Promoting global stability in data-driven models of quadratic nonlinear dynamics

Alan A. Kaptanoglu, Jared L. Callaham, Aleksandr Aravkin, Christopher J. Hansen, and Steven L. Brunton

Phys. Rev. Fluids **6**, 094401 — Published 7 September 2021

DOI: 10.1103/PhysRevFluids.6.094401

Promoting global stability in data-driven models of quadratic nonlinear dynamics

Alan A. Kaptanoglu*
Department of Physics, University of Washington, Seattle, WA 98195, United States

Jared L. Callaham and Aleksandr Aravkin
Department of Applied Mathematics, University of Washington, Seattle, WA 98195, United States

Christopher J. Hansen
Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195, United States

Steven L. Brunton
Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, United States

Modeling realistic fluid and plasma flows is computationally intensive, motivating the use of reduced-order models for a variety of scientific and engineering tasks. However, it is challenging to characterize, much less guarantee, the global stability (i.e., long-time boundedness) of these models. Previous work provided a theorem outlining necessary and sufficient conditions to ensure global stability in systems with energy-preserving, quadratic nonlinearities, with the goal of evaluating the stability of projection-based models. In this work, we incorporate this theorem into modern data-driven models obtained via machine learning. First, we propose that this theorem should be a standard diagnostic for the stability of projection-based and data-driven models, examining the conditions under which it holds. Second, we illustrate how to modify the objective function in machine learning algorithms to promote globally stable models, with implications for the modeling of fluid and plasma flows. Specifically, we introduce a modified "trapping SINDy" algorithm based on the sparse identification of nonlinear dynamics (SINDy) method. This method enables the identification of models that, by construction, only produce bounded trajectories. The effectiveness and accuracy of this approach are demonstrated on a broad set of examples of varying model complexity and physical origin, including the vortex shedding in the wake of a circular cylinder.

Keywords: fluid mechanics, reduced-order modeling, Galerkin projection, machine learning, global stability, nonlinear systems, magnetohydrodynamics, data-driven models, SINDy

I. INTRODUCTION

Modeling the full spatio-temporal evolution of natural processes is often computationally expensive, motivating the use of reduced-order models (ROMs) that capture only the dominant behaviors of a system [1–6]. Projection-based model reduction is a common approach for generating such models; a high-dimensional system, such as a spatially discretized set of partial differential equations (PDEs), is projected onto a low-dimensional basis of modes [7, 8]. This projection leads to a computationally efficient system of ordinary differential equations (ODEs) that describes how the mode amplitudes evolve in time [9]. However, these models often suffer from stability issues, causing solutions to diverge in finite-time. To address this issue, Schlegel and Noack [10] developed a "trapping theorem" with necessary and sufficient conditions for long-term model stability for systems that exhibit quadratic, energy-preserving nonlinearities.

Quadratic nonlinearity is pervasive in nature, with common examples including convection in the Navier-Stokes equations and the Lorentz force in magnetohydrodynamics (MHD). The trapping theorem provides conditions for the existence of a global trapping region, towards which every system trajectory asymptotically and monotonically converges; once a trajectory enters this region, it remains inside for all time, guaranteeing that all trajectories are bounded. These types of guarantees are ideal for the application of real-time flow-control strategies. An example trapping region is illustrated by the blue sphere in Fig. 1 for the Lorenz system. For convenience, we will use the terms "global stability", "long-term boundedness", and "monotonically trapping region" interchangeably, although systems exhibiting trapping regions are a strict subset of globally stable systems (see Fig. 1 of Schlegel and Noack [10] for a useful organizational diagram of these various notions of stability). In this work, we adapt the trapping theorem from projection-based modeling to promote global stability in data-driven machine learning models.

Increasingly, reduced-order models of complex systems, such as fluids and plasmas, are discovered from data with modern machine learning algorithms [11–36], rather than classical projection-based methods that are intrusive and require intricate knowledge of the governing equations. These data-driven approaches for modeling fluid dynamics [37, 38] range from generalized regression techniques [11, 15, 19] to deep learning [25, 31, 32, 34, 35, 39]. It is often possible to improve the stability and performance of data-driven models by incorporating partially known physics, such as conservation laws and symmetries [19, 30, 36], or known physical structure [40]. Thus, incorporating physics into machine learning and developing hybrid data-driven and operator-

* Corresponding author (akaptano@uw.edu).

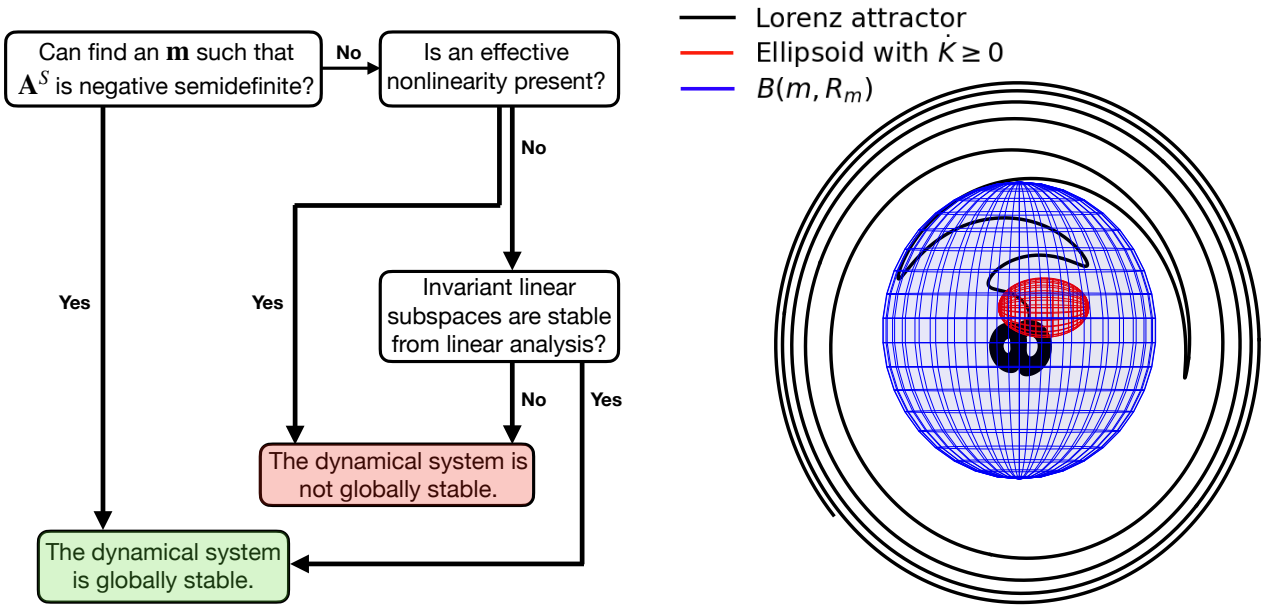


FIG. 1: Left: Decision diagram to determine global stability, modified from Schlegel and Noack [10] and described in Section II C. Right: Illustration of a trapping region (blue sphere) for the Lorenz system; all outside trajectories monotonically approach this region, and after entering, remain inside. Trajectories inside the red ellipsoid experience positive energy growth, in this case precluding convergence to a fixed point.

based approaches are rapidly growing fields of research [19, 40–48]. Physics can be incorporated into machine learning algorithms through model structure, by augmenting training data with known symmetries, by adding constraints to the optimization, or by adding custom loss functions [38]. However, even physics-informed data-driven models often lack global stability guarantees, and the ability of these methods to find long-term bounded models depreciates as the state dimension increases.

In this work, we use the Schlegel and Noack [10] trapping theorem to diagnose and promote global stability of data-driven models with quadratic nonlinearities. Even though their theorem was developed in the context of projection-based ROMs, we emphasize that it can be applied directly to analyze data-driven model stability *post hoc*, and we examine conditions under which it holds. Next, we describe how to use this theorem to promote global stability in machine learned models by modifying the optimization loss function. We illustrate this approach on the sparse identification of nonlinear dynamics (SINDy) algorithm [15, 17] by implementing a custom optimization loss term that promotes models that are globally stable by construction. A constrained version of the SINDy optimization was previously developed to enforce energy-preserving quadratic structure in incompressible fluids [19] and it has since been extended for arbitrary state size and global conservation laws in magnetohydrodynamic systems [30, 36]. These constrained SINDy variants generally produce more stable models, and reflect a broader trend that stability issues can often be improved by building physical constraints into system identification methods [19, 49]. Our "trapping SINDy" algorithm generalizes previous stabilized or constrained reduced-order models for fluids by considering global rather than local stability, allowing for both transients and long-time attracting sets. Promoting global stability also improves robustness to noise over unconstrained or constrained SINDy. Recent works by Erichson et al. [50] and Sawant et al. [51] promote a more restrictive Lyapunov stable origin in fluid flows by adding similar loss terms to the optimization problem. Additionally, much of the literature has focused on the long-time energy properties of a dynamic attractor [52] by either prescribing that the system be *fully* energy-preserving (or Hamiltonian) [53–58] or applying real-time control [59]. Mohebujjaman et al. [46] also used a simple version of the trapping theorem in order to constrain a hybrid projection-based and data-driven method. The present work builds on these studies, providing a framework for addressing the long-standing challenge of promoting global stability in data-driven models.

The remainder of this paper is organized as follows: in Section II, we introduce the general class of systems with energy-preserving quadratic nonlinearities, investigate the circumstances under which the trapping theorem holds, and indicate connections with other stability descriptions in fluid mechanics. In Section III, we define our "trapping SINDy" algorithm. Our trapping SINDy implementation is open-source and available through the PySINDy software package [60]. This is a rather technical section on nonconvex optimization; the reader may skip this section and proceed to the results if the algorithmic details are not of interest. In Section IV, we demonstrate the effectiveness of this new system identification technique on a wide range of examples. Abridged versions of all of the results have been incorporated into a single PySINDy example notebook and can be reproduced in a few minutes on a laptop. In Section V, we conclude with suggestions for future work. Similar trapping theorems are promising for data-driven models in fields such as neuroscience, epidemiology, and population dynamics.

II. REDUCED-ORDER MODELING

Before describing how we incorporate the trapping theorem of Schlegel and Noack [10] into data-driven models, here we briefly describe the family of projection-based ROMs for which the trapping theorem was introduced, and investigate the circumstances under which this theorem holds. It is helpful to first motivate this work by reviewing the many scenarios under which energy-preserving quadratic nonlinearities can arise. In fluid dynamics, the quadratic nonlinearity often represents the convective derivative $(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}$ in the Navier-Stokes equations. This quadratic nonlinearity is energy-preserving for a large number of boundary conditions. Examples include no-slip conditions, periodic boundary conditions [9, 61], mixed no-slip and periodic boundary conditions [62], and open flows in which the velocity magnitude decreases faster than the relevant surface integrals expand (e.g., two-dimensional rigid body wake flows and three-dimensional round jets) [63]. In magnetohydrodynamics, there are additional quadratic nonlinearities through $\nabla\times(\boldsymbol{u}\times\boldsymbol{B})$ and $\boldsymbol{J}\times\boldsymbol{B}$, which are also energy-preserving with common experimental boundary conditions such as a conducting wall [64], or a balance between dissipation and actuation in a steady-state plasma device [36, 65]. Notably, dissipationless Hall-MHD has four invariants; energy, cross-helicity, magnetic helicity, and generalized helicity [66], providing a wealth of potential model constraints for Hall-MHD ROMs. Here $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{J}$ is the electromagnetic current, and $\boldsymbol{B}$ is the magnetic field.

A. Projection-based ROMs

In modern scientific computing, a set of governing partial differential equations is typically discretized into a high-dimensional system of coupled ordinary differential equations. In this work we will explicitly consider dynamics with linear plus quadratic structure, as are found in many fluid and plasma systems:

$$\dot{\boldsymbol{u}} = \boldsymbol{L}^0\boldsymbol{u} + \boldsymbol{Q}^0(\boldsymbol{u}). \tag{1}$$

Here we assume that the PDE has already been discretized for numerical computation, resulting in a coupled system of n differential equations. The state of the system $\boldsymbol{u}(\boldsymbol{x},t)\in\mathbb{R}^n$ is a high-dimensional vector that represents the fluid velocity or other set of spatio-temporal fields, for example sampled on a high-resolution spatial grid. Thus, $\boldsymbol{L}^0$ and $\boldsymbol{Q}^0$ are high-dimensional operators used to perform the numerical simulation. The zero subscript distinguishes these operators from the Galerkin coefficients defined below in Eq. (3).

The goal of a projection-based ROM is to transform this high-dimensional system into a lower-dimensional system of size $r \ll n$ that captures the essential dynamics. One way to reduce the set of governing equations to a set of ordinary differential equations is by decomposition into a desired low-dimensional basis $\{\boldsymbol{\varphi}_i(\boldsymbol{x})\}$ in a process commonly referred to as Galerkin expansion:

$$\boldsymbol{u}(\boldsymbol{x},t) = \overline{\boldsymbol{u}}(\boldsymbol{x}) + \sum_{i=1}^{r} a_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}). \tag{2}$$

Here, $\overline{\boldsymbol{u}}(\boldsymbol{x})$ is the mean field, $\boldsymbol{\varphi}_i(\boldsymbol{x})$ are spatial modes, and $a_i(t)$ describe how the amplitude of these modes vary in time. The proper orthogonal decomposition (POD) [9, 67] is frequently used to obtain the basis, since the modes $\boldsymbol{\varphi}_i(\boldsymbol{x})$ are orthogonal and ordered by maximal energy content. The governing equations are then Galerkin projected onto the basis $\{\boldsymbol{\varphi}_i(\boldsymbol{x})\}$ by substituting Eq. (2) into the PDE and using inner products to remove the spatial dependence. Orthogonal projection onto POD modes is the simplest and most common procedure, resulting in *POD-Galerkin* models, although Petrov-Galerkin projection [3, 4] improves model performance in some cases.

In this paper, we rely on a theorem that is derived explicitly from modal expansions of the form of Eq. (2), i.e. linear and separable decompositions, although non-orthogonal modes can be straightforwardly handled in the following analysis. This reliance on a simple basis is not ideal, because there are many other modal expansions and bases that have been introduced for reduced-order fluid [7, 8] and plasma models [68–71], including balanced POD [72, 73], spectral POD [20], dynamic mode decomposition (DMD) [11, 12, 74], the Koopman decomposition [13, 75, 76], resolvent analysis [77, 78], and autoencoders [32, 79, 80]. Future work is required to expand our strategy to these alternative and nonlinear bases, and we provide some potential avenues for progress in the conclusion in Sec. V.

Now, if the governing equations for $\boldsymbol{u}(\boldsymbol{x},t)$ are at most quadratic in nonlinearity, Galerkin projection produces the following system of ODEs for the set of temporal functions $a_i(t)$,

$$\dot{a}_i(t) = E_i + \sum_{j=1}^{r} L_{ij}a_j + \sum_{j,k=1}^{r} Q_{ijk}a_j a_k. \tag{3}$$

E_i, L_{ij}, and Q_{ijk} are tensors of static coefficients, obtained from spatial inner products between the $\boldsymbol{\varphi}_i(\boldsymbol{x})$ and the operators $\boldsymbol{L}^0$ and $\boldsymbol{Q}^0$, that define the model dynamics. The class of systems we consider are those with energy-preserving nonlinearity, for which

$$\sum_{i,j,k=1}^{r} Q_{ijk}a_i a_j a_k = 0, \tag{4}$$

or equivalently, for all $i,j,k\in\{1,\ldots,r\}$,

$$Q_{ijk} + Q_{jik} + Q_{kji} = 0. \tag{5}$$

Q_{ijk} is symmetric in swapping j and k without loss of generality.

