

Information theory for data-driven model reduction in physics and biology

Matthew S. Schmitt*,^{1,2} Maciej Koch-Janusz*,^{1,3,4} Michel Fruchart,^{1,5}

Daniel S. Seara,¹ Michael Rust,^{6,2} and Vincenzo Vitelli^{1,2,7}

¹*University of Chicago, James Franck Institute, 929 E 57th Street, Chicago, IL 60637*

²*University of Chicago, Department of Physics, 929 E 57th Street, Chicago, IL 60637*

³*Haiqu, Inc., 95 Third Street, San Francisco, CA 94103, USA*

⁴*Department of Physics, University of Zurich, 8057 Zurich, Switzerland*

⁵*ESPCI, Laboratoire Gulliver, 10 rue Vauquelin, 75231 Paris cedex 05*

⁶*University of Chicago, Department of Molecular Genetics and Cell Biology, Chicago, IL, 60637*

⁷*University of Chicago, Kadanoff Center for Theoretical Physics, 933 E 56th St, Chicago, IL 60637*

Model reduction is the construction of simple yet predictive descriptions of the dynamics of many-body systems in terms of a few relevant variables. A prerequisite to model reduction is the identification of these variables, a task for which no general method exists. Here, we develop an approach to identify relevant variables, defined as those most predictive of the future, using the so-called information bottleneck. We elucidate analytically the relation between these relevant variables and the eigenfunctions of the transfer operator describing the dynamics. In the limit of high compression, the relevant variables are directly determined by the slowest-decaying eigenfunctions. Our results provide a firm foundation to interpret deep learning tools that automatically identify reduced variables. Combined with equation learning methods this procedure yields the hidden dynamical rules governing the system's evolution in a data-driven manner. We illustrate how these tools work in diverse settings including model chaotic and quasiperiodic systems in which we also learn the underlying dynamical equations, uncurated satellite recordings of atmospheric fluid flows, and experimental videos of cyanobacteria colonies in which we discover an emergent synchronization order parameter.

The exhaustive description of a biological or physical system is usually impractical due to the sheer volume of information involved. As an example, the fluids in a steam engine may be described by a 10^{27} -dimensional state vector containing the positions and momenta of every particle in the engine. Yet, for most practical purposes, its working can be effectively described using only a small number of thermodynamic variables like pressure and temperature. Likewise, some aspects of the behavior of a nematode or a fruit fly, with 385 and $\sim 140,000$ neurons respectively, may be effectively captured by orders of magnitude fewer degrees of freedom [1–5]. Similar reductions can be achieved for systems ranging from diffusing particles to biochemical molecules and complex networks. In all cases, certain *relevant variables* can be predicted far into the future even though individual degrees of freedom in the system are effectively unpredictable.

The process by which one goes from the complete description of a system to a simpler one is known as model reduction. Diverse procedures for model reduction exist across the natural sciences. They range from analytical methods, such as adiabatic elimination and multiple-scale analysis [6–14], to data-driven methods such as dynamic mode decomposition and other linear projection-based methods [15–20], diffusion maps [21], spectral submanifolds [22, 23], independent component analysis [24], and deep encoder-decoder neural networks [25–32].

Model reduction consists of two ingredients: a decomposition of the full system into relevant and irrelevant variables, and a self-contained dynamical equation which governs the evolution of the relevant variables. In the absence of prior knowledge and intuition (e.g. a clear

separation of scales), identifying a decomposition that produces a minimal set of variables which are maximally descriptive of the full system is an open problem [9]. The answer to this question depends both on the time scale and precision with which one aims to describe the system [33, 34]. In addition, it also depends on one's ability to identify an adequate dynamical equation for the relevant variables, especially in data-driven settings, where the relevant variables and the dynamical equation are to be learned simultaneously.

To make progress, we develop an information-theoretic framework for model reduction. Very much like MP3 compression is about retaining information that matters most to the human ear [35], model reduction is about keeping information that matters most to predict the future [36, 37]. However, formalizing this intuitive statement requires some care in order to avoid trivial results, that manual approaches usually avoid using physical intuition. By carefully formulating model reduction within this information-theoretic perspective, we show that model reduction (the identification of both relevant variables and their effective dynamics) can be decomposed into a two-step process. First, the problem of variable identification can be reformulated in a way that removes any reference to the effective dynamics, and solved on its own. Second, once the relevant variables are known, an independent dynamical-equation learning step can yield the effective dynamics.

Our key step is to relate the model reduction objective to a lossy compression problem known as the information bottleneck (IB)[36, 38, 39], and to analytically show how and under what conditions the standard operator-

theoretic formalism of dynamical systems [27, 40], which underlies most methods of model reduction, naturally emerges from the optimal compression solution. This allows us to give a precise answer to the question of how to identify relevant and irrelevant variables. Crucially, our framework provides a criterion for when to stop increasing the complexity of a minimal model by directly measuring how informative relevant variables are of their own future and of the future of the entire system. Further, it provides a firm foundation to address a practical problem: the construction of deep learning tools to perform model reduction that are guaranteed to be interpretable. This practical realization of our protocol is based on recently-developed deep learning tools that allow scaling our approach to high dimensional systems [41].

The remainder of this paper is organized as follows. Section 1 formalizes model reduction and gives our information-theoretic starting point (see also SI Sections 1-3). Section 2 shows how the optimal information-bottleneck solution identifies the dominant (left) eigenfunctions of the system’s transfer operator (see also SI Section 4), and delineates a procedure to extract the reduced variables. In Sections 3 and 4 we show how information measures capture features of the system’s transfer operator spectrum and how this relates to the question of “when to stop” increasing the complexity of the reduced model. Section 5 (and SI Sections 5-6) shows that our framework can be used to learn relevant variables directly from data using neural networks. We illustrate our approach on several benchmark dynamical systems and demonstrate that it works even on uncurated datasets, *e.g.* satellite movies of atmospheric flows downloaded from YouTube. SI Section 7 shows applications to chaotic dynamics including the Lorenz-63 system, the Kuramoto-Sivashinsky equation and a hyperchaotic system. In Section 6 we show how our approach, combined with equation learning algorithms, forms a full model reduction pipeline yielding both the reduced variables describing the system and the equations governing its dynamics. Finally, in Section 7 (and SI Section 8) we apply our method to experimental microscopy videos of cyanobacteria colonies, discovering an emergent synchronization order parameter.

I. MODEL REDUCTION AS A COMPRESSION PROBLEM

Consider a system whose state is described by a (random) variable X_t , which might correspond to anything from the position of a single particle to an image of a fluid flow or the fluorescent molecules in a living system (Fig. 1a,b). The full state can be high dimensional, with a number of dimensions equal to the number of particles in a gas, or the number of pixels in an image. It is often difficult to predict the future state $X_{t+\Delta t}$ of such high-dimensional systems due to the enormous computational cost of evolving the system forward in time to

a desired accuracy. However, in many cases there exist some functions of the full state $H_t = h(X_t)$ which are low-dimensional and very informative of the future state of the system. These features often correspond to collective variables which evolve slowly in time. While one may guess the appropriate choice of variables $h(X_t)$ in contexts where all microscopic details of the system are known, we are interested in how to find such variables without such *a priori* knowledge.

