

Degree-of-Freedom and Optimization-Dynamic Effects on the Observability of Kuramoto–Sivashinsky Systems

Noah B. Frank¹, Joshua L. Pughe-Sanford², and Samuel J. Grauer^{1,*}

¹Department of Mechanical Engineering, Pennsylvania State University

²Flatiron Institute, Simons Foundation

Abstract

Simulations of chaotic systems can only produce high-fidelity trajectories when the initial and boundary conditions are well specified. When these conditions are unknown but measurements are available, adjoint-variational state estimation can reconstruct a trajectory that is consistent with both the data and the governing equations. A key open question is how many measurements are required for accurate reconstruction, making the full system trajectory observable from sparse data. We establish observability criteria for adjoint state estimation applied to the Kuramoto–Sivashinsky equation by linking its observability to embedding theory for dissipative dynamical systems. For a system whose attractor lies on an inertial manifold of dimension $d_{\mathcal{M}}$, we show that $m \geq d_{\mathcal{M}}$ measurements ensures local observability from an arbitrarily good initial guess, and $m \geq 2d_{\mathcal{M}} + 1$ guarantees global observability and implies the only critical point on \mathcal{M} is the global minimum. We also analyze optimization-dynamic limitations that persist even when these geometric conditions are met, including drift off the manifold, Hessian degeneracy, negative curvature, and vanishing gradients. To address these issues, we introduce a robust reconstruction strategy that combines non-convex Newton updates with a novel pseudo-projection step. Numerical simulations of the Kuramoto–Sivashinsky equation validate our analysis and show practical limits of observability for chaotic systems with low-dimensional inertial manifolds.

Keywords: data assimilation; adjoint state estimation; chaotic dynamics; embedding theory; Kuramoto–Sivashinsky equation

1 Introduction

Chaotic dynamics arise across natural settings and technological systems: emerging in the whorls of turbulent fluid flow at high Reynolds numbers, in the fiery coupling of heat release and acoustic resonance that drives combustion instability, in the tremor of flexible wings at the precipice of flutter, in the jitter of micro-electromechanical resonators, and in the fluctuating frequencies of power grids. Although these systems differ in size and setting, they share a common character: nonlinear interactions that span many scales, a marked sensitivity to initial and boundary conditions, and aperiodic long-time dynamics. And yet, for all their apparent unruliness, the governing equations of many chaotic systems are well established, and modern numerical solvers can accurately integrate these systems forward in time from a prescribed starting point. Indeed, where computational resources suffice, scale-resolving simulations, such as direct numerical simulations of turbulent fluid flow, can map out dynamical structures. Such simulations reveal mechanisms of instability [1] and energy transfer [2, 3], they trace out the pathways through which disturbances amplify or decay, and they capture coherent structures [4, 5], along with low-dimensional manifolds that organize long-time behavior. Fully resolved states from high-fidelity simulations also furnish the data from which reduced-order models are extracted [6, 7] and from which closures for filtered simulations are calibrated [8, 9], thereby extending our predictive capability into regimes pertinent to engineering design and control.

Unfortunately, accurately simulating chaos comes at a steep computational cost. Direct simulations at device-relevant scales are often prohibitively expensive [10, 11], and reduced-order or closure models require calibration with data that may not exist in practical regimes. One compromise is to restrict simulations to

*Corresponding author: sgrauer@psu.edu

smaller subdomains [12], for example, resolving a shock wave–boundary layer interaction on a test article without modeling the entire wind tunnel environment. Although less expensive, such computations are strongly influenced by uncertain and usually unsteady inflow and outflow conditions, heat fluxes, material responses, and so forth. Experimental measurements, by contrast, provide access to the true dynamics of chaotic systems, free from assumptions and compute limitations, but the data are most often sparse, noisy, and indirect, whether they be from line-of-sight integrated schlieren or chemiluminescence images, pointwise thermocouples, pressure transducers, or hot-wire rakes, etc. Fortunately, data assimilation (DA) can blend partial observations like these with a system’s governing equations, using a numerical solver to produce high-fidelity trajectories that are anchored to real data [13–17]. In this paper, we focus on *state estimation*, where the goal is to reconstruct the full system evolution within an observation window (not necessarily to forecast future behavior). With this approach, DA can deliver accurate and dynamically consistent approximations of chaos in regimes where direct simulation is hindered by uncertain initial conditions, boundary conditions, or system parameters.

This perspective brings us to a central question: under what circumstances do available measurements provide enough information to permit accurate reconstruction of a chaotic trajectory in state space? In other words, when is the system *observable* via state estimation? For our purposes—distinct from the classical controls perspective—a system must satisfy two criteria to be observable. (1) The data uniquely determine the underlying trajectory. (2) The reconstruction method can recover that trajectory from those data. Taken together, observability requires that the inverse problem be numerically well posed for the chosen solver.

1.1 Data assimilation methods for state estimation

Observability depends on the dynamics of the target system and the available data, including their density, fidelity, and relation to the system state. As indicated above, it also depends on the chosen reconstruction scheme. Broadly, there are three families of DA methods for state estimation: filters, nudging, and smoothers. Filters, such as the Kalman filter [18] and extensions thereof [19], evolve the governing equations exactly between observation times but introduce discontinuous updates during the analysis steps, so the trajectory does not satisfy the dynamics across the entire assimilation window. Nudging and synchronization observers add feedback terms that drive the modeled system toward the measurements; this can be effective, but these solvers also perturb the true system dynamics since the feedback is not physical [20–22]. Variational methods, by contrast, reconstruct the entire trajectory at once by minimizing a loss function defined over the full measurement window. They assimilate all the data simultaneously, thereby producing a trajectory that is dynamically consistent (or can be, depending on the solver) while sustaining some discrepancies with the data. Hence, these methods “smooth out” said discrepancies by imposing dynamical constraints.

Some smoothers enforce the governing equations in an approximate manner via soft penalty terms, while others impose hard constraints embedded in the solver. Physics-informed neural networks (PINNs) [23], for instance, use soft constraints (for the most part). PINNs represent the system’s full trajectory with a global model that comprises one or more neural networks. The network parameters are tuned to minimize both measurement error and residuals of the governing equations, and the dynamics are weakly constrained by minimizing these residuals. In experimental fluid mechanics, related DA strategies approximate trajectories of flow states using B-splines [24], radial basis functions [25], or empirical modal bases [26], sometimes enforcing linear constraints like mass continuity as exact conditions.

A second category of smoothers enforces the system dynamics in full: the system is parameterized solely by initial and boundary conditions, for instance, and a high-fidelity solver is used to propagate it forward in time. The loss function, which compares predicted and experimental observations, is differentiated with respect to the unknown conditions. These conditions are then tuned to minimize the loss via gradient-based optimization. One way to compute these gradients is adjoint–variational DA [27, 28], often referred to as “4DVar” for unsteady systems in three spatial dimensions. 4DVar solves an adjoint equation that propagates measurement residuals backward in time to yield the gradient. An alternative is ensemble–variational DA, which avoids adjoint equations by estimating gradients statistically from an ensemble of forward model realizations [29]. For high-dimensional chaotic systems, adjoint–variational DA has been shown to be accurate and efficient [16, 30], since it provides exact gradients and its computational cost does not scale with the dimension of the control vector—in this case, initial and boundary conditions—which becomes very large in 4DVar state estimation problems.

Our goal in this work is to assess fundamental limits of observability, so we seek to minimize

solver-induced biases and avoid data–physics trade-offs, where possible. Since adjoint state estimation can enforce the governing equations exactly, it provides a natural framework to probe observability limits.

1.2 Application of embedding theory to observability in state estimation

Although many dissipative chaotic systems, ranging from the relatively simple Kuramoto–Sivashinsky (KS) equation to fluid turbulence governed by the three-dimensional (3D) Navier–Stokes equations, formally evolve in an infinite-dimensional state space, their long-time dynamics are expected to collapse onto a finite-dimensional invariant subset of state space, known as the global attractor and denoted by \mathcal{A} [31]. The box counting dimension of this attractor, $d_{\mathcal{A}}$, quantifies the system’s *effective degrees of freedom* and provides a measure of its complexity. It naturally follows that $d_{\mathcal{A}}$ should influence the number of measurements required to uniquely determine the system state.

Embedding theory makes this notion precise. It considers mappings of the form $\Phi : \mathcal{A} \rightarrow \mathbb{R}^m$, which takes a point on the attractor to an m -dimensional vector of observations $\mathbf{y} \in \mathbb{R}^m$. When the observations are sufficiently rich, Φ becomes an *embedding*, meaning that it is a smooth, one-to-one, and invertible mapping from points on the attractor to \mathbb{R}^m and it has a smooth inverse. In this case, the image $\Phi(\mathcal{A})$ is topologically equivalent to the attractor itself, so the geometry of the dynamics can be unfolded in measurement space without self-intersections [32]. Takens’ embedding theorem and its extensions [33] formalize this principle, showing that if the observation dimension m exceeds twice the attractor dimension ($m > 2d_{\mathcal{A}}$), then Φ is almost always an embedding. (These theorems hold under assumptions that we shall discuss later.) These results underpin the field of *state space reconstruction*,³ which leverages the topological equivalence of \mathcal{A} and $\Phi(\mathcal{A})$ to determine features of chaotic dynamics directly from measured data.

Although embedding theory has primarily been developed for state space reconstruction, we suggest that its insights can likewise inform adjoint state estimation. An embedding, by definition, is a smooth injective map with a smooth inverse Φ^{-1} , which implies that the initial state of the system, and thus its full trajectory over an observation window, can be uniquely identified from the data in \mathbf{y} . Embedding theorems specify when such a correspondence exists in principle, linking the number of measurements m to the attractor dimension $d_{\mathcal{A}}$. This is the first component of observability that we introduced above. What these theorems do not provide is a practical means of recovering Φ^{-1} , nor do they address the complications introduced by noise, limited observation windows, sensor placement, or model error [34]. These gaps motivate our study. We ask under what measurement conditions, and with which DA schemes, the observability conditions suggested by embedding theory can be achieved in practice.

1.3 Roadmap to the paper

To address the observability question posed above, we require a model system that is both chaotic and computationally accessible. The KS equation with periodic boundary conditions provides such a testbed. It is a nonlinear partial differential equation, often regarded as a minimal model of spatiotemporal chaos, that transitions from intermittent disorder to fully developed chaos with increasing domain lengths [35]. Despite its simplicity, the KS equation exhibits many hallmarks of more complex systems of engineering interest, including multiscale interactions and an energy cascade. Its long-term dynamics are well characterized in the literature [36–41], which provides benchmarks for testing predictions from embedding theory. At the same time, the modest computational cost of KS simulations enables systematic studies of measurement configurations and optimization strategies that would be prohibitive in higher-dimensional systems such as 3D fluid turbulence.

We apply embedding theory to the problem of observability in DA-based state estimation, with the goal of testing whether the reconstruction limits suggested by theory can be realized in practice. Our aim is not to recover invariants of chaotic attractors, per se, as in state space reconstruction, but to reconstruct full system trajectories that resolve the underlying fields, thereby enabling physical interpretation and modeling. To this end, we adopt an adjoint–variational state estimation method in which the state is parameterized by its initial condition and marched forward by a high-fidelity solver. Because the KS equation is one-dimensional (1D) in space and evolves in time, we refer to the approach as “2DVar.” We use 2DVar reconstructions to examine how accuracy varies with the density, spacing, and repetition rate of “sensors,” as well as on the

³State space reconstruction recovers an attractor by assembling delay-coordinate vectors $\mathbf{y}(t), \mathbf{y}(t - \tau), \mathbf{y}(t - 2\tau), \dots$ from one or more observed time series, typically to examine the attractor geometry or to estimate dynamical invariants of the system, like its Lyapunov exponents or fractal dimension, without explicitly enforcing the governing equations. State estimation in DA, by contrast, seeks to reconstruct the system’s trajectory in the original state space. This work is concerned with the latter.

duration of the observation window. First, we analyze our results in relation to predictions from embedding theory, expressed through a manifold dimension that bounds $d_{\mathcal{A}}$. We refer to dependencies on the attractor dimension as *degree-of-freedom effects*. After that, we examine how the topology of the loss landscape and the behavior of gradient-based optimizers influence the outcome of state estimation. We call this influence *optimization-dynamic effects*.

In the remainder of this paper, Sec. 2 introduces the KS equation as well as our numerical framework for forward and inverse computations. Section 3 lays out our procedure for generating test cases and presents sample reconstructions. Section 4 investigates how reconstruction accuracy depends on the sensor network, relating these results to the system's degrees of freedom and embedding-based observability criteria. Section 5 examines the role of optimization dynamics, showing how key features of the loss landscape influence convergence, and demonstrating that most difficulties arise from vanishing gradients and negative curvature rather than spurious local minima. Finally, Sec. 6 presents our conclusions and discusses broader implications of our work for higher-dimensional systems. Pertinent details on numerical methods and supporting derivations are provided in the appendices.

2 Adjoint state estimation for Kuramoto–Sivashinsky systems

We begin by introducing the model system and reconstruction framework used throughout the paper, starting with the KS equation, its key properties, and our numerical solver. Next, we present our 2DVar formulation and examine how chaos affects gradient computations. We then review first- and second-order optimization strategies and introduce a novel stabilization method for adjoint marching termed *pseudo-projection*.

2.1 Formulation and dynamics of the Kuramoto–Sivashinsky equation

In its derivative-form, the KS equation reads

$$\frac{\partial u}{\partial t} = - \underbrace{\frac{\partial^2 u}{\partial x^2}}_{\text{(I)}} - \underbrace{\frac{\partial^4 u}{\partial x^4}}_{\text{(II)}} - \underbrace{u \frac{\partial u}{\partial x}}_{\text{(III)}}, \quad (2.1)$$

for positions $x \in [-L/2, L/2]$, where L is the domain length, and times $t \in [0, \infty)$. In practice, we consider finite observation windows $[t_0, t_0 + T]$, where t_0 is an arbitrary start time and T is the window length. Observation times are relative to t_0 , with $t \in [0, T]$. Periodic boundary conditions are imposed,

$$\left(\frac{\partial^a u}{\partial x^a} \right)_{x=-L/2} = \left(\frac{\partial^a u}{\partial x^a} \right)_{x=L/2}, \quad \forall a \in \mathbb{N}_0, \quad (2.2)$$

and the dynamics are fully specified by the initial condition $u(x, 0)$.

Terms (I)–(III) can be analyzed in Fourier space, where periodic solutions admit the expansion

$$u(x, t) = \sum_{j \in \mathbb{Z}} \hat{u}_j(t) e^{ik_j x}, \quad (2.3)$$

with wavenumbers $k_j = 2\pi j/L$ and Fourier coefficients \hat{u}_j . Substituting this series into Eq. (2.1) yields

$$\frac{\partial \hat{u}_j}{\partial t} = (k_j^2 - k_j^4) \hat{u}_j - \frac{ik_j}{2} \sum_{i \in \mathbb{Z}} \hat{u}_i \hat{u}_{j-i}. \quad (2.4)$$

Here, $(k_j^2 - k_j^4) \hat{u}_j$ combines the energy-producing anti-diffusion and the energy-dissipating hyper-diffusion terms, i.e., (I) and (II). It amplifies low-wavenumber modes ($|k_j| < 1$) and damps high-wavenumber ones ($|k_j| > 1$), with maximum growth near $k_{\text{crit}} = \pm 1/\sqrt{2}$. The conservative convective term (III) redistributes energy across wavenumbers, coupling the stable and unstable modes and orchestrating the balance between the production, dissipation, and redistribution of energy that gives rise to chaos.

For all L , solutions of the KS equation have been proven to rapidly approach a smooth, finite-dimensional manifold \mathcal{M} , called the inertial manifold (IM) [42], which contains the global attractor $\mathcal{A} \subset \mathcal{M}$. The manifold is invariant under the system dynamics, so long-time trajectories of u are embedded in \mathcal{M} [43]. The dimension of this manifold, $d_{\mathcal{M}}$, has been extensively studied and provides an upper bound for the

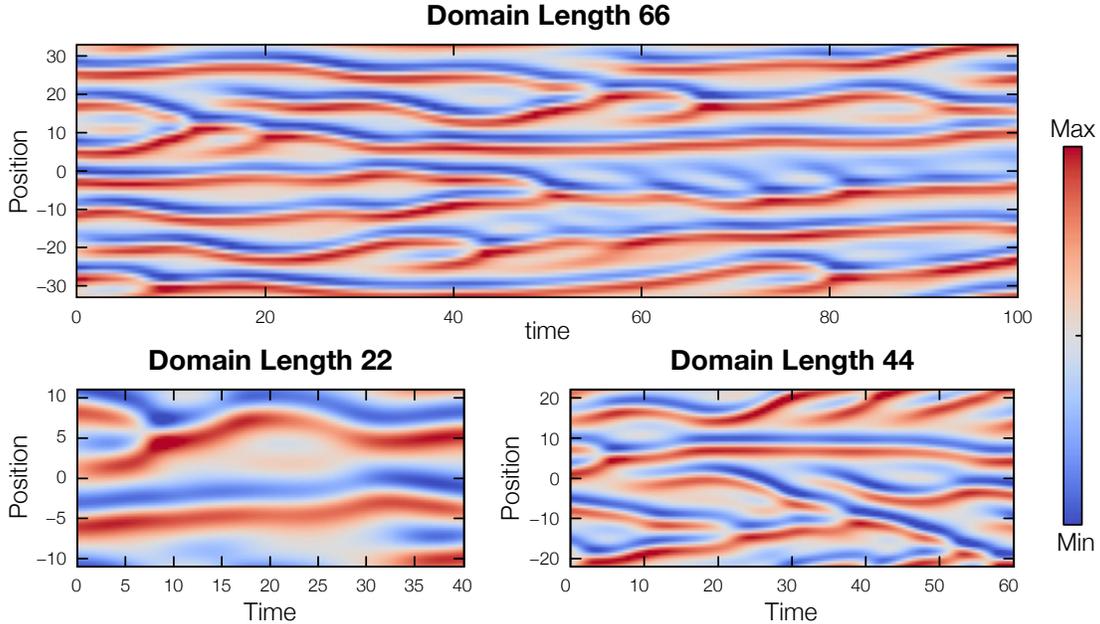


Figure 1: Representative trajectories of Kuramoto–Sivashinsky systems for domain lengths $L = 22$ (bottom left), $L = 44$ (bottom right), and $L = 66$ (top). All cases exhibit meandering streaks whose spatial and temporal complexity increases with L , reflecting the additional active degrees of freedom.

attractor’s box counting dimension, $d_A \leq d_M$, where d_A is generally intractable to compute. Thus, d_M serves as a rigorous measure of the system’s effective degrees of freedom, and embedding theory allows us to connect it to the number of measurements needed for reliable state estimation.

Forward solutions to the KS equation are obtained with a custom solver implemented in JAX, with full details provided in Appendix A. The solver employs a uniform grid with a Fourier pseudo-spectral discretization in space and a fourth-order Runge–Kutta exponential time-differencing scheme [44] to handle the stiff linear terms [45]. JAX also provides efficient gradient computations via automatic differentiation (AD), which we exploit in our 2DVar formulation. For $L \in \{22, 44\}$ we use 64 grid points with a time step of 0.1, while for $L = 66$ we use 72 points with a time step of 0.05. These discretizations are consistent with prior studies on KS systems [37, 45], and the Kaplan–Yorke dimension computed with our solver agrees with the results of Edson et al. [35].

Table 1: Inertial manifold dimension and leading Lyapunov exponent for different domain lengths.

Domain Length L	IM Dimension d_M	Leading Lyapunov Exponent ℓ_1
22	8	0.05
44	18	0.083
66	28	0.087

Numerical DA experiments are carried out in domains of length $L \in \{22, 44, 66\}$, each of which sustains chaotic dynamics [45]. As L grows larger, KS systems display behavior of increasing complexity: $L = 22$ lies just beyond the onset of structurally stable chaos [45], while $L = 66$ exhibits strongly chaotic behavior. Representative trajectories are shown in Fig. 1, characterized by undulating waveforms that manifest suddenly, drift and mingle across the domain, and merge.

To verify our solver, we benchmark our solutions against known chaotic invariants. Table 1 reports the IM dimension and leading Lyapunov exponent for each domain length. Lyapunov spectra are computed from long-time simulations using the QR method [46], with additional details provided in Appendix B. Estimates of d_M are obtained using the autoencoder-based approach of Zeng et al. [39]. An autoencoder

couples an encoder $E : \mathcal{M} \rightarrow \mathcal{L}$ to a decoder $D : \mathcal{L} \rightarrow \mathcal{M}$, where the composite map $A = D \circ E$ learns an identity function on \mathcal{M} and \mathcal{L} is a low-dimensional latent space of dimension $d_{\mathcal{L}}$. The encoder and decoder are jointly trained to minimize reconstruction error, such that A learns to represent states on \mathcal{M} in the compressed latent space \mathcal{L} . We set $d_{\mathcal{L}}$ conservatively so that $d_{\mathcal{M}} \leq d_{\mathcal{L}} < n$, where n is the dimension of the discrete state vector. The architecture of A is designed to promote models which only use a low-dimensional subset of the $d_{\mathcal{L}}$ -dimensional latent space. After training, the autoencoder provides mappings to and from a low-dimensional embedding of KS states. A principal component analysis (PCA) of the encoded states in \mathcal{L} yields a covariance matrix whose effective rank is taken as an estimate of $d_{\mathcal{M}}$. Figure 2 shows singular values of the centered data matrix and inferred dimensions for the $L = 22, 44,$ and 66 domains. These estimates of $d_{\mathcal{M}}$, computed with our solver and autoencoders, are consistent with rigorous analyses [36], physical-mode counts [40, 41], and previous autoencoder-based studies [37–39]. Further information on our autoencoder architectures and training procedures is provided in Appendix C.

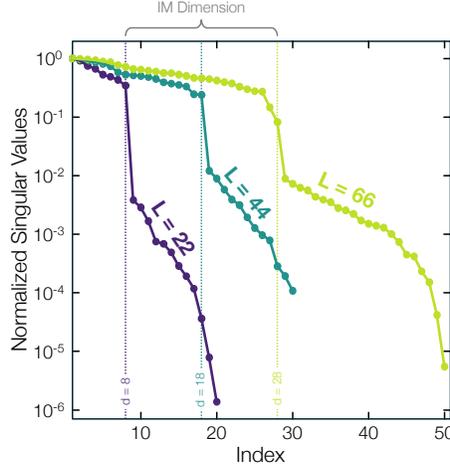


Figure 2: Autoencoder-based estimates of $d_{\mathcal{M}}$ for $L \in \{22, 44, 66\}$. Singular values of the centered latent state data matrix are shown, with vertical dashed lines marking the inferred IM dimension identified by the sharp drop in eigenvalues.