B. Schlegel and Noack Trapping Theorem

This theorem provides necessary and sufficient conditions for energy-preserving, effectively nonlinear, quadratic systems to exhibit a trapping region $B(\boldsymbol{m}, R_m)$, a ball centered at $\boldsymbol{m} \in \mathbb{R}^r$ with radius R_m. Outside this region the rate of change of energy K is negative everywhere, producing a Lyapunov function that renders this system globally stable. Recentering the origin by an arbitrary constant vector $\boldsymbol{m}$, the energy may be expressed in terms of the shifted state vector $\boldsymbol{y}(t) = \boldsymbol{a}(t) - \boldsymbol{m}$ as

$$K = \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}. \tag{6}$$

Taking a derivative and substituting in Eq. (3) produces

$$\frac{d}{dt}K = \boldsymbol{y}^T\boldsymbol{A}^S\boldsymbol{y} + \boldsymbol{d}_m^T\boldsymbol{y}, \tag{7}$$

$$\boldsymbol{A}^S = \boldsymbol{L}^S - \boldsymbol{m}^T\boldsymbol{Q},\ \ \boldsymbol{L}^S = \frac{1}{2}(\boldsymbol{L} + \boldsymbol{L}^T),\ \text{and} \tag{8}$$

$$\boldsymbol{d}_m = \boldsymbol{E} + \boldsymbol{L}\boldsymbol{m} + \boldsymbol{Q}\boldsymbol{m}\boldsymbol{m}. \tag{9}$$

$\boldsymbol{m}^T\boldsymbol{Q}$ refers to $m_i Q_{ijk}$ and $\boldsymbol{Q}\boldsymbol{m}\boldsymbol{m}$ to $Q_{ijk}m_j m_k$. The trapping theorem may now be stated as:

Theorem 1. *There exists a monotonically trapping region at least as small as the ball $B(\boldsymbol{m}, R_m)$ if and only if the real, symmetric matrix $\boldsymbol{A}^S$ is negative definite with eigenvalues $\lambda_r \leq \cdots \leq \lambda_1 < 0$; the radius is then given by $R_m = \|\boldsymbol{d}_m\|/|\lambda_1|$.*[*]

In practice, the goal is then to find an origin $\boldsymbol{m}$ so that the matrix $\boldsymbol{A}^S$ is negative definite, guaranteeing a trapping region and global stability. Without effective nonlinearity, described at the beginning of Section II C, only the backwards direction holds; if we can find an $\boldsymbol{m}$ so that $\boldsymbol{A}^S$ is negative definite, the system exhibits a trapping region. However, such systems can be globally stable without admitting such an $\boldsymbol{m}$. Subsequently, the goal of Section III is to use this theorem to define a constrained machine learning optimization that identifies a reduced-order model with a guaranteed trapping region. Even when the requirements of the trapping theorem are not fully satisfied, the algorithm results in Section IV indicate that this approach tends to produce models with improved stability properties.

To understand the R_m bound in Thm. 1, we transform into eigenvector coordinates $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{y}$, $\boldsymbol{h} = \boldsymbol{d}_m\boldsymbol{T}^T$, where $\boldsymbol{T}$ are the eigenvectors of $\boldsymbol{A}^S$. Now Eq. (7) becomes

$$\frac{d}{dt}K = \sum_{i=1}^{r} h_i z_i + \lambda_i z_i^2 = \sum_{i=1}^{r} \lambda_i \left(z_i + \frac{h_i}{2\lambda_i}\right)^2 - \frac{h_i^2}{4\lambda_i}. \tag{10}$$

We can see that the trapping region will be determined by the equation of the ellipsoid where $\dot{K} = 0$,

$$1 = \sum_{i=1}^{r} \frac{1}{\alpha_i^2}\left(z_i + \frac{h_i}{2\lambda_i}\right)^2, \tag{11}$$

$$\alpha_i = \frac{1}{2}\sqrt{\frac{1}{\lambda_i}\sum_{j=1}^{r}\frac{h_j^2}{\lambda_j}} \leq \frac{1}{2|\lambda_1|}\|\boldsymbol{d}_m\|. \tag{12}$$

The origin at $\boldsymbol{y} = 0$ ($\boldsymbol{a} = \boldsymbol{m}$) lies on the ellipsoid, and in the worst case scenario lies at the tip of the major axis. Thus, to guarantee that a ball centered at this origin entirely contains this region, we estimate R_m as twice the size of the largest possible value of the half-axes α_i. Note that our definition of α_i differs from Schlegel and Noack [10]; we believe that there is a minor typo in their Eq. 3.14. Fortunately, the only consequence is a change in the estimate of R_m. Lastly, recall that long-term bounded (not necessarily exhibiting a monotonically trapping region) and effectively nonlinear systems only guarantee an $\boldsymbol{m}$ exists such that $\boldsymbol{A}^S$ is negative semidefinite. In the case of mixed zero and nonzero eigenvalues, the ellipsoid becomes a paraboloid. The infinite extent of the paraboloid precludes a monotonic trapping region but not other forms of global stability. This edge case is not further discussed because in practice (numerically) there is no chance of arriving at an eigenvalue of exactly zero.

C. Interpretation of the trapping theorem

The Schlegel and Noack [10] theorem, summarized in Theorem 1, provides necessary and sufficient conditions for the projected ROM in Eq. (3) to be globally stable by admitting a trapping region. This theorem is necessary and sufficient for systems that exhibit effective nonlinearity, i.e., the system does not manifest invariant manifolds where there exists some i such that $Q_{ijk}a_j a_k = 0$ for all time, for which a linear stability analysis must be adopted. In other words, systems that start in purely linear model subspaces, and remain in those subspaces, do not exhibit effective nonlinearity. Fortunately, realistic fluid flows exhibit effective nonlinearity, although there are some subtleties that we discuss in Section II D. In this case, we can always use the total fluid kinetic energy K as a Lyapunov function for the trapping region. This is ideal, as finding a suitable Lyapunov function is often the most difficult task in stability analysis. It is possible that other Lyapunov functions exist with tighter bounds on the size of a trapping region, but this paper is primarily concerned with promoting models with long-term boundedness (i.e. models that exhibit a trapping region of any kind), rather than an algorithm for precisely capturing the shape and size of a trapping region. For a post-fit algorithm to find the optimal ellipsoidal estimate of the stability domain, see Kramer [81].

A generic nonlinear system may exhibit multiple fixed points, limit cycles, and other equilibrium point behavior. However, any physical system should produce bounded trajectories, and the global stability property from the trapping

[*] If a system is long-term bounded (not necessarily exhibiting a monotonically trapping region) and effectively nonlinear, only the existence of an $\boldsymbol{m}$ producing a negative *semi*definite $\boldsymbol{A}^S$ is guaranteed.

theorem is agnostic to any *local* stability properties. This manuscript solely considers systems that are globally stable, or equivalently, long-term (ultimately) bounded, by virtue of exhibiting globally trapping regions. Long-term boundedness means that there exists some T_0 and R_0 such that $\|\boldsymbol{a}(t)\| < R_0$ for all $t > T_0$. A trapping region encompasses an attractor or attracting set, which is typically defined as a set of the system phase space that many trajectories converge towards; this can be an equilibrium point, periodic trajectory, Lorenz's "strange attractor", or some other chaotic trajectory. Whenever it does not detract from the discussion, we omit the qualifiers "globally", "monotonically" and "long-term", as this is the only characterization of stability considered in the present work. Examples of physical systems that are globally stable but do not exhibit a trapping region include Hamiltonian systems and systems that do not fit into the trapping theorem assumptions (examined further in Section II D and summarized in Fig. 1). For instance, fully energy-preserving systems satisfy $\dot{K} = 0$, so trajectories represent shells of constant distance from the origin; these trajectories are globally bounded but no trapping region exists.

D. Model truncation, effective nonlinearity, and closure

Before implementing the trapping theorem into system identification, we investigate the circumstances under which truncated projection-based models will exhibit effective nonlinearity; the reader may skip this section if the subtleties of the trapping theorem are not of interest, although the discussion here is pertinent to Section IV E. Effectively nonlinear dynamics are ideal because they can be decisively classified as globally stable or not, requiring no additional stability analysis. To proceed, consider a Fourier-Galerkin model of Burgers' equation derived from the Fourier expansion $u(x,t) = \sum a_k(t)e^{ikx}$, and further examined in Section IV E,

$$\dot{u}=-u\partial_x u+\nu\partial_{xx}u \quad \Longrightarrow \quad \dot{a}_k=-\nu k^2 a_k-\sum_{\ell=-\infty}^{\infty} i\ell a_\ell a_{k-\ell} \tag{13}$$

$$\Longrightarrow \quad \dot{K}=-\nu\sum_{k=-\infty}^{\infty} k^2 a_k^2-\sum_{k,\ell=-\infty}^{\infty} i\ell a_\ell a_{k-\ell} a_k. \tag{14}$$

The particular "triadic" structure of the nonlinear term in the spectral domain, where the only nonzero terms acting on a_k are those whose wavenumbers sum to k, is identical to that arising in isotropic turbulence [82]. The triadic term in $\dot{K}$ transfers energy between length scales. Since the viscous term scales with k^2, energy is most effectively dissipated at the smallest scales; the combination of the two terms leads to the traditional energy cascade in which energy flows "downhill" from larger to smaller scales. This description implies that heavily truncating the Galerkin system leads to underresolving the dissipation rate and a closure scheme may be required to re-introduce the full dissipation. Towards this goal, modern sparse regression and deep learning methods have been used to produce new closures for fluid models [25, 83–87]. While the traditional explanation for unstable Galerkin models derives from these truncated dissipative scales, increasingly there are alternate explanations including fundamental numerical issues with the Galerkin framework (potentially resolved in a Petrov-Galerkin framework) [88] and the Kolmogorov width issues of linear subspaces more generally [32]. If true, this is probably good news for (incompressible, dissipationless) Hall-MHD, where the conservation of energy and the cross, magnetic, and generalized helicities leads to direct, inverse, and even bidirectional cascades [89]. Interestingly, the notion of effective nonlinearity appears to be another approach from which we can attempt to resolve these disagreements about the sources of ROM instability.

To proceed with this theme, we show that the triadic structure of the model has repercussions for the presence of effective nonlinearity. Consider the truncated model

$$\dot{a}_k = -\nu k^2 a_k - \sum_{\ell=-r}^{r} i\ell a_\ell a_{k-\ell}, \qquad k \in \{1, ..., r\} \tag{15}$$

with the initial condition $a_j = 1$ for any $j \in \{\pm(\frac{r}{2}+1), \pm(\frac{r}{2}+2), ..., \pm r\}$, and $a_k = 0$, $k \neq j$. For simplicity we have assumed r is divisible by two. In this case the system has r invariant 1D subspaces for which

$$\dot{a}_j = -\nu j^2 a_j. \tag{16}$$

These invariant linear subspaces exist because higher wavenumber modes that *could* interact to transfer energy between coefficients have been discarded. In other words, *Fourier-Galerkin models with finite truncation do not exhibit effective nonlinearity*. In contrast, POD-Galerkin models weakly break the triadic structure of the nonlinearity [90], and therefore in general will weakly satisfy the trapping theorem criteria for effective nonlinearity, to the extent that they differ from the Fourier modes because of inhomogeneity in the system. There are also modern ROMs which attempt to retain the full dissipation by utilizing bases that intentionally mix length scales [91] – these types of models should be more likely to satisfy effective nonlinearity. Lastly, numerical errors appear to weakly restore effective nonlinearity, since errors break any triadic structure. Proceeding with this analysis is complicated because the numerical errors also weakly break our foundational assumption that Q_{ijk} is energy-preserving. Future investigations should be pursued to explore relationships between effective nonlinearity, the energy cascade, and closure models that reintroduce stabilizing dissipation to truncated models.

It is difficult to quantify "how close" a model is to exhibiting effective nonlinearity, since a lack of effective nonlinearity $Q_{ijk}a_j a_k = 0$ must hold for all time, for any i, and for any valid system trajectory. However, for an orthonormal set of temporal modes, and assuming there exists at least one index i such that $Q_{ijj} \neq 0$, we propose quantifying the average strength of model effective nonlinearity through the metric

$$S_e = \frac{\min_i |Q_{ijk}\overline{a_j a_k}|}{\max_i |Q_{ijk}\overline{a_j a_k}|} = \frac{\min_i |Q_{ijj}|}{\max_i |Q_{ijj}|}. \tag{17}$$

The bar in $\overline{a_j a_k}$ denotes a temporal average. We will show

in Section IV E that in system identification a lack of effective nonlinearity is not a terrible loss. Our trapping SINDy algorithm in Section III minimizes $\dot{K}$ whether or not a negative definite $\boldsymbol{A}^S$ can be realized. However, without additional stability analysis, such models are no longer provably stable for any initial condition. Although Eq. (16) is a linearly stable system, this is not guaranteed for more general fluid models than the simple Burgers' equation considered here.

E. Constraints in model reduction and system identification

Before moving on to system identification, it is worth noting that enforcing these types of existence-based stability conditions is subtle. There are modern techniques to implement implicit constraints of the form

$$\mathscr{C}_i(\dot{\boldsymbol{a}}, \boldsymbol{a}, t, ...) = 0, \qquad i \in \{1, 2, ...\} \tag{18}$$

into both model reduction [48, 92] and system identification [19, 49, 93, 94]. Precisely in this way, the energy-preserving constraint in Eq. (5) is cast as an affine version of Eq. (18) in our optimization in Section III.

However, enforcing stability in quadratic energy-preserving models is more complicated than Eq. (18). To see this, note that there a few different circumstances under which we might want to promote stability. If the true $\boldsymbol{A}^S$ *and* the optimal $\boldsymbol{m}$ are known, we can simply constrain the coefficients in Eq. (3) to produce this known negative definite $\boldsymbol{A}^S$. This would imply that we already know the optimally-shifted eigenvalues of the system and an $\boldsymbol{m}$ that produces these negative eigenvalues; if this is the case, so much information about the system of ODEs is already known that machine learning methods are likely unnecessary.

But far more interesting are the cases in which 1) the underlying system is known to be globally stable and effectively nonlinear, so we want to find the "correct" $\boldsymbol{m}$ and corresponding $\boldsymbol{A}^S$, or 2) it is not known if any $\boldsymbol{m}$ exists such that $\boldsymbol{A}^S$ is negative definite. In system identification, either of these cases can be addressed by searching for a model that both optimally fits the data and is globally stable. In this context, we adopt a mixed approach in the next section where we enforce the energy-preserving constraint and then separately bias the optimization towards models with a trapping region. This technique is a significant methodological extension because we can no longer rely on constraints of the form in Eq. (18).

III. TRAPPING SINDY ALGORITHM

We now describe how to incorporate the trapping theorem of Schlegel and Noack [10] into data-driven model identification, specifically for the SINDy algorithm. Before describing the modified algorithm in Sec. III B, we first present the standard SINDy algorithm [15] along with a recent variant that incorporates explicit constraints [19, 49]. We then build on this framework to incorporate the Schlegel–Noack trapping theorem.