We formulate the task of identifying the reduced variables and their evolution as an optimization problem. A “good” reduced model is one which entails a compressed description of the state, and which may be evolved in time in place of the full system to make predictions. The procedure consists of three steps: first, reduction of the full system by an encoding $h_{\text{enc}} : X_t \mapsto H_t$; second, evolution of the reduced state via an evolution operator $\mathcal{S} : H_t \mapsto H_{t+\Delta t}$; third, decoding the reduced state via a function $h_{\text{dec}}^{-1} : H_{t+\Delta t} \mapsto X_t$ to predict the full system’s state. These functions can be found by optimizing

$$\min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \dim h_{\text{enc}}(X_t) + \beta d(h_{\text{dec}}^{-1}(\mathcal{S}h_{\text{enc}}(X_t)), X_{t+\Delta t}). \quad (1)$$

The first term is a compressive constraint enforcing a “reduction” of the model, with the $\dim(\cdot)$ function measuring *e.g.* the number of bits used to represent the encoded state, or the sparsity of its representation in a preferred basis. The second term captures the error in our prediction of the full state when using the reduced model, measured by a “distance” $d(\cdot, \cdot)$. The parameter β sets the trade-off between dimensionality reduction and predictive power. Note that if the encoding h_{enc} is non-invertible (as is generally the case), the decoder h_{dec}^{-1} is defined probabilistically via a conditional distribution $h_{\text{dec}}^{-1}(H_{t+\Delta t}) \sim p(\cdot | H_{t+\Delta t})$. This is analogous to a procedure producing a thermodynamic ensemble compatible with the state variables. For instance, the canonical ensemble associates to every macrostate $h_{\text{enc}} = (N, V, T)$ defined by the number of particles, the volume, and the temperature a distribution $p_{\text{can}}(\cdot | N, V, T)$ over the set of microstates from which the decoded state is drawn, $h_{\text{dec}}^{-1} \sim p_{\text{can}}(\cdot | N, V, T)$.

The above objective depends on the choice of distance d , which incorporates the geometry of the state space and any desired biases about what is “important” in the system; it may differ from system to system. In what follows, we measure accuracy in terms of the mutual information I shared between the reduced model prediction and the full state. This abstracts away from the geometry of the space and characterizes the similarity of variables in terms of how predictive they are of each other, rather than how close they are, in a system-independent way. This has an important consequence: as a result of the data processing inequality, we can show that our objective function simplifies to one depending only on h_{enc} :

$$\min_{h_{\text{enc}}} \dim h_{\text{enc}}(X_t) - \beta I(h_{\text{enc}}(X_t), X_{t+\Delta t}) \quad (2)$$

The steps to this result are shown in detail in the Methods. This simplification is dramatic: instead of simultaneously learning the encoding function, evolution operator, and decoder, the only step we need to care about is the encoding. Consequently, we view model reduction in the rest of this paper as a task of finding the optimal reduced variables with which to represent our system. The evolution rule \mathcal{S} and decoder h_{dec}^{-1} are *induced* by the choice of h_{enc} , and we may view their identification as a secondary, downstream task to be performed after the proper variables have been found.

Before we proceed, we still need to specify the dimensionality reduction. This is subtle, as in different contexts different measures may seem natural, for instance, the entropy for discrete variables, or the number of components for continuous variables. We achieve our dimensionally-reduced variables with a two-step construction. First, we use a term explicitly forcing the reduced representation to lose information about the full state, $\min_h I(h(X_t), X_t)$ (for notational simplicity, we refer to h_{enc} as h). The minimal objective found in this way

$$\min_h \mathcal{L}_{\text{IB}} = I(X_t, h(X_t)) - \beta I(X_{t+\Delta t}, h(X_t)), \quad (3)$$

is known as the information bottleneck (IB) [38, 39]. It (and related variants [42, 43]) has been used in a variety of contexts, from document clustering [42] to neural codes [44] to (most relevant for us) dynamical systems [36, 45, 46], as well as in deep-learning contexts [41, 47, 48].

The IB objective ((3)) is a functional over stochastic functions, which may be represented by a conditional probability distribution $h(x_t) \sim p(h|x_t)$ (we denote random variables by uppercase X_t and their values by lowercase x_t). Crucially, the parameter β controls the trade-off between compression and prediction. For small β the compression term dominates and the optimal encoder is trivial, losing all information about the system. For intermediate β the compression term does not allow X_t to be completely represented by H_t , so features of X_t must compete to pass through to the encoding variable (Fig. 1b). These selected features are reflected in the IB encoder

$$p^*(h_t|x_t) = \arg \min \mathcal{L}_{\text{IB}} \quad (4)$$

which provides the optimal trade-off between retained information and predictability [37]. In the optimal encoder, information about X_t has been removed both by reducing the dimensionality, as well as by injecting noise into h_t . Increasing β can impact both of these, either by increasing the dimensionality to learn new features, or by reducing noise to better identify the features that have already been found. In order to “perfectly” identify the features given a fixed set of components, in the second step we modify the IB encoder by discarding the part attributable to noise alone. This effectively increases the coupling of the reduced variables to the already-learned

features. We shall see that this procedure isolates deterministic, interpretable reduced variables in both discrete and continuous settings.

II. THE OPTIMAL ENCODER IN TERMS OF THE TRANSFER OPERATOR

In any realistic experimental setting, the presence of noise or uncertainty means we cannot predict precisely the future state of a system but instead only a likely distribution of future states. Our prediction of the state at Δt in the future is then represented mathematically as $p(x_{t+\Delta t}|x_t)$, the probability of observing state $x_{t+\Delta t}$ given the current state x_t . For Markovian, or “memoryless” dynamics, this conditional probability distribution completely characterizes the dynamics of the system, and determines how probability distributions evolve in time. In this work we assume that the systems we consider are (or can be made) Markovian; for a detailed discussion of how to measure the memory and how it changes our below results, see SI Sections 3 and 4D. The evolution of probability distributions can then be understood as the action of a linear transfer operator $U^{\Delta t}$ on probability distributions:

$$p_{X_{t+\Delta t}}(x_{t+\Delta t}) = \int p(x_{t+\Delta t}|x_t)p_{X_t}(x_t)dx_t \equiv U^{\Delta t}[p_{X_t}] \quad (5)$$

$U^{\Delta t}$ can be decomposed as follows (using standard notation from quantum mechanics and explained in the SI):

$$U^{\Delta t} = \sum_n |\rho_n\rangle e^{\lambda_n \Delta t} \langle \phi_n| + U_{\text{ess}} \quad (6)$$

where $|\rho_n\rangle$ denote right eigenfunctions with eigenvalue $\Lambda_n \equiv e^{\lambda_n \Delta t}$ and $\langle \phi_n|$ are the corresponding left eigenfunctions. λ_n are the eigenvalues of the infinitesimal generator of $U^{\Delta t}$, known as the Fokker-Planck operator (Fig. 1d). The operator U_{ess} corresponds to the so-called essential spectrum, and we assume that it can be neglected. This is usually possible when the system is subjected to even a small amount of noise, or when some amount of uncertainty is present in the measurements [49, 50]. The eigenfunctions $\langle \phi_n|$ in (6) are in some sense “natural” features of the dynamics, as they evolve independently in time.