2.2 Adjoint-variational state estimation

We consider state estimation from point measurements of a KS system. In the general case, scalar observations y are derived from a point in state space at some time t relative to an initial time t_0 . An arbitrary observation operator for such measurements is

$$h_t : H \rightarrow \mathbb{R}, \quad u(-, t_0 + t) \mapsto y, \quad (2.5)$$

where h_t is at least twice differentiable and the state lives in a periodic Hilbert space $H = L^2_{\text{per}}(-L/2, L/2)$. If $t = 0$, the measurement is said to be *unlagged*, otherwise it is *lagged*. When observations depend on a spatial location, we include this dependence in the subscript: $h_{(x,t)}$. For unlagged pointwise measurements, we have

$$h_{(x,0)}[u(-, t_0)] = u(x, t_0). \quad (2.6)$$

Lagged observations are taken at later times; they are modeled by composing the unlagged operator with the system flow map,

$$h_{(x,t)}[u(-, t_0)] = h_{(x,0)} \circ f_t[u(-, t_0)], \quad (2.7)$$

where $f_t : H \rightarrow H$ is obtained by integrating Eq. (2.1) forward from t_0 to $t_0 + t$.

A set of m observations is available, denoted $\mathbf{y} \in \mathbb{R}^m$, with elements $y_i = u(x_i, t_0 + t_i)$. We consider spatial sensor locations $x \in \mathcal{X}$ and measurement times $t \in \mathcal{T} \subset [t_0, t_0 + T]$. Each sensor in \mathcal{X} records a value at every time in \mathcal{T} , and the observation coordinates $(x_i, t_i) \in \mathcal{X} \times \mathcal{T}$ are indexed by $i \in \mathcal{I}$. The objective is to reconstruct the true initial condition that produced the observed trajectory. A candidate initial condition $u_\theta \in H$, parameterized by θ , is used to generate reconstructions. In the variational setting, and assuming no

model error, measurement error, nor prior information about the initial condition, state estimation is posed as minimization of the loss functional (a.k.a. the cost functional or objective functional),

$$\mathcal{J}(\theta) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{G} \left[h_{(x_i, t_i)}(u_\theta), y_i \right], \quad (2.8)$$

where $\mathcal{G} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a smooth measure of residuals. In this work, \mathcal{G} is taken to be the squared error so that \mathcal{J} represents the mean squared error between the observations \mathbf{y} and the corresponding pointwise predictions of u generated from θ .

In our numerical experiments, the observation set $\mathcal{X} \times T$ is defined by placing sensors uniformly in space and sampling them at regular intervals. The spatial locations are evenly distributed, with m_x sensors separated by $\Delta x = L/m_x$ and centered within the domain. Each sensor records m_t samples at a constant rate $\Delta t = T/m_t$, beginning at $t_0 + \Delta t$ and including the final time, $t_0 + T$. Thus, the initial state is always excluded, the final time is always observed, and the total number of measurements is $m = m_x \times m_t$. Sample setups are shown in Fig. 3 for the $L = 22$ domain and a $T = 20$ time horizon. The left plot shows a sparse configuration with $m = 4$; the right plot shows a dense configuration with $m = 16$.

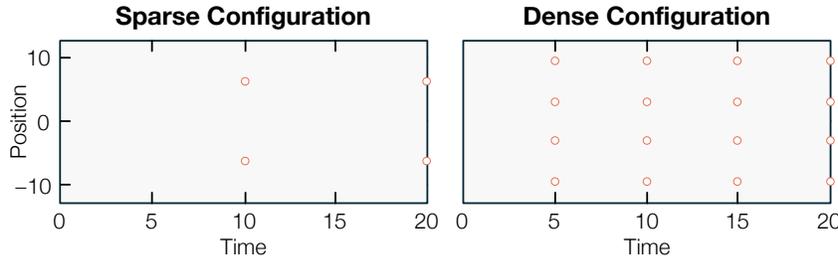


Figure 3: Exemplary measurement configurations in the $L = 22$ domain. The left panel shows a sparse layout with two spatial sensors and two observation times; the right panel shows a denser configuration with four spatial sensors and four observation times.

This problem setup naturally leads to a constrained optimization task,

$$\hat{\theta} = \arg \min_{\theta} \mathcal{J}(\theta). \quad (2.9)$$

Here, the constraint on the system dynamics is included in the definition of h , where the unlagged observation operator is composed with the flow map, and the candidate initial condition enters implicitly through h acting on u_θ . The minimizer $\hat{\theta}$ yields the initial condition whose trajectory best reproduces the observed data \mathbf{y} . Gradients and Hessians of \mathcal{J} with respect to the control parameter are obtained via adjoint equations, as described next. The optimization proceeds by calculus-based methods, starting from an initial guess θ_0 that is iteratively updated until convergence.

2.2.1 Discrete adjoint framework for gradients and Hessians

In practice, we discretize the KS equation and solve it numerically. The state is represented by a vector $\mathbf{u}_k \in \mathbb{R}^n$ that contains the solution at n uniformly spaced spatial nodes at time $k\Delta t$ for $k \in \mathcal{K} = \{0, \dots, K\}$. Here, \mathcal{K} is the set of all times and $K = T/\Delta t$ is the final time index. The system is advanced by a numerical solver,

$$\mathbf{u}_{k+1} = f_{\Delta t}(\mathbf{u}_k), \quad (2.10)$$

where $f_{\Delta t}$ is detailed in Appendix A. The continuous measurement operator is likewise replaced by a discrete analogue. For an unlagged point measurement at node i ,

$$h_{(i,0)}(\mathbf{u}_0) = \mathbf{e}_i^\top \mathbf{u}_0, \quad (2.11)$$

where \mathbf{e}_i is the i th standard basis vector in \mathbb{R}^n . The lagged observation at (i, j) is

$$h_{(i,j)}(\mathbf{u}_0) = \mathbf{e}_i^\top f_{\Delta t}^j(\mathbf{u}_0), \quad (2.12)$$

where $f_{\Delta t}^j$ denotes j successive applications of $f_{\Delta t}$, with $j = t/\Delta t \in \mathbb{N}_0$. We assume that all observation points (x_i, t_j) coincide with grid nodes and time steps. For brevity, we write h_i to indicate the discrete operator corresponding to the observation at $(x_i, t_i) \in \mathcal{X} \times \mathcal{T}$. The initial condition is parameterized via p Fourier coefficients $\boldsymbol{\theta} \in \mathbb{C}^p$, with the discrete field given by the inverse discrete Fourier transform,

$$\mathbf{u}_\theta = \mathbb{F}^{-1}(\boldsymbol{\theta}) \iff \boldsymbol{\theta} = \mathbb{F}(\mathbf{u}_\theta). \quad (2.13)$$

We use 15, 20, and 25 Fourier modes for the $L = 22, 44,$ and 66 domains. The cost functional for the discrete problem is

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \underbrace{[h_i(\mathbf{u}_\theta) - y_i]^2}_{\mathcal{G}(h_i(\mathbf{u}_\theta), y_i)}, \quad (2.14)$$

and Eq. (2.9) becomes

$$\tilde{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}). \quad (2.15)$$

We solve adjoint equations to compute first- and second-order derivatives of \mathcal{J} with respect to $\boldsymbol{\theta}$. The discrete adjoint recursion for the gradient is

$$\mathbf{u}_k^\dagger = \mathbf{u}_{k+1}^\dagger \frac{\partial \mathbf{u}_{\theta, k+1}}{\partial \mathbf{u}_{\theta, k}} + \frac{\partial M_k}{\partial \mathbf{u}_{\theta, k}}, \quad (2.16a)$$

for $k = K - 1, \dots, 0$, with terminal condition

$$\mathbf{u}_{\theta, K}^\dagger = \frac{\partial M_k}{\partial \mathbf{u}_{\theta, K}}. \quad (2.16b)$$

The adjoint variable \mathbf{u}_k^\dagger is a row vector and $\mathbf{u}_{\theta, k}$ denotes the observer system at the k th time index (the index is dropped for $k = 0$). For ease of notation, we define the iteration specific loss

$$M_k(\mathbf{u}_{\theta, k}, \mathbf{u}_k) = \frac{1}{m} (\mathbf{u}_{\theta, k} - \mathbf{u}_k)^\top \mathbf{M}_k (\mathbf{u}_{\theta, k} - \mathbf{u}_k),$$

where \mathbf{M}_k is a binary diagonal matrix selecting measurement positions at iteration k ,

$$M_{k,ii} = \begin{cases} 1, & \text{if } x_i \in \mathcal{X} \text{ and } k\Delta t \in \mathcal{T}, \\ 0, & \text{otherwise,} \end{cases}$$

and x_i is the spatial position corresponding to the i th cell. Using these definitions, the desired gradient of \mathcal{J} is obtained as

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}} = \mathbf{u}_0^\dagger \frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}}. \quad (2.16c)$$

The second-order system for the Hessian is

$$\frac{\partial (\mathbf{u}_k^\dagger)^\top}{\partial \mathbf{u}_\theta} = \left[\frac{\partial}{\partial \mathbf{u}_{\theta, k}} \left(\frac{\partial \mathbf{u}_{\theta, k+1}}{\partial \mathbf{u}_{\theta, k}} \right)^\top \right] \frac{\partial \mathbf{u}_{\theta, k}}{\partial \mathbf{u}_\theta} \mathbf{u}_{k+1}^\dagger + \left(\frac{\partial \mathbf{u}_{\theta, k+1}}{\partial \mathbf{u}_{\theta, k}} \right)^\top \frac{\partial (\mathbf{u}_{k+1}^\dagger)^\top}{\partial \mathbf{u}_\theta} + \frac{\partial^2 M_k}{\partial \mathbf{u}_{\theta, k}^2} \frac{\partial \mathbf{u}_{\theta, k}}{\partial \mathbf{u}_\theta}, \quad (2.17a)$$

for $k = K - 1, \dots, 0$, with terminal condition

$$\frac{\partial (\mathbf{u}_{\theta, K}^\dagger)^\top}{\partial \mathbf{u}_\theta} = \frac{\partial^2 M_k}{\partial \mathbf{u}_{\theta, K}^2} \frac{\partial \mathbf{u}_{\theta, K}}{\partial \mathbf{u}_\theta}. \quad (2.17b)$$

The Hessian of the loss is given by

$$\frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{\theta}^2} = \left(\frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}} \right)^\top \frac{\partial (\mathbf{u}_0^\dagger)^\top}{\partial \mathbf{u}_\theta}. \quad (2.17c)$$

In general, we adopt the following convention for higher-order tensor contractions:

$$\frac{\partial^2 \mathbf{a}}{\partial \mathbf{b}^2} \frac{\partial \mathbf{b}}{\partial \mathbf{c}} \iff \frac{\partial^2 a_i}{\partial b_j \partial b_k} \frac{\partial b_k}{\partial c_\ell}, \quad (2.18)$$

where the tensor notation on the left is equivalent to the Einstein notation on the right. The sensitivities $\partial \mathbf{u}_{\theta,k} / \partial \mathbf{u}_\theta$ are computed incrementally during the forward pass and stored alongside $\mathbf{u}_{\theta,k}$.

Automatic differentiation can be applied to full rollouts in JAX, where a ‘‘rollout’’ is a trajectory produced by autoregressive application of the forward solver. Computing gradients and Hessians via AD is equivalent to solving the adjoint systems described above. We implemented both methods and used AD to validate our adjoint formulation. Although explicit adjoint solvers can be far more memory efficient than AD, and thus indispensable for very-high-dimensional systems and long assimilation windows, the KS systems considered here have small state vectors and short rollouts. AD is thus simpler and faster to use for these cases, and we do so for the results reported in this paper.

2.2.2 Impact of chaos on adjoint marching

Numerical simulations of chaotic systems necessarily diverge from the true system trajectory over long time horizons due to the exponential growth of errors. The adjoint system inherits the Lyapunov spectrum of the forward dynamics [47, 48], so gradients of \mathcal{J} are vulnerable to the amplification of measurement noise and numerical errors accumulated during backward integration. Consequently, optimizing a single trajectory over long time horizons is not feasible [16]. A common strategy is to restrict the assimilation window to be on the order of the Lyapunov timescale, $T_\ell = 1/\ell_1$, where ℓ_1 is the leading Lyapunov exponent [49, 50]. For measurements spanning a total duration $T > T_\ell$, the problem is divided into $\text{ceil}(T/T_\ell)$ segments of length T_ℓ or less [50]. In order to speed up convergence in a multi-window reconstruction, the terminal condition from one segment may be used as the initial guess for the next [51]. In this paper, we follow this practice and take the assimilation windows to be of duration T_ℓ .

2.3 Optimization methods

With exact gradients and Hessians of the loss in hand, the performance of adjoint state estimation depends on the optimization algorithm. The choice of optimizer is pivotal in chaotic systems, where ill-conditioning and instability are endemic.

It is often assumed that optimization is difficult because descent methods become trapped in spurious local minima [52, 53]. However, in many high dimensional problems, such as dictionary learning [54], tensor decomposition [55], matrix completion [56], and training certain (restrictive) classes of neural networks [57], *all* minima are global minima with the same loss. Theory for random high-dimensional error surfaces, which are good surrogates for practical loss landscapes, likewise suggests that nearly all critical points of high loss are saddle points rather than minima [58, 59]. Ergo, the main challenge in most high-dimensional, gradient-based state estimation problems is escaping saddle points [52, 60, 61]. As a corollary, any local minimum can be regarded as a satisfactory solution. Similar challenges arise in adjoint state estimation, namely, effective handling of negative curvature and achieving convergence despite an inherently ill-conditioned Hessian.

We consider four calculus-based minimizers in this work: vanilla gradient descent, Newton’s method, a quasi-Newton algorithm, and a regularized variant of Newton’s method. The first three illustrate common failure modes in adjoint state estimation, whilst the last mitigates these pathologies. Throughout our discussion, the control vector at iteration k is denoted $\boldsymbol{\theta}_k$; it is initialized at $\boldsymbol{\theta}_0$ and iteratively updated with increasing k . The gradient and Hessian of the cost functional are

$$\mathbf{g} = \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}} \right)^\top, \quad \mathbf{H} = \frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{\theta}^2}, \quad (2.19)$$

where $\mathbf{g} \in \mathbb{R}^n$ and $\mathbf{H} \in \mathbb{R}^{n \times n}$ is symmetric.

2.3.1 Vanilla gradient descent

Gradient descent updates the control vector along the steepest slope of the loss landscape,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{g}, \quad (2.20)$$

where $\eta > 0$ is the step size, which may vary with k . Unfortunately, progress can slow to a crawl when the Hessian is ill-conditioned [53]. To see this, consider a quadratic loss with ordered Hessian eigenvalues $\lambda_1 \geq \dots \geq \lambda_n > 0$. The condition number of \mathbf{H} is $\kappa_{\mathbf{H}} = \lambda_1/\lambda_n$. Using the optimal step size, given by $\eta = 2/(\lambda_1 + \lambda_n)$, progress near a critical point $\boldsymbol{\theta}^*$ satisfies

$$\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}^*\|_2 < \frac{\kappa_{\mathbf{H}} - 1}{\kappa_{\mathbf{H}} + 1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_2, \quad (2.21)$$

where $\|\cdot\|_2$ is the Euclidean norm [62]. Thus, when $\kappa_{\mathbf{H}} \gg 1$, the contraction factor approaches unity, updates shrink extremely slowly, and the optimizer lingers near $\boldsymbol{\theta}^*$.

2.3.2 Newton's method

Gradients can be rescaled using local curvature information to deal with ill-conditioned Hessians. Newton's method, which achieves quadratic local convergence when \mathbf{H} is full rank [62], modifies gradient descent by taking smaller steps along directions of high curvature and larger steps along directions of low curvature,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{H}^{-1} \mathbf{g}. \quad (2.22)$$

When \mathbf{H} has negative eigenvalues, however, the Newton step can move uphill, converging to a nearby saddle point or local maximum. Moreover, if \mathbf{H} is singular or nearly so, regularization or other modifications are required to approximate \mathbf{H}^{-1} .

2.3.3 Quasi-Newton methods

Quasi-Newton methods are employed when computing, storing, or inverting the full Hessian is too costly. Instead, an approximation to the inverse Hessian $\mathbf{B}_k^{-1} \approx \mathbf{H}^{-1}$ is built from past gradients and iterates, so that the update step is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{B}_k^{-1} \mathbf{g}. \quad (2.23)$$

Like in Newton's method, \mathbf{B}_k^{-1} is meant to accelerate convergence in locally convex regions by rescaling gradients according to the estimated curvature at $\boldsymbol{\theta}_k$. The theoretical foundations of quasi-Newton methods break down in regions with strong negative curvature [52]. The most widely used variant, i.e., the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, builds a positive definite approximation \mathbf{B}_k^{-1} whenever the curvature condition is satisfied. At points of negative curvature this condition can fail, producing a non-positive definite or poorly conditioned update. A common remedy is to reset $\mathbf{B}_k^{-1} = \mathbf{I}$, thereby reverting to a gradient descent step at iteration k . As a result, BFGS can perform poorly in regions where negative curvature is prevalent. We use BFGS as our representative quasi-Newton method.

2.3.4 Regularized Newton methods

Several modifications to Newton's method have been proposed to address ill-conditioning of the Hessian and negative eigenvalues. One family of methods takes the absolute value and/or thresholds the eigenvalues of \mathbf{H} prior to inversion [52, 53, 63, 64]. We adopt one such technique called the non-convex Newton (NCN) method [53], which conditions the gradient using the "positive definite truncated inverse Hessian," denoted $|\mathbf{H}|^{-1}$. NCN steps are given by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta |\mathbf{H}|^{-1} \mathbf{g}. \quad (2.24)$$

To compute $|\mathbf{H}|^{-1}$, we perform an eigendecomposition of the Hessian,

$$\mathbf{H} = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top, \quad (2.25)$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ contains the ordered eigenvalues and \mathbf{Q} is an orthonormal matrix of the corresponding eigenvectors. The eigenvalues are regularized:

$$\lambda'_i = \max(|\lambda_i|, \delta), \quad (2.26)$$

where $\delta \in (0, \lambda_1]$ is a threshold, which yields the diagonal matrix $|\boldsymbol{\Lambda}| = \text{diag}(\lambda'_1, \dots, \lambda'_n)$. Note that $\delta > 0$ ensures that $|\mathbf{H}|$ is full rank. The regularized inverse is

$$|\mathbf{H}|^{-1} = \mathbf{Q} |\boldsymbol{\Lambda}|^{-1} \mathbf{Q}^\top. \quad (2.27)$$

The threshold δ controls the condition number of $|\mathbf{H}|^{-1}$, which equals λ_1/δ .

Unlike Newton’s method, NCN can descend along directions of negative curvature and is proven to escape saddle points exponentially fast [53]. Furthermore, because $|\mathbf{H}|^{-1}$ is inherently positive definite, NCN can always decrease the loss function for a sufficiently small step, provided that $\mathbf{g} \neq \mathbf{0}$. Since NCN can handle ill-conditioned Hessians and negative concavity, and because the Hessian is easy to compute for our KS state estimation problem, NCN is the default optimizer in this work.

Selection of the threshold δ and step size η are key considerations when using NCN. We first set $\delta = \lambda_1/\kappa_{|\mathbf{H}|}$, where $\kappa_{|\mathbf{H}|}$ is the targeted condition number, and we then determine η through a standard backtracking line search that satisfies the Armijo condition. Setting $\kappa_{|\mathbf{H}|}$ to 1 removes curvature-based scaling from the step direction, i.e., $|\mathbf{H}|^{-1}\mathbf{g} = \lambda_1^{-1}\mathbf{g}$, which is equivalent to gradient descent (i.e., robust to noise in \mathbf{g} but slow and prone to stagnation near critical points). Conversely, as $\delta \rightarrow 0$, we have $\kappa_{|\mathbf{H}|} \rightarrow \kappa_{\mathbf{H}}$ and all components of the Hessian are retained. However, when the Hessian is ill-conditioned, NCN steps amplify gradient components aligned with directions of low curvature, which we observe are typically associated with high-wavenumber components in \mathbf{u}_0 (see Sec. 4). Such components are usually non-physical because the hyper-diffusion term (II) in Eq. (2.1) rapidly damps high-wavenumber content and the true state lies near the attractor (by assumption), which primarily contains low-wavenumber modes. Selection of $\kappa_{|\mathbf{H}|}$ is thus a balancing act: it must be large enough to accelerate convergence by exploiting curvature information but small enough to avoid amplifying non-physical gradient components. In practice, we constrain $\kappa_{|\mathbf{H}|}$ to $\{10^3, 10^5\}$ to simplify the regularization procedure.

We set $\kappa_{|\mathbf{H}|} = 10^3$ at the start of each optimization, which biases updates toward states dominated by low-wavenumber modes. However, if $\mathbf{g}^\top |\mathbf{H}|^{-1}\mathbf{g}$ becomes much smaller than the objective loss, optimization effectively stalls (see Sec. 5.5). To address this, whenever $\mathcal{J}^{-1}(\mathbf{g}^\top |\mathbf{H}|^{-1}\mathbf{g}) < 0.01$, we increase $\kappa_{|\mathbf{H}|}$ to 10^5 , enabling larger steps along “higher-wavenumber directions.” This adjustment typically gets the optimizer unstuck and reduces the loss, though it also tends to increase early-time reconstruction errors.

2.4 Pseudo-projection

Most initial conditions in state space do not lie on the attractor, and many points off the attractor converge toward trajectories that are nearly indistinguishable from those on it. As a result, when optimizing the initial condition, the estimate can drift away from the attractor along directions that yield similar trajectories but contain erroneous early transients. This tendency is exacerbated when $\kappa_{|\mathbf{H}|}$ is large, since curvature scaling amplifies gradient components aligned with directions of low curvature (typically associated with high-wavenumber content in \mathbf{u}_0 , as discussed above). We refer to this behavior as a blow-up of the initial condition, since values of \mathbf{u}_θ can become unphysically large along these directions.

Because all trajectories approach the attractor exponentially fast, it is reasonable to assume that the true initial condition lies on the attractor or very near it. To incorporate this knowledge into our state estimation algorithm and to prevent blow-up, we introduce *pseudo-projection*. This operation projects a state onto the attractor (or at least close to it) by integrating the governing equations forward in time for a short duration. Pseudo-projection is given by

$$\boldsymbol{\theta}_{k+1} = \mathbb{F} \left\{ \underbrace{f_{\Delta t}^j[\mathbb{F}^{-1}(\boldsymbol{\theta}_k)]}_{\mathbf{u}_\theta} \right\}, \quad (2.28)$$

where j is chosen such that $j\Delta t \ll T$. By keeping the rollout short, pseudo-projection acts as a dynamical filter that removes non-physical, high-wavenumber modes while leaving the long-time trajectory essentially unchanged. We use pseudo-projection in conjunction with NCN. The optimization is performed for 350 iterations, with pseudo-projection applied at steps 50, 100, and 150. We set $\kappa_{|\mathbf{H}|} = 10^3$ after the final application of pseudo-projection, allowing the optimizer to refine physical modes.