A. Standard and constrained SINDy algorithms

The goal of system identification is to identify a system of ODEs or PDEs that describe how a given data set evolves dynamically in time. The SINDy algorithm [15] identifies sparse, parsimonious models that remain interpretable and avoid the overfitting issues that are common in this field. As in Loiseau et al. [19], we develop SINDy models for the dynamics of $\boldsymbol{a}$, representing the coefficients or amplitudes of a modal Galerkin expansion in Eq. (2). We assume that the dynamics of $\boldsymbol{a}$ will be described as a sparse linear combination of elements from a library $\boldsymbol{\Theta}$ containing candidate terms such as:

$$\frac{d}{dt}\boldsymbol{a} \approx \boldsymbol{\Theta}(\boldsymbol{a})\boldsymbol{\Xi}, \qquad \boldsymbol{\Theta}(\boldsymbol{a}) = \begin{bmatrix} | & | & | \\ \mathbf{1} & \boldsymbol{a} & \boldsymbol{a}\otimes\boldsymbol{a} \\ | & | & | \end{bmatrix}. \tag{19}$$

Here $\boldsymbol{a}\otimes\boldsymbol{a}$ contains all combinations of $a_i a_j$ without duplicates. The $\boldsymbol{\Theta}$ matrix may contain any desired candidate terms, but in this work we consider only terms up to quadratic polynomials in $\boldsymbol{a}$ because we are searching for energy-preserving quadratic models. The expressions in Eq. (19) are typically evaluated on a data matrix $\boldsymbol{X}$ obtained from time-series data of the state, $\boldsymbol{a}(t_1), \boldsymbol{a}(t_2), ..., \boldsymbol{a}(t_M)$:

$$\boldsymbol{X} = \begin{array}{c} \xrightarrow{\qquad\qquad\text{state}\qquad\qquad} \\ \begin{bmatrix} a_1(t_1) & a_2(t_1) & \cdots & a_r(t_1) \\ a_1(t_2) & a_2(t_2) & \cdots & a_r(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ a_1(t_M) & a_2(t_M) & \cdots & a_r(t_M) \end{bmatrix} \end{array} \Bigg\downarrow \text{time}. \tag{20}$$

A matrix of derivatives in time, $\dot{\boldsymbol{X}}$, is defined similarly and can be numerically computed from $\boldsymbol{X}$. In this case, Eq. (19) becomes $\dot{\boldsymbol{X}} = \boldsymbol{\Theta}(\boldsymbol{X})\boldsymbol{\Xi}$. The goal of SINDy is to determine a sparse matrix of coefficients $\boldsymbol{\Xi} = \begin{bmatrix}\boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_r\end{bmatrix}$, also written in vectorized form as $\boldsymbol{\Xi}[:] = \boldsymbol{\xi}$,

$$\boldsymbol{\xi} = [\xi_1^{a_1}, \ldots, \xi_1^{a_r}, \xi_2^{a_1}, \ldots, \xi_2^{a_r}, \ldots, \xi_N^{a_1}, \ldots, \xi_N^{a_r}], \tag{21}$$

where N is the number of candidate functions and r is the state space size; nonzero elements in each column $\boldsymbol{\xi}_j$ indicate which terms are active in the dynamics of $\dot{a}_j(t)$. The matrix of coefficients $\boldsymbol{\Xi}$ is determined via the following sparse optimization problem:

$$\arg\min_{\boldsymbol{\xi}} \left[\frac{1}{2}\|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|^2 + \lambda\|\boldsymbol{\xi}\|_0\right]. \tag{22}$$

We deviate from the typical SINDy definitions by explicitly formulating the problem in terms of the vectorized $\boldsymbol{\xi} \in \mathbb{R}^{rN}$, $\boldsymbol{\Theta}(\boldsymbol{X}) \in \mathbb{R}^{rM\times rN}$, and $\dot{\boldsymbol{X}} \in \mathbb{R}^{rM}$. The first term in the SINDy optimization problem in Eq. (22) fits a system of ODEs $\boldsymbol{\Theta}\boldsymbol{\xi}$ to the given data in $\dot{\boldsymbol{X}}$. The $\|\boldsymbol{\xi}\|_0$ term counts the number of nonzero elements of $\boldsymbol{\Xi}$, and the elements of $\boldsymbol{\Xi}$ smaller than a threshold value, λ, are zeroed out. However, it is not technically a norm and leads to a non-convex optimization, so sev-

eral convex relaxations have been proposed [15, 17, 49].

Since the original SINDy algorithm, Loiseau et al. [19] introduced an extension to directly enforce constraints on the coefficients in Ξ. In particular, they enforced energy-preserving, skew-symmetry constraints on the quadratic terms for incompressible fluid flows, demonstrating improved model performance over standard Galerkin projection. The quadratic library in Eq. (19) has $N = \frac{1}{2}(r^2+3r)+1$ terms. With the energy-preserving structure, it can be shown that the number of constraints is $p = r(r+1)(r+2)/6$ and therefore the number of free parameters is $rN - p = 2p$. This constraint is encoded as $\boldsymbol{C}\boldsymbol{\xi} = \boldsymbol{d}$, $\boldsymbol{C} \in \mathbb{R}^{p\times rN}$, $\boldsymbol{d} \in \mathbb{R}^p$, and the constrained SINDy algorithm solves the following minimization,

$$\arg\min_{\boldsymbol{\xi}} \left[\frac{1}{2}\|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 + \delta_0(\boldsymbol{C}\boldsymbol{\xi} - \boldsymbol{d})\right]. \quad (23)$$

In general we can use nonconvex regularizers that promote sparsity in $\boldsymbol{\xi}$, but the trapping SINDy modifications below require a convex regularizer such as the L^1 norm. The third term δ_0 is an indicator function that encodes the constraint $\boldsymbol{C}\boldsymbol{\xi} = \boldsymbol{d}$, guaranteeing the energy-preserving structure in the quadratic nonlinearity is retained in the identified model. There are also variants of the constrained SINDy objective function in Eq. (23) that utilize sparse relaxed regularized regression (SR3) in order to improve performance [49, 95].

B. Proposed trapping SINDy algorithm

Model constraints in system identification, such as global conservation laws or other physical considerations, often result in improved models, but do not generally guarantee global stability. Here, we will additionally promote globally stable models that exhibit a monotonically trapping region. Recall from Thm. 1 that $\boldsymbol{m}$ is an arbitrary, constant vector, of the same state size as $\boldsymbol{a}$, that specifies the center of a possible trapping region. Stability promotion is then achieved by jointly determining the sparse model coefficients Ξ and state vector $\boldsymbol{m}$ such that $\boldsymbol{A}^S$ from Eq. (8) is negative definite.

To proceed with our trapping SINDy formulation, we must relate the model coefficients in $\boldsymbol{\xi}$ to the matrix $\boldsymbol{A}^S$ appearing in the trapping theorem. We first define the projection operators $\boldsymbol{P}^L \in \mathbb{R}^{r\times r\times rN}$, $\boldsymbol{P}^Q \in \mathbb{R}^{r\times r\times r\times rN}$, and $\boldsymbol{P} \in \mathbb{R}^{r\times r\times rN}$. The operator $\boldsymbol{P}^L$ projects out the symmetric part of the linear coefficients through $\boldsymbol{L}^S = \boldsymbol{P}^L\boldsymbol{\xi}$. The same is true for the quadratic coefficients, $\boldsymbol{Q} = \boldsymbol{P}^Q\boldsymbol{\xi}$. The operator $\boldsymbol{P} = \boldsymbol{P}^L - \boldsymbol{m}^T\boldsymbol{P}^Q$ provides a concise representation of $\boldsymbol{A}^S$ through the following equation:

$$\boldsymbol{A}^S = \boldsymbol{L}^S - \boldsymbol{m}^T\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{\xi} = (\boldsymbol{P}^L - \boldsymbol{m}^T\boldsymbol{P}^Q)\boldsymbol{\xi}. \quad (24)$$

We now phrase a tentative version of the trapping SINDy optimization problem, in analogy to the constrained SINDy optimization in Eq. (23), that incorporates an additional loss term to reduce the maximal (most positive) eigenvalue λ_1 of the real, symmetric matrix $\boldsymbol{A}^S$:

$$\arg\min_{\boldsymbol{\xi},\boldsymbol{m}} \left[\frac{1}{2}\|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 + \delta_0(\boldsymbol{C}\boldsymbol{\xi} - \boldsymbol{d}) + \frac{\lambda_1}{\eta}\right]. \quad (25)$$

Note that we have introduced a new hyperparameter η, which modulates the strength of the λ_1 loss term. Although λ_1 is a convex function of the matrix elements [96], $\boldsymbol{A}^S = (\boldsymbol{P}^L - \boldsymbol{m}^T\boldsymbol{P}^Q)\boldsymbol{\xi}$ is not affine in $\boldsymbol{\xi}' = [\boldsymbol{\xi}, \boldsymbol{m}]$. The result is that this new term is not convex, but *convex composite*. It is possible to approximately solve this problem with a variable projection technique, where we essentially treat $\boldsymbol{\xi}$ and $\boldsymbol{m}$ as independent, solve the convex problem in $\boldsymbol{\xi}$, and then substitute $\boldsymbol{\xi}^*$, the solution at each iteration, into the optimization for $\boldsymbol{m}$. In practice this algorithm performs fairly well, although the convergence properties are unclear. Eq. (25) is also amenable to other approaches, such as Gauss-Newton [97] or the prox-linear algorithm [98], because λ_1 is a convex function of the elements of $\boldsymbol{A}^S$ and $\boldsymbol{P}\boldsymbol{\xi}$ is smooth in $\boldsymbol{m}$ and $\boldsymbol{\xi}$. Although we institute a modified algorithm below, these convex-composite approaches are a promising future direction for effectively solving this nonconvex optimization problem.

In order to produce an algorithm with better performance and better understood convergence properties, we adopt a relax-and-split approach [99], similar to the approach taken in Champion et al. [49]. We introduce an auxiliary variable $\boldsymbol{A}$ that represents the projection of $\boldsymbol{A}^S = \boldsymbol{P}\boldsymbol{\xi}$ onto the space of negative definite matrices, and introduce two new terms in the optimization:

$$\arg\min_{\boldsymbol{\xi},\boldsymbol{m},\boldsymbol{A}} \left[\frac{1}{2}\|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 + \delta_0(\boldsymbol{C}\boldsymbol{\xi} - \boldsymbol{d}) + \frac{1}{2\eta}\|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2 + \delta_{\mathscr{I}}(\boldsymbol{\Lambda})\right]. \quad (26)$$

The new least-squares term enforces a "soft" constraint (or bias) towards $\boldsymbol{A}^S = \boldsymbol{P}\boldsymbol{\xi}$ being negative definite by minimizing the difference between $\boldsymbol{P}\boldsymbol{\xi}$ and its projection into the space of negative definite matrices. The auxiliary variable $\boldsymbol{A}$ is updated to approximate $\boldsymbol{A}^S = \boldsymbol{P}\boldsymbol{\xi}$, and then, through the $\delta_{\mathscr{I}}$ term, enforced to be negative definite by requiring that the diagonalized matrix $\boldsymbol{\Lambda} = \boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{V}$ lies in $\mathscr{I} = (-\infty, -\gamma]$, $\gamma > 0$. Directly enforcing $\boldsymbol{P}\boldsymbol{\xi}$ to be negative definite tends to badly distort the model fit to the data. Instead, the auxiliary variable $\boldsymbol{A}$ in Eq. (26) allows the algorithm to accurately fit the data with $\boldsymbol{\xi}$ and then relax the coefficients towards a negative definite $\boldsymbol{A}^S$ to promote global stability.

This flexible formulation also allows $\boldsymbol{A}$, our proxy for the projection of $\boldsymbol{P}\boldsymbol{\xi}$ onto the space of negative definite matrices, to vary, and therefore fit the particular eigenvalues of the system in question. In other words, the proposed approach pushes $\boldsymbol{A}^S$ into the space of negative definite matrices in $\mathbb{R}^{r\times r}$ with minimal assumptions about the eigenvalues, only assuming that they are negative. Contrast our algorithm to a more restrictive approach that prescribes an $\boldsymbol{A}$, meaning we already know a set of negative eigenvalues of $\boldsymbol{P}\boldsymbol{\xi}$ that is compatible with the data. A description of each of the hyperparameters λ, η, and γ, is provided in Table I. Note that Eq. (26) is not

convex in $\boldsymbol{A}$, and this is the most challenging aspect of this formalism.

Now that we have defined our problem in Eq. (26), we need to solve it. If we denote the convex part of the optimization,

$$F(\boldsymbol{\xi}, \boldsymbol{m}, \boldsymbol{A}) = \|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2/2 + \lambda\|\boldsymbol{\xi}\|_1 + \delta_0(\boldsymbol{C}\boldsymbol{\xi} - \boldsymbol{d}) + \|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2/2\eta, \tag{27}$$

and fix initial guesses for $\boldsymbol{m}$ and $\boldsymbol{A}$, then we can define the solution vector $\boldsymbol{\xi}^*$ through

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \left[F(\boldsymbol{\xi}, \boldsymbol{m}, \boldsymbol{A})\right]. \tag{28}$$

If $\lambda = 0$, $\boldsymbol{\xi}^*$ is structurally identical to the $\boldsymbol{\xi}^*$ in Champion et al. [49]:

$$\boldsymbol{H} = (\boldsymbol{\Theta}^T\boldsymbol{\Theta} + \frac{1}{\eta}\boldsymbol{P}^T\boldsymbol{P})^{-1}, \tag{29}$$

$$\boldsymbol{\xi}^* = \boldsymbol{H}\left[\boldsymbol{I} - \boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{H}\boldsymbol{C}^T)^{-1}\boldsymbol{C}\boldsymbol{H}\right]\left[\boldsymbol{\Theta}^T\dot{\boldsymbol{X}} + \frac{1}{\eta}\boldsymbol{P}^T\boldsymbol{A}\right] + \boldsymbol{H}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{H}\boldsymbol{C}^T)^{-1}\boldsymbol{d}. \tag{30}$$

$\boldsymbol{H}$ is positive definite, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{C}\boldsymbol{\xi}^* = \boldsymbol{d}$ can be verified using Eq. (30). The minimization over $\boldsymbol{\xi}$ with $\lambda \neq 0$ is still convex but not analytically tractable as in Eq. (30). Since it is convex, it can be solved with standard convex optimization libraries such as CVXPY [100]. It can also be shown to reduce to a constrained quadratic program over the unit box with a positive semidefinite cost matrix. A barrier to this route is that typical numerical solvers either assume that the quadratic cost matrix is sparse or positive definite. Neither assumption is true here.