Our key observation is that the optimal IB encoder in (4), which we will use as a “filter” to extract relevant features, can be expressed in terms of the eigenvalues λ_n and left eigenfunctions ϕ_n of (the generator of) U ,

$$p_\beta^*(h_t|x_t) = \frac{p_\beta^*(h_t)}{\mathcal{N}(x_t)} \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x_t) f_n(h_t) \right] \quad (7)$$

where $f_n(h_t)$ are factors that do not depend on x_t . For an outline of the mathematical steps leading to this see Methods, as well as SI Section 4. The combined effect of

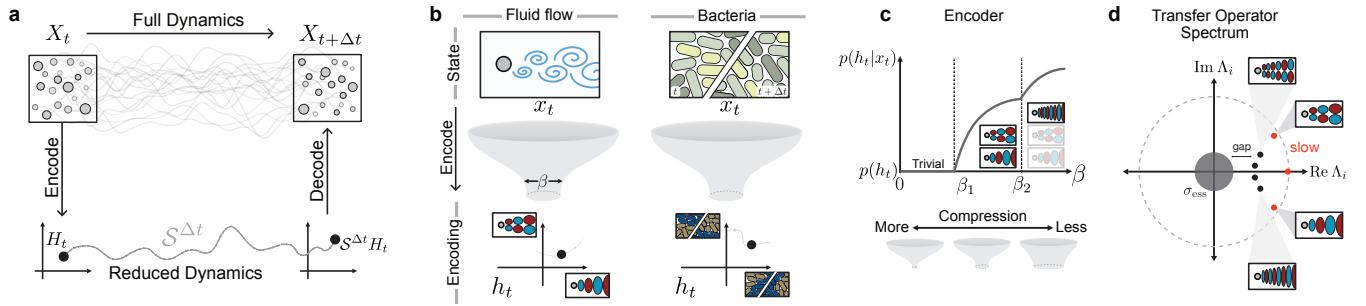


FIG. 1. Interpretable dynamical variables in reduced models via the information bottleneck. (a) In general model reduction aims to both learn an encoding and reduced dynamics, so that the reduced variable serves as a useful proxy that may be decoded for an estimate of the full system. Requiring the reduced variable to only be informative of the full state decouples these two steps (see SI Section 2). (b) The information bottleneck compresses high-dimensional state variables x_t , into simpler encoding variables h_t with a controllable trade-off between the degree of compression and the predictive power about the system's future. With deep neural networks, the encoding can be computed directly from data of observed fluid flows (left) or biological datasets, such as fluorescently labeled bacteria colonies (right). In general, the state of the variable x_t may comprise time-lagged variables of the intensity field, $x_t = \{I_t, I_{t+\Delta t}\}$ (right). The amount of compression is determined by the “width” of the bottleneck β (see (3)). The resulting compressed, or encoded, variables h_t represent collective variables most predictive of the system's future. (c) Schematic evolution of the encoder $p(h_t|x_t)$ for varying compression strength β . For low β (high compression), the encoder is trivial and forgets everything about the input x_t . After the first IB transition at β_1 , the encoder becomes non-trivial by gaining some dependence on x_t ; some features of the input are able to pass through the bottleneck. At subsequent IB transitions, additional features are learned. (d) The point spectrum of the transfer operator contains several slowly decaying modes (red). We show that the most predictive variables that IB systematically extracts correspond to the slowest eigenfunctions of the transfer operator, associated to eigenvalues Λ_i with $|\Lambda_i| \approx 1$. In fluid flows, the slowest-decaying eigenfunctions typically represent large-scale coherent patterns of the flow field, while faster-decaying eigenfunctions correspond to variations over shorter length scales. In general the system may exhibit an essential spectrum, however for noisy systems this is small (see main text).

the factors in the exponent determines what features the encoder learns about the state x_t . In general, there may be a large number of non-zero factors f_n so that the learned features are difficult to interpret in terms of individual eigenfunctions ϕ_n . However, things become simple in the limit of small β , or high compression. When β is small the encoder is trivial: $p(h_t|x_t) = p(h_t)$ so the value h_t is assigned at random with no regard to the state of the system. No feature has been learned, and all factors f_n are equal to zero. As β is increased, the encoder undergoes a series of transitions at $\beta = \beta_1 < \beta_2 < \beta_3 \dots$ where new features are allowed to pass through the bottleneck (Fig. 1c) [51–54]. The first transition happens at a finite value of β_1 when the first, most predictive, feature is learned.

Surprisingly, we find that at the first IB transition the vector of f_n coefficients is dominated by a single term f_1 .

$$p_\beta^*(h_t|x_t) \approx \frac{p_\beta^*(h_t)}{\mathcal{N}(x_t)} \exp\left(\beta e^{\lambda_1 \Delta t} \phi_1(x_t) f_1(h_t)\right) \quad (8)$$

This is our main mathematical result, which we derive by considering a perturbative expansion of the IB objective for small f_n . A proof of Eq. 8 with clearly specified technical assumptions may be found in Section 4 of the SI. One notable assumption underpinning this result, in addition to Markovianity, is that the system is in a statistical steady state, so that x_t is sampled from the steady state distribution. When this assumption is violated,

eigenfunctions of the transfer operator may cease to carry information about the system's dynamics and instead its pseudospectrum may become relevant [55]; see SI Section 4D.

The above result shows that in the limit of high compression the encoder's dependence on x_t is given by the first left eigenfunction $\phi_1(x_t)$, which is the slowest-varying function of the state under dynamics given by U . Therefore, Eq. 8 makes precise the intuitive statement that slow features are the most relevant for predicting the future. Our analytical result, while applying only to the dominant eigenfunction, is valid for arbitrary, including non-Gaussian, variables.

We further observe numerically that this picture holds true more generally: at successive IB transitions, the learned features correspond to successive modes of the transfer operator. This is consistent with the exact results known for the specific case of Gaussian IB, where the encoder learns successive eigenvectors of a matrix related to the covariance of the joint $X_t, X_{t+\Delta t}$ distribution at each IB transition [53].

Due to the smallness of f_1 at the transition, the optimal IB encoder couples only weakly to the eigenfunction ϕ_1 , so that the dynamics of h_t will be dominated by noise. Given the maximal number of available bits in memory, or variables used in an analytical model, the IB solution will thus not make the best use of the resources (IB is optimal with respect to the amount of information retained, but

not the size of its representation [43]). However, we can use the optimal IB encoding as a filter to identify ϕ_1 , and then “distill” it by discarding the noisy part of the encoding, effectively increasing the coupling between h_t and $\phi_1(x_t)$ while keeping the number of components fixed. In practice, this is done by taking the most likely value for the reduced state, $h_t = \arg \max_h p(h|x_t)$ (see SI Section 4E). Similar procedures are performed in other probabilistic dimensionality reduction approaches, *e.g.* probabilistic principal component analysis [56]. There one finds a stochastic mapping from the full state x to the reduced representation h , and then discards the noise to obtain the deterministic reduction.

Together this lays the foundation of an operational prescription for model reduction which is built on IB for identifying relevant, slow, features. As we show later, this insight can be leveraged to systematically learn these slow variables directly from data with neural networks [41] which in turn may be used as inputs to an equation learning pipeline.

III. INFORMATION DECAY AND THE SPECTRUM OF THE TRANSFER OPERATOR

To develop intuition for information in a dynamical system, we turn to the simple example of a Brownian particle in a double-well potential. This may represent, for example, a molecule with a single degree of freedom that transitions between two metastable configurations [57]. In the overdamped limit the state of the particle is completely determined by its position $X_t \in \mathbb{R}$, with dynamics given by the Langevin equation

$$\dot{x}_t = -\partial_x V(x_t) + \sigma \eta_t. \quad (9a)$$

$$V(x) = \frac{1}{4}(\mu - x^2)^2 \quad (9b)$$

Here, η_t is unit-variance white noise, σ controls its strength, and μ controls the shape of the potential $V(x)$.

The deterministic dynamical system undergoes a bifurcation at $\mu = 0$ (Fig. 2a). To quantify the amount of information about the future state $X_{t+\Delta t}$ contained in the initial state X_t we compute their mutual information (see SI for details). The dynamics of X_t are Markovian, so that for any sequence of times $t_0 < t_1 < t_2$, $p(X_{t_2}|X_{t_1}, X_{t_0}) = p(X_{t_2}|X_{t_1})$. From the data processing inequality, one has [58]

$$I(X_{t_2}, X_{t_0}) \leq I(X_{t_1}, X_{t_0}),$$

which implies that information can only decrease in time.