3 Test cases and sample reconstructions

This section presents representative test cases to illustrate typical outcomes of adjoint state estimation. We begin by describing the procedure used to generate the dataset of trajectories and initial guesses employed throughout our study, followed by the error metrics used to assess reconstruction quality. We then report representative reconstructions for the $L = 22$ domain to contextualize these metrics, compare optimizer performance, and demonstrate the effects of pseudo-projection, in that order.

3.1 Generation of cases

Adjoint state estimation for KS systems is a highly non-convex problem that depends strongly on both the reference trajectory and the initial guess for the observer trajectory. To marginalize these dependencies and obtain representative reconstruction statistics, we perform reconstructions across a large ensemble of ground truth trajectories and guesses. For each domain size, $L \in \{22, 44, 66\}$, we generate a collection of states on the attractor by integrating a single system forward for 10 000 time units. The first 1000 time units are discarded to ensure convergence to \mathcal{A} , and the remaining 9000 time units are retained at intervals of $\Delta t = 1$. The center and radius of the attractor are approximated as

$$\mathbf{u}_{\mathcal{A}} \approx \frac{1}{9000} \sum_{k=1001}^{10000} \mathbf{u}_k, \quad R_{\mathcal{A}} \approx \frac{1}{9000} \sum_{k=1001}^{10000} \|\mathbf{u}_k - \mathbf{u}_{\mathcal{A}}\|_2, \quad (3.1)$$

where k indicates time units. The radius serves as a characteristic scale in state space, and we use it to normalize errors and sample initial guesses at prescribed distances from the true initial condition.

For each domain size, we define test cases using 20 random initial conditions and 400 random initial guesses per initial condition, yielding 8000 cases per domain. All cases are reconstructed using data from multiple sensor configurations, with $m_x \in \{2, \dots, 16\}$ spatial sensors and $m_t \in \{2, \dots, 8\}$ measurement times. The initial conditions $\mathbf{u}_0^{(i)}$ and guesses $\mathbf{u}_{\theta}^{(ij)}$ are drawn from the long-time rollout for the corresponding domain. When generating guesses for a given condition, we compute the distances

$$D_{ij} = \|\mathbf{u}_{\theta}^{(ij)} - \mathbf{u}_0^{(i)}\|_2. \quad (3.2)$$

We sample states with distances $D_{ij} \in [0.01R_{\mathcal{A}}, R_{\mathcal{A}}]$ to ensure a mixture of good guesses and poor ones. To do this, random target distances are drawn from $[0.01R_{\mathcal{A}}, R_{\mathcal{A}}]$ with uniform probability, and the state for which D_{ij} most closely matches the sample is selected. Duplicates are redrawn until we have 400 unique starting points for our observer system.

3.2 Error metrics

Reconstruction accuracy is primarily evaluated using two metrics: a normalized Euclidean distance between initial conditions of the observer and reference systems as well as the cosine similarity between the full trajectories. The initial condition error is

$$e_u = R_{\mathcal{A}}^{-1} \|\mathbf{u}_0 - \mathbf{u}_{\theta}\|_2. \quad (3.3)$$

Because distinct initial conditions can yield nearly indistinguishable trajectories on the attractor, we also quantify accuracy at the trajectory level via the cosine similarity,

$$\text{CS}_{\mathbf{U}} = \frac{\mathbf{U}^{\top} \mathbf{U}_{\theta}}{\|\mathbf{U}\|_2 \|\mathbf{U}_{\theta}\|_2}, \quad (3.4)$$

where $\mathbf{U} = (\mathbf{u}_0; \dots; \mathbf{u}_K)$ and $\mathbf{U}_{\theta} = (\mathbf{u}_{\theta}; \dots; \mathbf{u}_{\theta,K})$ are the ground truth and reconstructed trajectories in \mathbb{R}^{nK} .

We also consider whether an embedding is well conditioned by computing the largest loss below which trajectory estimates are accurate with high probability. Specifically, we define

$$\varepsilon^* = \sup \left\{ \varepsilon \mid \underbrace{p(\overbrace{\text{CS}_{\mathbf{U}} \geq \tau}^{\text{accurate est.}} \mid \overbrace{\mathcal{J} < \varepsilon}^{\text{of low loss}})}_{\text{with high probability}} \geq 1 - \delta \right\}, \quad (3.5)$$

where $\tau \approx 1$ indicates an accurate trajectory and $0 < \delta \ll 1$. Thus, any loss below ε^* almost certainly corresponds to an accurate reconstruction. Equivalently, let $\mathbf{U}_{\mathbf{A}}$ and $\mathbf{U}_{\mathbf{B}}$ denote trajectories initialized at $\mathbf{u}_{\mathbf{A}}$ and $\mathbf{u}_{\mathbf{B}}$, respectively, with measurements $\mathbf{y}_{\mathbf{A}} = \Phi(\mathbf{u}_{\mathbf{A}})$ and $\mathbf{y}_{\mathbf{B}} = \Phi(\mathbf{u}_{\mathbf{B}})$. Well-conditioned embeddings require that trajectories which are also close in measurement space are necessarily close in state space. In particular, accurate estimation occurs whenever

$$\frac{1}{m} \|\mathbf{y}_{\mathbf{A}} - \mathbf{y}_{\mathbf{B}}\|_2 < \varepsilon^*.$$

For state estimation to be well posed, ε^* should exist and it should be reasonably large.

3.3 Representative reconstructions

State estimation has three characteristic outcomes: poor generalizations, failed optimizations, and successful reconstructions. Figure 4 presents examples of all three for the $L = 22$ domain. The top row shows reconstructed trajectories with sensor positions superimposed on the estimates and the bottom row shows absolute error fields. All reconstructions were computed using the same reference state \mathbf{u}_0 and the same optimizer configuration, namely, our default sequence of 350 NCN iterations with pseudo-projection applied at iterations 50, 100, and 150. The only differences among the examples in Fig. 4 are the number of observations and the initial guess. For the cases shown (left to right), the initial distances D_{ij} are 0.58, 0.80, and 0.89. The values of \mathcal{J} and CS_U reported in this subsection correspond to these reconstructions.

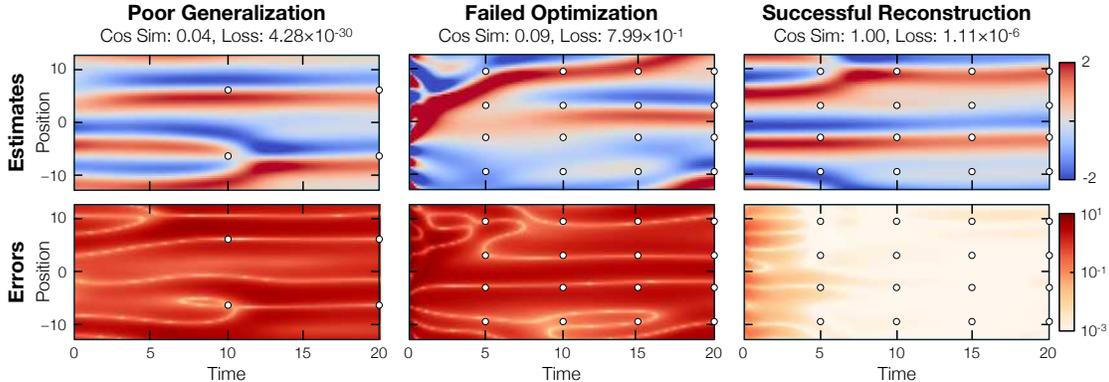


Figure 4: Sample reconstructions for $L = 22$. The top row shows reconstructed trajectories with sensor locations superimposed; the bottom row shows absolute error fields. Final loss values and cosine similarities are reported above each column. From left to right: poor generalization (low loss, high error), failed optimization (high loss, high error), and successful reconstruction (low loss, low error).

Poor generalization (left) occurs when the observations are sparse ($m_x = 2$ and $m_t = 2$) and the initial guess lies far from the truth. Although the optimizer drives the loss to an extremely low value (4.3×10^{-30}), the reconstructed trajectory differs markedly from the reference, yielding a cosine similarity of 0.04. The problem is underdetermined: many trajectories can reproduce this sparse set of observations, so a low loss does not imply an accurate reconstruction. Indeed, as shown in Sec. 4.3.2, even for very good guesses, as $D_{ij} \rightarrow 0$, one generally requires $m \geq d_M$ measurements for the initial state to be observable, where $d_M = 8$ for the $L = 22$ domain. In this case, the error field exhibits two shallow valleys of extremely low loss centered on the sensor positions, which is a common feature of low-sensor-count reconstructions.

Even with a denser set of observations, the optimization can still fail, as shown in the middle panel. Here, the measurement density ($m = 16$) is close to the embedding criterion $m \geq 2d_M + 1$ discussed in Sec. 4.3.3, yet the optimizer converges to a spurious solution with a high loss (7.9×10^{-1}) and a low cosine similarity (0.09). Once again, the error field contains valleys of low loss, though not nearly as deep as in the first reconstruction, and the reconstructed field bears little resemblance to the true system. Such failures arise when the optimizer becomes trapped in high plateaus on the loss landscape, stalling convergence.

Lastly, the right panel shows a successful reconstruction for the same sensor configuration as the middle panel, where the optimizer converges to a physically consistent solution with low loss (1.1×10^{-6}) and a cosine similarity near unity. Notably, the initial guess in this case was *further* from \mathbf{u}_0 than in the second example ($D_{ij} = 0.89$ as compared to 0.80). The relationship between the initial separation and the conditions required for accurate reconstruction is discussed in detail in Secs. 4 and 5.

3.4 Characteristic optimizer behavior

To illustrate the behavior of different optimizers, Fig. 5 shows representative loss traces for a case with $m_x = 4$, $m_t = 4$, and $L = 22$, where all methods start from the same initial guess. We compare gradient descent, BFGS, and NCN, as described in Sec. 2.3, but we do not include a bona fide Newton method. Due to dissipative dynamics and measurement sparsity the true Hessian is very ill-conditioned or entirely degenerate, per Sec. 5, so exact Newton steps are not well defined. Even with minimal regularization—i.e., retaining only non-zero eigenvalues in Eq. (2.27) (setting $\lambda_i^{-1} = 0$ when $\lambda_i = 0$) without enforcing positivity

or applying a threshold—Newton iterations quickly diverge. To isolate the effects of negative curvature in our comparison, therefore, we include a modified Newton scheme that applies the same cutoff as NCN but does not enforce positivity of Λ . Pseudo-projections are omitted from these tests to highlight the intrinsic behavior of each optimizer.

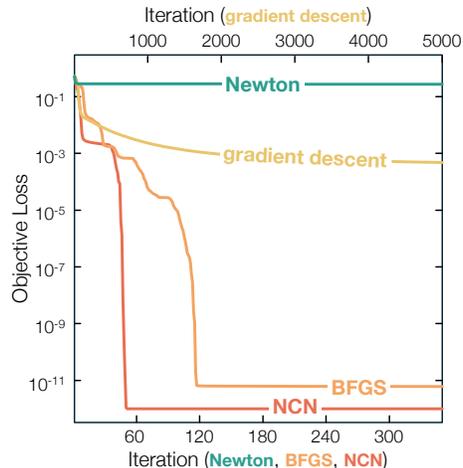


Figure 5: Optimization loss versus iteration for gradient descent, modified Newton, BFGS, and NCN applied to the same $L = 22$ case with $m_x = 4$ and $m_t = 4$. The lower axis (0–350 iterations) corresponds to Newton, BFGS, and NCN; the upper axis (0–5000 iterations) corresponds to gradient descent.

Due to severe ill-conditioning of the loss landscape, gradient descent converges at a glacial pace: even after 5000 iterations, its loss remains orders of magnitude higher than those achieved by BFGS and NCN in far fewer steps (note the separate x -axes). The modified Newton method also plateaus, despite having access to exact curvature information. Because the Hessian can become indefinite, Newton steps point uphill whenever the gradient overlaps with directions of negative curvature. Such steps would increase the loss, so the backtracking line search sets the step size to zero, causing the optimizer to get stuck. By contrast, BFGS preconditions the gradient with a positive definite matrix B_k^{-1} , ensuring that the loss necessarily decreases provided the step is sufficiently small. Even when the curvature condition fails, our implementation resets B_k to I , which is positive definite and thus allows descent to continue. Finally, NCN exhibits the fastest convergence and the lowest final loss, in line with its relative performance across all the test cases we examined.

3.5 Reconstructions with pseudo-projection

Recall that pseudo-projection involves a short forward integration of the system dynamics that is meant to bring \mathbf{u}_θ closer to \mathcal{M} . Figure 6 illustrates its effect for a representative case with $m_x = 4$, $m_t = 4$, and $L = 22$. As throughout this paper, we use 350 NCN iterations with pseudo-projection applied at iterations 50, 100, and 150. The plots compare two otherwise identical DA runs: one with pseudo-projection (solid lines) and the other without (dashed lines). The left panel shows traces of the loss \mathcal{J} and cosine similarity CS_U , the middle panel shows the loss and the initial condition error e_u , and the right panel displays initial conditions of the reconstructed and reference systems (top right) as well as the residuals (bottom right).

Pseudo-projection events are indicated by vertical dotted lines. At each instance, the loss spikes up and the cosine similarity dips down. Both effects are expected because the action of the system dynamics per se does not account for observations of the reference system. By contrast, the initial condition error consistently decreases with pseudo-projection, meaning that the dynamics do indeed pull \mathbf{u}_θ toward \mathcal{M} . Notably, the loss always remains below its initial value after projection, and the cosine similarity stays relatively high. Hence, pseudo-projection moves \mathbf{u}_θ closer to the IM without fundamentally degrading the estimated trajectory.

Between pseudo-projections, the optimizer makes limited progress in reducing e_u , mainly due to the small number of measurements and the moderately high NCN threshold used in this work. The case without pseudo-projection clearly highlights this limitation: although the optimization achieves a low loss, the final e_u is worse than at initialization, and the cosine similarity is lower than in the pseudo-projection case. Thus,

for this example, pseudo-projection yields a more accurate trajectory even though the final measurement match is slightly worse. More broadly, pseudo-projection almost always makes the optimization harder for a few steps, i.e., because the spike in loss must be brought back down, but it also introduces perturbations that help the optimizer to escape plateaus or shallow valleys in \mathcal{J} . Across all of our tests, we find that pseudo-projection is the primary mechanism for reducing e_u .

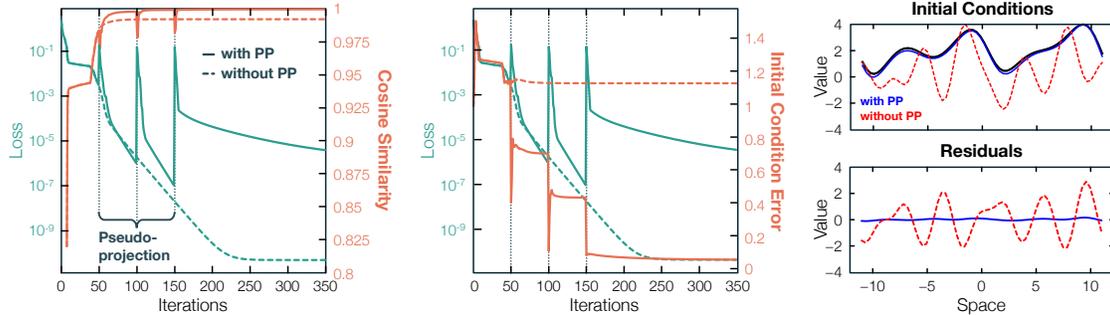


Figure 6: Effect of pseudo-projections for a case with $m_x = 4$, $m_t = 4$, and $L = 22$. Shown are the loss and cosine similarity (left), the loss and initial-condition error (middle), true and estimated initial conditions (top right), and initial condition residuals (bottom right). Vertical dashed lines indicate pseudo-projection steps, which reduce the initial-state error by nudging the estimate back toward the attractor.

Next, we examine the global effect of pseudo-projection using all 8000 trials of the $m_x = 4$, $m_t = 4$, $L = 22$ case. Figure 7 shows joint probability density functions (PDFs) $p(\text{CS}_U \geq \tau, \mathcal{J})$, evaluated for $\tau = 0.95$. The left and middle panels compare optimizations performed with pseudo-projection (left) and without it (middle). The right panel shows the same analysis, but we restrict the cosine similarity metric to the latter 75% of the trajectory, thereby excluding early-time transients that take place before the first measurement time in \mathcal{T} . For each plot, we also indicate the supremum threshold ϵ^* from Eq. (3.5), computed using $\delta = 0.001$.

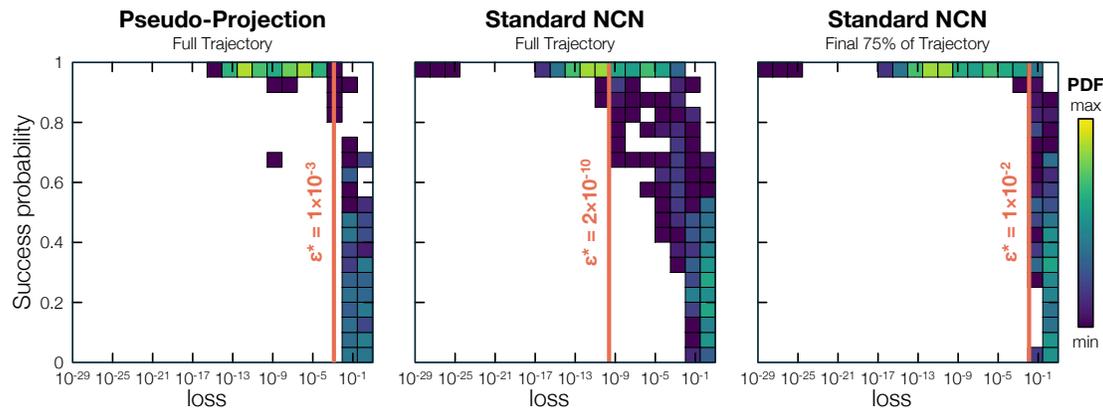


Figure 7: State estimation with and without pseudo-projections for $L = 22$. Each panel shows $p(\text{CS}_U = \tau, \mathcal{J})$ in the (\mathcal{J}, τ) plane. From left to right: with pseudo-projection, without pseudo-projection, and without pseudo-projection but computing CS_U using only the latter 75% of the trajectory. Vertical lines indicate ϵ^* .

Large values of ϵ^* indicate that, for a given optimization scheme, low-loss solutions correspond to accurate reconstructions with high probability. Naturally, small values of ϵ^* suggest the opposite, where low-loss solutions can arise from distinct trajectories that nearly match the reference observations. Such cases can occur when trajectories begin off the IM but closely shadow trajectories on it. This behavior is apparent in the middle panel of Fig. 7, wherein many runs produce a low loss yet have a low probability of accurate reconstruction. With pseudo-projection (left), these spurious *low-loss–low-accuracy* cases are greatly reduced, and ϵ^* is much higher. We interpret this increase as improved numerical robustness due to

pseudo-projection, since the underlying problem is unchanged across these cases. The right panel confirms that the difficulty originates in early-time reconstruction errors. When the cosine similarity is only computed for the latter 75% of the trajectory, the bulk of the problematic low-loss–low-accuracy region vanishes. This suggests that pseudo-projection primarily improves observability of the initial condition, as opposed to the full trajectory, by pulling it closer to the IM.

4 Degree-of-freedom effects

In this section, we assume the existence of a smooth compact manifold in state space that contains the global attractor with minimal possible dimension and on which the flow map is a diffeomorphism. In other words, we assume an *inertial manifold*, which is known to exist for KS systems. It is also reasonable to posit its existence for more complex dissipative systems such as Navier–Stokes flows, where an IM has not yet been rigorously proven but is expected to exist [65]. The results of this section therefore have potential applicability to such systems. A detailed discussion of the existence and properties of IMs in dissipative systems is provided by Zelik [31].

Kuramoto–Sivashinsky dynamics on the IM can be expressed as a system of $d_{\mathcal{M}}$ ordinary differential equations. Trajectories are determined by state vectors in $\mathbb{R}^{d_{\mathcal{M}}}$ that specify initial positions on \mathcal{M} . Hence, $d_{\mathcal{M}}$ provides a natural measure of the information necessary to define the system state. Given sufficient knowledge of the system dynamics, $d_{\mathcal{M}}$ should correspond to the number of measurements m needed for state estimation. Embedding theory formalizes this connection by relating $d_{\mathcal{M}}$ to the m required for a smooth, invertible mapping $\Phi : \mathcal{M} \rightarrow \mathbb{R}^m$ to exist, which holds when $m \geq 2d_{\mathcal{M}} + 1$. While embedding theory has been widely applied to state space reconstruction, we employ it here for the first time to analyze the well-posedness of adjoint state estimation. Because $d_{\mathcal{M}}$ increases with the domain length, the number of measurements required for reconstruction likewise grows. We refer to this dependence as a *degree-of-freedom effect on observability*.

Figure 8 provides a graphical summary of the spaces relevant to state estimation and their relationships to one another. At the center lies the inertial manifold \mathcal{M} , which contains the system’s long-time dynamics and is assumed to include both the reference trajectory starting at u_0 and the observer trajectory starting at u_θ . To the right appears the measurement manifold $\mathcal{Y} = \Phi(\mathcal{M})$, where the observation operator Φ maps states $u \in \mathcal{M}$ to measurements $y \in \mathcal{Y}$. When Φ is an embedding, this mapping is smooth and invertible, so \mathcal{Y} and \mathcal{M} are topologically equivalent. On the left is a local embedding: a chart ψ defined on an open patch $\mathcal{U} \subset \mathcal{M}$ that maps $u \in \mathcal{U}$ to manifold coordinates $z \in \mathcal{V} = \psi(\mathcal{U}) \subset \mathbb{R}^{d_{\mathcal{M}}}$. By definition, such patches form an open cover of \mathcal{M} . The remainder of this section develops these spaces and mappings and substantiates their role in adjoint state estimation.

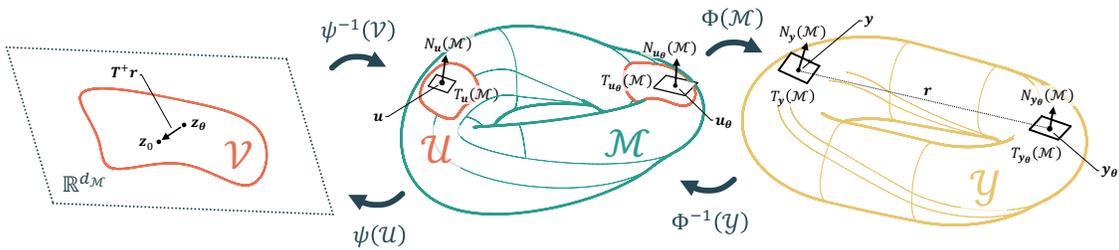


Figure 8: Schematic illustrating the relationships among the inertial manifold \mathcal{M} , the measurement manifold \mathcal{Y} , and a local Euclidean parametrization \mathcal{V} . The observation map Φ takes states on \mathcal{M} to measurements on \mathcal{Y} , and the chart ψ provides local coordinates on $\mathcal{U} \subset \mathcal{M}$. Tangent and normal spaces are shown for representative states and observations.