Now that we have solved the minimization over $\boldsymbol{\xi}$, we can use prox-gradient descent on $(\boldsymbol{m}, \boldsymbol{A})$; each algorithm iteration we alternate between solving for $\boldsymbol{\xi}^*$ and solving for $(\boldsymbol{m}^*, \boldsymbol{A}^*)$. Again, we can think about this problem as a variable projection [101, 102], which is a value function optimization over the remaining variables $(\boldsymbol{m}, \boldsymbol{A})$. To make this viewpoint more precise, we define

$$\widetilde{F}(\boldsymbol{m}, \boldsymbol{A}) = F(\boldsymbol{\xi}^*, \boldsymbol{m}, \boldsymbol{A}), \tag{31}$$

The problem we want to solve is now written more simply as

$$\arg\min_{\boldsymbol{m}, \boldsymbol{A}} \left[\widetilde{F}(\boldsymbol{m}, \boldsymbol{A}) + \delta_{\mathscr{I}}(\boldsymbol{\Lambda})\right].$$

We apply prox-gradient descent to this nonconvex problem, so that

$$\boldsymbol{m}^* = \boldsymbol{m} - \alpha_m \nabla_{\boldsymbol{m}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A}), \tag{32}$$

$$\boldsymbol{A}^* = \text{proj}_{\mathscr{I}}\left[\boldsymbol{A} - \alpha_A \nabla_{\boldsymbol{A}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A})\right], \tag{33}$$

where α_m and α_A are step sizes. All that remains is to compute the gradients of the value function $\widetilde{F}$,

$$\nabla_{\boldsymbol{A}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A}) = \frac{1}{\eta}(\boldsymbol{A} - \boldsymbol{P}\boldsymbol{\xi}^*), \tag{34}$$

$$\nabla_{\boldsymbol{m}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A}) = \frac{1}{\eta}\boldsymbol{P}^Q\boldsymbol{\xi}^*(\boldsymbol{A} - \boldsymbol{P}\boldsymbol{\xi}^*). \tag{35}$$

These are Lipschitz continuous functions with Lipschitz constants L_A, L_m satisfying

$$\alpha_A \leq \frac{1}{L_A} \leq \eta, \tag{36}$$

$$\alpha_m \leq \frac{1}{L_m} \leq \frac{\eta}{\|(\boldsymbol{P}^Q\boldsymbol{\xi}^*)_{ijk}(\boldsymbol{P}^Q\boldsymbol{\xi}^*)_{ljk}\|_{\text{F}}}, \tag{37}$$

in order for guaranteed convergence of fixed step-size, prox-gradient descent [103]. While the denominator in Eq. (37) varies with the update in $\boldsymbol{\xi}$, in practice, one can reduce α_m until convergence is found. The full trapping SINDy optimization is illustrated in Algorithm 1.

Algorithm 1 Trapping SINDy

Input: Numerical data $\dot{\boldsymbol{X}}$ and optional initial guesses for $\boldsymbol{m}$ and $\boldsymbol{A}$.
Output: Optimal model coefficients $\boldsymbol{\xi}^*$ and shift vector $\boldsymbol{m}^*$.

1: **procedure** SINDy($\dot{\boldsymbol{X}}$, λ, η, γ)
2: Construct $\boldsymbol{\Theta}(\boldsymbol{X})$, $\boldsymbol{P}$, $\boldsymbol{C}$, and $\boldsymbol{d}$.
3: **while** $|\boldsymbol{\xi}_k - \boldsymbol{\xi}_{k+1}| > \varepsilon_{\text{tol}}^{\xi}$ and $|\boldsymbol{m}_k - \boldsymbol{m}_{k+1}| > \varepsilon_{\text{tol}}^{m}$
4: $\boldsymbol{\xi}_{k+1} \Longleftarrow \arg\min_{\boldsymbol{\xi}_k}\left[F(\boldsymbol{\xi}_k, \boldsymbol{m}_k, \boldsymbol{A}_k)\right]$,
5: $\boldsymbol{V}_{k+1}\boldsymbol{\Lambda}_{k+1}(\boldsymbol{V}_{k+1})^{-1} \Longleftarrow \boldsymbol{A}_k - \alpha_A \nabla_{\boldsymbol{A}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A})|_{\boldsymbol{m}_k, \boldsymbol{A}_k}$,
6: $\boldsymbol{A}_{k+1} \Longleftarrow \boldsymbol{V}_{k+1}\text{proj}_{\mathscr{I}}\left[\boldsymbol{\Lambda}_{k+1}\right](\boldsymbol{V}_{k+1})^{-1}$,
7: $\boldsymbol{m}_{k+1} \Longleftarrow \boldsymbol{m}_k - \alpha_m \nabla_{\boldsymbol{m}} \widetilde{F}(\boldsymbol{m}, \boldsymbol{A})|_{\boldsymbol{m}_k, \boldsymbol{A}_k}$,
8: **end procedure**

In words, 2: initialize variables, 3: start iteration loop, 4: convex minimization for $\boldsymbol{\xi}_{k+1}$, 5: prox-gradient step for $\boldsymbol{A}_{k+1}$, 6: project $\boldsymbol{A}_{k+1}$ into $\mathscr{I}$, rotate into $\boldsymbol{P}\boldsymbol{\xi}_{k+1}$ basis, and 7: prox-gradient step for $\boldsymbol{m}_{k+1}$. Note that inequalities (36)–(37) should be satisfied, and there tends to be a sweet spot for η. It is often useful to start with $\eta \gg 1$ and then reduce η until the model coefficients are significantly affected.

$\varepsilon_{\text{tol}}^{\xi}$ and $\varepsilon_{\text{tol}}^{m}$ are convergence tolerances. The $\boldsymbol{V}_{k+1}$ are the eigenvectors of $\boldsymbol{P}\boldsymbol{\xi}_{k+1}$ and are used to transform $\boldsymbol{A}_{k+1}$ into the same basis as $\boldsymbol{P}\boldsymbol{\xi}_{k+1}$. An example of the algorithm iterating on noisy data from the chaotic Lorenz system is shown in Fig. 2, demonstrating how the algorithm transitions from a poor initial guess that decays to a fixed point to a stable model converging to the correct attractor. We also implement an optional FISTA method [104, 105] for reducing the convergence time in the $(\boldsymbol{m}, \boldsymbol{A})$ optimization. Algorithm 1 is computationally intensive, but it can be parallelized for speed in future work, following other SINDy variants [106]. Initial guesses are allowed for $\boldsymbol{m}$ and $\boldsymbol{A}$ in order to facilitate continuation of previous optimization runs. Along with these methods, we also implement the λ_1 variant of the trapping algorithm in Eq. (25) in the open-source PySINDy code [60].

A key insight to the trapping algorithm is that the energy-preserving constraint $\boldsymbol{C}\boldsymbol{\xi} = \boldsymbol{d}$ is non-negotiable. Although in practice small errors in $\boldsymbol{C}\boldsymbol{\xi} = \boldsymbol{d}$ do not significantly affect the

Trapping SINDy hyperparameters	
λ	Specifies the strength of sparsity-promotion through the regularizer $R(\boldsymbol{\xi})$. $\lambda = 0$ already works well for low-dimensional systems because the $\|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2$ term promotes stability.
η	Specifies how strongly to push the algorithm towards models with negative definite $\boldsymbol{A}^S$. If $\eta \gg 1$, $\boldsymbol{\xi}^*$ is unaffected by the minimization over $\boldsymbol{m}$. If $\eta \ll 1$, the problem is increasingly nonconvex.
γ	Determines how far to push the eigenvalues of $\boldsymbol{A}^S$ towards being negative definite. Typically $\gamma \lesssim 0.1$ works for a variety of problems regardless of the true eigenvalues of $\boldsymbol{A}^S$.

TABLE I: Description of the trapping SINDy hyperparameters.

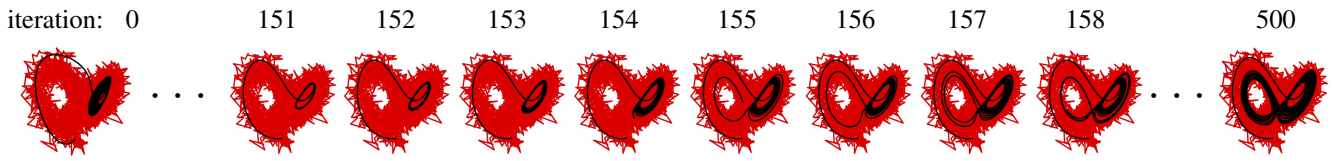


FIG. 2: Illustration of trapping SINDy progress on noisy Lorenz data. The minimization results in the transition from a poor initial guess to identification of the correct attractor dynamics.

optimization problem, the $\|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2$ term in the optimization loses its physical interpretation if the coefficients are not exactly energy-preserving. Thus, the goal is to satisfy $\boldsymbol{C}\boldsymbol{\xi} = \boldsymbol{d}$ *exactly*, and then to push a potential model towards a more refined model that exhibits a trapping region, potentially at the expense of the fit to the data (this can also mitigate overfitting).

With regards to choosing hyperparameters, there is some work to do for each new problem. Fortunately, the results in Sec. IV are fairly insensitive to the *precise* hyperparameter values. Rather, we rediscover a common occurrence in sparse regression techniques – the existence of cutoffs in the hyperparameter space where the model quality sharply drops. These sharp boundaries are actually grounded in reality; if sparsity-promotion is increased in the regression, it will eventually start to truncate out the smallest physical scales in the system, and at "large enough" values, it truncates the primary dynamics of interest!

More specifically for our Algorithm 1, if the system has some small scales, poor choices of λ (sparsity-promotion) or γ (smallest eigenvalue of $\boldsymbol{A}^S$) can truncate these scales during the regression. A reasonable strategy, assuming no prior system knowledge, is to start with $\lambda = 0$, $\gamma \sim 0$, $\eta \gg 1$, and then scan the values. For η (the strength of the long-term boundedness term), there tends to be a "sweet spot" regime. If $\eta^{-1}\|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2 \ll \|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2$, then $\boldsymbol{\xi}^*$ is essentially unaffected by the minimizations over $\boldsymbol{m}$ and $\boldsymbol{A}$. In practice, this means that poor initial guesses for $\boldsymbol{\xi}^*$ do not improve as the full optimization problem is solved. In the opposite extreme, $\eta^{-1}\|\boldsymbol{P}\boldsymbol{\xi} - \boldsymbol{A}\|_2^2 \gg \|\boldsymbol{\Theta}\boldsymbol{\xi} - \dot{\boldsymbol{X}}\|_2^2$, the optimization in Eq. (26) is increasingly nonconvex and potentially pulls $\boldsymbol{\xi}^*$ far away from the data. Finding the η regime where updating $\boldsymbol{m}$ perturbs $\boldsymbol{\xi}^*$, instead of leaving $\boldsymbol{\xi}^*$ unaffected or mangled, requires scanning η. Because each problem requires some data-craftsmanship, there are plans to parallelize the algorithm to efficiently scan large ranges in the hyperparameters. Finally, prior system knowledge can also constrain the hyperparameter space.

IV. RESULTS

We now investigate the utility of our trapping SINDy algorithm to identify stable, sparse, nonlinear models for a number of canonical systems. These examples illustrate that it is possible to both effectively discover stable models that exhibit trapping regions and improve the discovery of systems that do not satisfy Thm. 1 or the requirement of effective nonlinearity. For each system, we train SINDy on a single trajectory with a random initial condition and evaluate the model on a different trajectory of the same temporal duration with a new random initial condition. It is difficult to quantity model performance for chaotic systems, such as the Lorenz system, where lobe-switching is extremely sensitive to initial conditions and the coefficients of the identified model, and for systems with transients, for which the precise timing of instability must be matched to achieve the correct phase. Two reasonable definitions for the model quality are the relative Frobenius error in the model coefficients (for models with closed forms) and the time-averaged error in $\dot{\boldsymbol{X}}$,

$$\mathrm{E_m} = \frac{\|\boldsymbol{\Xi}_{\mathrm{True}} - \boldsymbol{\Xi}_{\mathrm{SINDy}}\|_{\mathrm{F}}}{\|\boldsymbol{\Xi}_{\mathrm{True}}\|_{\mathrm{F}}}, \tag{38}$$

$$\mathrm{E_f} = \frac{\overline{\|\dot{\boldsymbol{X}}_{\mathrm{True}} - \dot{\boldsymbol{X}}_{\mathrm{SINDy}}\|^2}}{\|\dot{\boldsymbol{X}}_{\mathrm{True}}\|^2}. \tag{39}$$

It should be understood that the time-average in $\mathrm{E_f}$ is computed after dividing the numerator and denominator. When appropriate, we also report a far more demanding relative prediction error,

$$\mathrm{E_{pred}} = \frac{\|\boldsymbol{X}_{\mathrm{True}} - \boldsymbol{X}_{\mathrm{SINDy}}\|_F}{\|\boldsymbol{X}_{\mathrm{True}}\|_F}. \tag{40}$$

Table II summarizes the sampling, hyperparameters, and identified trapping regions for each example discussed in Sections IV A–IV F. Table II is intended to be instructive rather than exhaustive. For clarity, the training and testing trajecto-

ries used to generate this table do not have added noise, although Fourier modes from the Burgers' Equation and POD modes from the Von Kàrmàn street are obtained from direct numerical simulation (DNS), and subsequently contain minor numerical noise; the performance on noisy data will be explored further in Sec. IV C. To compare trapping region sizes R_m across different examples, we also report $R_{\text{eff}} = R_m/\sqrt{\sum_{i=1}^{r} \bar{y}_i^2}$, which is normalized to the approximate radius of the training data. The denominator denotes the root-mean-square of the temporal average of each component of the trajectory.

A. Mean field model

Often the trajectories of a nonlinear dynamical system, which has a linear part exhibiting some stable directions, will approach a slow manifold of reduced dimension with respect to the full state space. As an example of this behavior, consider the following linear-quadratic system originally proposed by Noack et al. [1] as a simplified model of the von Kàrmàn vortex shedding problem explored further in Sec. IV F:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \mu & -1 & 0 \\ 1 & \mu & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} -xz \\ -yz \\ x^2+y^2 \end{bmatrix}. \tag{41}$$

Systems of this form commonly arise in PDEs with a pair of unstable eigenmodes represented by x and y. The third variable z models the effects of mean-field deformations due to nonlinear self-interactions of the instability modes. The system undergoes a supercritical Hopf bifurcation at $\mu = 0$; for $\mu \ll 1$ trajectories quickly approach the parabolic manifold defined by $z = x^2 + y^2$. All solutions asymptotically approach a stable limit cycle on which $z = x^2 + y^2 = \mu$. It is enough to notice that $\boldsymbol{m} = [0, 0, \mu + \varepsilon]$, $\varepsilon > 0$ produces

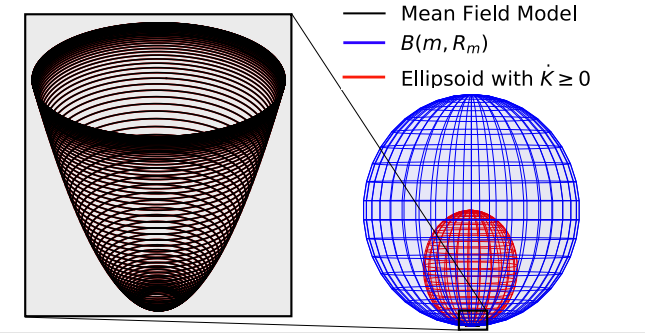
$$\boldsymbol{A}^S = \boldsymbol{L}^S - \boldsymbol{m}^T\boldsymbol{Q} = \begin{bmatrix} -\varepsilon & 0 & 0 \\ 0 & -\varepsilon & 0 \\ 0 & 0 & -1 \end{bmatrix}, \tag{42}$$

so this system exhibits a trapping region. We illustrate a stable and accurate model identified by our trapping SINDy algorithm in Fig. 3a.