What governs the rate at which information decays? Here we can already see the role of the spectrum of the dynamics’ transfer operator. By exploiting the spectral expansion of the conditional distribution $p(x_{t+\Delta t}|x_t)$ one finds that for long times the information decays as

$$I(X_t, X_{t+\Delta t}) = e^{2\lambda_1 \Delta t} \langle \phi_1^2 \rangle \langle \rho_1^2 / \rho_0^2 \rangle + \mathcal{O}(e^{2\lambda_2 \Delta t}) \quad (10)$$

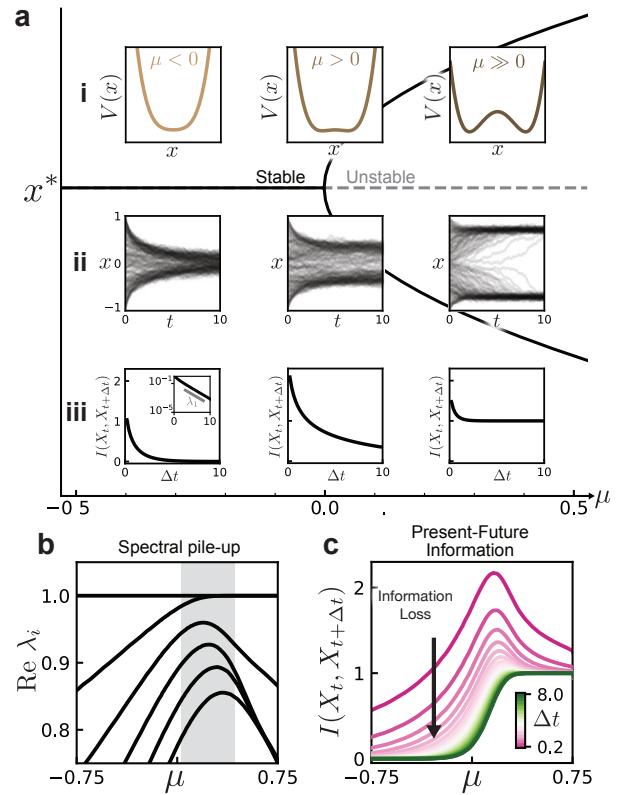


FIG. 2. Information loss of a Brownian particle in a double well potential. (a) Fixed point (FP) diagram of the dynamics given by Eq. 9 for zero noise. There is a bifurcation at $\mu = 0$ where the stable FP at $x = 0$ becomes unstable and two new stable FPs appear at $\pm \sqrt{\mu}$. (a,i) The corresponding potential $V(x)$ for varying μ , with the emergence of a double-well structure for $\mu > 0$. (a,ii) Dynamics of the system Eq. 9 for varying values of μ corresponding to the potentials above, with uniformly-distributed initial conditions. (a,iii) Loss of information between the initial condition and the future state. Inset shows scaling given by the first eigenvalue of the transfer operator. (b) Spectrum of the transfer operator U , showing a pile-up of eigenvalues for $\mu \gtrsim 0$. These are related to the eigenvalues of its infinitesimal generator by $\Lambda_i = e^{\lambda_i \Delta t}$. (c) Mutual information between the present and future state for varying time delay Δt and bifurcation parameter μ .

where expectations are taken over the steady state distribution (see SI Section 3). Asymptotically, the information decay is set by the value of λ_1 , the rate of decay of the slowest-varying function $\phi_1(x)$ under the dynamics of U . In the limit of infinite time, for any value of μ even weak noise will cause the mutual information to become zero as there is a non-zero (perhaps exponentially small) probability of hopping between the wells [49].

The loss of information in time depends on the bifurcation parameter μ as summarized in Fig. 2c. Note the peak in $I(X_t, X_{t+\Delta t})$ for small, positive μ . This corresponds to dynamics where observation of X_t strongly informs the future state; recall that the mutual information is maximized when the conditional entropy $\mathcal{H}(X_{t+\Delta t}|X_t) \approx 0$

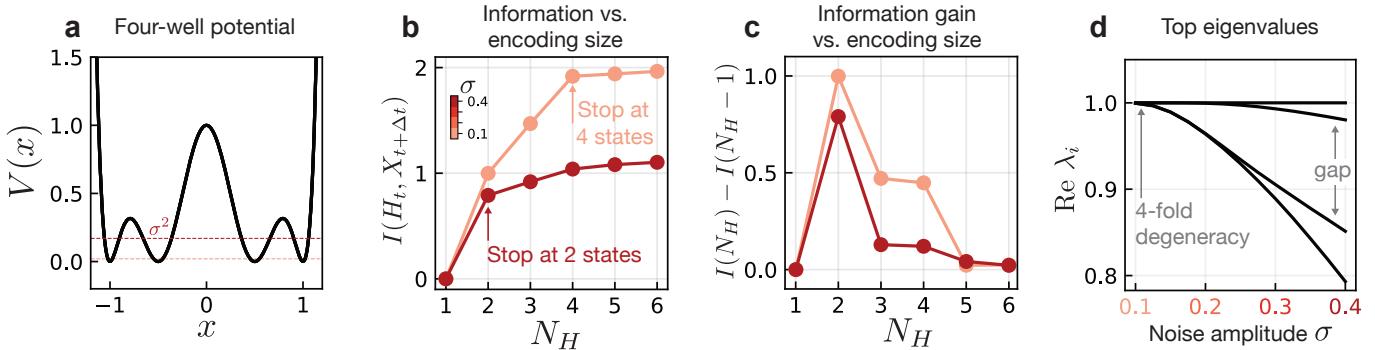


FIG. 3. “Knowing when to stop”. The spectral properties of the transfer operator determine the necessary complexity (i.e. “when to stop” [33]) of the reduced model, which we show is also visible in information theoretic metrics. (a) Four-well potential in which a Brownian particle fluctuates. The magnitude σ of the fluctuating noise is related to an energy scale $E_\sigma = \sigma^2$. (b) Information contained in the encoding variable H_t about the future state $X_{t+\Delta t}$ for varying levels of noise and alphabet sizes N_H . (c) Information gain achieved by increasing the alphabet size by a single variable. This is the discrete derivative of the curve in (b). (d) Spectrum of the transfer operator for changing values of noise amplitude.

(see SI Section 1). In contrast, for large positive or negative μ , X_t is not as informative of $X_{t+\Delta t}$ even for small times: the initial state is quickly forgotten as the particle approaches the bottom of the single (for $\mu < 0$) or double (for $\mu > 0$) well.

This phenomenon is reminiscent of critical slowing down, which occurs in the noise-free system as μ passes through the bifurcation at $\mu = 0$. For the deterministic dynamics, the slowing down is reflected in the spectrum as a “pile up” of eigenvalues to form a continuous spectrum [49]. In the presence of noise, although the continuous spectrum becomes discrete [49, 50] there is still a pile-up of eigenvalues characterized by several eigenvalues becoming close to 1 (Fig. 2b). This pile-up gives rise to the information peak seen in Fig. 2c. The peak is not solely due to the closing spectral gap $\lambda_1 - \lambda_2$, but is also impacted by the subdominant eigenvalues which accumulate at $\mu \approx 0.2$ (SI Fig. S2).