4.1 Mappings from states to measurements

We begin by introducing the notation used in this section and by briefly reviewing some relevant aspects of embedding theory. Consider the mapping

$$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad (4.1)$$

which takes an initial condition in state space to a vector of m observations,

$$\Phi(\mathbf{u}) = [h_1(\mathbf{u}); h_2(\mathbf{u}); \dots; h_m(\mathbf{u})] = \mathbf{y}. \quad (4.2)$$

To analyze the properties of Φ , we restrict its domain to states on \mathcal{M} , such that $\mathcal{Y} = \Phi(\mathcal{M})$ defines the corresponding shadow manifold. We denote by \mathcal{P}^m the space of all such mappings with output dimension m . When the domain is further restricted to a subset $\mathcal{B} \subset \mathcal{M}$, we write

$$\Phi_{\mathcal{B}} : \mathcal{B} \rightarrow \mathcal{Y}. \quad (4.3)$$

While the forward problem $\mathbf{y} = \Phi(\mathbf{u})$ is well posed, the inverse problem $\mathbf{u} = \Phi^{-1}(\mathbf{y})$ may not be, since Φ^{-1} may not exist and is typically unavailable in closed form, regardless. Adjoint state estimation implicitly approximates this inverse through a constrained optimization. The problem can be well posed when each \mathbf{y} corresponds to a unique initial condition, which holds for all $\mathbf{u} \in \mathcal{M}$ when Φ is an embedding, ensuring that Φ^{-1} does indeed exist.

Numerous results in the literature on state space reconstruction establish bounds on the number of measurements required for an embedding to exist. Takens' pioneering work showed that if \mathbf{y} is a scalar time series obtained from $\mathbf{u} \in \mathcal{M}$,⁴ then the delay-coordinate map $\Phi : \mathbf{u} \mapsto \mathbf{y}$ is a diffeomorphic embedding when the number of delays satisfies $m \geq 2d_{\mathcal{M}} + 1$ [66, 67]. Sauer, Yorke, and Casdagli [68] later extended this result to strange attractors, showing that a generic observation function yields an embedding when $m > 2d_{\mathcal{A}}$. Deyle and Sugihara [33] further generalized these results to multivariate time series, demonstrating that if $m \geq 2d_{\mathcal{M}} + 1$, then $\Phi \in \mathcal{P}^m$ is generically an embedding for sufficiently smooth measurement functions, under mild assumptions about periodic points. They also showed that in a probabilistic formulation, any $\Phi_{\mathcal{B}} \in \mathcal{P}^m$ is almost surely an embedding when $m > 2d_{\mathcal{B}}$, where $d_{\mathcal{B}} < d_{\mathcal{M}}$ is the box-counting dimension of a compact subset $\mathcal{B} \subset \mathcal{M}$. Finally, if Φ is an embedding, then \mathcal{M} and \mathcal{Y} are topologically equivalent, so that $d_{\mathcal{Y}} = d_{\mathcal{M}}$, which suggests a lower bound $m \geq d_{\mathcal{M}}$ on the number of measurements required for state estimation. The implications of $m \geq d_{\mathcal{M}}$ and $m \geq 2d_{\mathcal{M}} + 1$ for adjoint state estimation are derived in Sec. 4.3. Lastly, we note that these theorems establish when Φ is *almost always* an immersion or an embedding, but they do not provide universal guarantees.

4.2 State space reconstruction

In state space reconstruction, the goal is to determine invariants of a dynamical system from sparse measurements and to predict their evolution [69]. If Φ is an embedding, then one can define dynamics of \mathbf{y} on \mathcal{Y} that are equivalent to the dynamics of \mathbf{u} on \mathcal{M} such that both systems share the same invariants. The measurement space dynamics are written as

$$\mathbf{y}_{k+1} = \mathbf{g}_{\Delta t}(\mathbf{y}_k), \quad (4.4)$$

where $\mathbf{g}_{\Delta t}$ is simply

$$\mathbf{g}_{\Delta t} = \Phi \circ \mathbf{f}_{\Delta t} \circ \Phi^{-1}. \quad (4.5)$$

Hence, the system dynamics can be examined entirely in measurement space when Φ is an embedding. The existence of Φ^{-1} is sufficient for this purpose, which stands in contrast to state estimation, where we seek a functional approximation to Φ^{-1} .

4.2.1 Sensor placement and repetition rate

Although embedding theorems specify how many measurements are needed to establish a diffeomorphic mapping from \mathcal{M} to \mathcal{Y} , they offer no guidance on where to place sensors or how rapidly they should record observations of \mathbf{u} . These are critical considerations in practice, especially for noisy measurements [32]. A standard approach for selecting the time lag τ between measurements is to analyze the average mutual information between measurements at times t and $t + \tau$. The lag is often chosen as the first minimum of mutual information with increasing τ [70], minimizing redundancy while ensuring that successive measurements are still correlated [71, 72]. Alternatively, one can fix the measurement time horizon T , from which τ is determined by the number of measurements as $\tau = T/m$. Rosenstein et al. [72] showed that the optimal lag scales with m such that T remains approximately constant. By fixing T , one can keep the earliest and latest measurements within a window where their dynamical relationship remains

⁴A vector of delay coordinates contains measurements of u at a fixed spatial position x and at times $t, t + \tau, t + 2\tau, \dots$

computable. A myriad of methods have been proposed to optimize T or τ [72–74], all aiming to strike a reasonable compromise between *redundancy* (short τ , strongly correlated measurements) and *irrelevance* (long τ , decorrelated measurements that convey little information about the initial state) [34].

For adjoint state estimation, prior studies recommended restricting the assimilation window to the Lyapunov time T_ℓ [49, 50]. Beyond this scale, the exponential sensitivity to initial conditions causes gradient calculations to rapidly deteriorate. When $K\Delta t \gg T_\ell$, therefore, the probability density $p(\mathbf{u}_K | \mathbf{u}_0)$ approaches the unconditional distribution $p(\mathbf{u}_0)$, and as a corollary, $p(\mathbf{u}_0 | \mathbf{u}_K) \rightarrow p(\mathbf{u}_0)$ since the adjoint system inherits the Lyapunov spectrum of the forward dynamics. Therefore, setting $T = T_\ell$ is a pragmatic choice for the assimilation window, independent of m , and we follow this convention throughout the present work. Because the literature on state space reconstruction primarily concerns 1D time series, however, there is little precedent for spatial sensor placement. We thus adopt uniform spatial coverage under the assumption that all spatial locations are equally informative.

4.3 Critical points on \mathcal{J}

In Sec. 2.3, we discuss evidence that critical points of high loss are rare in high-dimensional non-convex optimization problems, while critical points of low loss are typically saddle points or global minima (possibly with multiple minima of equal loss). Here, we examine the conditions under which critical points can arise in the loss landscape on \mathcal{M} . We show that, under suitable assumptions, the global minimum is the only critical point on manifold, which holds locally for $m \geq d_{\mathcal{M}}$ and globally for $m \geq 2d_{\mathcal{M}} + 1$.

4.3.1 Some definitions

Several geometric quantities must be defined to assess critical points on the IM. The mapping Φ must be an *immersion* to qualify as an embedding, and an *atlas of charts* is needed to parameterize the loss landscape. An immersion is simply a local embedding: for every $\mathbf{u} \in \mathcal{M}$, there exists a neighborhood such that $\mathbf{u} \in \mathcal{U} \subset \mathcal{M}$, wherein the restricted mapping $\Phi_{\mathcal{U}} : \mathcal{U} \rightarrow \mathcal{Y}$ is an embedding [75]. Immersions are known to exist generically when the number of measurements satisfies $m \geq d_{\mathcal{M}}$ [33].

Equivalently, immersions can be characterized using the *tangent spaces* of \mathcal{M} and \mathcal{Y} , which are depicted in Fig. 8. The tangent space of a smooth manifold \mathcal{B} at x is denoted $T_x(\mathcal{B})$, with a Euclidean dimension that necessarily equals the manifold dimension $d_{\mathcal{B}}$, and the normal space is $N_x(\mathcal{B})$. For states \mathbf{u} and measurements \mathbf{y} , the Jacobian of Φ with respect to \mathbf{u} must relate both the tangent and normal spaces of \mathcal{M} and \mathcal{Y} ,

$$\frac{\partial \Phi}{\partial \mathbf{u}} : \underbrace{T_{\mathbf{u}}(\mathcal{M}) \oplus N_{\mathbf{u}}(\mathcal{M})}_{\mathbb{R}^n} \rightarrow \underbrace{T_{\mathbf{y}}(\mathcal{Y}) \oplus N_{\mathbf{y}}(\mathcal{Y})}_{\mathbb{R}^m}. \quad (4.6)$$

The immersion property pertains solely to the restricted mapping

$$\left(\frac{\partial \Phi}{\partial \mathbf{u}} \right)_{T_{\mathbf{u}}(\mathcal{M})} : T_{\mathbf{u}}(\mathcal{M}) \rightarrow T_{\mathbf{y}}(\mathcal{Y}). \quad (4.7)$$

If Φ is an immersion, this mapping is *injective*, which implies that $d_{\mathcal{M}} \leq d_{\mathcal{Y}}$. Since the measurement manifold is the image of \mathcal{M} , i.e., $\mathcal{Y} = \Phi(\mathcal{M})$, we also have $d_{\mathcal{Y}} \leq d_{\mathcal{M}}$. Hence, $d_{\mathcal{M}}$ must equal $d_{\mathcal{Y}}$ and the tangent map is a bijection. A first-order Taylor series expansion of Φ gives

$$\delta \mathbf{y} \approx \frac{\partial \Phi}{\partial \mathbf{u}} \delta \mathbf{u}, \quad (4.8)$$

so if Φ is an immersion, then any non-zero perturbation $\delta \mathbf{u} \in T_{\mathbf{u}}(\mathcal{M})$ produces a non-zero measurement perturbation $\delta \mathbf{y} \in T_{\mathbf{y}}(\mathcal{Y})$ and vice versa. In other words, as an immersion, Φ resolves all the intrinsic directions on the inertial and shadow manifolds.

Next, we define an atlas, which allows us to map from a state $\mathbf{u} \in \mathcal{M}$ to a vector of manifold coordinates $\mathbf{z} \in \mathbb{R}^{d_{\mathcal{M}}}$ corresponding to the system’s intrinsic degrees of freedom. An atlas is a collection of charts whose domains $\mathcal{U} \subset \mathcal{M}$ form an open cover of \mathcal{M} , where \mathcal{U} is one of many such domains. Each chart

$$\psi : \mathcal{U} \rightarrow \mathcal{V} \subset \mathbb{R}^{d_{\mathcal{M}}} \quad (4.9)$$

is a diffeomorphism from \mathcal{U} onto an open subset of $\mathbb{R}^{d_{\mathcal{M}}}$, such that $\mathbf{u} = \psi^{-1}(\mathbf{z})$ for $\mathbf{z} \in \mathcal{V}$. The tangent space of a manifold can also be obtained by differentiating a chart. Specifically,

$$T_{\mathbf{u}}(\mathcal{M}) = \text{span} \left(\frac{\partial \psi^{-1}}{\partial \mathbf{z}} \right), \quad (4.10)$$

where $\mathbf{u} \in \mathcal{U}$, $\mathbf{z} \in \mathcal{V}$, and the rank of the Jacobian is $d_{\mathcal{M}}$. An example of this mapping is shown in Fig. 8.

4.3.2 Local behavior

To begin, we show that if Φ is an immersion, and if the initial conditions of the observer and reference systems, \mathbf{u}_θ and \mathbf{u}_0 , are confined to a sufficiently small region on a chart domain \mathcal{U} , then there exists a single critical point in this region at $\mathbf{u}_\theta = \mathbf{u}_0$. The proximity of \mathbf{u}_θ and \mathbf{u}_0 is required to justify a first-order Taylor expansion. In what follows, $\mathbf{z}_\theta = \psi^{-1}(\mathbf{u}_\theta)$ and $\mathbf{z}_0 = \psi^{-1}(\mathbf{u}_0)$ are representations of \mathbf{u}_θ and \mathbf{u}_0 in manifold coordinates, with $\mathbf{z}_\theta, \mathbf{z}_0 \in \mathcal{V} \subset \mathbb{R}^{d_{\mathcal{M}}}$. We thus define the measurements as a function of \mathbf{z}_0 ,

$$\mathbf{y} = \Phi \circ \psi^{-1}(\mathbf{z}_0), \quad (4.11)$$

and so too for \mathbf{y}_θ and \mathbf{z}_θ . The first-order Taylor series expansion about \mathbf{z}_θ gives

$$\mathbf{y} = \mathbf{y}_\theta + \mathbf{T}(\mathbf{z}_0 - \mathbf{z}_\theta), \quad (4.12)$$

where

$$\mathbf{T} = \frac{\partial \Phi}{\partial \mathbf{u}_\theta} \frac{\partial \psi^{-1}}{\partial \mathbf{z}_\theta}. \quad (4.13)$$

The row space of $\partial \Phi / \partial \mathbf{u}_\theta \in \mathbb{R}^{m \times n}$, which has a rank greater than or equal to $d_{\mathcal{M}}$, contains $T_{\mathbf{u}_\theta}(\mathcal{M})$ since Φ is an immersion. The column space of $\partial \psi^{-1} / \partial \mathbf{z}_\theta \in \mathbb{R}^{n \times d_{\mathcal{M}}}$, whose rank is exactly $d_{\mathcal{M}}$, is identically $T_{\mathbf{u}_\theta}(\mathcal{M})$. Therefore, the rank of $\mathbf{T} \in \mathbb{R}^{m \times d_{\mathcal{M}}}$ is $d_{\mathcal{M}}$.

We see this scenario on the left side of Fig. 8. Two nearby states that fall within the same chart domain \mathcal{U} are mapped into $\mathcal{V} \subset \mathbb{R}^{d_{\mathcal{M}}}$, yielding manifold coordinates \mathbf{z}_0 and \mathbf{z}_θ . From Eq. (4.12), the points are separated by

$$\underbrace{(\mathbf{T}^\top \mathbf{T})^{-1}}_{\mathbf{T}^+} \mathbf{T}^\top \underbrace{(\mathbf{y} - \mathbf{y}_\theta)}_{\mathbf{r}} = \mathbf{z}_0 - \mathbf{z}_\theta,$$

where \mathbf{T}^+ is the pseudoinverse of \mathbf{T} and \mathbf{r} is the measurement residual.

The loss functional may be written as

$$\mathcal{J} = \frac{1}{2}(\mathbf{y} - \mathbf{y}_\theta)^\top (\mathbf{y} - \mathbf{y}_\theta) = \frac{1}{2} \mathbf{r}^\top \mathbf{r}, \quad (4.14)$$

which is equivalent to Eq. (2.14) up to a constant. Differentiating it with respect to \mathbf{z}_θ gives

$$\frac{\partial \mathcal{J}}{\partial \mathbf{z}_\theta} = \mathbf{r}^\top \mathbf{T}. \quad (4.15)$$

Substituting the first-order expansion from Eq. (4.12) yields

$$\frac{\partial \mathcal{J}}{\partial \mathbf{z}_\theta} = (\mathbf{z}_0 - \mathbf{z}_\theta)^\top \mathbf{T}^\top \mathbf{T}. \quad (4.16)$$

Note that $\text{rank}(\mathbf{T}^\top \mathbf{T}) = \text{rank}(\mathbf{T}) = d_{\mathcal{M}}$, so the gradient only vanishes when $\mathbf{z}_\theta = \mathbf{z}_0$. Hence, so long as Φ is an immersion, the only critical point local to the global minimum \mathbf{u}_0 is in fact \mathbf{u}_0 itself.

This result establishes a lower bound on the number of measurements required for state estimation. If the initial guess $\mathbf{u}_\theta \in \mathcal{M}$ is close enough to \mathbf{u}_0 to justify the Taylor expansion, then $\mathbf{u}_0 - \mathbf{u}_\theta \in T_{\mathbf{u}}(\mathcal{M})$ and the corresponding measurement residual satisfies $\mathbf{r} \in T_{\mathbf{y}}(\mathcal{Y})$. Consequently, gradient based optimization in this neighborhood is guaranteed to converge to \mathbf{u}_0 if \mathbf{T} is full rank, which is true when Φ is an immersion. Therefore, $m \geq d_{\mathcal{M}}$ gives the minimum number of measurements needed for \mathbf{u}_0 to be observable. This may be a practical limit for sequential smoothers or filters, where the initial guess for each segment can become accurate after several assimilation windows or analysis steps.

4.3.3 Global behavior

Next, we look into the properties of critical points when \mathbf{u}_θ and \mathbf{u}_0 need not be close. In particular, we show that \mathbf{u}_0 is the only critical point on \mathcal{M} . Starting from the gradient of Eq. (4.14) with respect to \mathbf{u}_θ ,

$$\frac{\partial \mathcal{J}}{\partial \mathbf{u}_\theta} = \mathbf{r}^\top \frac{\partial \Phi}{\partial \mathbf{u}_\theta}, \quad (4.17)$$

critical points arise either when $\mathbf{y} = \mathbf{y}_\theta$ or when the residual lies in the left null space of $\partial\Phi/\partial\mathbf{u}_\theta$. The column space of this Jacobian generically spans \mathbb{R}^m whenever $m \leq n$, so all critical points satisfy $\mathbf{y} = \mathbf{y}_\theta$.⁵ This conclusion holds even if Φ is not an immersion or an embedding. Going further, when Φ is indeed an embedding, then $\mathbf{y} = \mathbf{y}_\theta$ can occur only if $\mathbf{u}_\theta = \mathbf{u}_0$ for $\mathbf{u}_\theta, \mathbf{u}_0 \in \mathcal{M}$ since $\Phi : \mathcal{M} \rightarrow \mathcal{Y}$ is a bijection. Therefore, the only critical point on the manifold is the global minimum.

Recall that Φ is generically an embedding when $m \geq 2d_{\mathcal{M}} + 1$ [33]. Consequently, when m satisfies this bound, one might expect the state estimation problem to be well posed.

Regrettably, we note that the existence of a single critical point at \mathbf{u}_0 on \mathcal{M} does not guarantee convergence to that point via adjoint state estimation. Even when the optimization begins on \mathcal{M} , the gradient may have components orthogonal to $T_{\mathbf{u}_\theta}(\mathcal{M})$, pushing \mathbf{u}_θ off the manifold, where the above analysis no longer holds and where additional minima may exist. Constraining the optimization to \mathcal{M} by projecting \mathbf{g}_u onto the local tangent space—where $\mathbf{g}_u = \partial \mathcal{J} / \partial \mathbf{u}_\theta$ is the state space gradient—which is loosely approximated by our pseudo-projection procedure, helps to mitigate this issue. However, if \mathbf{g}_u happens to be in $N_{\mathbf{u}_\theta}(\mathcal{M})$, then NCN steps counteract pseudo-projection and even a true manifold-constrained (i.e., Riemannian) optimization would stall. Thus, although \mathbf{u}_0 is the only critical point on \mathcal{M} when Φ is an embedding, the gradient need not lie within the local tangent space, and specialized optimization strategies may be required to handle such pathologies.

To show that gradients can in fact point off the manifold, we invoke Whitney’s strong embedding theorem, which states that the measurement manifold $\mathcal{Y} = \Phi(\mathcal{M})$, of intrinsic dimension $d_{\mathcal{Y}} \leq d_{\mathcal{M}}$, can be smoothly embedded in a Euclidean space $D \subset \mathbb{R}^m$ of dimension d , where for any nonlinear manifold we have $d_{\mathcal{Y}} < d \leq \min(m, 2d_{\mathcal{Y}})$. Here, D represents the *minimal* Euclidean space that embeds \mathcal{Y} . The residual \mathbf{r} necessarily lies in D , and because $d > d_{\mathcal{Y}}$ for a nonlinear manifold, there must exist residuals with components in the normal space $N_{\mathbf{y}_\theta}(\mathcal{Y})$. A visual example of this is provided on the right-hand side of Fig. 8, where the residual \mathbf{r} does not fully reside in $T_{\mathbf{y}_\theta}(\mathcal{Y})$. In such instances, when Φ is an immersion or an embedding, \mathbf{g}_u necessarily contains components in $N_{\mathbf{u}_\theta}(\mathcal{M})$, as argued next.

For any immersion Φ , the restricted mapping

$$\left(\frac{\partial \Phi}{\partial \mathbf{u}_\theta} \right)_{T_{\mathbf{u}_\theta}(\mathcal{M})} : T_{\mathbf{u}_\theta}(\mathcal{M}) \rightarrow T_{\mathbf{y}_\theta}(\mathcal{Y})$$

is bijective, even though the full mapping between ambient spaces $\mathbb{R}^n \rightarrow \mathbb{R}^m$ need not be. Consequently, the gradient

$$\mathbf{g}_u = \left(\frac{\partial \Phi}{\partial \mathbf{u}_\theta} \right)^\top \mathbf{r}, \quad (4.18)$$

must take any component of \mathbf{r} that lies in $N_{\mathbf{y}_\theta}(\mathcal{Y})$ to $N_{\mathbf{u}_\theta}(\mathcal{M})$, because mapping such a component into $T_{\mathbf{u}_\theta}(\mathcal{M})$ would contradict the bijectivity of $\partial\Phi/\partial\mathbf{u}_\theta$ restricted to the tangent spaces. Thus, Whitney’s theorem implies that residuals with normal components exist, and therefore some gradients must point off the IM when Φ is an immersion or an embedding.

While the residual \mathbf{r} does not generally lie in $N_{\mathcal{Y}}(\mathcal{Y})$, there exist points on many manifolds for which this occurs. For instance, on a circular manifold, the displacement between any pair of antipodal points is normal to the manifold. We hypothesize that *some* such configurations could act as attractors in manifold-constrained optimization, posing a potential but likely uncommon pathology for adjoint state estimation.

⁵It has been shown that one can independently perturb each observation function h_i to obtain m linearly independent tangent vectors $\partial h_i / \partial \mathbf{u}_\theta$ [33]. Additional justification is required for cases with $m > n$.

4.4 Tangent spaces on \mathcal{M} and \mathcal{Y}

Even if Φ is an embedding, the stability of adjoint state estimation depends on two additional factors: (1) the numerical conditioning of the measurement map when restricted to the IM and (2) the extent to which gradients of the loss remain aligned with the manifold. Because both the reference and observer trajectories lie on \mathcal{M} , these effects are governed by the local geometric structure of \mathcal{M} and its image \mathcal{Y} . Up next, we empirically investigate the condition number of the Jacobian restricted to the tangent spaces of \mathcal{M} and \mathcal{Y} . Since good conditioning alone does not prevent the optimizer from drifting off the manifold, we then quantify how often gradients possess nontrivial components in the normal directions.