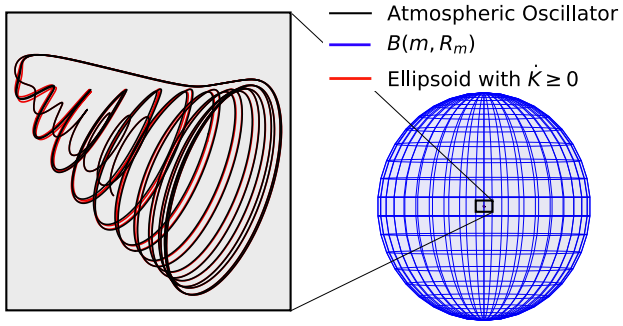
This system is of particular interest because it is a prototypical example of how quadratic interactions in a multi-scale system can give rise to effective higher-order nonlinearities. If the dynamics are restricted to the slow manifold, the system reduces to the cubic Hopf normal form [1, 107]

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mu - (x^2+y^2) & -1 \\ 1 & \mu - (x^2+y^2) \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{43}$$

Systems of this type arise in weakly nonlinear pattern-forming systems and are often called Stuart-Landau equations. In this case, the nonlinear interactions are no longer energy-preserving, since the manifold restriction removes the fast, dissipative degree of freedom. We might intuitively expect that this type of manifold reduction would inherit the trapping properties of the underlying system, but to our knowledge a general theory of such situations has not yet been worked out, even for the quadratic energy-preserving case.



(a) Trapping SINDy model (black) of a mean field system trajectory (red) with $\mu = 0.01$ and initial condition $[\mu, \mu, 0]$. The trajectory is shown within the estimated trapping region and ellipsoid where $\dot{K} \geq 0$. The prediction error is $\text{E}_{\text{pred}} \approx 0.6\%$.



(b) Same illustration for the atmospheric oscillator with random initial condition chosen from the unit ball. There is large scale separation in this system, so that $|\lambda_1| \ll |\lambda_2|, |\lambda_3|$. This leads to an overestimate of the trapping region size. The prediction error is $\text{E}_{\text{pred}} \approx 6\%$.

FIG. 3: Identified models and trapping regions for the mean field and atmospheric oscillator systems.

B. Atmospheric oscillator model

Here we examine a more complicated Lorenz-like system of coupled oscillators that is motivated from atmospheric dynamics:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & \omega \\ 0 & -\omega & \mu_2 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \sigma xy \\ \kappa yz + \beta z^2 - \sigma x^2 \\ -\kappa y^2 - \beta yz \end{bmatrix}. \tag{44}$$

For comparison, we use the parameters in Tuwankotta et al. [108], $\mu_1 = 0.05$, $\mu_2 = -0.01$, $\omega = 3$, $\sigma = 1.1$, $\kappa = -2$, and $\beta = -6$, for which a limit cycle is known to exist. The trapping SINDy algorithm finds $\boldsymbol{m}$ such that $\boldsymbol{A}^S$ is negative

Dynamic system	r	Δt	M	λ	η	γ	$\boldsymbol{m}^*$	R_m	R_{eff}	λ_1	E_m	E_f
Mean field	3	0.01	50K	0	10^{10}	1	$[0, 0, 1.3]$	1.3	218	-1	10^{-5}	10^{-12}
Atmospheric oscillator	3	0.005	50K	0	10^{8}	0.1	$[0, -0.9, 0.4]$	300	597	-0.01	10^{-4}	10^{-7}
Lorenz attractor	3	0.005	50K	0	0.1	1	$[-1.2, 0.1, 38]$	106	4.4	-1	10^{-3}	10^{-5}
Triadic MHD	6	0.001	50K	0	10^{3}	0.1	$[0, ..., 0]$	–	–	0	10^{-6}	10^{-10}
Burgers' Equation	10	0.1	30K	0	500	0.1	$[-0.2, 0, ...]$	–	–	0.1	–	10^{-3}
Cylinder wake	5	0.1	30K	0.1	1	0.1	$[-1.2, ..., 1.1]$	29	17	-0.1	–	10^{-3}

TABLE II: Description of the sampling, trapping SINDy hyperparameters, and identified trapping region for the dynamic systems examined in Section IV. Trajectory data does not include any added noise so $\lambda = 0$ works for most of the systems. The SINDy models are identified from a single trajectory. These parameters produce reasonable results for these systems, but a hyperparameter scan can lead to further improvements. The errors in the last two columns are approximate up to $\mathcal{O}(1)$ factors.

definite for a wide range of parameter and hyperparameter choices, and accurate model results are illustrated in Fig. 3b alongside the mean-field model results.

So far, we have illustrated that the trapping algorithm successfully produces accurate and provably stable models on simple systems that exhibit well-behaved attractors. In the next sections, we investigate progressively noisier (Section IV C) and higher-dimensional (Sections IV D–IV F) systems that typically provide significant challenges for model discovery algorithms.

C. Noisy Lorenz attractor

The Lorenz 1963 system [109] is among the simplest systems exhibiting chaotic dynamics, developed to model thermal convection in the atmosphere based on computer simulations from his graduate students Ellen Fetter and Margaret Hamilton:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ -xz \\ xy \end{bmatrix}. \tag{45}$$

For this system, it is possible to write $\boldsymbol{A}^S$ explicitly as

$$\boldsymbol{A}^S = \begin{bmatrix} -\sigma & \frac{1}{2}(\rho + \sigma - m_3) & \frac{1}{2}m_2 \\ \frac{1}{2}(\rho + \sigma - m_3) & -1 & 0 \\ \frac{1}{2}m_2 & 0 & -\beta \end{bmatrix}. \tag{46}$$

For Lorenz's choice of parameters, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, this system is known to exhibit a stable attractor. For $\boldsymbol{m} = [0, m_2, \rho + \sigma]$ (m_1 does not contribute to $\boldsymbol{A}^S$ so we set it to zero),

$$\boldsymbol{A}^S = \begin{bmatrix} -\sigma & 0 & \frac{1}{2}m_2 \\ 0 & -1 & 0 \\ \frac{1}{2}m_2 & 0 & -\beta \end{bmatrix}, \tag{47}$$

$$\lambda_1 = -1, \quad \lambda_\pm = -\frac{1}{2}\left[\beta + \sigma \mp \sqrt{m_2^2 + (\beta - \sigma)^2}\right],$$

so that if $\lambda_\pm < 0$, then $-2\sqrt{\sigma\beta} < m_2 < 2\sqrt{\sigma\beta}$. Our algorithm successfully identifies the optimal $\boldsymbol{m}$, and identifies the inequality bounds on m_2 for stability. As this analysis is invariant to m_1, in principle the trapping region is given by a cylinder, extruded in the m_1 direction, rather than a sphere.

We can show further improvements in model quality. We train unconstrained, constrained, and trapping SINDy models four times; the data for each is a single Lorenz attractor with four different noise instantiations. Then we test the performance of the resulting models with a random initial condition in $[-10, 10] \times [-10, 10] \times [-10, 10]$. For direct comparison, we use the L^1 regularizer for each method. Fig. 4 illustrates the increased performance with our trapping SINDy algorithm over the constrained SINDy algorithm on noisy Lorenz data for varying threshold levels $\lambda = \{0, 0.01, 0.1\}$. The unconstrained method is not pictured because most of the identified models diverge at these high noise levels. At all values of λ and most initial conditions, the unconstrained method overfits to the data and produces unstable and diverging models. The traditional constrained SINDy variant mostly manages to produce stable models but produces increasingly poor data fits as λ increases. In contrast, the trapping version continues to produce stable models that lie on the correct attractor. In this way, the additional optimization loss terms that promote stable models provide both a trapping region of known size and additional robustness to noise, even when the models appear otherwise stable, as with many of the constrained SINDy models that incorrectly decay to a fixed point.

D. Triadic MHD model

Magnetohydrodynamic systems exhibit quadratic nonlinearities that are often energy-preserving with typical boundary conditions. We consider a simple model of the nonlinearity in two-dimensional incompressible MHD, which can be obtained from Fourier-Galerkin projection of the governing equations onto a single triad of wave vectors. For the Fourier wave vectors $\boldsymbol{k}_1 = (1, 1)$, $\boldsymbol{k}_2 = (2, -1)$, and $\boldsymbol{k}_3 = (3, 0)$ and no background magnetic field, the Carbone and Veltri [110]

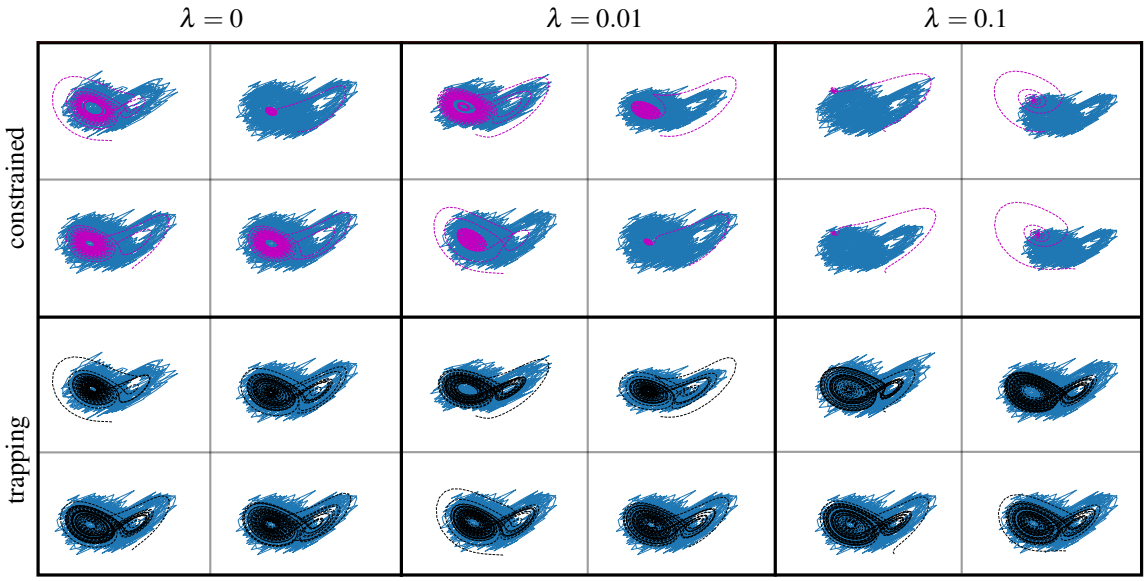


FIG. 4: Comparison between the constrained SINDy (magenta) and trapping SINDy (black) results for the Lorenz system using three different values of the sparsity-promotion strength λ. Unconstrained SINDy results are not pictured because most of the models diverge. Each model is trained on a single Lorenz attractor with noise sampled from $\mathcal{N}(0,4)$ and an initial condition of $[1,-1,20]$ (blue). The illustrations depict the model performance on data evolved from four random initial conditions between $[-10,10]$ (this testing data is not shown but the attracting set is unchanged). Trapping SINDy produces stable models that follow the underlying attractor for all values of λ.

system is

$$\begin{bmatrix}\dot{V}_1\\ \dot{V}_2\\ \dot{V}_3\\ \dot{B}_1\\ \dot{B}_2\\ \dot{B}_3\end{bmatrix} = \begin{bmatrix}-2\nu & 0 & 0 & 0 & 0 & 0\\ 0 & -5\nu & 0 & 0 & 0 & 0\\ 0 & 0 & -9\nu & 0 & 0 & 0\\ 0 & 0 & 0 & -2\mu & 0 & 0\\ 0 & 0 & 0 & 0 & -5\mu & 0\\ 0 & 0 & 0 & 0 & 0 & -9\mu\end{bmatrix}\begin{bmatrix}V_1\\ V_2\\ V_3\\ B_1\\ B_2\\ B_3\end{bmatrix} + \begin{bmatrix}4(V_2V_3-B_2B_3)\\ -7(V_1V_3-B_1B_3)\\ 3(V_1V_2-B_1B_2)\\ 2(B_3V_2-V_3B_2)\\ 5(V_3B_1-B_3V_1)\\ 9(V_1B_2-B_1V_2)\end{bmatrix}, \quad (48)$$

where $\nu \geq 0$ is the viscosity and $\mu \geq 0$ is the resistivity. Without external forcing, this system is stable, dissipating to zero, so we consider the inviscid limit $\nu = \mu = 0$. The system is now Hamiltonian and our algorithm correctly converges to $\boldsymbol{m} = 0$, $\boldsymbol{A}^S = 0$. The results in Fig. 5 provide a useful illustration that trapping SINDy converges to stable energy-preserving models even when the trapping theorem is not satisfied. These results also provide a reminder that there are a large number of dynamical systems beyond fluids, such as MHD, which may benefit from these types of techniques. The reason our algorithm converges to the correct behavior is because it is still minimizing $\dot{K}$; in this case trapping SINDy converges to $\dot{K} \approx 0$ and can make no further improvement.

E. Forced Burgers' equation

The viscous Burgers' equation has long served as a simplified one-dimensional analogue to the Navier-Stokes equations [111, 112]. The forced, viscous Burgers' equation on a periodic domain $x \in [0, 2\pi)$ is:

$$\frac{d}{dt}u = -(U+u)\partial_x u + \nu\partial_{xx}^2 u + g(x,t), \quad (49)$$

where ν is viscosity and the constant U models mean-flow advection. We project this system onto a Fourier basis and assume constant forcing acting on the largest scale, i.e., $g(x,t) = \sigma\left(a_1(t)e^{ix} + a_{-1}(t)e^{-ix}\right)$, as in Noack et al. [113]. After Fourier projection, the evolution of the coefficients $a_k(t)$ is given by the Galerkin dynamics

$$\dot{a}_k = \left(\delta_{|k|1}\sigma - \nu k^2 - ikU\right)a_k - \sum_{\ell=-r}^{r} i\ell a_\ell a_{k-\ell}. \quad (50)$$

In the subcritical case $\sigma < \nu$, the origin of this system is stable to all perturbations and all solutions decay for long times. However, in the supercritical case $\sigma > \nu$, the excess energy input from the forcing cascades to the smaller dissipative scales. The "absolute equilibrium" limit $\sigma = \nu = 0$ has a Hamiltonian structure; for long times the coefficients approach thermodynamic equilibrium and equipartition of energy [114]. This structure does not correspond to any physical behavior of the Navier-Stokes equations, although it does approximate some properties of the inviscid Euler equations [115]. Due to its rich dynamics, this modified Burgers' equation has also been investigated in the context of closure schemes for Galerkin models [113]. We simulate the PDE in Eq. (49) with a high-resolution Godunov-type finite volume method using a van Leer flux limiter, implemented in the open-source Clawpack solver [116].