IV. KNOWING WHEN TO STOP

For discrete encoding variables h , the information bottleneck partitions state space and reduces the dynamics on x to a discrete dynamics on h . Such reductions of complex systems to symbolic sequences via partitioning of state space has attracted attention for more than half a century in both theoretical and data-driven contexts [4, 59–64]. Several works have approached this partition problem from a dynamical systems perspective, linking optimal partitions to eigenfunctions of the (adjoint) transfer operator [33, 65]. In this setting, a central question is “when to stop” [4, 33, 34, 63]: how many states does h need in order to capture statistical properties of the original dynamics?

We consider this question by finding the optimal IB encoder in the limit of low compression, $\beta \gg 1$, but fixed encoding capacity N_H (where $H_t \in \{0, \dots, N_H - 1\}$), i.e.

the encoder is only restricted by the number of symbols it can use. An analogous setup was used in the context of renormalization group (RG) transformations in [66–68], which results in effective model reduction due to the “sloppiness”, or irrelevance, of certain system variables [69, 70]. In this regime, the encoder learned by IB is deterministic; we are learning an optimal hard partition of state space. This can be seen by noting that $I(H_t; X_{t+\Delta t}) = H(H_t) - H(H_t|X_{t+\Delta t})$ is maximized when the latter term is zero, which happens when x_t unambiguously determines h_t , i.e. when $p(h_t|x_t) \in \{0, 1\}$ for all x_t . The details of how the encoder is computed are discussed in the next section.

Consider a fluctuating Brownian particle as in the double well above, where now each of the wells is split into two smaller wells, giving a total of four potential minima (Fig. 3a). As the system is in steady state, the variance of the fluctuations defines an energy scale $E_\sigma = \sigma^2 = 2k_B T$. For small E_σ , the system rarely transitions between the four potential minima. In this case, knowledge of the initial minimum is very informative of the future state of the particle. In contrast, for large fluctuations the particle can spontaneously jump between shallow minima in each large well, so that the system immediately forgets about the precise potential minimum it was in. Information about the shallow minima has been “washed out”, and only the information about the larger double-well structure remains.

To see this reflected in the information, we again consider an encoding of the initial state into a discrete variable $H_t \in \{0, \dots, N_H - 1\}$. In both the small and large noise scenarios, a variable with $N_H = 2$ encodes approximately one bit of information (Fig. 3b), corresponding to an H_t which distinguishes the two large wells for $x \leq 0$. For large noise this is essentially all the information that can be learned; increasing the capacity of the encoding variable beyond this provides only marginally more information about the future state (Fig. 3c). In the small

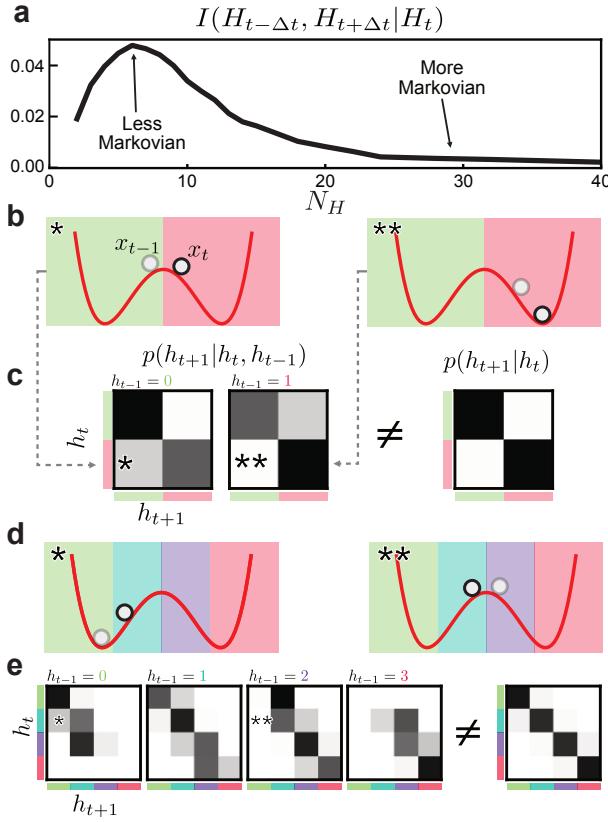


FIG. 4. Non-Markovianity for a Brownian particle in a double well potential. (a) Conditional mutual information $I(H_{t-\Delta t}, H_{t+\Delta t} | H_t)$ as a function of N_H . (b) Two possible scenarios (left and right) with the same value of h_t , but different h_{t-1} . Here, the circles represent the value of the full state x_t while the colored regions represent the value of the reduced state $h_t \in \{1, 2\}$ which are mapped to for each x_t . (c) Illustration of the transition probability which takes into account the previous state h_{t-1} (left), compared to the Markovian transition probability obtained by marginalizing out h_{t-1} (right). In a Markovian system, these transition probabilities would coincide. (d-e) Same as in (h), but for $N_H = 4$ states.

noise case, the information between the encoding and the future state continues to increase to approximately two bits at $N_H = 4$, after which it plateaus. The encoding has learned to distinguish each of the four potential wells. These observations are reflected in the transfer operator spectrum shown in Fig. 3d. For small noise, the eigenvalue $\lambda = 1$ is nearly four-fold degenerate, indicating the existence of four regions that can evolve independently under U , giving rise to four steady state distributions satisfying $U\rho = \rho$. These regions correspond to the potential minima. Hops between the separate minima are exceedingly rare, so that the dynamics essentially take place in the four minima independently. With larger σ the degeneracy is lifted, resulting in one dominant subleading eigenvalue followed by a gap. The corresponding eigenfunction is one which is positive (negative) on the right (left) side of the large potential barrier at $x = 0$: the only relevant

piece of information is which of the large wells the initial condition is contained in, and all other information is lost exponentially quickly.

In many cases, our goal is not only to find a reduced representation H_t that is predictive of the future state, but one which may be evolved forward for long times as a surrogate for the full system. This is a practical but much more stringent requirement of our reduced state, and it may result in a different answer to the question of “when to stop.”

In general the dynamics of the reduced variable will be non-Markovian, meaning that its evolution will be determined not only by its current state, but also its past. With a fixed encoder $p(h_t|x_t)$, transition probabilities $p(h_{t+1}|h_t)$ are induced by decoding h_t to x_t , evolving x_t forward in time to x_{t+1} , and then encoding x_{t+1} to h_{t+1} (here we assume discrete time dynamics with $\Delta t = 1$ for simplicity). Information theory provides a natural quantification of the (non-)Markovianity of the dynamics as measured by the conditional mutual information $I(H_{t-1}, H_{t+1} | H_t)$ (see SI Section 3 and alternative approaches in Refs. [63, 71, 72]). This quantity tells us how much additional information the past state H_{t-1} contains about the future H_{t+1} , on top of what is contained in the present state H_t .

To illustrate this, we again consider a particle in a double well, which we compress into a discrete variable $H_t \in \{0, \dots, N_H - 1\}$. Because the full system is Markovian by design, we expect that for large N_H the conditional mutual information $I(H_{t-1}, H_{t+1} | H_t)$ should go to zero as H_t begins to more closely approximate the full state X_t (Fig. 4a). Interestingly, the approach to Markovianity is highly non-monotonic, and the system is more non-Markovian with $N_H = 4$ than with $N_H = 2$.