4.4.1 Conditioning of $\partial\Phi/\partial\mathbf{u}_\theta$ restricted to $T_u(\mathcal{M}) \rightarrow T_y(\mathcal{Y})$

To restrict our analysis of $\partial\Phi/\partial\mathbf{u}_\theta$ to the mapping between tangent spaces, we must construct a projection operator that maps $\mathbb{R}^n \rightarrow T_u(\mathcal{M})$ for any state $\mathbf{u} \in \mathcal{M}$. Theoretically, such an operator can be obtained by differentiating the inverse of a chart with respect to \mathbf{u} , since the span of this Jacobian equals $T_u(\mathcal{M})$, per Eq. (4.10). To this end, we first recall the definition of a chart:

$$\mathbf{z} = \psi(\mathbf{u}), \quad \mathbf{u} = \psi^{-1}(\mathbf{z}).$$

Although ψ and its inverse are not available in closed form, we approximate these mappings using the encoder E and decoder D introduced in Sec. 2.1 and detailed in Appendix C.

The autoencoder’s latent space \mathcal{L} generally has an oversized dimension, $d_{\mathcal{L}} \geq d_{\mathcal{M}}$. In order to obtain a reduced representation that is consistent with the manifold dimension, we perform a PCA on the latent states from our long-time rollout and retain the first $d_{\mathcal{M}}$ principal components, storing them in $\mathbf{P} \in \mathbb{R}^{d_{\mathcal{L}} \times d_{\mathcal{M}}}$, as well as the mean latent vector $\boldsymbol{\ell}$. Together, these elements define the affine transformation used to approximate the mappings $\mathcal{L} \rightarrow \mathcal{V}$ and $\mathcal{V} \rightarrow \mathcal{L}$. We express the chart and its inverse as

$$\mathbf{z} \approx \mathbf{P}^\top [\mathbf{E}(\mathbf{u}) - \boldsymbol{\ell}], \quad \mathbf{u} \approx \mathbf{D}(\mathbf{P}\mathbf{z} + \boldsymbol{\ell}).$$

While an atlas of charts is needed to cover \mathcal{M} , corresponding to a set of encoders and decoders with one pair per chart, the manifolds considered in this work are well represented by a single pair of global mappings, $\mathbf{E} : \mathcal{M} \rightarrow \mathcal{L}$ and $\mathbf{D} : \mathcal{L} \rightarrow \mathcal{M}$. The method proposed by Floryan and Graham [76] can be employed when multiple local mappings are required.

Given a differentiable approximation to ψ^{-1} , we sample $\mathbf{u} \in \mathcal{M}$ from the rollout and compute

$$\frac{\partial \mathbf{D}(\mathbf{P}\mathbf{z} + \boldsymbol{\ell})}{\partial \mathbf{u}} \approx \frac{\partial \psi^{-1}}{\partial \mathbf{u}} \quad (4.19)$$

via AD. The Jacobian $\partial\Phi/\partial\mathbf{u}$ is also obtained by AD. Applying a QR decomposition to our approximation of $\partial\psi^{-1}/\partial\mathbf{u}$ yields an orthonormal basis $\mathbf{Q} : \mathbb{R}^n \rightarrow T_u(\mathcal{M})$ whose columns span the tangent space. We may therefore restrict the Jacobian to the mapping between tangent spaces as follows:

$$\frac{\partial\Phi}{\partial\mathbf{u}} \mathbf{Q} \mathbf{Q}^\top \approx \left(\frac{\partial\Phi}{\partial\mathbf{u}} \right)_{T_u(\mathcal{M})} : T_u(\mathcal{M}) \rightarrow T_y(\mathcal{Y}), \quad (4.20)$$

since the null space of $\mathbf{Q} \mathbf{Q}^\top$ is $N_u(\mathcal{M})$. We finally compute the singular value decomposition (SVD) of $(\partial\Phi/\partial\mathbf{u}) \mathbf{Q}$ to obtain the spectrum of the restricted mapping.⁶

Figure 9 summarizes the condition numbers obtained from the tangent space mapping for 1000 snapshots sampled from the long-time rollout in the $L = 22$ domain. Mean condition numbers are shown as solid dots, and vertical lines indicate the corresponding ranges. Results are plotted as a function of the number of measurement times m_t , with a separate curve for each number of spatial sensors m_x . Although the condition numbers are large, they remain finite and are well below the inverse of machine precision, indicating that the mapping from $T_u(\mathcal{M})$ to $T_y(\mathcal{Y})$ is indeed bijective. This behavior is consistent with the immersion criterion, whereby $\Phi \in \mathcal{P}^m$ is generically an immersion if $m \geq d_{\mathcal{M}}$. However, the large condition numbers imply poor numerical conditioning and could lead to vanishing gradients. As expected, we see that conditioning improves with additional spatial and temporal observations, reflecting a more stable mapping between the IM and measurement space. However, improvements in the condition number level off at large m because it is bounded from below by the conditioning of the flow map Jacobian restricted to $T_u(\mathcal{M})$.

⁶The operator $\mathbf{Q} \mathbf{Q}^\top$ could be used to perform a manifold-constrained optimization. We successfully implemented a related approach with \mathbf{z} as the control vector, i.e., by applying AD to the computational graph from $\mathbf{z} \rightarrow \mathbf{y}$. However, since the manifold is not known a priori for most reconstruction problems, we do not employ such techniques in the present work.

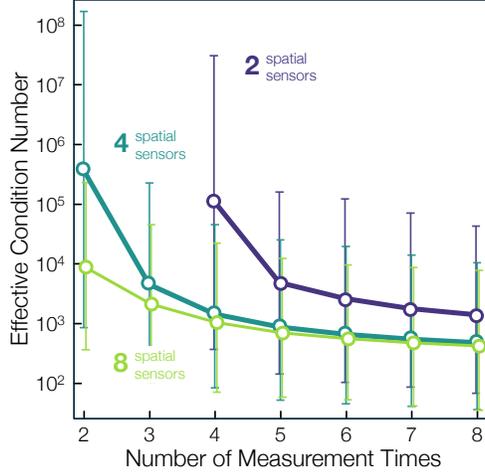


Figure 9: Mean (dots) and range (vertical lines) of condition numbers for mappings $T_u(\mathcal{M}) \rightarrow T_y(\mathcal{Y})$ for the $L = 22$ domain. Curves correspond to different numbers of spatial sensors m_x and are plotted against the number of observation times m_t . All maps are full rank, consistent with the immersion criterion, and show improved conditioning with increased spatial and temporal sampling.

4.4.2 Gradient components in $T_u(\mathcal{M})$ and $N_u(\mathcal{M})$

Section 4.3.3 shows that reference–observer pairs exist for which $\mathbf{g}_u \notin T_{u_\theta}(\mathcal{M})$. Moreover, if a direction in $N_{y_\theta}(\mathcal{Y})$ intersects \mathcal{Y} , then gradients with $\mathbf{g}_u \in N_{u_\theta}(\mathcal{M})$ can occur. The existence of such cases would prevent the theoretical global convergence of manifold-constrained optimization, and in practice any component of the gradient lying in $N_{u_\theta}(\mathcal{M})$ can push the initial observer state off the manifold. The frequency of these events, however, is not known a priori.

We estimate this frequency numerically, using the basis \mathbf{Q} to project gradients into the local tangent space and computing the cosine similarity

$$\text{CS}_g = \frac{\mathbf{g}_u^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{Q} \mathbf{Q}^\top \mathbf{g}_u\|_2},$$

which is unity when $\mathbf{g}_u \in T_{u_\theta}(\mathcal{M})$ and zero when $\mathbf{g}_u \in N_{u_\theta}(\mathcal{M})$. We evaluate CS_g for 20 random initial conditions in the $L = 22$ domain, using 1000 random initial guesses for each reference state. Gradients are computed for the $m_x = 16$, $m_t = 16$, and $L = 22$ case with $T = 20$. The resulting PDF is plotted in Fig. 10. The distribution is strongly skewed toward unity, with a mean of 0.78, indicating that gradients are usually well aligned with the tangent space, although cases with substantial normal components do occur.

4.5 Well posedness of adjoint state estimation

The preceding sections examined key geometric factors that influence adjoint reconstructions. We now shift to a direct assessment of how state estimation transitions from an ill-posed problem to a well-posed one as the immersion and embedding criteria are satisfied. These criteria do not determine the behavior of numerical optimization, per se: rank-deficient Hessians, negative curvature, and vanishing gradients (all analyzed in the next section) can obscure the observability of the reference system, independent of $d_{\mathcal{M}}$ or the sensor configuration. Nevertheless, when optimization is stable, embedding criteria should govern the “posedness” of state estimation, as can be seen through the ε^* metric. When ε^* is small, many distinct trajectories yield nearly indistinguishable measurements, so a low loss need not imply an accurate reconstruction. To evaluate the practical onset of well-posed reconstruction, therefore, we conduct an empirical survey of reconstruction accuracy and compare these results with the theoretical criteria from Sec. 4.3.

To isolate degree-of-freedom effects from the optimization dynamics, we exclude cases with poor convergence, i.e., those with a final loss above 10^{-3} (or a mean pointwise error over 3%, roughly). These cases are limited by failures of optimization rather than the geometric relationship between the IM and its image. For every $(d_{\mathcal{M}}, m)$ point in our dataset—spanning all the sensor configurations and reference–observer pairs described in Sec. 3.1—we compute the probability of successful reconstruction conditioned on $\mathcal{J} < 10^{-3}$.

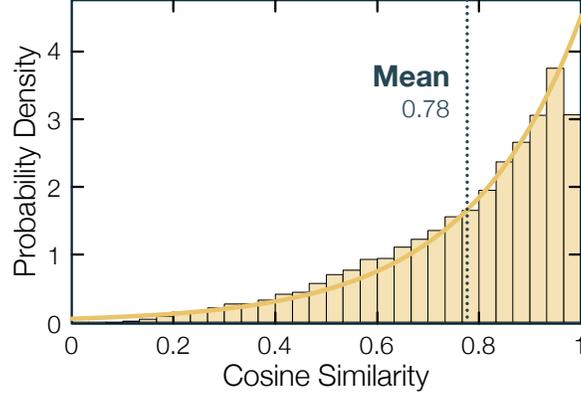


Figure 10: PDF of the cosine similarity between the full gradient and the gradient projected onto the tangent space of \mathcal{M} , quantifying the extent to which optimization directions point off the manifold.

Success is defined by $CS_U \geq 0.95$. Figure 11 plots these probabilities in the $d_{\mathcal{M}}-m$ plane, along with the immersion line $m = d_{\mathcal{M}}$ and the embedding line $m = 2d_{\mathcal{M}} + 1$. Below the immersion line, reconstruction accuracy collapses to just a few percent. Above the embedding line, the probability approaches unity. Between these bounds, the probabilities vary smoothly with m , reflecting a dependence on the specific reference trajectory and initial guess. The structure of Fig. 11 supports the applicability of the immersion and embedding criteria. An immersion marks the onset of feasible reconstruction, and an embedding marks the regime in which a low loss reliably corresponds to an accurate reconstruction.

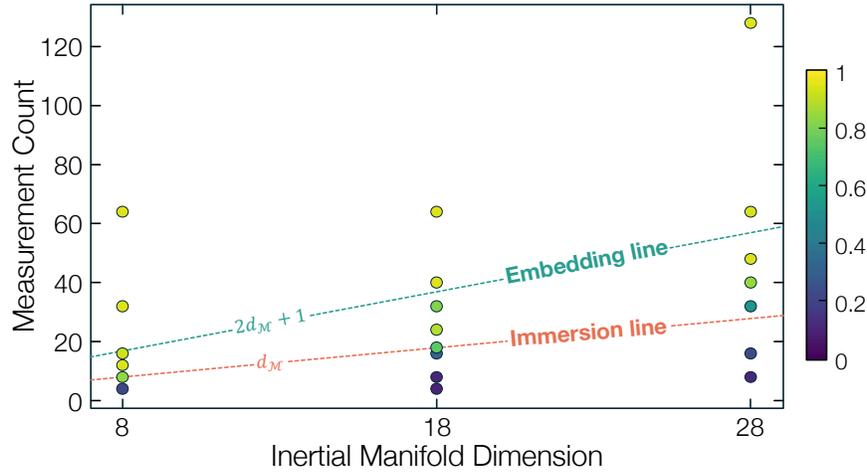


Figure 11: Summary of embedding quality across domain lengths and normalized measurement counts. Reconstructions are classified as accurate for $CS_U \geq 0.95$. Accuracy is low below the immersion line, high above the embedding line, and transitions smoothly between them, consistent with theory.

Next, we look at trends in ε^* and their relation to the embedding criterion to explicate the relationship between $(d_{\mathcal{M}}, m)$ and $p(CS_U \geq \tau)$. For each domain size and sensor configuration, we plot the joint density $p(CS_U = \tau, \mathcal{J})$ in the $\mathcal{J}-\tau$ plane. If a threshold $\varepsilon^* \in [10^{-10}, 10^{-3}]$ exists for $\tau = 0.95$ and $\delta = 0.001$, it is indicated by a vertical line. Figure 12 shows these trends for the $L = 22$ domain, where $d_{\mathcal{M}} = 8$, using three configurations spanning $m = 4$ to $m = 32$. For $m = 4$, where Φ cannot be an embedding since $m < d_{\mathcal{M}}$, there is essentially no relationship between the probability of accurate reconstruction and the loss; ε^* does not exist for such configurations and the reconstruction problem is hopelessly ill posed. As m increases and eventually exceeds the embedding threshold, a clear correlation between $p(CS_U)$ and \mathcal{J} emerges: all the probability mass for $\mathcal{J} < \varepsilon^*$ concentrates near $CS_U = 1$, indicating uniformly accurate reconstructions. With $m = 32$, the ε^* threshold reaches 9×10^{-3} .

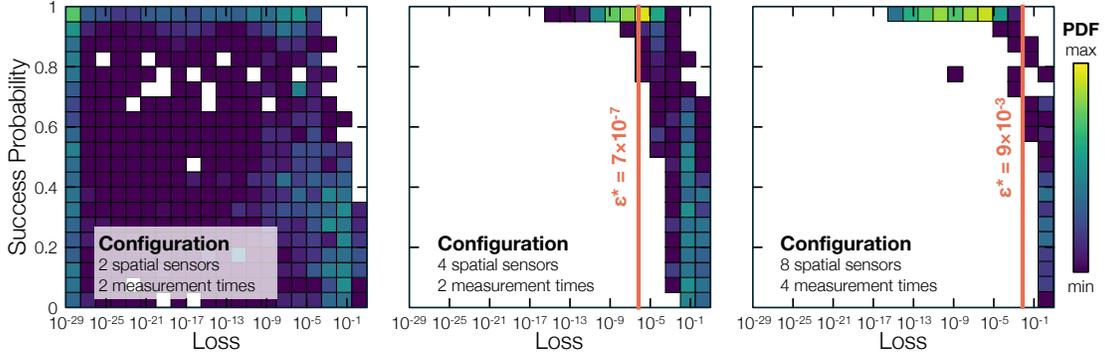


Figure 12: Joint PDF $p(\text{CS}_U = \tau, \mathcal{J})$ for various measurement configurations in the $L = 22$ domain. Vertical lines mark ϵ^* , which is expected to be large whenever the measurement map is an embedding.

Figure 13 presents the same analysis for the $L = 44$ domain, for which $d_M = 18$. Here, $m = 8$ shows no evidence of an embedding (in contrast with the transitional behavior observed at $m = 8$ for the $L = 22$ domain). At $m = 16$, the relationship is transitional, although the threshold of $\epsilon^* \approx 5 \times 10^{-9}$ remains small. At $m = 64$, however, the results evince a robust embedding, with $\epsilon^* \approx 10^{-2}$, indicating that adjoint state estimation is theoretically well posed for such sensor arrangements.

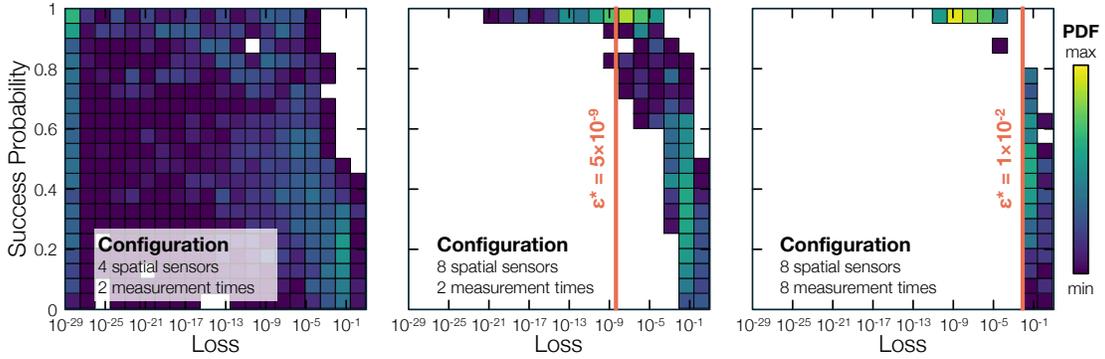


Figure 13: Joint PDF $p(\text{CS}_U = \tau, \mathcal{J})$ for measurement configurations in the $L = 44$ domain. Vertical lines mark ϵ^* , which is expected to be large in cases admitting an embedding.

These trends were computed for all domain sizes and sensor configurations and are summarized in Fig. 14. For each configuration, the figure reports the value of ϵ^* for $\tau = 0.95$ and $\delta = 0.001$. If no such ϵ^* exists above 10^{-10} , the cell is labeled “DNE.” The cell colors indicate a normalized measurement count,

$$\tilde{m} = \begin{cases} 0, & m < d_M, \\ \frac{m - (d_M - 1)}{2d_M + 1 - (d_M - 1)}, & d_M < m < 2d_M + 1, \\ 1, & m \geq 2d_M + 1, \end{cases}$$

which equals zero below the immersion criterion, increases linearly between the immersion and embedding criteria, and saturates at unity thereafter. Larger values of \tilde{m} correlate strongly with larger ϵ^* , mirroring the trends in Fig. 11. Configurations with $\tilde{m} = 0$, for which Φ is neither an immersion nor an embedding, either do not admit a computable value of ϵ^* or else yield a trivial value. Once \tilde{m} reaches 1, ϵ^* is consistently large, indicating that Φ acts as an embedding and that the state estimation problem is well posed.

5 Optimization dynamics

Numerical optimizations can still fail when Φ is an embedding, due in large part to degeneracy of the Hessian, negative curvature, or vanishing gradients. We now analyze *optimization-dynamic effects* caused

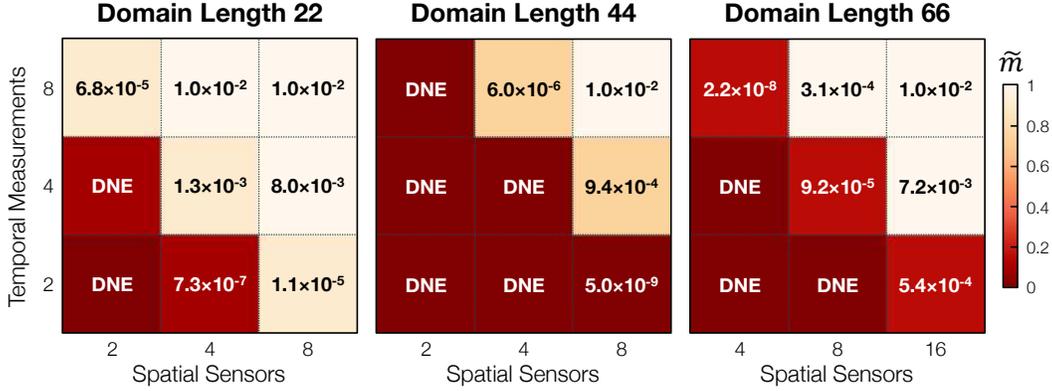


Figure 14: Summary of ε^* for different domain lengths and normalized measurement counts, plotted in (m_x, m_t) space. Values $\tilde{m} = 1$ mark guaranteed embeddings, $\tilde{m} \in [0, 1)$ mark immersions (and possible embeddings), and $\tilde{m} = 0$ cannot support an immersion. Trends in ε^* are consistent with these classifications.

by these issues and their role in determining whether a theoretically well-posed reconstruction problem is numerically tractable. To start, we show the ways in which chaotic dynamics cause adjoint state estimation to fail. We then use analytical and empirical results to demonstrate why the Hessian creates a poorly condition loss landscape both near optimality and far away from it. These observations motivate our use of NCN, which leverages curvature information to stabilize adjoint marching. Lastly, we present an upper bound for the loss reduction from a single NCN step, and we show numerical results that clarify when and why the optimizer becomes trapped.

5.1 Expressions for the gradient and Hessian of \mathcal{J}

To frame our discussion of optimization dynamics, we begin by recalling our loss function, which equals

$$\mathcal{J} = \frac{1}{2} \sum_{k \in \mathcal{K}} \left[f_{\Delta t}^k(\mathbf{u}_\theta) - \mathbf{u}_k \right]^\top \mathbf{M}_k \left[f_{\Delta t}^k(\mathbf{u}_\theta) - \mathbf{u}_k \right] \quad (5.1)$$

up to a constant. We differentiate it to obtain

$$\mathbf{g} = \sum_{k \in \mathcal{K}} \mathbf{g}_k = \sum_{k \in \mathcal{K}} \left(\frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}} \right)^\top \mathbf{J}_k^\top \mathbf{M}_k \left[f_{\Delta t}^k(\mathbf{u}_\theta) - \mathbf{u}_k \right], \quad (5.2)$$

where $\mathcal{K} = \{0, \dots, K\}$ and $\mathbf{J}_k = \partial f_{\Delta t}^k / \partial \mathbf{u}_\theta$ is the Jacobian of the flow map at time index k . The vector \mathbf{g}_k is the contribution to the gradient from time k , and the full gradient \mathbf{g} is simply the sum of these “sub-gradients.” Finally, we note that $\partial \mathbf{u}_\theta / \partial \boldsymbol{\theta}$ is the inverse discrete Fourier transform, and we have $\partial^2 \mathbf{u}_\theta / \partial \boldsymbol{\theta}^2 = \mathbf{0}$.

