).
[8] C. Foias, O. Manley, and R. Temam, Modelling of the interaction of small and large eddies in two dimensional turbulent flows, ESAIM: Math. Modell. Numer. Anal. **22**, 93 (1988).
[9] R. Temam, Do inertial manifolds apply to turbulence? Physica D **37**, 146 (1989).
[10] S. Zelik, Attractors. Then and now, arXiv:2208.12101 (2022).
[11] J. M. Lee, Smooth manifolds, in *Introduction to Smooth Manifolds* (Springer, New York, 2013), pp. 1–31.
[12] H. Whitney, The self-intersections of a smooth n-manifold in 2n-space, Ann. Math. **45**, 220 (1944).
[13] D. Floryan and M. D. Graham, Data-driven discovery of intrinsic dynamics, Nat. Mach. Intell. **4**, 1113 (2021).
[14] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, Predictions of turbulent shear flows using deep neural networks, Phys. Rev. Fluids **4**, 054603 (2019).
[15] J. Moehlis, H. Faisst, and B. Eckhardt, A low-dimensional model for turbulent shear flows, New J. Phys. **6**, 56 (2004).
[16] J. Page, M. P. Brenner, and R. R. Kerswell, Revealing the state space of turbulence using machine learning, Phys. Rev. Fluids **6**, 034402 (2021).
[17] T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata, Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow, Phys. Fluids **33**, 025116 (2021).
[18] N. A. K. Doan, W. Polifke, and L. Magri, Auto-encoded reservoir computing for turbulence learning, in *International Conference on Computational Science* (Springer, Krakow, Poland, 2021), pp. 344–351.
[19] A. J. Linot and M. D. Graham, Deep learning to discover and predict dynamics on an inertial manifold, Phys. Rev. E **101**, 062209 (2020).
[20] K. Zeng, A. J. Linot, and M. D. Graham, Data-driven control of spatiotemporal chaos with reduced-order neural ODE-based models and reinforcement learning, Proc. R. Soc. A **478**, 20220297 (2022).
[21] C. J. Crowley, J. L. Pughe-Sanford, W. Toler, M. C. Krygier, R. O. Grigoriev, and M. F. Schatz, Turbulence tracks recurrent solutions, Proc. Natl. Acad. Sci. USA **119**, e2120665119 (2022).
[22] D. Armbruster, B. Nicolaenko, N. Smaoui, and P. Chossat, Symmetries and dynamics for 2-D Navier-Stokes flow, Physica D **95**, 81 (1996).
[23] D. Armbruster, R. Heiland, E. J. Kostelich, and B. Nicolaenko, Phase-space analysis of bursting behavior in Kolmogorov flow, Physica D **58**, 392 (1992).
[24] C. W. Rowley and S. T. M. Dawson, Model reduction for flow analysis and control, Annu. Rev. Fluid Mech. **49**, 387 (2017).
[25] G. J. Chandler and R. R. Kerswell, Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow, J. Fluid Mech. **722**, 554 (2013).
[26] V. I. Iudovich, Example of the generation of a secondary stationary or periodic flow when there is loss of stability of the laminar flow of a viscous incompressible fluid, J. Appl. Math. Mech. **29**, 527 (1965).

[27] L. D. Meshalkin and Ia G. Sinai, Investigation of the stability of a stationary solution of a system of equations for the plane movement of an incompressible viscous liquid, J. Appl. Math. Mech. **25**, 1700 (1961).
[28] J. S. A. Green, Two-dimensional turbulence near the viscous limit, J. Fluid Mech. **62**, 273 (1974).
[29] A. Thess, Instabilities in two-dimensional spatially periodic flows. Part I: Kolmogorov flow, Phys. Fluids **4**, 1385 (1992).
[30] P. Bartello and T. Warn, Self-similarity of decaying two-dimensional turbulence, J. Fluid Mech. **326**, 357 (1996).
[31] N. Platt, L. Sirovich, and N. Fitzmaurice, An investigation of chaotic Kolmogorov flows, Phys. Fluids **3**, 681 (1991).
[32] B. Nicolaenko and Z.-S. She, Symmetry-breaking homoclinic chaos in Kolmogorov flows, in *Nonlinear World* (World Scientific, Singapore, 1990).
[33] N. B. Budanur, D. Borrero-Echeverry, and P. Cvitanović, Periodic orbit analysis of a system with continuous symmetry—A tutorial, Chaos **25**, 073112 (2015).
[34] N. B. Budanur, P. Cvitanović, R. L. Davidchack, and E. Siminos, Reduction of SO(2) Symmetry for Spatially Extended Dynamical Systems, Phys. Rev. Lett. **114**, 084102 (2015).
[35] S. Kneer, T. Sayadi, D. Sipp, P. Schmid, and G. Rigas, Symmetry-Aware Autoencoders: s-PCA and s-nlPCA, arXiv:2111.02893 (2021).
[36] M. Inubushi, M. U. Kobayashi, S.-i. Takehiro, and M. Yamada, Covariant Lyapunov analysis of chaotic Kolmogorov flows, Phys. Rev. E **85**, 016331 (2012).
[37] M. Farazmand and T. P. Sapsis, A variational approach to probing extreme events in turbulent dynamical systems, Sci. Adv. **3**, e1701533 (2017).
[38] B. E. Boser, I. M. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, in *Proceedings of the fifth annual workshop on Computational learning theory (COLT '92)* (Association for Computing Machinery, New York, USA, 1992), pp. 144–152.
[39] M. A. Nayak and S. Ghosh, Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier, Theor. Appl. Climatol. **114**, 583 (2013).
[40] A. J. Linot, K. Zeng, and M. D. Graham, Turbulence control in plane couette flow using low-dimensional neural ode-based models and deep reinforcement learning, arXiv:2301.12098 (2023).