The role of memory is illustrated in the two cases shown in Fig. 4b. In both cases, the current state is in the right well, so that $h_t = 1$. However, more precise information about the state (and hence its future) is revealed by knowing the system’s past. If $h_{t-1} = 0$ (so the state was in the left well), it is likely that the full state is near the boundary between wells, so that $h_{t+1} = 0$ is more likely than if $h_{t-1} = 1$, which would suggest that the state is deep in the right well. This behavior can be quantified by computing the transfer probabilities $p(h_{t+1}|h_t, h_{t-1})$ (Fig. 4c). These differ significantly from $p(h_{t+1}|h_t)$, which would not be the case for Markovian dynamics. From these, we see that the primary contribution to the non-Markovian dynamics occurs when $(h_{t-1}, h_t) = (0, 1)$ or $(1, 0)$, which can only happen when the state is very near to the boundary between the wells. As N_H increases, the likelihood that the state is near the boundary of two regions with different h_t labels will increase, which will increase the non-Markovianity of the system (Fig. 4d-e). However, at the same time the different h_t regions will shrink in size so that h_{t-1} will no longer provide any meaningful information about the system’s location x_t . The trade-off between these two effects leads to the observed non-monotonic behavior of $I(H_{t-1}, H_{t+1} | H_t)$.

V. DATA-DRIVEN DISCOVERY OF SLOW VARIABLES

IB finds a reduced state variable by optimizing an information theoretic-objective that makes no reference to physics or dynamics. This suggests it may be used for the discovery of slow variables in situations where one lacks physical intuition. In the regime of small β , or high compression, features of the state x_t are forced to compete to make it through the bottleneck h_t , which allows us to extract relevant features of the system.

These variables coincide with left eigenfunctions of the transfer operator, as discussed in Section II. We verify this theoretical result numerically in cases where the IB objective (3) can be solved exactly by using an iterative scheme known as the Blahut-Arimoto algorithm [38, 58] (see SI Section 3). By taking the logarithm of the encoder as it begins to deviate from the trivial uniform encoder above β_1 we can confirm that it depends on x only through $\phi_1(x)$ (SI Fig. S3). This can be independently verified by studying the stability of the trivial encoder with respect to perturbations in the parameters $f_n(h_t)$ in Eq. 7, where we see that the encoder becomes unstable to perturbations in f_1 , as predicted by our theoretical results (SI Section 4).

The utility of exact IB for variable discovery is limited because it requires knowledge of the exact conditional distribution $p(x_{t+\Delta t}|x_t)$ which is difficult to estimate in many real-world scenarios. Fortunately however, the IB optimization problem can be replaced by an approximate variational objective introduced in Ref. [41] that can be solved with neural networks and which we refer to as variational IB. This is achieved by introducing a tractable Ansatz for the form of $p(h_t|x_t)$; we show in SI Section 4 that this does not change the encoder's dependence on transfer operator eigenfunctions.

Our core result linking the behavior of the learned encoder with transfer operator eigenfunctions remains valid even for variational IB applied to high-dimensional systems, which we show by considering a simulated data of fluid flow past a disk [74]. The state of the system is given by a two-dimensional velocity field $\mathbf{v}(\mathbf{x}) \in \mathbb{R}^{2 \times N_{\text{pixels}}}$, where $N_{\text{pixels}} \sim \mathcal{O}(10^5)$ (Fig. 5a). Fluid flows in from the left boundary with a constant velocity $v_0 \hat{\mathbf{e}}_x$ past a disk of unit diameter. At Reynolds number $\text{Re} \gtrsim 150$, the fluid undergoes periodic vortex shedding behind the disk, forming what is known as a von Kármán street.

The eigenfunctions for this system are in general complex, and come in conjugate pairs: $\phi_2(x) = \phi_1^*(x)$. In this situation any linear combination of ϕ_1 and ϕ_2 will decay at the same rate, and hence we expect to learn some arbitrary combination of the two dominant eigenfunctions, or equivalently a combination of the real and imaginary parts of ϕ_1 . We therefore take a two-dimensional encoding variable $[h_0, h_1]$, so that it can represent the full complex eigenfunction.

Our learned latent variables are oscillatory with the correct frequency as shown in Fig. 5b. However, this

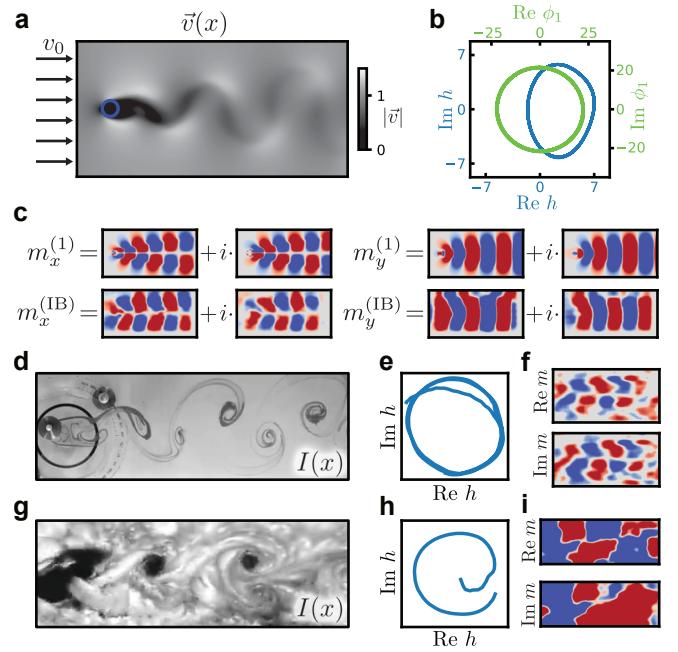


FIG. 5. Variational IB for high-dimensional simulated fluid flow. (a) Fluid flow past a disk with uniform in-flow velocity v_0 exhibits vortex shedding behind the object in a so-called von Kármán street. The state of the system is given by a spatially varying two-component vector field $\mathbf{v}(\mathbf{x})$. (b) Dynamics in latent space (blue) traverse a nearly circular trajectory (see Supplementary Movie 1). For comparison we show the evolution of the mode amplitudes obtained by projecting the velocity field onto the first DMD mode (green). (c) Comparison of the first Koopman mode obtained from DMD ($\mathbf{m}^{(1)}$) and from VIB ($\mathbf{m}^{(\text{IB})}$). Koopman modes from VIB are computed as gradients of the latent encoding variables as described in the main text. Red corresponds to positive values and blue to negative; the magnitudes of the modes are not directly comparable. (d) Intensity field $I(x)$ from a video of water flowing past an obstacle; video from Ref. [73]. (e) Trajectory of the system in latent space. (f) Gradients of the latent encodings with respect to the input field $I(x)$. (g-i) Same as (d-f), but using satellite images of clouds near the island of Guadalupe as our state $I(x)$.

alone does not indicate whether we are learning the correct eigenfunctions. Because the system is well approximated by linear dynamics, eigenfunctions of the adjoint transfer operator are linear functions of the state variable, and may hence be represented as $\phi_n[\mathbf{v}] = \langle \mathbf{v}(\mathbf{x}), \mathbf{m}^{(n)}(\mathbf{x}) \rangle$, where angled brackets denote an average over space and $\mathbf{m}^{(n)}(\mathbf{x})$ are so-called Koopman modes. The true $\mathbf{m}^{(n)}$ (and hence the eigenfunctions) can be thus computed via dynamic mode decomposition (DMD) [15, 16] as described in the SI. For the functions $h[\mathbf{v}]$ learned using variational IB, we can infer the corresponding $\mathbf{m}^{(\text{IB})}$ using gradients of the neural network with respect to the input field. Fig. 5c shows these inferred modes $\mathbf{m}^{(\text{IB})}$ compared to the true mode $\mathbf{m}^{(1)}$ (see SI Section 5 for details). This shows that variational IB not only recovers the essential oscillatory nature of the dynamics, but does so by learning the correct

slowly varying functions of the state variable given by the adjoint transfer operator eigenfunctions.