Equation (5.2) may be differentiated once more to obtain the Hessian,

$$\mathbf{H} = \underbrace{\left(\frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}} \right)^\top \left(\sum_{k \in \mathcal{K}} \mathbf{J}_k^\top \mathbf{M}_k \mathbf{J}_k \right) \frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}}}_{\mathbf{H}^{\text{GN}}} + \underbrace{\left(\frac{\partial \mathbf{u}_\theta}{\partial \boldsymbol{\theta}} \right)^\top \sum_{k \in \mathcal{K}} \frac{\partial (\mathbf{J}_k^\top)}{\partial \mathbf{u}_\theta} \mathbf{M}_k \left[f_{\Delta t}^k(\mathbf{u}_\theta) - \mathbf{u}_k \right]}_{\mathbf{H}^{\text{C}}} \quad (5.3)$$

where \mathbf{H}^{GN} is the positive semidefinite Gauss–Newton component and \mathbf{H}^{C} is the second-order term stemming from the curvature of the flow map. Near optimality, the residuals $f_{\Delta t}^k(\mathbf{u}_\theta) - \mathbf{u}_k$ vanish and the Hessian reduces to the positive semidefinite component $\mathbf{H} \approx \mathbf{H}^{\text{GN}}$.

5.2 Optimization failure modes in adjoint state estimation

When a reconstruction fails, it is either because Φ is not an embedding—so that ε^* is extremely small (or not computable) and many trajectories of low loss exhibit large error—or because the optimizer fails to attain a sufficiently low loss. Section 4.5 shows that ε^* increases with m , but ε^* characterizes the probability that a low loss yields an accurate reconstruction, not the probability of attaining low loss in the first place.

It is therefore natural to ask whether the latter probability also increases with m . This would be intuitive, and results in Sec. 4.3 demonstrate that the conditioning of Φ improves with added measurements, which *should* increase the likelihood of successful optimization, though this hypothesis must still be verified. We address this question by comparing the probability of achieving a low loss to that of achieving an accurate reconstruction. Trends in these probabilities reveal distinct failure modes of adjoint state estimation, which are explained in the remainder of Sec. 5.

Figure 15 shows the probability of $CS_U \geq 0.95$ (solid lines) and $\mathcal{J} < 10^{-3}$ (dashed lines) for all three domains $L \in \{22, 44, 66\}$. Results are plotted against $m - (2d_M + 1)$, so that $x = 0$ corresponds to the embedding criterion for all domains (marked by a red vertical line). Immersion thresholds are shown as vertical dotted lines that are color-coded by L . To emphasize challenging cases where negative curvature and departures from \mathcal{M} are more likely, we report results for cases with an initial distance in $D_{ij} \in [0.8, 1]$.

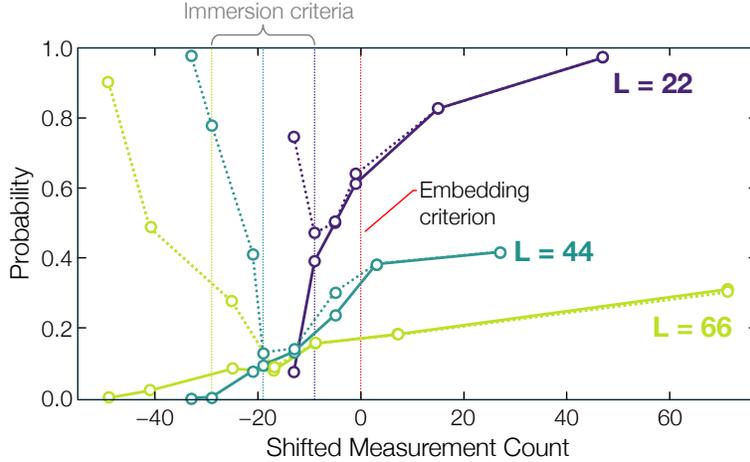


Figure 15: Probabilities of achieving high accuracy ($CS_U \geq 0.95$, solid lines) and low loss ($\mathcal{J} < 10^{-3}$, dashed lines) plotted against $m - (d_M + 1)$ for $L \in \{22, 44, 66\}$. Vertical lines show the embedding ($x = 0$) and immersion thresholds. All cases use far-off initial conditions with $D_{ij} \in [0.8, 1]$.

The trends in Fig. 15 are consistent across all three domains. The probability of accurate reconstruction (taken as $CS_U \geq 0.95$) begins near zero and increases with m , whereas the probability of achieving a low loss exhibits a “U” shape: initially high, dipping to a minimum between the immersion and embedding lines, and rising again thereafter.

The initial drop in $p(\mathcal{J} < 10^{-3})$ occurs because, when Φ is not an embedding and m is very low, adding measurements eliminates spurious minima from the loss landscape, making cases of low loss rarer. As m continues to increase and crosses the immersion and embedding thresholds, the loss and accuracy curves converge, and low loss becomes a reliable indicator of accurate reconstructions thereafter (i.e., ε^* increases). Two effects drive this transition. First, ε^* rises with increasing m because observation vectors \mathbf{y} from different states on \mathcal{M} become more distinct, making accurate reconstructions more likely, even at moderate loss levels. Second, the probability of attaining low loss itself increases because further measurements improve the conditioning of the optimization problem. In short, accuracy gains at low m are due to degree-of-freedom effects that govern the mapping from \mathcal{M} to \mathcal{Y} , whereas gains at high m come from better optimizer performance. Still, the absolute probability $p(\mathcal{J} < 10^{-3})$ remains low for large- L domains, even at high m , which underscores the need to understand why optimizations fail in the embedding regime.

5.3 Condition and curvature of the loss landscape near optimality

Equations (5.2) and (5.3) reveal how the flow map Jacobian J_k (defined by $\delta u_k = J_k \delta u_0$) affects gradients and curvature of the loss landscape. The singular values of J_k are directly determined by the Lyapunov spectrum, i.e., the i th singular value scales as $\sigma_i \sim e^{\ell_i k \Delta t}$, where ℓ_i is the i th Lyapunov exponent [77]. As the assimilation window becomes longer, J_k becomes severely ill-conditioned: it quickly becomes singular and its nullity grows steadily with T . Figure 16 illustrates this behavior via normalized singular value spectra of J_k for time horizons of increasing duration, averaged over 1000 initial conditions for the $L = 22$ domain.

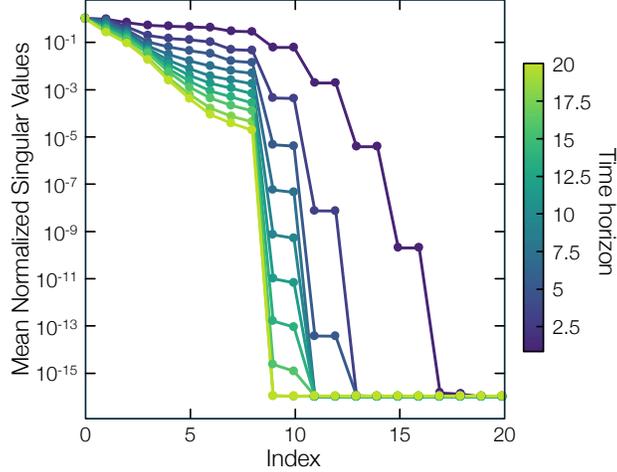


Figure 16: Normalized singular value spectra of J_k averaged over 1000 initial conditions from the $L = 22$ dataset. Line color shows the time horizon $T \in [1, 20]$.

The rapid divergence of J_k 's singular values creates a fundamental trade-off in adjoint state estimation. Increasing T initially reduces redundancy among measurements and increases ε^* , but the exponential amplification of perturbations progressively weakens the numerical link between early measurements and later ones. Beyond the Lyapunov time, this amplification dominates and the state estimation problem becomes badly ill-posed. A second complication arises from the unweighted MSE loss: because the operator norm $\|J_k\|_2^2$ grows exponentially with k , later measurements have a disproportionate influence on the gradient, causing the optimizer to match observations at the end of the assimilation window first, as observed in prior studies on adjoint-variational state estimation [48, 49]. Consequently, extending the window beyond T_ℓ , even when adding more measurements, can hinder optimization, since numerical errors accumulate, gradients with respect to u_0 become unreliable, and corrupted gradient components from later times dominate the step direction.

Ill-conditioning of J_k also manifests in the curvature of the loss. Near optimality, the Hessian is dominated by H^{GN} , whose rank is at most m . Because H^{GN} depends quadratically on J_k , any ill-conditioning in J_k is inherited by and amplified in the Hessian. In practice, the rank of H is usually less than m ; in cases where it manages to attain rank m , its effective condition number (excluding the null space) remains extremely large. These features can slow or stall the optimizer near critical points and can limit the observability of u_0 , even when the initial guess lies arbitrarily close to it.

5.4 Curvature of the loss landscape away from optimality

The previous section shows that for the KS systems of interest, the flow map Jacobian guarantees that H has null eigenvalues at optimality, even for short assimilation windows, thereby limiting observability. We now turn to the prevalence of negative eigenvalues away from optimality, which indicate directions of negative curvature. We show that they are ubiquitous in regions of moderate loss. To do this, we first develop a mathematical intuition for why negative eigenvalues arise, and we then present numerical evidence to support this result.

Far from optimality, H^{C} becomes important in Eq. (5.3). Because the residual $f_{\Delta t}^k(u_\theta) - u_k$ has no preferred sign, H^{C} is indefinite in expectation and possesses both positive and negative eigenvalues. Meanwhile, H^{GN} has a nontrivial null space due to singularity of the flow map Jacobian and the sparsity of observations, with a nullity that almost always exceeds $n - m$ when $m < n$ (as is always the case in practical state estimation problems). To understand how H^{GN} and H^{C} contribute to the eigenvalues of H , let $\lambda_i(\cdot)$ denote the i th ordered eigenvalue, with $\lambda_1 \geq \dots \geq \lambda_n$. Weyl's inequality provides tight bounds on the eigenvalues of a sum of symmetric matrices. Applied to H^{GN} and H^{C} , it yields

$$\lambda_i(H^{\text{GN}}) + \lambda_n(H^{\text{C}}) \leq \lambda_i(H^{\text{GN}} + H^{\text{C}}) \leq \lambda_i(H^{\text{GN}}) + \lambda_1(H^{\text{C}}). \quad (5.4)$$

For many indices i , the Gauss-Newton term satisfies $\lambda_i(H^{\text{GN}}) \approx 0$. Substituting such an index into the

inequality gives

$$\lambda_i(\mathbf{H}^{\text{GN}} + \mathbf{H}^{\text{C}}) = \lambda_i(\mathbf{H}) \in [\lambda_n(\mathbf{H}^{\text{C}}), \lambda_1(\mathbf{H}^{\text{C}})], \quad (5.5)$$

Since $\lambda_n(\mathbf{H}^{\text{C}}) < 0 < \lambda_1(\mathbf{H}^{\text{C}})$, where λ_1 and λ_n are comparable in expected magnitude, Eq. (5.5) implies that the corresponding $\lambda_i(\mathbf{H})$ will take both positive and negative values. Consequently, away from optimality, where the residual is non-zero and often large, negative eigenvalues of \mathbf{H} arise with high probability and are expected to be prevalent.

Figure 17 provides empirical evidence for this claim. It shows the probability of \mathbf{H} containing at least one negative eigenvalue below -10^{-8} , conditioned on the loss, for the $L = 22$ dataset under two sensor configurations: $m_x = 2$ and $m_t = 2$ and $m_x = 8$ and $m_t = 8$. From this plot, we see that the loss landscape almost always exhibits negative curvature in regions of moderate loss. All remaining sensor configurations across all domain sizes display the same qualitative behavior. Taken together with the severe ill-conditioning of \mathbf{H} , these observations motivate our use of the NCN optimizer.

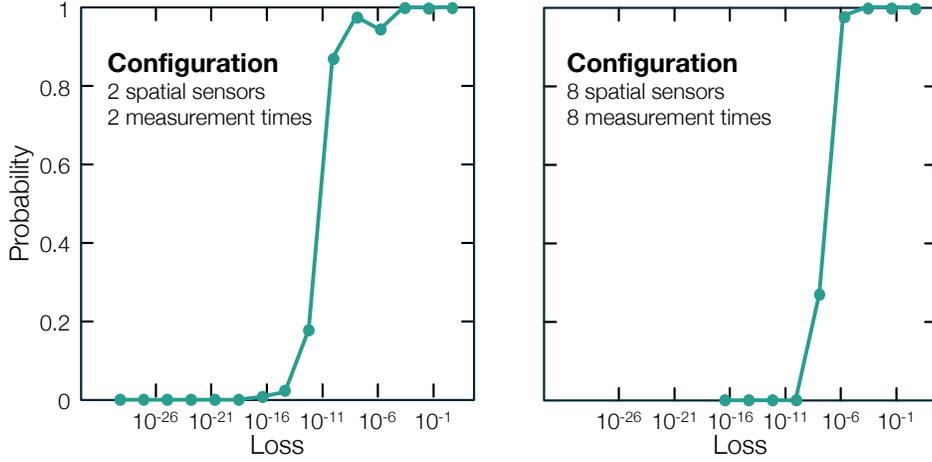


Figure 17: Probability that the terminal Hessian has a negative eigenvalue of magnitude exceeding 10^{-8} for the $L = 22$ dataset. Left: $m_x = 2$ and $m_t = 2$. Right: $m_x = 8$ and $m_t = 8$.

5.5 When does NCN optimization stall?

Up to this point, we have established several favorable properties of adjoint state estimation: when Φ is an immersion, local reconstruction is well posed; when it is an embedding, the global optimum is the only critical point on \mathcal{M} ; negative curvature is effectively handled by NCN steps; and pseudo-projection suppresses gradient components that point off \mathcal{M} , thereby stabilizing the optimization. Still, NCN optimization with pseudo-projection can proceed very slowly when $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$ is orders of magnitude smaller than \mathcal{J} . To explain why, we present an upper bound on the reduction in \mathcal{J} produced by a single NCN step.

The upper bound in question begins with a Taylor expansion of the loss increment:

$$\mathcal{J}(\boldsymbol{\theta}_{k+1}) \leq \mathcal{J}(\boldsymbol{\theta}_k) + \mathbf{g}^\top (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) + \frac{1}{2} (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k)^\top \mathbf{H} (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) + \frac{M}{6} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2, \quad (5.6)$$

where M is the Lipschitz constant of \mathbf{H} . Substituting the NCN step $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta |\mathbf{H}|^{-1} \mathbf{g}$ yields

$$\mathcal{J}(\boldsymbol{\theta}_{k+1}) - \mathcal{J}(\boldsymbol{\theta}_k) < \underbrace{-\eta \mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}}_{\text{first-order term}} + \underbrace{\frac{1}{2} \eta^2 \mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{H} |\mathbf{H}|^{-1} \mathbf{g}}_{\text{second-order term}} + \underbrace{\frac{M}{6} \|\eta |\mathbf{H}|^{-1} \mathbf{g}\|_2^3}_{\text{correction term}}. \quad (5.7)$$

Here, first and second order indicates the origin of these terms in the Taylor series. The eigenvectors of $|\mathbf{H}|^{-1} \mathbf{H} |\mathbf{H}|^{-1}$ are the same as those of \mathbf{H} , and the eigenvalues are similar. Specifically, they are

$$\lambda_i(|\mathbf{H}|^{-1} \mathbf{H} |\mathbf{H}|^{-1}) = \begin{cases} 1/\lambda_i(\mathbf{H}), & \lambda_i(\mathbf{H}^{-1}) > \delta, \\ \lambda_i(\mathbf{H})/\delta^2, & \lambda_i(\mathbf{H}^{-1}) \leq \delta. \end{cases} \quad (5.8)$$

Since $|\mathbf{H}|^{-1}$ is positive definite, the first-order term in Eq. (5.7) always acts to decrease the loss so long as $\mathbf{g} \neq \mathbf{0}$ and the step size η is sufficiently small. However, the magnitude of the reduction is governed by that of $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$, which must be comparable to \mathcal{J} to ensure meaningful progress. The second-order term can also reduce the loss, but only when \mathbf{g} aligns with directions of negative curvature. If $\|\mathbf{g}\|_2$ is small relative to the loss, progress becomes extremely slow unless the gradient happens to point along very flat or negatively curved directions of the loss landscape.

We now present evidence that the optimizer does not become trapped in local minima. If it were getting stuck at true critical points, we would expect little or no correlation between the loss and the gradient norm; a plot of $\|\mathbf{g}\|_2$ versus \mathcal{J} would show no discernible trend. Instead, Fig. 18, which plots the gradient norm versus the loss for all cases at $L = 22$ with the $m_x = 2$ and $m_t = 2$ and $m_x = 8$ and $m_t = 8$ sensor configurations, shows a strong, nearly linear relationship on a log-log scale. Across all sensor configurations and domain lengths tested, the minimum correlation between $\log(\mathcal{J})$ and $\log(\|\mathbf{g}\|_2)$ is 0.84. This behavior is consistent with a power-law relation of the form $\|\mathbf{g}\|_2 \sim \mathcal{J}^a$, where a is a constant. Given this strong correlation, we do not attribute stalled convergence to local minima. Instead, we believe the limiting factor is the regime in which $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g} \ll \mathcal{J}$, as illustrated in Fig. 19. This figure uses the same cases as Fig. 18, but the y -axis is replaced by $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$. For the overwhelming majority of points, we observe that $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$ is less than $\mathcal{J}(\theta_k)$, and for cases of high loss, it falls orders of magnitude below \mathcal{J} . Finally, we note that the results in Figs. 18 and 19 were essentially unchanged after an additional 650 NCN iterations, confirming that the optimizations in these figures are converged.

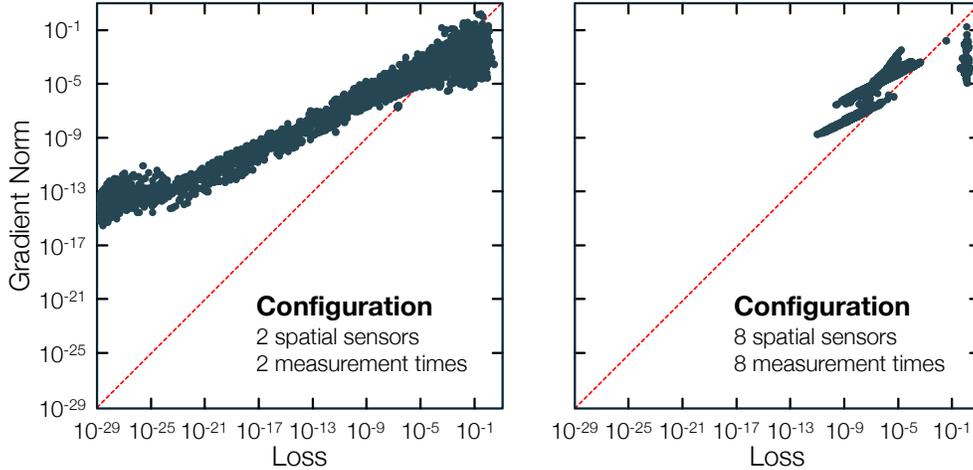


Figure 18: Gradient norm versus optimization loss for $L = 22$. Left: cases with $m_x = 2$ and $m_t = 2$. Right: cases with $m_x = 8$ and $m_t = 8$.

To illustrate these ideas, we examine the optimizer’s behavior for a failed case with $m_x = 4$, $m_t = 4$, $L = 22$. The left panel of Fig. 20 shows the loss together with the magnitudes of \mathbf{g} , $|\mathbf{H}|^{-1} \mathbf{g}$, and $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$ as functions of iteration. The loss barely decreases because $\|\mathbf{g}\|_2$ is several orders of magnitude smaller than \mathcal{J} , which in turn forces the quadratic term to be even smaller still. A plausible explanation for why the gradient becomes so small relative to the loss is the presence of conflicting sub-gradients. That is, contributions \mathbf{g}_k from different observation times remain large at the end of the optimization, but they almost perfectly cancel out in aggregate. The right panel of Fig. 20 illustrates this effect for the same case shown on the left; sub-gradients from all four observation times at the final iteration are mapped into state space and plotted. Although magnitudes of the individual curves are substantial, their sum, corresponding to the $\|\mathbf{g}\|_2$ curve in the left panel (multiplied by \sqrt{n} for the conversion to state space), is nearly zero.

Motivated by this observation, we note that minimizing measurement residuals at different times can be viewed as a multi-task optimization problem in which sub-gradients may conflict with one another. This perspective suggests that techniques from multi-task learning, such as dynamic loss weighting [78] or gradient-conflict resolution [79], could improve the global behavior of optimizers in adjoint state estimation.

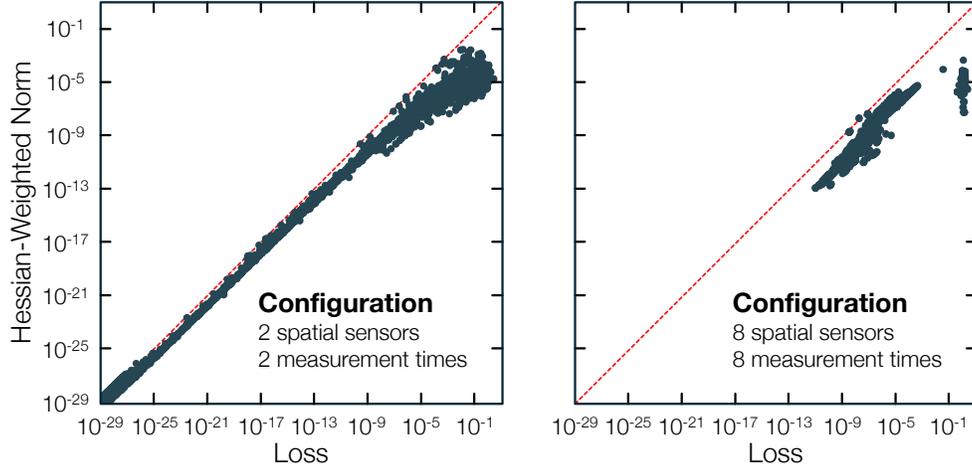


Figure 19: Hessian-weighted gradient norm, $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$, versus loss for $L = 22$. Left: cases with $m_x = 2$ and $m_t = 2$. Right: cases with $m_x = 8$ and $m_t = 8$.

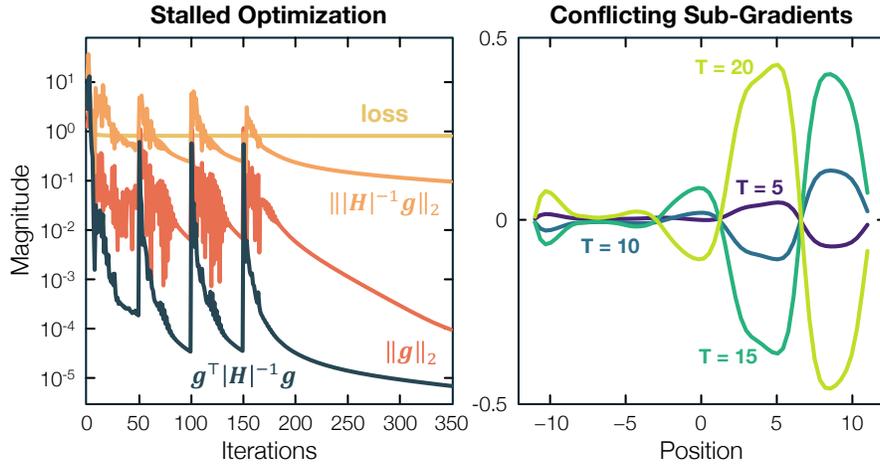


Figure 20: Representative optimization failure caused by vanishing gradients for a case from the $L = 22$ domain. Left: gradient norm $\|\mathbf{g}\|_2$, step $\|\mathbf{H}^{-1} \mathbf{g}\|_2$, and Hessian-weighted norm $\mathbf{g}^\top |\mathbf{H}|^{-1} \mathbf{g}$ versus iteration. The gradient is several orders of magnitude smaller than the loss, stalling progress. Right: sub-gradients at the final iteration visualized in state space; they nearly cancel when summed to produce the full gradient \mathbf{g} .