We illustrate the model performance in Fig. 6a for the subcritical case with $\sigma = 0.01$ and $\nu = 0.025$, the supercritical case with $\sigma = 0.1$ and $\nu = 0.025$, and the absolute equilibrium. In all cases $U = 1$. For the subcritical condition, all the eigenvalues of $\boldsymbol{L}^S$ are negative, and thus the algorithm finds stable models. For the supercritical condition $\sigma > \nu$,

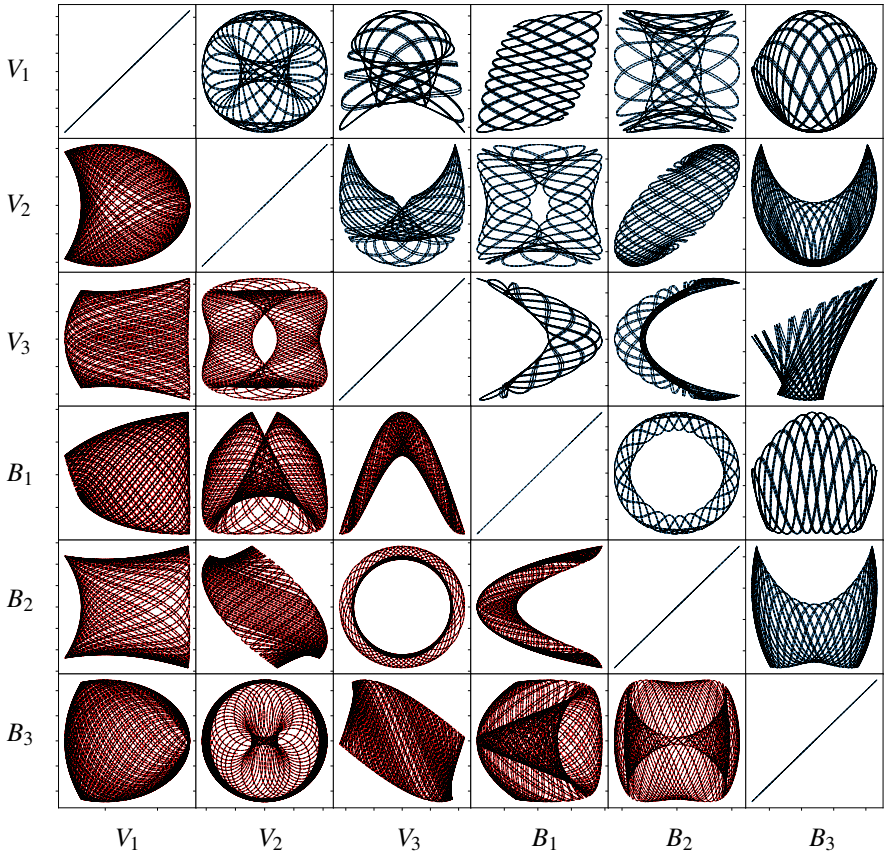


FIG. 5: The triad model for 2D inviscid MHD training data (blue, upper triangle) and a trapping SINDy model (black) capturing Hamiltonian dynamics on testing data (red, lower triangle).

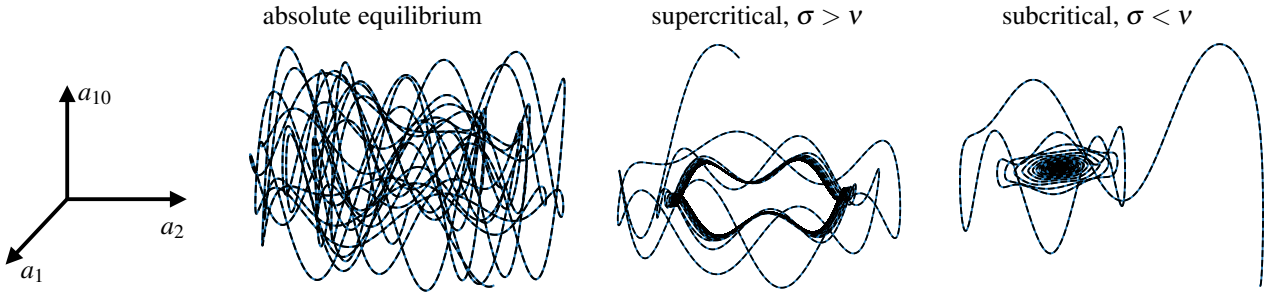
there is some subtlety. The algorithm does not converge to a negative definite $\boldsymbol{A}^S$, although it finds a solution with $\dot{K} \leq 0$. As mentioned in Section II D, this system does not exhibit effective nonlinearity. This lack of effective nonlinearity was also true for the MHD example in Section IV D, since the initial condition with no magnetic field perturbation, $B_1(0) = B_2(0) = B_3(0) = 0$, remains on the purely hydrodynamic manifold. In the inviscid limit, we did not need to consider this subspace because the system already does not satisfy the trapping theorem by virtue of being Hamiltonian. Lastly, in the absolute equilibrium regime the trapping SINDy algorithm correctly identifies vanishing eigenvalues of $\boldsymbol{A}^S$. In practice, we find excellent models for all of the aforementioned systems and for all practical purposes these models are typically stable, regardless of effective nonlinearity or Hamiltonian dynamics, because the SINDy trapping algorithm is minimizing $\dot{K}$. However, without effective nonlinearity we are not guaranteed to produce a stable model for every possible initial condition.

In Fig. 6b we illustrate the $r = 10$ model built from the DNS data in the supercritical regime with $\sigma = 0.1$, $\nu = 0.025$. It struggles a bit with the transient but otherwise the performance is accurate. Part of the reason for the poor fit to the transient is that $\lambda = 0$ is used here. The biasing towards stability appears to mitigate some of the need for sparsity-promotion; in other words, sparsity-promotion is not necessarily needed to produce a stable model, but may be needed for a more accurate or interpretable model, since the number of coefficients in Q_{ijk} is $\mathcal{O}(r^3)$ despite the constraints. Using finite λ may improve the model further, especially the transients, but instead of further investigating this example, we move on and conclude the results by addressing the challenging von Kàrmàn vortex shedding behind a circular cylinder.

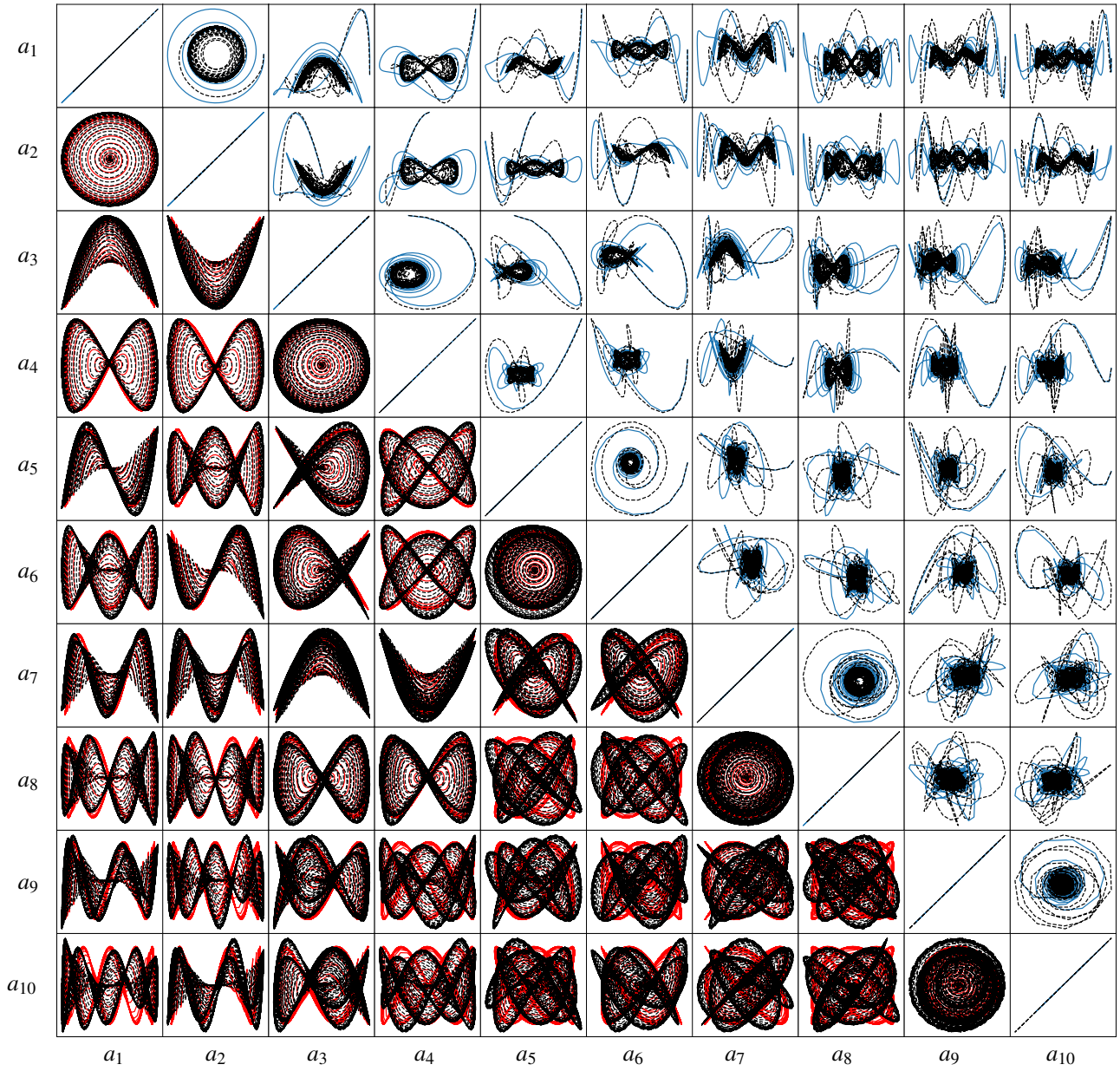
F. Von Kàrmàn vortex street

Here we investigate the fluid wake behind a bluff body, characterized by a periodic vortex shedding phenomenon known as a von Kàrmàn street. The two-dimensional incompressible flow past a cylinder is a stereotypical example of such behavior, and has been a benchmark problem for Galerkin models for decades [1]. The transition from a steady laminar solution to vortex shedding is given by a Hopf bifurcation, as a pair of eigenvalues of the linearized Navier-Stokes operator cross the real axis.

The transient energy growth and saturation amplitude of this instability mode is of particular interest and has historically posed a significant modeling challenge. Early Galerkin models of vortex shedding, based on a POD expansion about the mean flow, captured the oscillatory behavior but were structurally unstable [117]. This was later resolved by Noack et al. [1], who recognized that the transient behavior could be explained by Stuart-Landau nonlinear stability theory, in



(a) Trapping SINDy model (black) for the modified Burgers' equation in the three dynamic regimes. For improved illustration, the ground truth data (blue) is generated from the 10D Noack et al. [113] model rather than DNS.



(b) Temporal evolutions of each (a_i, a_j) pair for $i, j = 1, ..., 10$ obtained from DNS training data (blue, upper triangle), DNS testing data (red, lower triangle), and trapping SINDy prediction on both DNS datasets (black). The trapping algorithm struggles a bit with the transients, but obtains the correct attractor behavior.

FIG. 6: Summary of trapping SINDy performance for the forced Burgers' equation.

which the unsteady symmetric flow is deformed to the neutrally stable mean flow via a nonlinear self-interaction of the instability mode. In that work, an 8-mode POD basis was augmented with a ninth "shift mode" parameterizing this mean flow deformation. This approach was later formalized with a perturbation analysis of the flow at the threshold of bifurcation [118].

This modification encodes the intuition that the dynamics take place on the parabolic manifold associated with the Hopf bifurcation; without it, the energy quadratic models tends to overshoot and oscillate before approaching the post-transient limit cycle. Nevertheless, the 9-mode quadratic Galerkin model does resolve the transient dynamics, nonlinear stability mechanism, and post-transient oscillation, accurately reproducing all of the key physical features of the vortex street. Moreover, in Schlegel and Noack [10] stability of the quadratic model was proven with $m_9 = m_{\text{shift}} = \varepsilon$, $\varepsilon > 1$, and $m_i = 0$ for $i = \{1, ..., 8\}$. Recall from the discussion in Section II D that POD-Galerkin models will generally weakly satisfy the effective nonlinearity criteria and it is unclear if the shift-mode complicates this picture.

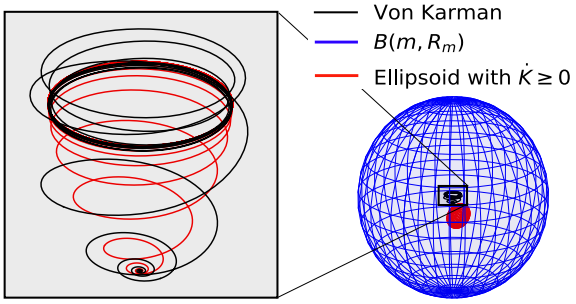
Although the POD-Galerkin model is an accurate description of the flow past a cylinder, it is an intrusive model, in the sense that evaluating the projected dynamics requires evaluating individual terms in the governing equations, such as spatial gradients of the flow fields. POD-Galerkin models therefore tend to be highly sensitive to factors including mesh resolution, convergence of the POD modes, and treatment of the pressure and viscous terms. Recent work by Loiseau et al. [19, 119, 120] has bypassed the Galerkin projection step by using the SINDy algorithm to directly identify the reduced-order dynamics. This approach has been shown to yield compact, accurate models for low-dimensional systems ($r = 2$ or 3), but preserving accuracy and stability for higher-dimensional systems remains challenging. Higher-dimensional regression problems often become ill conditioned; for example, in the cylinder wake example, the higher modes 3-8 are essentially harmonics of the driving modes 1-2, and so it is difficult to distinguish between the various polynomials of these modes during regression. Because these higher harmonics are driven by modes 1-2, the 3D constrained quadratic SINDy model with modes 1-2 plus the shift mode from Loiseau et al. [19] already performs well enough to capture the energy evolution with minor overshoot and correct long-time behavior. Details of the DNS and the POD-Galerkin technique used to reproduce the 9D shift-mode model can be found in Appendix A.

With the trapping SINDy algorithm, we obtain new 5-dimensional and 9-dimensional models for the cylinder wake and compare the performance against the same-size analytic POD-Galerkin models. The 5D trapping SINDy model is provably stable and we illustrate the identified trapping region in Fig. 7a. We also compare the 5D SINDy and 9D POD-Galerkin models in Fig. 7c. The 5D trapping SINDy model outperforms the 9D POD-Galerkin model by significantly improving the transient and improving the identification of the long-term attractor. For the 9D trapping SINDy model, we managed to reduce the largest eigenvalue of $\boldsymbol{A}^S$ to $\mathcal{O}(10^{-2} - 10^{-4})$ but were unable to produce accurate trapping SINDy models with fully negative definite $\boldsymbol{A}^S$. In practice, these models are functionally stable; we tested a large set of random initial conditions and did not find unbounded trajectories. Further searching in the hyperparameter space, or more algorithm iterations for better convergence, could potentially produce fully stable models.

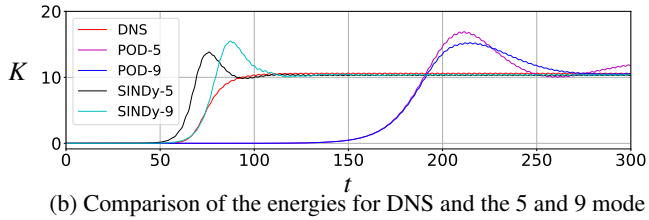
Despite this setback, the 9D trapping SINDy model performs quite well. The Galerkin model and the trapping SINDy model exhibit comparable performance and the SINDy model improves the transient prediction. The energies in Fig. 7b illustrate convergence to the true fluid flow energy for all the SINDy and POD-Galerkin models, with only the 9D trapping SINDy model capturing the precise timing of the transient. The flow reconstructions in Fig. 7d are quite accurate for both models. This is surprisingly strong performance with SINDy; recall that: 1) the Galerkin model is far more invasive a procedure than SINDy, requiring computation of spatial derivatives and inner products from the DNS, 2) the Galerkin model can still be quite sensitive to the DNS data, boundary conditions, and mesh size, and 3) the 9D trapping SINDy model is far sparser and has far fewer "active" terms than the 9D POD-Galerkin model.