Our framework provides interpretability for the learned latent variables even in messy real-world fluid flow datasets scraped directly from videos on YouTube [73, 75] (Supplementary Movie 1). The first shows a von Kármán street which forms as water passes by a cylindrical obstacle at Reynolds number 171, with flow visualized by a dye injected at the site of the obstacle [73]. We take a background-subtracted grayscale image of the flow field as our input (Fig. 5d) and task VIB with learning a two-dimensional latent variable as above. Also here, variational IB learns oscillatory dynamics of the latent variables (Fig. 5e). We visualize the function learned by the encoder by considering gradients of the latent variables, which show the same structure as those obtained for the x component of the simulated data (Fig. 5f). This is expected, as the x -component of the velocity field has similar glide reflection symmetry as the intensity image.

We also apply variational IB to a von Kármán street arising due to air flow around Guadalupe Island, which was imaged by a National Oceanic and Atmospheric Administration (NOAA) satellite [75] (Fig. 5g). Although the video consists of only 62 frames and less than one full oscillation, the variational IB neural network learns latent variables which capture this oscillation and have the expected dependence on the input variables (Fig. 5h-i). As in the first experimental example, the gradients of the encoding variables show the glide symmetry of m_x due to the symmetry of the intensity pattern in Fig. 5g.

VI. LEARNING THE DYNAMICS OF LATENT VARIABLES

In Section I we showed how using an information-theoretic measure of predictability reduces the problem of model reduction to the problem of identifying relevant variables. In this setting, the dynamical rule governing the evolution of H_t is not explicitly learned, but is instead induced by the choice of encoding variable. We now show how variational IB can be incorporated into a complete model reduction pipeline by combining it with sparse equation learning methods such as SINDy [76, 77]. We consider a system described by an underdamped, driven, one-dimensional PDE known as the sine-Gordon equation [78] (Fig. 6a; see SI Section 9). For a particular choice of driving frequency and strength, this system undergoes quasiperiodic dynamics, and hence lives on a toroidal manifold.

To identify the reduced variables for the system, we must make a choice about the dimensionality of H_t . As discussed above, the mutual information between H_t and $X_{t+\Delta t}$ provides an indicator of “when to stop” (Fig. 6b). This metric plateaus already at $\dim H_t = 2$. The fractal dimension of the latent manifold, however, does not plateau until $\dim H_t = 3$ (Fig. 6c). Interestingly, when using the encoding to learn a deterministic equation for

the evolution of H_t using SINDy, the prediction error is only minimized at $\dim H_t = 4$ (Fig. 6d).

Each of these indicators of “when to stop” tells us something different about the system. The encoding learned for $\dim H_t = 2$ is shown in Figure 6e, with a portion of the trajectory highlighted. Although the full dynamics takes place on a torus, the encoding neural network is able to “cut” the torus to embed it in two dimensions. The dynamics of the latent variables are completely deterministic: at any point on the latent manifold it is known where the system will evolve next, so that the information about the future is complete. However, the dynamics will be highly irregular, featuring large jumps in latent space (Fig. 6f).

The “fraying” evident at the boundary of the torus embedded in two dimensions leads to a fractal dimension which is less than two, although this difference may disappear in the limit of infinite data. At three dimensions, the latent manifold is two dimensions everywhere (Fig. 6c). However, while the torus may be embedded in three dimensions, it does not admit simple, linear dynamics until $\dim H_t = 4$. In this dimension, SINDy is able to identify the correct linear equations of motion

$$\begin{aligned} \dot{h}_0 &= -\omega_1 h_1 & \dot{h}_2 &= \omega_2 h_3 \\ \dot{h}_1 &= \omega_1 h_0 & \dot{h}_3 &= -\omega_2 h_2 \end{aligned} \quad (11)$$

with $\omega_1 = 1.5$ and $\omega_2 = 8.7$ (Fig. 6g-h).

The above example illustrates the subtle difference between a model being the simplest in the sense of providing a minimal predictive set of variables, and the simplest in terms of the governing equations. The latter may benefit from the inclusion of informationally redundant variables. The IB objective, being purely information based, is indifferent to the geometric or analytic properties of the encodings. Such model properties, often desirable, as e.g. smoothness or linearity must be introduced as additional constraints.

VII. RELEVANT VARIABLE DISCOVERY IN CYANOBACTERIAL POPULATIONS

We now demonstrate how variational IB may be used as an aid for collective variable discovery in situations where physical intuition may not be a useful guide – collective behavior of biological organisms (Supplementary Movie 2). Here, we ask what the most predictive variables are for predicting the evolution of populations of cyanobacteria (*Synechococcus elongatus*). The dynamics of the colonies are driven by several factors: growth and division of individual bacteria, translational motion of groups of bacteria as they are pushed by their neighbors, as well as the circadian oscillations within each bacterium (Fig. 7a). These oscillations are controlled by three Kai proteins [79] and depend in particular on the ratios of the copy number of these proteins which can be tuned experimentally [80].

We were provided with videos of 10 cyanobacteria colonies that were grown under various conditions that im-