6 Conclusions and outlook

Adjoint state estimation provides a powerful framework for combining simulations and experiments to reconstruct high-fidelity trajectories of a dynamical system that are anchored to real-world observations. However, the number of measurements required for accurate reconstruction of a chaotic system is not known a priori. We address that gap using tools from dynamical systems and embedding theory. For a dissipative system whose attractor lies on an inertial manifold \mathcal{M} of dimension $d_{\mathcal{M}}$, we show that $m \geq d_{\mathcal{M}}$ measurements are sufficient for local observability from an arbitrarily good initial guess, and $m \geq 2d_{\mathcal{M}} + 1$ are required for global observability on \mathcal{M} . These classical bounds determine whether the observation map Φ is an immersion or an embedding, respectively, guaranteeing the local or global existence of Φ^{-1} . While such criteria are well established in the literature on state space reconstruction, we demonstrate their applicability to adjoint state estimation by translating the existence and conditioning of Φ^{-1} into geometric conditions for a well-posed reconstruction problem.

Specifically, we show that when Φ is an embedding, the global optimum is the only critical point of the loss on \mathcal{M} . Moreover, the induced geometry is well conditioned, such that similar measurements correspond

to nearby states and dissimilar measurements to distant states. These theoretical results are validated through extensive simulations of Kuramoto–Sivashinsky systems in domains of length 22, 44, and 66. In general, when the embedding criterion is satisfied, adjoint state estimation becomes well posed and reconstruction accuracy improves steadily with increasing m .

Despite these guarantees, embedding theory alone does not determine practical limits on observability. Even when Φ is an embedding, the loss landscape can remain severely ill conditioned due to singularity of the flow map Jacobian and sparsity of the sensor configuration. As a result, trajectories may be theoretically observable yet difficult to recover by numerical means. Two key challenges are identified. First, gradients can have large components normal to \mathcal{M} , causing iterates in an optimization to drift off the manifold, at which point guarantees from embedding theory no longer apply. To counteract this, we introduce a pseudo-projection step that periodically pulls the estimate back toward \mathcal{M} , helping to stabilize adjoint marching. Second, the Hessian is degenerate at optimality and becomes indefinite away from it. Indeed, directions of negative curvature are nearly ubiquitous in the loss landscape at moderate loss levels. These pathological features undermine first-order, Newton, and quasi-Newton methods alike. To address this, we employ a “non-convex Newton” technique that explicitly handles negative curvature while preserving descent directions for indefinite Hessians. When combined with pseudo-projection, NCN enables robust state estimation once the embedding criterion is satisfied.

Nevertheless, optimization can still stall when the gradient norm becomes much smaller than the loss. We attribute this behavior to destructive interference among sub-gradients from different observation times. This effect accounts for all failures we examined in the embedding regime. Future work will address this limitation by incorporating ideas from multi-task learning, and we will pursue extensions of the present framework to account for measurement noise and operator error, enable its application to higher-dimensional systems, and handle non-stationary measurement operators.

Appendix A Numerical simulation

A.1 Pseudo-spectral scheme with exponential time-differencing

The hyper-diffusion term in the KS equation causes Fourier coefficients associated with high-wavenumber modes to have large values of $\partial \hat{u}_j / \partial t$, leading to rapid transients. The characteristic time scale of the j th Fourier mode scales as $O(j^{-4})$ for large j , whereas low-wavenumber modes evolve much more slowly. Differential equations that exhibit such a wide separation of scales are deemed to be “stiff,” and stiffness poses major challenges for classical explicit time-stepping schemes.

Explicit integrators require a time step that is small enough to stabilize the fastest modes, but simulations must run for long enough to resolve the slow dynamics of low-frequency modes. This combination of small Δt and long integration times leads to a high computational cost. To overcome this issue, we employ exponential time-differencing, which analytically integrates the stiff linear terms in the KS equation and numerically integrates the nonlinear term [80–82]. This approach enables large time steps and long-time integrations without compromising stability.

To start, we discretize the periodic spatial domain $[-L/2, L/2]$ into n uniformly spaced points,

$$\mathbf{u}(t) = [u(-L/2, t), \dots, u(L/2 - \Delta x, t)], \quad (\text{A.1})$$

where $\Delta x = L/n$. The semi-discrete KS equation is written as

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{D}^{(2)} + \mathbf{D}^{(4)}) \mathbf{u} - \frac{1}{2} \mathbf{D}^{(1)} \mathbf{u}^{\circ 2}, \quad (\text{A.2})$$

with $\mathbf{D}^{(i)}$ being the i th-order discrete derivative operator and $(\cdot)^{\circ 2}$ the element-wise square.

Applying the discrete Fourier transform

$$\hat{\mathbf{u}} = \mathbf{F}(\mathbf{u})$$

yields the KS equation in Fourier space,

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} = -(\hat{\mathbf{D}}^{(2)} + \hat{\mathbf{D}}^{(4)}) \hat{\mathbf{u}} - \frac{1}{2} \hat{\mathbf{D}}^{(1)} \mathbf{F}(\mathbf{u}^{\circ 2}), \quad (\text{A.3})$$

where $\hat{\mathbf{D}}^{(i)}$ are diagonal derivative operators in Fourier space with entries

$$\hat{D}_{jj}^{(i)} = (ik_j)^i.$$

Here, i is the imaginary unit, $k_j = 2\pi j' / L$ are wavenumbers, and j' is the signed mode index,

$$j' = \begin{cases} j, & 0 \leq j \leq n/2, \\ j - n, & n/2 < j < n. \end{cases}$$

We next define the linear and nonlinear terms,

$$\mathbf{C} = -\widehat{\mathbf{D}}^{(2)} - \widehat{\mathbf{D}}^{(4)}, \quad (\text{A.4a})$$

$$\mathbf{N}(\mathbf{u}) = -\frac{1}{2}\widehat{\mathbf{D}}^{(1)}\mathbf{F}(\mathbf{u}^{\circ 2}), \quad (\text{A.4b})$$

and we apply the one-third dealiasing rule by zeroing out all modes for which $|j'| > n/3$ when evaluating the nonlinear term. With these elements in hand, the KS equation becomes

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} = \mathbf{C}\widehat{\mathbf{u}} + \mathbf{N}(\mathbf{u}). \quad (\text{A.5})$$

Multiplying both sides by e^{-Ct} gives

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} e^{-Ct} - \mathbf{C}\widehat{\mathbf{u}} e^{-Ct} = \mathbf{N}(\mathbf{u}) e^{-Ct}.$$

We rearrange this as

$$\frac{\partial}{\partial t} (e^{-Ct} \widehat{\mathbf{u}}) = \mathbf{N}(\mathbf{u}) e^{-Ct},$$

and integrate it from t_{k-1} to t_k , where the solver time step is $\Delta t = t_k - t_{k-1}$,

$$\begin{aligned} \widehat{\mathbf{u}}(t_k) e^{-Ct_k} - \widehat{\mathbf{u}}(t_{k-1}) e^{-Ct_{k-1}} &= \int_{t_{k-1}}^{t_k} \mathbf{N}[\mathbf{u}(t)] e^{-Ct} dt, \\ \widehat{\mathbf{u}}(t_k) e^{-Ct_k} - \widehat{\mathbf{u}}(t_{k-1}) e^{-Ct_{k-1}} &= e^{-Ct_{k-1}} \int_0^{\Delta t} \mathbf{N}[\mathbf{u}(t_{k-1} + \tau)] e^{-C\tau} d\tau. \end{aligned}$$

Multiplying by e^{Ct_k} results in the final expression,

$$\widehat{\mathbf{u}}(t_n) = \widehat{\mathbf{u}}(t_{n-1}) e^{C\Delta t} + e^{C\Delta t} \int_0^{\Delta t} \mathbf{N}[\mathbf{u}(t_{n-1} + \tau)] e^{-C\tau} d\tau. \quad (\text{A.6})$$

Various numerical schemes can be used to approximate the integral. We employ the fourth-order exponential time-differencing Runge–Kutta method of Cox and Matthews [44].

A.2 Solver validation

Long-time KS trajectories exhibit predominantly low-frequency content, so only a modest number of spatial nodes are required for accurate simulation. In the literature, Linot et al. [37] used 64 nodes for $L \in \{22, 44, 66\}$, while Cvitanović [45] used 32 nodes for $L = 22$. In our numerical experiments, we adopt 64 nodes for $L \in \{22, 44\}$ and 72 nodes for $L = 66$. Figure 21 plots the average Fourier-coefficient magnitude versus mode number (left) and versus wavenumber (right) using snapshots from our dataset. These spectra confirm that the chosen spatial resolutions are sufficient to resolve the dynamically relevant frequency content of long-time solutions to the KS equation. In particular, the dominant energy lies at wavenumbers between 0 and 1, with a peak near k_{crit} , as predicted from Eq. (2.4).

Our main requirement is that the asymptotic statistical properties of our numerical solutions match those of the true dynamics. To evaluate this, Fig. 22 shows the Kaplan–Yorke dimension d_{KY} computed with our solver for $L \in \{22, 44, 66\}$ as a function of time step. The attractor dimension is nearly invariant with respect to Δt , indicating that the solver is stable and statistically consistent across a wide range of time steps. Edson et al. [35] report $d_{\text{KY}} = 5.198$ for $L = 22$. Using time steps $\Delta t \in \{0.01, 0.1, 0.5\}$, our solver yields values of ~ 5.23 , in close agreement with Edson and co. Our solver also reproduces the expected linear growth of d_{KY} with L , as reported in [35]. These comparisons support our supposition that our solver resolves the long-time statistics well. To balance computational cost and accuracy, we use $\Delta t = 0.1$ for $L \in \{22, 44\}$ and $\Delta t = 0.05$ for $L = 66$. Finally, we note that trajectories produced by our solver yield correct estimates of the IM dimension, per Fig. 2, further validating the fidelity of our scheme.

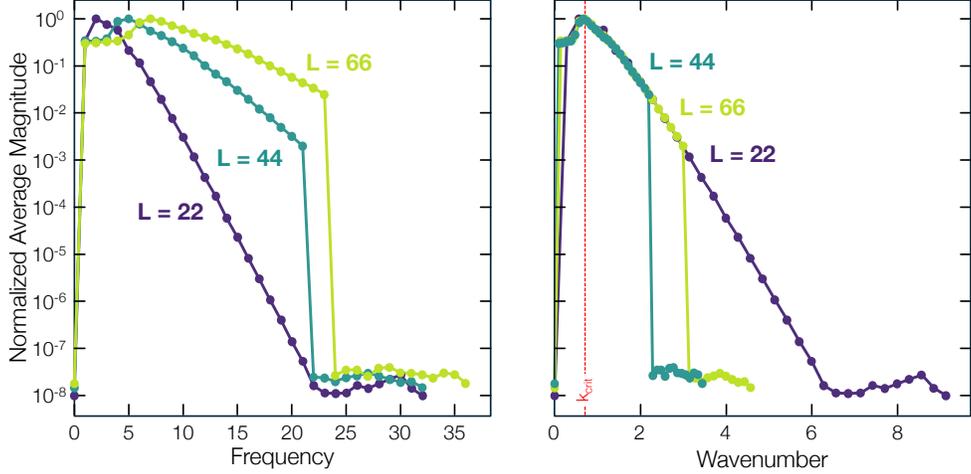


Figure 21: Average Fourier coefficient magnitude plotted versus frequency index (left) and versus wavenumber (right) for long-time trajectories in domains of length 22, 44, and 66.

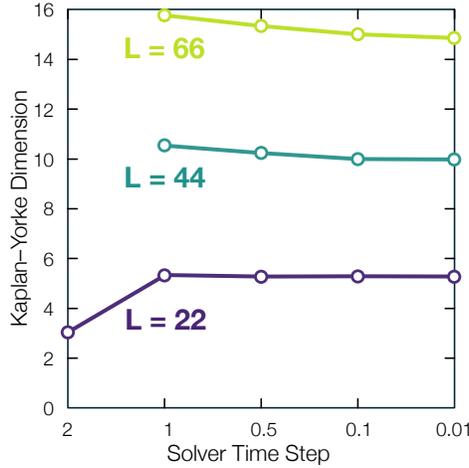


Figure 22: Kaplan–Yorke dimension d_{KY} versus solver time step Δt for domain lengths $L = 22, 44,$ and 66 .

Appendix B Lyapunov spectra

We use the algorithm of Benettin et al. [46] to compute the Lyapunov spectrum for the KS equation (see also Sandri [83]). The method begins with a set of orthonormal tangent vectors \mathbf{Q}_0 . They are advanced forward in time by k steps using the variational equation

$$\mathbf{V}_j = \mathbf{J}_k \mathbf{Q}_j, \quad (\text{B.1})$$

where \mathbf{J}_k is the flow map Jacobian for k time units of advancement. A QR decomposition is applied, $\mathbf{V}_j = \mathbf{Q}_{j+1} \mathbf{R}^{(j)}$, and the process is repeated with \mathbf{Q}_{j+1} . We perform K iterations of this cycle. The i th Lyapunov exponent ℓ_i , which measures the average exponential growth rate of the i th most unstable tangent direction, is computed as

$$\ell_i = \frac{1}{T} \sum_{j=0}^{K-1} \log(|R_{ii}^{(j)}|), \quad (\text{B.2})$$

where $R_{ii}^{(j)}$ is the i th diagonal entry of $\mathbf{R}^{(j)}$ and $T = Kk\Delta t$ is the integration time. Periodic application of the QR decomposition is essential to prevent the tangent vectors from collapsing onto the dominant mode of \mathbf{J}_k . Figure 23 shows Lyapunov spectra that we computed for $L \in \{22, 44, 66\}$, using a total time horizon of $T = 5 \times 10^5$ and performing QR decomposition every 2 time units, i.e., $k = 2/\Delta t$.

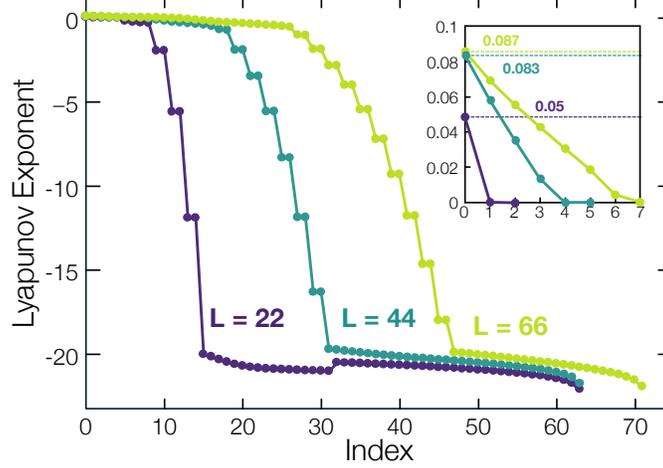


Figure 23: Lyapunov exponent spectra ℓ_i for $L \in \{22, 44, 66\}$ computed over a time horizon of $T = 5 \times 10^5$ with reorthogonalization every 2 time units.

Appendix C Autoencoder architecture and training

An autoencoder is a neural network composed of an encoder $E : \mathcal{M} \rightarrow \mathcal{L}$, which for us maps discrete states on the IM, $\mathbf{u} \in \mathcal{M} \subset \mathbb{R}^n$, into a lower-dimensional latent space \mathcal{L} , and a decoder $D : \mathcal{L} \rightarrow \mathcal{M}$, which approximates the inverse of E . Their composition, $A = D \circ E$, is trained to approximate the identity on \mathcal{M} such that all information in \mathbf{u} is preserved when compressed into the latent representation. Parameters of the autoencoder are learned by minimizing the loss

$$\mathcal{J} = \frac{1}{K} \sum_{k=1}^K \|\mathbf{u}_k - D[E(\mathbf{u}_k)]\|_2^2, \quad (\text{C.1})$$

where \mathbf{u}_k are the training samples. For each domain, these samples are drawn from the corresponding long-time rollout described in Sec. 3.1, and the networks are trained for 2000 epochs using the Adam optimizer.

The autoencoders used in this work comprise a sequence of fully connected layers. In addition, at the end of the encoder, we append a “linear block” composed of several fully connected linear layers, each with an output dimension equal to the latent dimension and with no biases or activation functions. This block encourages a latent space of low-rank [39, 84]. The architectures employed for each domain are summarized in Table 2, where n is the dimension of the state space and $d_{\mathcal{L}}$ is the latent dimension. We set $d_{\mathcal{L}}$ to 20 for $L = 22$, to 30 for $L = 44$, and to 50 for $L = 66$, although the results are insensitive to this hyperparameter.

In order to use the autoencoder for inference and to estimate $d_{\mathcal{M}}$, we perform a PCA in the latent space. First, we approximate the mean latent space vector as

$$\boldsymbol{\ell} = \frac{1}{K} \sum_{k=1}^K E(\mathbf{u}_k). \quad (\text{C.2})$$

We then construct the centered data matrix

$$\mathbf{X} = [E(\mathbf{u}_0) - \boldsymbol{\ell}, \dots, E(\mathbf{u}_K) - \boldsymbol{\ell}] \quad (\text{C.3})$$

and compute its SVD. The number of nontrivial singular values provides an estimate of $d_{\mathcal{M}}$. During inference, we restrict the latent representation to the dominant subspace by projecting out directions associated with negligible singular values. To do so we build a matrix $\mathbf{P} \in \mathbb{R}^{d_{\mathcal{L}} \times d_{\mathcal{M}}}$ using the leading $d_{\mathcal{M}}$ left singular vectors of \mathbf{X} . The reduced latent coordinate is obtained as

$$\mathbf{z} = \mathbf{P}^{\top} [E(\mathbf{u}) - \boldsymbol{\ell}], \quad (\text{C.4})$$

with approximate inverse

$$\mathbf{u} = D(\mathbf{P}\mathbf{z} + \boldsymbol{\ell}). \quad (\text{C.5})$$

Table 2: Autoencoder architecture. Here, n denotes the dimension of the state space, and $d_{\mathcal{L}}$ denotes the dimension of the latent space, set to 20 for $L = 22$, 30 for $L = 44$, and 50 for $L = 66$.

Component	Input Dim. \rightarrow Output Dim.	Activation	Bias
Encoder	$n \rightarrow 512$	Swish	Yes
	$512 \rightarrow 320$	Swish	Yes
	$320 \rightarrow d_{\mathcal{L}}$	No	Yes
Linear Block	$d_{\mathcal{L}} \rightarrow d_{\mathcal{L}}$	No	No
	$d_{\mathcal{L}} \rightarrow d_{\mathcal{L}}$	No	No
Decoder	$d_{\mathcal{L}} \rightarrow 320$	Swish	Yes
	$320 \rightarrow 512$	Swish	Yes
	$512 \rightarrow n$	No	Yes

Appendix D Discrete adjoint systems

In our adjoint state estimation problem, we seek to minimize the objective

$$\mathcal{J} = \sum_{k=0}^K M_k(\mathbf{u}_{\theta,k}, \mathbf{u}_k), \quad (\text{D.1})$$

where each term measures the discrepancy between the observer trajectory and the true trajectory at time index k ,

$$M_k(\mathbf{u}_{\theta,k}, \mathbf{u}_k) = \frac{1}{m} (\mathbf{u}_{\theta,k} - \mathbf{u}_k)^\top \mathbf{M}_k (\mathbf{u}_{\theta,k} - \mathbf{u}_k), \quad (\text{D.2})$$

and $\mathbf{M}_k \in \mathbb{R}^{n \times n}$ is a diagonal binary matrix selecting the measurement positions at measurement times. The observer trajectory is constrained by the discrete KS dynamics,

$$\mathbf{u}_{\theta,k+1} = f_{\Delta t}(\mathbf{u}_{\theta,k}). \quad (\text{D.3})$$

We solve this constrained minimization problem by introducing a Lagrangian that enforces the dynamics and deriving the associated adjoint equations. Below, we present the resulting systems for computing the gradient and Hessian of the loss.