The difficulty in producing provably stable, 9D trapping SINDy models here appears to reveal an interesting optimization tradeoff. While sparsity-promotion tends to promote more accurate models and reduce the complexity of the nonconvex optimization problem (since there are fewer active terms to manage), it also deemphasizes our proposed metric for the strength of effective nonlinearity, S_e from Eq. (17), by reducing the values of unimportant model terms. For instance, the SINDy model here exhibits weak effective nonlinearity, $S_e \approx 10^{-5}$, compared with $S_e \approx 10^{-2}$ for the POD-Galerkin model. This small value of S_e may indicate increased difficulty in obtaining a fully negative definite $\boldsymbol{A}^S$. SINDy models with weaker sparsity-promotion exhibit larger S_e, but then it becomes exceedingly difficult to obtain accurate models in the nonconvex optimization problem. Without any sparsity-promotion this is an ill-conditioned, nonconvex optimization in a 330-dimensional space. In this way, there appears to be some tradeoff between sparsity-promotion and the strength of effective nonlinearity. Given these points, we consider the sparse 5-mode and 9-mode SINDy models to be promising first steps towards incorporating stability constraints into higher-dimensional data-driven models.

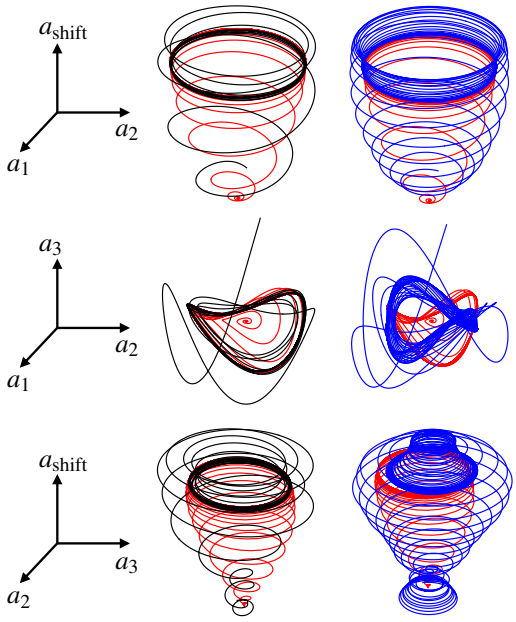
Before concluding, we should note that the eight-mode (no shift mode) POD-Galerkin model from Noack et al. [1], and all eight-mode models found by trapping SINDy, do not exhibit global stability. The problem fundamentally stems from the marginal stability of the mean flow and the very weak effective nonlinearity, both of which are somewhat addressed by the shift mode in the 9-mode model. This should be taken as a cautionary warning; success of these algorithms still relies on useful representations that capture the stability information of the underlying dynamics. This may require high-resolution data or the alternative dynamic bases mentioned in Section II A.



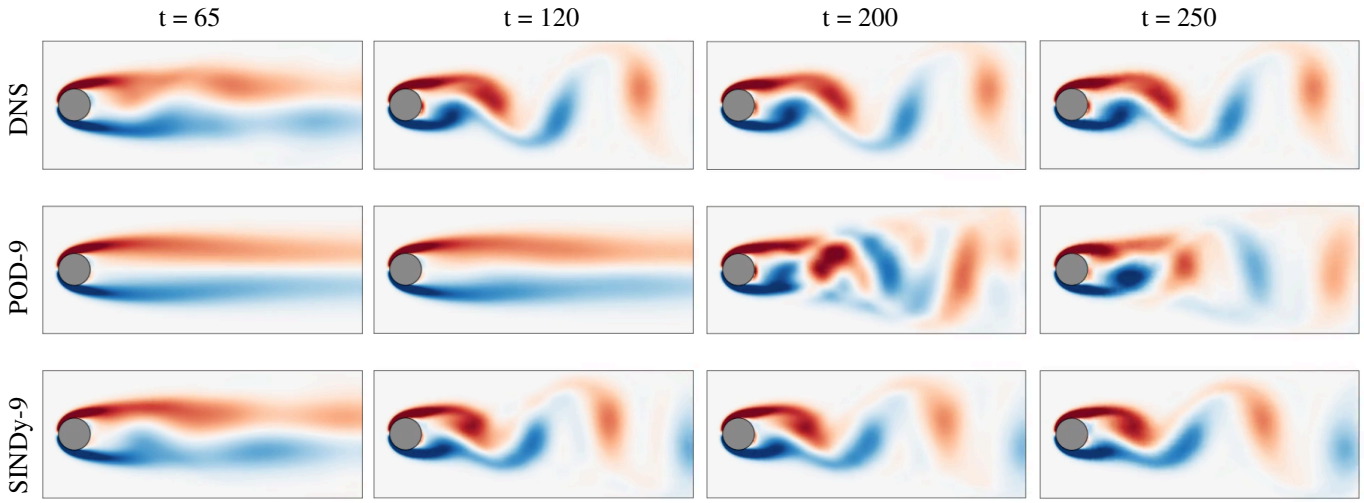
(a) Trapping SINDy 5D model (black) of a Von Kàrmàn trajectory (red). The trajectory is shown within the estimated trapping region and ellipsoid where $\dot{K} \geq 0$.



(b) Comparison of the energies for DNS and the 5 and 9 mode POD-Galerkin and trapping SINDy models.



(c) 5-mode trapping SINDy (black) and 9-mode POD-Galerkin (blue) models with a random initial condition, and the Von Kàrmàn trajectory used for training (red).



(d) Predictions of the vorticity field for the von Kàrmàn street at four snapshots in time, with a movie available [121]. The trapping SINDy model outperforms the 9D POD-Galerkin model, although an initial phase error in the trapping SINDy prediction (visible in the first snapshot) persists throughout the prediction.

FIG. 7: Summary of the differences between DNS, POD-Galerkin models, and trapping SINDy models.

V. CONCLUSION

The present work develops physics-constrained system identification by biasing models towards fulfilling global stability criteria, and subsequently produces long-term bounded models with no extra assumptions about the stability properties of equilibrium points and equilibrium trajectories. In order to produce globally stable models, we have implemented a new trapping SINDy algorithm based on the Schlegel-Noack trapping theorem [10]. Biasing models towards stability, and post-fit, proving that identified models are globally stable, will likely become increasingly important for both projection-based and data-driven models of fluids and plasmas. Our approach, which relies on using the energy as a Lyapunov function for an entire class of models with fixed nonlinear structure, is challenging for application to higher-order nonlinearities where generic Lyapunov functions are often unknown. Fortunately, data-driven methods are now increasingly used to discover Lyapunov functions and barrier functions for nonlinear control [93, 122–129]. These methods build a heuristic

Lyapunov function for a given dataset, rendering the search for a Lyapunov function tractable but possibly at the cost of model generality.

We demonstrated the effectiveness of this optimization to identify stable models and additionally managed to improve the discovery of models that do not conform to the assumptions of the trapping theorem. Our trapping SINDy algorithm resulted in more accurate and stable models for a range of systems, including simple benchmark problems, noisy data from chaotic systems, and DNS from full spatial-temporal PDEs. In these examples, we found that our modified SINDy algorithm could effectively discover stable, accurate, and sparse models from significantly corrupted data. Even when an explicit stable trapping region was not found, improved stability was observed. Finally, we explored relatively high-dimensional reduced-order models, with $\mathcal{O}(10)$ degrees of freedom, which are typically challenging for unconstrained data-driven algorithms.

There is considerable future work for biasing machine learning methods to discover models that satisfy existence-style proofs of stability, especially those that require non-convex optimization; we find that the lack of convexity in the trapping SINDy algorithm leads to deprecating algorithm speed and tractability as the size of the problem increases. There are many fluid flows which have known stable and unstable projection-based and data-driven reduced-order models, and which would benefit from a larger class of models with trapping region guarantees. Future work should apply this methodology to heavily-researched systems such as the fluidic pinball [21, 130] and the lid-cavity flow [131, 132]. Other promising future work includes adapting this structure to N-body coupled Stuart-Landau equations for which stability theorems already exist [133]. However, the nonconvexity of this formulation may require adaptation to a deep learning approach for high-dimensional N-body problems that occur in fluids and modern neuronal models.

For all of the examples in this work, we train our trapping SINDy algorithm on a single trajectory, although most data-driven methods can improve performance by processing data from multiple trajectories. Very large data can be effectively addressed with modern approaches, such as manifold Galerkin projection [120] and autoencoder [32, 79, 80, 134, 135] methods. These approaches may also address the significant Kolmogorov width limitations of linear transformations [136], and help ease the nonconvexity of our new optimization problem. There are also modern reduced-order modeling techniques, such as "lift & learn" [29], which produce quadratic ROMs regardless of the nonlinear structure of the underlying governing equations. Similarly Koopman analysis aims to produce a map from the original state-space, where the dynamics are nonlinear, to a new coordinate system, typically infinite dimensional, where the dynamics become linear [13, 16, 75, 79, 137–140].

However, adapting our methodology to these alternative bases requires additional work to understand how the trapping theorem, or similar theorems, change under these (often nonlinear) coordinate transformations. For instance, Pan et al. [28] builds stable Koopman models by requiring that the real parts of the eigenvalues of the linear Koopman operator are non-positive, although the relationship between this linear stability and the trapping theorem is unclear. In related work, neural-network-based encoders are often used to reverse this mapping; encoders can input quadratically nonlinear fluid flow data and apply nonlinear transformations to find useful reduced-order models beyond what is capable with traditional projection-based methods [141]. A natural question that arises is: assuming the original energy-preserving, quadratically nonlinear fluid flow exhibits a trapping region, under what conditions can we conclude that global stability holds in a new coordinate system given by $\boldsymbol{b} = \boldsymbol{g}(\boldsymbol{y})$, for some map $\boldsymbol{g} : \mathbb{R}^r \to \mathbb{R}^s$? The transformation could be an encoder, the reverse lifting map [29], or some other coordinate transform. Understanding how the stability properties manifest in the transformed system is a promising future direction for extending this stability theorem for ROMs with alternative dynamic bases.

ACKNOWLEDGEMENTS

This work was supported by the Army Research Office (ARO W911NF-19-1-0045) and the Air Force Office of Scientific Research (AFOSR FA9550-18-1-0200). JLC acknowledges support from the National Defense Science and Engineering Fellowship. SLB would like to acknowledge valuable discussions with Tulga Ersal, Ronan Fablet, Alireza Goshtabi, Nathan Kutz, JC Loiseau, Bernd Noack, and Said Ouala related to SINDy, Galerkin models, and stability constraints.

Appendix A: Von Kàrmàn DNS and POD-Galerkin details

We simulate the flow past a circular cylinder at Reynolds number $\mathrm{Re} = 100$ with unsteady incompressible DNS using the open source spectral element solver Nek5000 [142]. The domain consists of $17,432$ seventh order spectral elements ($\sim 850,000$ grid points) on $x, y \in (-20, 50) \times (-20, 20)$, refined close to a cylinder of unit diameter centered at the origin. Diffusive terms are integrated with third order backwards differentiation, while convective terms are advanced with a third order extrapolation. The 9-mode augmented POD-Galerkin model is computed following Noack et al [1], using gradients extracted directly from the DNS code. Mean-subtracted POD modes are computed from a set of 100 equally-spaced snapshots over one period of vortex shedding. The shift mode is calculated as the difference between an unsteady base flow, obtained with the selective frequency damping algorithm [143] and the mean of the snapshots, orthonormalized with respect to the remaining POD modes with a Gram-Schmidt procedure. The transient DNS is initialized with the unstable steady state perturbed by the leading linear instability mode with energy 10^{-8}. We compute the transient POD coefficients by projecting 3000 snapshots sampled at $\Delta t = 0.1$ onto this POD basis.

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

[2] B. R. Noack, M. Morzynski, and G. Tadmor, *Reduced-order modelling for flow control*, Vol. 528 (Springer Science & Business Media, 2011).

[3] K. Carlberg, C. Bou-Mosleh, and C. Farhat, Efficient nonlinear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, International Journal for numerical methods in engineering **86**, 155 (2011).

[4] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics **242**, 623 (2013).

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

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

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

[8] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. Dawson, and C.-A. Yeh, Modal analysis of fluid flows: Applications and outlook, AIAA journal **58**, 998 (2020).

[9] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry* (Cambridge university press, 2012).

[10] M. Schlegel and B. R. Noack, On long-term boundedness of Galerkin models, Journal of Fluid Mechanics **765**, 325 (2015).

[11] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics **656**, 5 (2010).

[12] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson, Spectral analysis of nonlinear flows, Journal of fluid mechanics **641**, 115 (2009).

[13] I. Mezic, Analysis of fluid flows via spectral properties of the Koopman operator, Annual Review of Fluid Mechanics **45**, 357 (2013).

[14] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Osth, S. Krajnovic, and R. K. Niven, Cluster-based reduced-order modelling of a mixing layer, J. Fluid Mech. **754**, 365 (2014).

[15] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the National Academy of Sciences **113**, 3932 (2016).

[16] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, Data-driven model reduction and transfer operator approximation, Journal of Nonlinear Science **28**, 985 (2018).

[17] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, Science Advances **3**, e1602614 (2017).

[18] M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model, Physics of Plasmas **24**, 022310 (2017).

[19] J.-C. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, Journal of Fluid Mechanics **838**, 42 (2018).

[20] A. Towne, O. T. Schmidt, and T. Colonius, Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis, Journal of Fluid Mechanics **847**, 821 (2018).

[21] N. Deng, B. R. Noack, M. Morzyński, and L. R. Pastur, Low-order model for successive bifurcations of the fluidic pinball, Journal of Fluid Mechanics **884**, A37 (2020).

[22] M. Raissi and G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, Journal of Computational Physics **357**, 125 (2018).

[23] 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, Physical review letters **120**, 024102 (2018).

[24] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner, Learning data-driven discretizations for partial differential equations, Proceedings of the National Academy of Sciences **116**, 15344 (2019).

[25] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annual Reviews of Fluid Mechanics **51**, 357 (2019).

[26] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, Universal differential equations for scientific machine learning, arXiv preprint arXiv:2001.04385 (2020).

[27] E. P. Alves and F. Fiuza, Data-driven discovery of reduced plasma physics models from fully-kinetic simulations, arXiv preprint arXiv:2011.01927 (2020).

[28] S. Pan and K. Duraisamy, Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability, SIAM Journal on Applied Dynamical Systems **19**, 480 (2020).

[29] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems, Physica D: Nonlinear Phenomena **406**, 132401 (2020).

[30] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, Physics-constrained, low-dimensional models for magnetohydrodynamics: First-principles and data-driven approaches, Physical Review E **104**, 015206 (2021).

[31] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895 (2020).

[32] K. Lee and K. T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, Journal of Computational Physics **404**, 108973 (2020).

[33] B. Herrmann, P. J. Baddoo, R. Semaan, S. L. Brunton, and B. J. McKeon, Data-driven resolvent analysis, arXiv preprint arXiv:2010.02181 (2020).

[34] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, Learning to simulate complex physics with graph networks, in *International Conference on Machine Learning* (PMLR, 2020) pp. 8459–8468.

[35] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, Machine learning accelerated computational fluid dynamics, arXiv preprint arXiv:2102.01010 (2021).

[36] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, The structure of global conservation laws in Galerkin plasma models, arXiv preprint arXiv:2101.03436 (2021).