D.1 Adjoint system for the gradient

We denote by \mathcal{A} the discrete initial condition and dynamical constraints,

$$\mathcal{A} = \underbrace{\mathbf{u}_0^\dagger (\mathbf{u}_\theta - \boldsymbol{\beta})}_{\text{initial condition}} + \underbrace{\sum_{k=0}^{K-1} \mathbf{u}_{k+1}^\dagger [\mathbf{u}_{\theta,k+1} - f_{\Delta t}(\mathbf{u}_{\theta,k})]}_{\text{system dynamics}}, \quad (\text{D.4})$$

where the adjoint variables \mathbf{u}_k^\dagger are Lagrange multipliers, arranged as row vectors, and $\boldsymbol{\beta}$ is the design parameter which determines the initial state, $\mathbf{u}_\theta = \boldsymbol{\beta}$. The Lagrangian is built as

$$\mathcal{L} = \mathcal{J} - \mathcal{A}. \quad (\text{D.5})$$

Substituting \mathcal{J} and \mathcal{A} , we get

$$\mathcal{L} = \sum_{k=0}^K M_k(\mathbf{u}_{\theta,k}, \mathbf{u}_k) - \mathbf{u}_0^\dagger (\mathbf{u}_\theta - \boldsymbol{\beta}) - \sum_{k=0}^{K-1} \mathbf{u}_{k+1}^\dagger [\mathbf{u}_{\theta,k+1} - f_{\Delta t}(\mathbf{u}_{\theta,k})]. \quad (\text{D.6})$$

This is rearranged to obtain

$$\mathcal{L} = M_K(\mathbf{u}_{\theta,K}, \mathbf{u}_K) - \mathbf{u}_0^\dagger (\mathbf{u}_\theta - \boldsymbol{\beta}) - \sum_{k=0}^{K-1} \left[\mathbf{u}_{k+1}^\dagger (\mathbf{u}_{\theta,k+1} - f_{\Delta t}(\mathbf{u}_{\theta,k})) - M_k(\mathbf{u}_{\theta,k}, \mathbf{u}_k) \right]. \quad (\text{D.7})$$

Because the dynamics are enforced during the simulation, the system constraint in \mathcal{A} is always satisfied, i.e.,

$$\mathcal{A} = 0, \quad (\text{D.8})$$

and hence

$$\mathcal{L} = \mathcal{J}. \quad (\text{D.9})$$

Differentiating \mathcal{L} with respect to β gives

$$\frac{\partial \mathcal{L}}{\partial \beta} = \frac{\partial M_K}{\partial \mathbf{u}_{\theta,K}} \frac{\partial \mathbf{u}_{\theta,K}}{\partial \beta} + \mathbf{u}_0^\dagger - \sum_{k=0}^{K-1} \left[\mathbf{u}_{k+1}^\dagger \left(\frac{\partial \mathbf{u}_{\theta,k+1}}{\partial \beta} - \frac{\partial f_{\Delta t}}{\partial \mathbf{u}_{\theta,k}} \frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \right) - \frac{\partial M_k}{\partial \mathbf{u}_{\theta,k}} \frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \right]. \quad (\text{D.10})$$

Pulling $\mathbf{u}_k^\dagger (\partial \mathbf{u}_{\theta,k} / \partial \beta)$ out of the summation, we get

$$\frac{\partial \mathcal{L}}{\partial \beta} = \mathbf{u}_0^\dagger + \left(\frac{\partial M_K}{\partial \mathbf{u}_{\theta,K}} - \mathbf{u}_K^\dagger \right) \frac{\partial \mathbf{u}_{\theta,K}}{\partial \beta} - \sum_{k=0}^{K-1} \left(\mathbf{u}_k^\dagger - \mathbf{u}_{k+1}^\dagger \frac{\partial f_{\Delta t}}{\partial \mathbf{u}_{\theta,k}} - \frac{\partial M_k}{\partial \mathbf{u}_{\theta,k}} \right) \frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta}, \quad (\text{D.11})$$

where

$$\frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} = \frac{\partial \mathbf{u}_{\theta,K}}{\partial \mathbf{u}_{\theta,K-1}} \frac{\partial \mathbf{u}_{\theta,K-1}}{\partial \mathbf{u}_{\theta,K-2}} \cdots \frac{\partial \mathbf{u}_{\theta,1}}{\partial \beta}. \quad (\text{D.12})$$

Unfortunately, direct computation of $\partial \mathbf{u}_{\theta,k} / \partial \beta \in \mathbb{R}^{n \times n}$ is prohibitively expensive. Since $\mathcal{A} = 0$ for all choices of the adjoint variables, we may select a sequence of \mathbf{u}_k^\dagger that annihilates the bracketed coefficients in Eq. (D.11). This yields the discrete adjoint recursion

$$\mathbf{u}_k^\dagger = \mathbf{u}_{k+1}^\dagger \frac{\partial \mathbf{u}_{\theta,k+1}}{\partial \mathbf{u}_{\theta,k}} + \frac{\partial M_k}{\partial \mathbf{u}_{\theta,k}}, \quad (\text{D.13a})$$

for $k = K-1, \dots, 0$, with terminal condition

$$\mathbf{u}_{\theta,K}^\dagger = \frac{\partial M_K}{\partial \mathbf{u}_{\theta,K}}, \quad (\text{D.13b})$$

Finally, because $\mathbf{u}_{\theta,0} = \beta$, we end up with

$$\frac{\partial \mathcal{J}}{\partial \mathbf{u}_\theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{u}_\theta} = \mathbf{u}_0^\dagger. \quad (\text{D.14})$$

D.2 Adjoint system for the Hessian

Second-order adjoints are commonly used to compute Hessian–vector products, but the modest dimension of the KS systems considered in this work allows us to form full Hessians. To derive an adjoint system for this purpose, we start by differentiating the transpose of Eq. (D.11) with respect to β :

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial \beta^2} &= \frac{\partial (\mathbf{u}_0^{\dagger \top})}{\partial \beta} + \frac{\partial}{\partial \beta} \left(\frac{\partial \mathbf{u}_{\theta,K}}{\partial \beta} \right)^\top \left(\frac{\partial M_K}{\partial \mathbf{u}_{\theta,K}} - \mathbf{u}_K^\dagger \right)^\top + \left(\frac{\partial \mathbf{u}_{\theta,K}}{\partial \beta} \right)^\top \left(\frac{\partial^2 M_K}{\partial \mathbf{u}_{\theta,K}^2} \frac{\partial \mathbf{u}_{\theta,K}}{\partial \beta} - \frac{\partial (\mathbf{u}_K^{\dagger \top})}{\partial \beta} \right) \\ &\quad - \sum_{k=0}^{K-1} \frac{\partial}{\partial \beta} \left(\frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \right)^\top \left(\mathbf{u}_k^\dagger - \mathbf{u}_{k+1}^\dagger \frac{\partial f_{\Delta t}}{\partial \mathbf{u}_{\theta,k}} - \frac{\partial M_k}{\partial \mathbf{u}_{\theta,k}} \right)^\top \\ &\quad - \sum_{k=0}^{K-1} \left(\frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \right)^\top \left(\frac{\partial (\mathbf{u}_k^{\dagger \top})}{\partial \beta} - \left[\frac{\partial}{\partial \mathbf{u}_{\theta,k}} \left(\frac{\partial f_{\Delta t}}{\partial \mathbf{u}_{\theta,k}} \right) \right]^\top \frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \mathbf{u}_{k+1}^{\dagger \top} - \left(\frac{\partial f_{\Delta t}}{\partial \mathbf{u}_{\theta,k}} \right)^\top \frac{\partial (\mathbf{u}_{k+1}^{\dagger \top})}{\partial \beta} - \frac{\partial^2 M_k}{\partial \mathbf{u}_{\theta,k}^2} \frac{\partial \mathbf{u}_{\theta,k}}{\partial \beta} \right). \end{aligned} \quad (\text{D.15})$$

After substituting $\mathbf{u}_\theta = \beta$ and $\mathbf{u}_{\theta,k+1} = f_{\Delta t}(\mathbf{u}_{\theta,k})$, we get our second-order adjoint system:

$$\frac{\partial (\mathbf{u}_k^{\dagger \top})}{\partial \mathbf{u}_\theta} = \left[\frac{\partial}{\partial \mathbf{u}_{\theta,k}} \left(\frac{\partial \mathbf{u}_{\theta,k+1}}{\partial \mathbf{u}_{\theta,k}} \right) \right]^\top \frac{\partial \mathbf{u}_{\theta,k}}{\partial \mathbf{u}_\theta} \mathbf{u}_{k+1}^{\dagger \top} + \left(\frac{\partial \mathbf{u}_{\theta,k+1}}{\partial \mathbf{u}_{\theta,k}} \right)^\top \frac{\partial (\mathbf{u}_{k+1}^{\dagger \top})}{\partial \mathbf{u}_\theta} + \frac{\partial^2 M_k}{\partial \mathbf{u}_{\theta,k}^2} \frac{\partial \mathbf{u}_{\theta,k}}{\partial \mathbf{u}_\theta}, \quad (\text{D.16a})$$

for $k = K - 1, \dots, 0$, with terminal condition

$$\frac{\partial(\mathbf{u}_{\theta,K}^{\dagger\top})}{\partial\mathbf{u}_{\theta}} = \frac{\partial^2 M_k}{\partial\mathbf{u}_{\theta,K}^2} \frac{\partial\mathbf{u}_{\theta,K}}{\partial\mathbf{u}_{\theta}}. \quad (\text{D.16b})$$

This adjoint system provides the Hessian via

$$\frac{\partial^2 \mathcal{J}}{\partial\mathbf{u}_{\theta}} = \frac{\partial^2 \mathcal{L}}{\partial\mathbf{u}_{\theta}} = \frac{\partial(\mathbf{u}_0^{\dagger\top})}{\partial\mathbf{u}_{\theta}}. \quad (\text{D.17})$$

References

- [1] N. Smith and W. D. Arnett, “Preparing for an explosion: hydrodynamic instabilities and turbulence in presupernovae,” *Astrophys. J.* **785**, 82 (2014).
- [2] N. Marati, C. M. Casciola, and R. Piva, “Energy cascade and spatial fluxes in wall turbulence,” *J. Fluid Mech.* **521**, 191–215 (2004).
- [3] I. A. Bolotnov, R. T. Lahey Jr, D. A. Drew, K. E. Jansen, and A. A. Oberai, “Spectral analysis of turbulence based on the DNS of a channel flow,” *Comput. Fluids* **39**, 640–655 (2010).
- [4] S. K. Robinson, “Coherent motions in the turbulent boundary layer,” *Annu. Rev. Fluid Mech.* **23**, 601–639 (1991).
- [5] A. J. Smits, B. J. McKeon, and I. Marusic, “High-Reynolds number wall turbulence,” *Annu. Rev. Fluid Mech.* **43**, 353–375 (2011).
- [6] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *One-dimensional “turbulence”* (Cambridge University Press, 2012), p. 214–235, Cambridge Monographs on Mechanics.
- [7] A. J. Linot, K. Zeng, and M. D. Graham, “Turbulence control in plane Couette flow using low-dimensional neural ODE-based models and deep reinforcement learning,” *Int. J. Heat Fluid Flow* **101**, 109139 (2023).
- [8] K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annu. Rev. Fluid Mech.* **51**, 357–377 (2019).
- [9] C. D. Argyropoulos and N. Markatos, “Recent advances on the numerical modelling of turbulent flows,” *Appl. Math. Modell.* **39**, 693–732 (2015).
- [10] J. P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis, “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences,” Tech. Rep. NF1676L-18332, National Aeronautics and Space Administration (2014).
- [11] A. W. Cary, J. Chawner, E. P. Duque, W. Gropp, W. L. Kleb, R. M. Kolonay, E. Nielsen, and B. Smith, “Cfd vision 2030 road map: Progress and perspectives,” in “AIAA aviation 2021 forum,” (2021), p. 2726.
- [12] A. Gronsksis, D. Heitz, and E. Mémin, “Inflow and initial conditions for direct numerical simulation based on adjoint data assimilation,” *J. Comput. Phys.* **242**, 480–497 (2013).
- [13] M. Asch, M. Bocquet, and M. Nodet, *Data Assimilation: Methods, Algorithms, and Applications* (SIAM, 2016).
- [14] T. Hayase, “Numerical simulation of real-world flows,” *Fluid Dyn. Res.* **47**, 051201 (2015).
- [15] T. A. Zaki and M. Wang, “Data assimilation and flow estimation,” in “Data Driven Analysis and Modeling of Turbulent Flows,” (Elsevier, 2025), pp. 129–181.
- [16] T. A. Zaki, “Turbulence from an observer perspective,” *Annu. Rev. Fluid Mech.* **57** (2025).

- [17] C. He, S. Li, and Y. Liu, "Data assimilation: new impetus in experimental fluid dynamics," *Exp. Fluids* **66**, 1–24 (2025).
- [18] G. Welch and G. Bishop, "An introduction to the Kalman filter," Tech. Rep. TR 95-041, University of North Carolina (1995).
- [19] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophys. Res.: Oceans* **99**, 10143–10162 (1994).
- [20] P. Clark Di Leoni, A. Mazzino, and L. Biferale, "Synchronization to big data: Nudging the Navier-Stokes equations for data assimilation of turbulent flows," *Phys. Rev. X* **10**, 011023 (2020).
- [21] A. Vela-Martín, "The synchronisation of intense vorticity in isotropic turbulence," *J. Fluid Mech.* **913**, R8 (2021).
- [22] C. C. Lalescu, C. Meneveau, and G. L. Eyink, "Synchronization of chaos in fully developed turbulence," *Phys. Rev. Lett.* **110**, 084102 (2013).
- [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.* **378**, 686–707 (2019).
- [24] S. Gesemann, F. Huhn, D. Schanz, and A. Schröder, "From noisy particle tracks to velocity, acceleration and pressure fields using B-splines and penalties," in "18th international symposium on applications of laser and imaging techniques to fluid mechanics, Lisbon, Portugal," , vol. 4 (2016), vol. 4.
- [25] L. Casa and P. Krueger, "Radial basis function interpolation of unstructured, three-dimensional, volumetric particle tracking velocimetry data," *Meas. Sci. Technol.* **24**, 065304 (2013).
- [26] H. Wang, Q. Gao, L. Feng, R. Wei, and J. Wang, "Proper orthogonal decomposition based outlier correction for PIV data," *Exp. Fluids* **56**, 43 (2015).
- [27] F.-X. Le Dimet and O. Talagrand, "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects," *Tellus A: Dyn. Meteorol. Oceanogr.* **38**, 97–110 (1986).
- [28] M. Wang, Q. Wang, and T. A. Zaki, "Discrete adjoint of fractional-step incompressible Navier-Stokes solver in curvilinear coordinates and application to data assimilation," *J. Comput. Phys.* **396**, 427–450 (2019).
- [29] C. Liu, Q. Xiao, and B. Wang, "An ensemble-based four-dimensional variational data assimilation scheme. part i: Technical formulation and preliminary test," *Mon. Weather Rev.* **136**, 3363–3373 (2008).
- [30] V. Mons, J.-C. Chassaing, T. Gomez, and P. Sagaut, "Reconstruction of unsteady viscous flows using data assimilation schemes," *J. Comput. Phys.* **316**, 255–280 (2016).
- [31] S. Zelik, "Inertial manifolds and finite-dimensional reduction for dissipative PDEs," *Proc. R. Soc. A* **144**, 1245–1327 (2014).
- [32] D. Kugiumtzis, "State space reconstruction parameters in the analysis of chaotic time series—the role of the time window length," *Physica D* **95**, 13–28 (1996).
- [33] E. R. Deyle and G. Sugihara, "Generalized theorems for nonlinear state space reconstruction," *Plos one* **6**, e18295 (2011).
- [34] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," *Physica D* **51**, 52–98 (1991).
- [35] R. A. Edson, J. E. Bunder, T. W. Mattner, and A. J. Roberts, "Lyapunov exponents of the Kuramoto–Sivashinsky PDE," *ANZIAM J.* **61**, 270–285 (2019).

- [36] X. Ding, H. Chaté, P. Cvitanović, E. Siminos, and K. Takeuchi, “Estimating the dimension of an inertial manifold from unstable periodic orbits,” *Phys. Rev. Lett.* **117**, 024101 (2016).
- [37] A. J. Linot and M. D. Graham, “Deep learning to discover and predict dynamics on an inertial manifold,” *Phys. Rev. E* **101**, 062209 (2020).
- [38] A. J. Linot and M. D. Graham, “Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations,” *Chaos* **32** (2022).
- [39] K. Zeng, C. E. P. De Jesus, A. J. Fox, and M. D. Graham, “Autoencoders for discovering manifold dimension and coordinates in data from complex dynamical systems,” *Mach. Learn.: Sci. Technol.* **5**, 025053 (2024).
- [40] H.-I. Yang, K. A. Takeuchi, F. Ginelli, H. Chaté, and G. Radons, “Hyperbolicity and the effective dimension of spatially extended dissipative systems,” *Phys. Rev. Lett.* **102**, 074102 (2009).
- [41] K. A. Takeuchi, H.-I. Yang, F. Ginelli, G. Radons, and H. Chaté, “Hyperbolic decoupling of tangent space and effective dimension of dissipative systems,” *Phys. Rev. E* **84**, 046214 (2011).
- [42] C. Foias, B. Nicolaenko, G. R. Sell, and R. Temam, “Inertial manifolds for the Kuramoto–Sivashinsky equation and an estimate of their lowest dimension,” *J. Math. Pures Appl.* **67**, 197–226 (1988).
- [43] D. A. Jones and E. S. Titi, “ C^1 Approximations of Inertial Manifolds for Dissipative Nonlinear Equations,” *Journal of Differential Equations* **127**, 54–86 (1996).
- [44] S. M. Cox and P. C. Matthews, “Exponential time differencing for stiff systems,” *J. Comput. Phys.* **176**, 430–455 (2002).
- [45] P. Cvitanović, R. L. Davidchack, and E. Siminos, “On the state space geometry of the Kuramoto–Sivashinsky flow in a periodic domain,” *SIAM J. Appl. Dyn. Syst.* **9**, 1–33 (2010).
- [46] G. Benettin, L. Galgani, A. Giorgilli, and J.-M. Strelcyn, “Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory,” *Meccanica* **15**, 9–20 (1980).
- [47] N. Chandramoorthy, P. Fernandez, C. Talnikar, and Q. Wang, “Feasibility analysis of ensemble sensitivity computation in turbulent flows,” *AIAA J* **57**, 4514–4526 (2019).
- [48] T. A. Zaki and M. Wang, “From limited observations to the state of turbulence: Fundamental difficulties of flow reconstruction,” *Phys. Rev. Fluids* **6**, 100501 (2021).
- [49] Y. Li, J. Zhang, G. Dong, and N. S. Abdullah, “Small-scale reconstruction in three-dimensional Kolmogorov flows using four-dimensional variational data assimilation,” *J. Fluid Mech.* **885**, A9 (2020).
- [50] P. Chandramouli, E. Mémin, and D. Heitz, “4D large scale variational data assimilation of a turbulent flow with a dynamics error model,” *J. Comput. Phys.* **412**, 109446 (2020).
- [51] M. Wang and T. A. Zaki, “Variational data assimilation in wall turbulence: from outer observations to wall stress and pressure,” *J. Fluid Mech.* **1008**, A26 (2025).
- [52] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” *Adv. Neural Inf. Process. Syst.* **27** (2014).
- [53] S. Paternain, A. Mokhtari, and A. Ribeiro, “A Newton-based method for nonconvex optimization with fast evasion of saddle points,” *SIAM J. Optim.* **29**, 343–368 (2019).
- [54] J. Sun, Q. Qu, and J. Wright, “Complete dictionary recovery over the sphere I: Overview and the geometric picture,” *IEEE Trans. Inf. Theory* **63**, 853–884 (2016).

- [55] R. Ge, F. Huang, C. Jin, and Y. Yuan, “Escaping from saddle points—online stochastic gradient for tensor decomposition,” in “Conference on learning theory,” (PMLR, 2015), pp. 797–842.
- [56] R. Ge, J. D. Lee, and T. Ma, “Matrix completion has no spurious local minimum,” *Adv. Neural Inf. Process. Syst.* **29** (2016).
- [57] K. Kawaguchi, “Deep learning without poor local minima,” *Adv. Neural Inf. Process. Syst.* **29** (2016).
- [58] A. J. Bray and D. S. Dean, “Statistics of critical points of Gaussian fields on large-dimensional spaces,” *Phys. Rev. Lett.* **98**, 150201 (2007).
- [59] Y. V. Fyodorov and I. Williams, “Replica symmetry breaking condition exposed by random matrix calculation of landscape complexity,” *J. Stat. Phys.* **129**, 1081–1116 (2007).
- [60] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples without local minima,” *Neural Networks* **2**, 53–58 (1989).
- [61] A. M. Saxe, J. L. McClellans, and S. Ganguli, “Learning hierarchical categories in deep neural networks,” in “Proceedings of the Annual Meeting of the Cognitive Science Society,” , vol. 35 (2013), vol. 35.
- [62] S. Saarinen, R. Bramley, and G. Cybenko, “Ill-conditioning in neural network training problems,” *SIAM J. Sci. Comput.* **14**, 693–714 (1993).
- [63] J. Greenstadt, “On the relative efficiencies of gradient methods,” *Math. Comput.* **21**, 360–367 (1967).
- [64] N. I. Gould and J. Nocedal, “The modified absolute-value factorization norm for trust-region minimization,” in “High Performance Algorithms and Software in Nonlinear Optimization,” (Springer, 1998), pp. 225–241.
- [65] E. Hopf, “A mathematical example displaying features of turbulence,” *Commun. Pure Appl. Math.* **1**, 303–322 (1948).
- [66] F. Takens, “Detecting Strange Attractors in Turbulence,” in “Dynamical Systems and Turbulence, Warwick 1980,” , vol. 898 of *Lecture Notes in Mathematics*, D. A. Rand and L.-S. Young, eds. (Springer, Berlin, Heidelberg, 1981), vol. 898 of *Lecture Notes in Mathematics*, pp. 366–381.
- [67] L. Noakes, “The Takens embedding theorem,” *Int. J. Bifurcation Chaos* **1**, 867–872 (1991).
- [68] T. Sauer, J. A. Yorke, and M. Casdagli, “Embedology,” *J. Stat. Phys.* **65**, 579–616 (1991).
- [69] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen, “Chaotic time series. Part I. Estimation of some invariant properties in state-space,” *Int. J. Modell. Identif. Control* **15** (1994).
- [70] A. M. Fraser and H. L. Swinney, “Independent coordinates for strange attractors from mutual information,” *Phys. Rev. A* **33**, 1134 (1986).
- [71] H. D. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, “The analysis of observed chaotic data in physical systems,” *Rev. Mod. Phys.* **65**, 1331 (1993).
- [72] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, “Reconstruction expansion as a geometry-based framework for choosing proper delay times,” *Physica D* **73**, 82–98 (1994).
- [73] J. Caputo, B. Malraison, and P. Atten, “Determination of attractor dimension and entropy for various flows: An experimentalist’s viewpoint,” in “Dimensions and Entropies in Chaotic Systems: Quantification of Complex Behavior,” (Springer, 1986), pp. 180–190.
- [74] J. F. Gibson, J. D. Farmer, M. Casdagli, and S. Eubank, “An analytic approach to practical state space reconstruction,” *Physica D* **57**, 1–30 (1992).
- [75] R. L. Bishop and S. I. Goldberg, *Tensor analysis on manifolds* (Courier Corporation, 2012).

- [76] D. Floryan and M. D. Graham, "Data-driven discovery of intrinsic dynamics," *Nat. Mach. Intell.* **4**, 1113–1120 (2022).
- [77] V. I. Oseledec, "A multiplicative ergodic theorem, Lyapunov characteristic numbers for dynamical systems," *Trans. Mosc. Math. Soc.* **19**, 197–231 (1968).
- [78] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in "Proceedings of the IEEE conference on computer vision and pattern recognition," (2018), pp. 7482–7491.
- [79] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *NeurIPS* **33**, 5824–5836 (2020).
- [80] R. Holland, "Finite-difference time-domain (FDTD) analysis of magnetic diffusion," *IEEE Trans. Electromagn. Compat.* **36**, 32–39 (2002).
- [81] P. G. Petropoulos, "Analysis of exponential time-differencing for FDTD in lossy dielectrics," *IEEE Trans. Antennas Propag.* **45**, 1054–1057 (2002).
- [82] C. Schuster, A. Christ, and W. Fichtner, "Review of FDTD time-stepping schemes for efficient simulation of electric conductive media," *Microwave Opt. Technol. Lett.* pp. 16–21 (2000).
- [83] M. Sandri, "Numerical calculation of Lyapunov exponents," *Mathematica Journal* **6**, 78–84 (1996).
- [84] L. Jing, J. Zbontar, and Y. LeCun, "Implicit rank-minimizing autoencoder," *Adv. Neural Inf. Process. Syst.* **33**, 14736–14746 (2020).