[37] M. Brenner, J. Eldredge, and J. Freund, Perspective on machine learning for advancing fluid mechanics, Physical Review Fluids **4**, 100501 (2019).

[38] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, Annual Review of Fluid Mechanics **52**, 477 (2020).

[39] M. Raissi, A. Yazdani, and G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science **367**, 1026 (2020).

[40] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, Lagrangian neural networks, arXiv preprint arXiv:2003.04630 (2020).

[41] A. J. Majda and J. Harlim, Physics constrained nonlinear regression models for time series, Nonlinearity **26**, 201 (2012).

[42] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering **102**, 1136 (2015).

[43] B. Peherstorfer and K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering **306**, 196 (2016).

[44] L. Yang, D. Zhang, and G. E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, SIAM Journal on Scientific Computing **42**, A292 (2020).

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

[46] M. Mohebujjaman, L. G. Rebholz, and T. Iliescu, Physically constrained data-driven correction for reduced-order modeling of fluid flows, International Journal for Numerical Methods in Fluids **89**, 103 (2019).

[47] F. Noé, S. Olsson, J. Köhler, and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science **365**, eaaw1147 (2019).

[48] K. Lee and K. Carlberg, Deep conservation: A latent-dynamics model for exact satisfaction of physical conservation laws, arXiv preprint arXiv:1909.09754 (2019).

[49] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, A unified sparse optimization framework to learn parsimonious physics-informed models from data, IEEE Access **8**, 169259 (2020).

[50] N. B. Erichson, M. Muehlebach, and M. W. Mahoney, Physics-informed autoencoders for Lyapunov-stable fluid flow prediction, arXiv preprint arXiv:1905.10866 (2019).

[51] N. Sawant, B. Kramer, and B. Peherstorfer, Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference, arXiv preprint arXiv:2107.02597 (2021).

[52] M. J. Balajewicz, E. H. Dowell, and B. R. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation, Journal of Fluid Mechanics **729**, 285 (2013).

[53] M. Balajewicz, Lyapunov stable Galerkin models of post-transient incompressible flows, arXiv preprint arXiv:1312.0284 (2013).

[54] K. Carlberg, R. Tuminaro, and P. Boggs, Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics, SIAM Journal on Scientific Computing **37**, B153 (2015).

[55] L. Peng and K. Mohseni, Symplectic model reduction of Hamiltonian systems, SIAM Journal on Scientific Computing **38**, A1 (2016).

[56] B. M. Afkham and J. S. Hesthaven, Structure preserving model reduction of parametric Hamiltonian systems, SIAM Journal on Scientific Computing **39**, A2616 (2017).

[57] H. S. Bhat, Learning and interpreting potentials for classical Hamiltonian systems, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2019) pp. 217–228.

[58] H. K. Chu and M. Hayashibe, Discovering interpretable dynamics by sparsity promotion on energy and the Lagrangian, IEEE Robotics and Automation Letters **5**, 2154 (2020).

[59] D. Lasagna, D. Huang, O. R. Tutty, and S. Chernyshenko, Sum-of-squares approach to feedback control of laminar wake flows, Journal of Fluid Mechanics **809**, 628 (2016).

[60] B. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. Brunton, Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data, Journal of Open Source Software **5**, 1 (2020).

[61] W. D. McComb, The physics of fluid turbulence, Oxford (1990).

[62] B. Rummler and A. Noske, Direct Galerkin approximation of plane-parallel-Couette and channel flows by Stokes eigenfunctions, Notes on numerical fluid mechanics **64**, 3 (1998).

[63] H. Schlichting and K. Gersten, *Boundary-layer theory* (Springer, 2016).

[64] J. P. Freidberg, *Ideal MHD* (Cambridge University Press, 2014).

[65] A. A. Kaptanoglu, T. E. Benedett, K. D. Morgan, C. J. Hansen, and T. R. Jarboe, Two-temperature effects in Hall-MHD simulations of the HIT-SI experiment, Physics of Plasmas **27**, 072505 (2020).

[66] S. Galtier, *Introduction to modern magnetohydrodynamics* (Cambridge University Press, 2016).

[67] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control* (Cambridge University Press, 2019).

[68] V. Škvára, V. Šmídl, T. Pevnỳ, J. Seidl, A. Havránek, and D. Tskhakaya, Detection of Alfvén eigenmodes on COMPASS with generative neural networks, Fusion Science and Technology **76**, 962 (2020).

[69] D. R. Ferreira, P. J. Carvalho, C. Sozzi, P. J. Lomas, and J. Contributors, Deep learning for the analysis of disruption precursors based on plasma tomography, Fusion Science and Technology **76**, 901 (2020).

[70] I. Nayak and F. L. Teixeira, Dynamic mode decomposition for prediction of kinetic plasma behavior, in *2020 International Applied Computational Electromagnetics Society Symposium (ACES)* (IEEE, 2020) pp. 1–2.

[71] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, Characterizing magnetized plasmas with dynamic mode decomposition, Physics of Plasmas **27**, 032108 (2020).

[72] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA journal **40**, 2323 (2002).

[73] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, International Journal of Bifurcation and Chaos **15**, 997 (2005).

[74] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, 2016).

[75] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, Proceedings of the National Academy of Sciences **17**, 315 (1931).

[76] S. Pan, N. Arnold-Medabalimi, and K. Duraisamy, Sparsity-promoting algorithms for the discovery of informative Koopman invariant subspaces, arXiv preprint arXiv:2002.10637 (2020).

[77] B. J. McKeon and A. S. Sharma, A critical-layer framework for turbulent pipe flow, Journal of Fluid Mechanics **658**, 336 (2010).

[78] M. Luhar, A. S. Sharma, and B. J. McKeon, Opposition control within the resolvent analysis framework, Journal of Fluid Mechanics **749**, 597 (2014).

[79] B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nature communications **9**, 1 (2018).

[80] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proceedings of the National Academy of Sciences **116**, 22445 (2019).

[81] B. Kramer, Stability domains for quadratic-bilinear reduced-order models, SIAM Journal on Applied Dynamical Systems **20**, 981 (2021).

[82] H. Tennekes and J. L. Lumley, *A First Course in Turbulence* (MIT Press, 1972).

[83] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, Journal of Fluid Mechanics **807**, 155 (2016).

[84] O. San and R. Maulik, Neural network closures for nonlinear model order reduction, Advances in Computational Mathematics **44**, 1717 (2018).

[85] S. Pan and K. Duraisamy, Data-driven discovery of closure models, SIAM Journal on Applied Dynamical Systems **17**, 2381 (2018).

[86] R. Maulik, O. San, A. Rasheed, and P. Vedula, Subgrid modelling for two-dimensional turbulence using neural networks, Journal of Fluid Mechanics **858**, 122 (2019).

[87] S. Beetham and J. Capecelatro, Formulating turbulence closures using sparse regression with embedded form invariance, Physical Review Fluids **5**, 084611 (2020).

[88] S. Grimberg, C. Farhat, and N. Youkilis, On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows, Journal of Computational Physics **419**, 109681 (2020).

[89] A. Pouquet, D. Rosenberg, J. E. Stawarz, and R. Marino, Helicity dynamics, inverse, and bidirectional cascades in fluid and magnetohydrodynamic turbulence: a brief review, Earth and Space Science **6**, 351 (2019).

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

[91] M. Balajewicz and E. H. Dowell, Stabilization of projection-based reduced order models of the Navier–Stokes, Nonlinear Dynamics **70**, 1619 (2012).

[92] A. Schein, K. T. Carlberg, and M. J. Zahr, Preserving general physical properties in model reduction of dynamical systems via constrained-optimization projection, International Journal for Numerical Methods in Engineering (2021).

[93] J. Z. Kolter and G. Manek, Learning stable deep dynamics models, in *Advances in Neural Information Processing Systems*, Vol. 32 (2019) pp. 11128–11136.

[94] V. Srivastava and K. Duraisamy, Generalizable physics-constrained modeling using learning and inference assisted by feature space engineering, arXiv preprint arXiv:2103.16042 (2021).

[95] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, A unified framework for sparse relaxed regularized regression: SR3, IEEE Access **7**, 1404 (2019).

[96] M. L. Overton, On minimizing the maximum eigenvalue of a symmetric matrix, SIAM Journal on Matrix Analysis and Applications **9**, 256 (1988).

[97] J. V. Burke and M. C. Ferris, A Gauss—Newton method for convex composite optimization, Mathematical Programming **71**, 179 (1995).

[98] D. Drusvyatskiy and C. Paquette, Efficiency of minimizing compositions of convex functions and smooth maps, Mathematical Programming **178**, 503 (2019).

[99] P. Zheng and A. Aravkin, Relax-and-split method for nonconvex inverse problems, Inverse Problems **36**, 095013 (2020).

[100] S. Diamond and S. Boyd, CVXPY: A Python-embedded modeling language for convex optimization, The Journal of Machine Learning Research **17**, 2909 (2016).

[101] A. Aravkin, D. Drusvyatskiy, and T. van Leeuwen, Variable projection without smoothness, arXiv preprint arXiv:1601.05011 (2016).

[102] J. Zhang, A. M. Pace, S. A. Burden, and A. Aravkin, Offline state estimation for hybrid systems via nonsmooth variable projection, Automatica **115**, 108871 (2020).

[103] H. Attouch, J. Bolte, and B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods, Mathematical Programming **137**, 91 (2013).

[104] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM journal on imaging sciences **2**, 183 (2009).

[105] Y. Nesterov, Gradient methods for minimizing composite functions, Mathematical Programming **140**, 125 (2013).

[106] K. Kaheman, J. N. Kutz, and S. L. Brunton, SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics, Proceedings of the Royal Society A **476**, 20200279 (2020).

[107] P. Holmes and J. Guckenheimer, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Applied Mathematical Sciences, Vol. 42 (Springer-Verlag, Berlin, Heidelberg, 1983).

[108] J. Tuwankotta, Chaos in a coupled oscillators system with widely spaced frequencies and energy-preserving nonlinearity, International Journal of Non-Linear Mechanics **41**, 180 (2006).

[109] E. N. Lorenz, Deterministic nonperiodic flow, Journal of atmospheric sciences **20**, 130 (1963).

[110] V. Carbone and P. Veltri, Relaxation processes in magnetohydrodynamics-a triad-interaction model, Astronomy and Astrophysics **259**, 359 (1992).

[111] J. M. Burgers, A mathematical model illustrating the theory of turbulence, Advances in Applied Mechanics **1**, 171 (1948).

[112] E. Hopf, A mathematical example displaying features of turbulence, Communications on Pure and Applied Mathematics **1**, 302 (1948).

[113] B. R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzyński, P. Comte, and G. Tadmor, A finite-time thermodynamics of unsteady fluid flows, Journal of Non-Equilibrium Thermodynamics **33**, 103 (2008).

[114] A. J. Majda and I. Timofeyev, Remarkable statistical behavior for truncated Burgers–Hopf dynamics, Proceedings of the National Academy of Sciences **97**, 12413 (2000).

[115] R. H. Kraichnan and S. Chen, Is there a statistical mechanics of turbulence?, Physica D: Nonlinear Phenomena **37**, 160 (1989).

[116] Clawpack Development Team, Clawpack software (2020), version 5.7.1.

[117] A. Deane, I. Kevrekidis, G. E. Karniadakis, and S. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders, Physics of Fluids A: Fluid Dynamics **3**, 2337 (1991).

[118] D. Sipp and A. Lebedev, Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows, Journal of Fluid Mechanics **593**, 333 (2007).

[119] J.-C. Loiseau, B. R. Noack, and S. L. Brunton, Sparse reduced-order modeling: sensor-based dynamics to full-state estimation, Journal of Fluid Mechanics **844**, 459 (2018).

[120] J.-C. Loiseau, S. L. Brunton, and B. R. Noack, From the POD-Galerkin method to sparse manifold models, Handbook of Model-Order Reduction **2**, 1 (2019).

[121] See supplemental material at [URL will be inserted by publisher] for a movie illustrating a methodological comparison of the vortex shedding behind a circular cylinder.

[122] K. Neumann, A. Lemme, and J. J. Steil, Neural learning of stable dynamical systems based on data-driven Lyapunov candidates, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013) pp. 1216–1222.

[123] S. M. Khansari-Zadeh and A. Billard, Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions, Robotics and Autonomous Systems **62**, 752 (2014).

[124] S. M. Richards, F. Berkenkamp, and A. Krause, The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems, in *Proceedings of the 2nd Conference on Robot Learning (CoRL 2018)*, Vol. 87 (PMLR, 2018) pp. 466–476.

[125] W. Jin, Z. Wang, Z. Yang, and S. Mou, Neural certificates for safe control policies, arXiv preprint arXiv:2006.08465 (2020).

[126] N. Takeishi and Y. Kawahara, Learning dynamics models with stable invariant sets, arXiv preprint arXiv:2006.08935 (2020).

[127] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani, Learning stability certificates from data, arXiv preprint arXiv:2008.05952 (2020).

[128] Y.-C. Chang, N. Roohi, and S. Gao, Neural Lyapunov control, arXiv preprint arXiv:2005.00611 (2020).

[129] S. Massaroli, M. Poli, M. Bin, J. Park, A. Yamashita, and H. Asama, Stable neural flows, arXiv preprint arXiv:2003.08063 (2020).

[130] C. Raibaudo, P. Zhong, B. R. Noack, and R. J. Martinuzzi, Machine learning strategies applied to the control of a fluidic pinball, Physics of Fluids **32**, 015108 (2020).

[131] F. Terragni, E. Valero, and J. M. Vega, Local POD plus Galerkin projection in the unsteady lid-driven cavity problem, SIAM Journal on Scientific Computing **33**, 3538 (2011).

[132] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, Computer Methods in Applied Mechanics and Engineering **311**, 151 (2016).

[133] E. Panteley, A. Loria, and A. El-Ati, Practical dynamic consensus of Stuart–Landau oscillators over heterogeneous networks, International Journal of Control **93**, 261 (2020).

[134] P. Baldi and K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural networks **2**, 53 (1989).

[135] M. Milano and P. Koumoutsakos, Neural network modeling for near wall turbulent flow, Journal of Computational Physics **182**, 1 (2002).

[136] A. Pinkus, *N-widths in Approximation Theory*, Vol. 7 (Springer Science & Business Media, 2012).

[137] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator, Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 103111 (2017).

[138] E. Yeung, S. Kundu, and N. Hodas, Learning deep neural network representations for Koopman operators of nonlinear dynamical systems, in *2019 American Control Conference (ACC)* (IEEE, 2019) pp. 4832–4839.

[139] N. Takeishi, Y. Kawahara, and T. Yairi, Learning Koopman invariant subspaces for dynamic mode decomposition, in *Advances in Neural Information Processing Systems* (2017) pp. 1130–1140.

[140] S. E. Otto and C. W. Rowley, Linearly recurrent autoencoder networks for learning dynamics, SIAM Journal on Applied Dynamical Systems **18**, 558 (2019).

[141] F. J. Gonzalez and M. Balajewicz, Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, arXiv preprint arXiv:1808.01346 (2018).

[142] P. F. Fischer, J. W. Lottes, and S. G. Kerkemeir, Nek5000 web pages, http://nek5000.mcs.anl.gov (2008).

[143] E. Åkervik, L. Brandt, D. S. Henningson, J. Hœpffner, O. Marxen, and P. Schlatter, Steady solutions of the Navier-Stokes equations by selective frequency damping, Physics of fluids **18**, 068102 (2006).