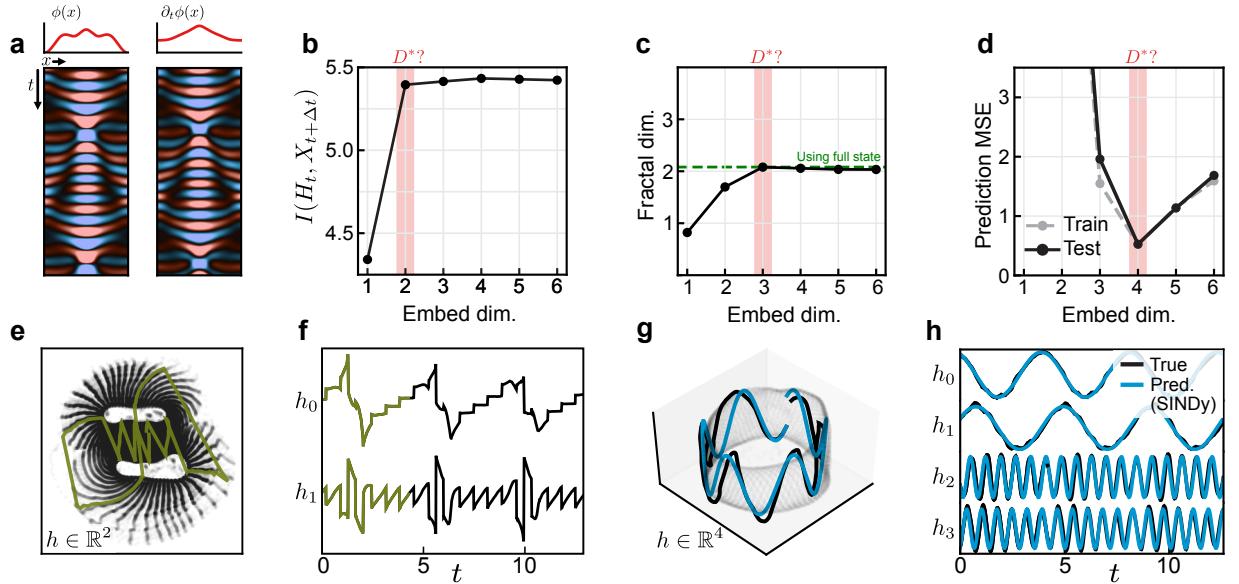


FIG. 6. IB as part of an equation learning pipeline. (a) Kymographs of the sine-Gordon system, discussed in the text. Left shows the field $\phi(x)$ while the right shows time-derivative of the field. Initial conditions are shown above as curves. (b) Information content contained in the encoding for varying dimensionality of the embedding space. While the information appears to plateau at $D^* = 2$, this does not necessarily mean a “proper” embedding dimension has been reached. (c) Estimated fractal dimension of the embedded dynamics for varying embedding dimension. The full fractal dimension of ≈ 2 is only attained with an encoding dimension $D^* = 3$. (d) Error of the predictions for the latent variable evolution produced by the dynamical system learned by SINDy. The error reaches a minimum at $D^* = 4$. Interestingly the error begins to rise for $d > 4$, likely due to the fact that the IB encoding begins to store other features of the full state variable which are irrelevant for the dynamics. (e) For $d = 2$, the encoding manages to project the full torus into two dimensions by cutting and unfolding it (left). This introduces discontinuous jumps in the evolution of h . (f) Time evolution of both components of h , with highlighted part corresponding to the trajectory highlighted in (e). (g) Trajectory for a four-dimensional latent variable, visualized by projected onto the first three principal components. True trajectory shown in black, trajectory predicted by SINDy shown in blue with the equations given in the main text. (h) Predicted trajectory for all the degrees of freedom of h .

pact their dynamics. However, as a test of our method, we were blinded to these conditions until we had performed our analysis. The videos are sequences of fluorescent images, taken once per hour, which show the clock state of each individual bacteria visualized with a fluorescent marker EYFP driven by the *kaiBC* promoter. Here, we focus on collective variables which are predictive of the state of the interior of the colony and not the growth in area of the colonies. We therefore crop the images to the interiors of each colony (SI Fig. S13). This allows us to isolate the motion of individual bacteria and fluorescence oscillations.

Our input to the variational IB neural network are these cropped images augmented with a time-lagged image of the same region (Fig. 7b-c). The purpose of this time lag is to make the dynamics Markovian: the intensity field oscillating, the observation at a single time point does not allow to determine whether the intensity is increasing or decreasing. In principle more time-lags may be necessary, and their number could be estimated from the behavior of the mutual information (see SI Section 3) or the transition entropy [63]. As these quantities are difficult to measure from images with limited computational resources, we keep only one lag: the state is given by pairs $X_t =$

$\{I(\mathbf{x}, t), I(\mathbf{x}, t + \tau)\}$, where τ is the lag duration. Here we take $\tau = 3$ hr and a prediction horizon $\Delta t = 8$ hr, but find that different Δt or τ does not change our results (SI Fig. S13).

With variational IB we compress the state X_t into a latent variable h of variable dimension. We train the neural network on the entire dataset of all 10 colonies simultaneously. The dynamics in latent space undergo clear oscillations, indicating that the relevant variables encode primarily the intensity fluctuations rather than, for example, the spatial locations of the bacteria. Notably, the trajectories are essentially two-dimensional, even when the encoding space is higher dimensional. This is reflected in the information retained about the future state, $I(X_{t+\Delta t}, H_t)$: increasing the dimension of the embedding space beyond two only marginally increases $I(X_{t+\Delta t}, H_t)$; this tells us “when to stop” (Fig. 7d). We independently verify this by using principal component analysis to characterize the geometry of embedded trajectories, and find that even in higher dimensions the trajectories occupy a two dimensional subspace (SI Fig. S10). In the following, we thus restrict our focus to $\dim H = 2$.

We noticed that there were notable differences in the radius of latent space oscillations from colony to colony,

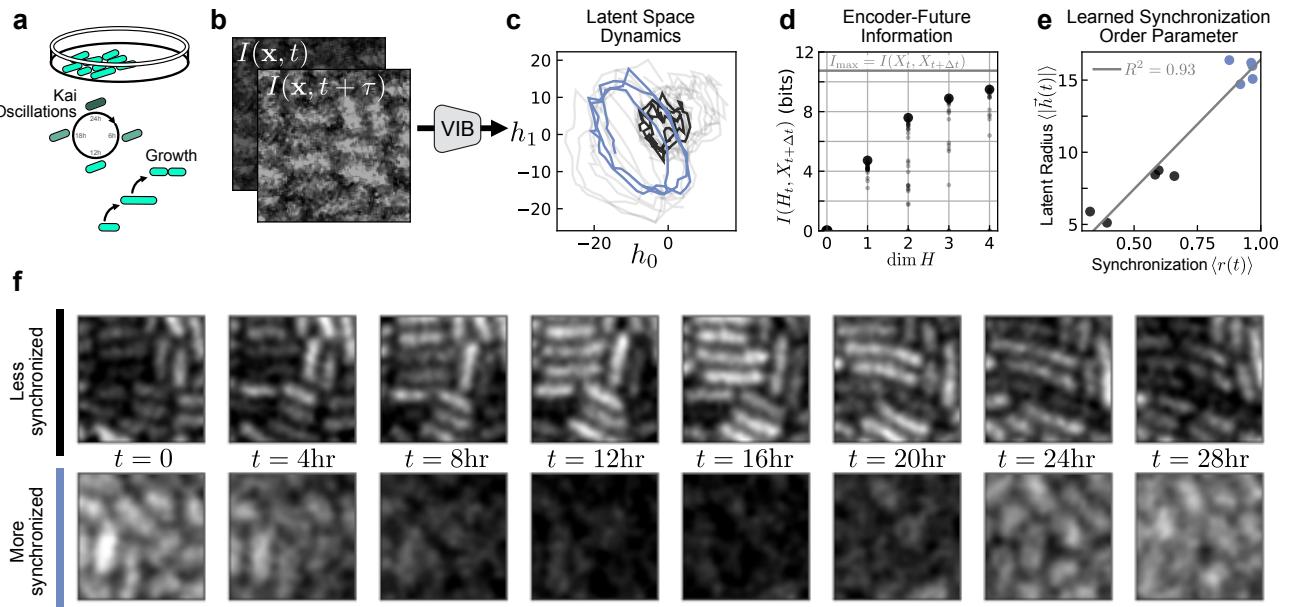


FIG. 7. Discovering slow collective variables in cyanobacteria populations (a) Fluorescent images of cyanobacteria colonies labeled with EYFP driven by the *kaiBC* promoter, allowing the visualization of Kai protein transcription. The colonies are imaged as they undergo cell growth and oscillations in Kai expression associated with the circadian rhythm. (b) The “state” used for variational IB is the time-lagged intensity field with lag time τ (see SI Fig. S13 for details). (c) Variational IB embeddings of time-lagged images into two dimensions (see Supplementary Movie 2). Every line corresponds to one colony’s evolution in time. Note the apparent oscillations of different radii. Three-dimensional embedding is shown in SI Fig. S15. One large-radius (blue) and one small-radius (black) trajectory are highlighted. (d) Mutual information between the future state and the encoding, given by $I(X_{t+\Delta t}, H_t)$, for varying dimension of the latent space. Each small point represents one training instance of the variational IB model, while the large point shows the maximum estimated value. $I_{\max} = I(X_t, X_{t+\Delta t})$ is the mutual information estimated for the full dynamics. (e) Mean radius versus synchronization parameter for each colony (see SI Section 8). VIB identifies clusters of cells characterized by high (blue) or low (black) synchronization. This clustering corresponds to differing theophylline concentrations across experiments. Within each experimental movie (each of which contains 2–3 colonies) the radii are mostly constant. (f) Sample time series of weakly (black) and highly (blue) synchronized colonies. We apply a slight Gaussian blur to better visualize the bacteria boundaries.

two of which are highlighted in (Fig. 7c). To understand this difference, we examined the original microscopy time series corresponding to both large and small latent radius (Fig. 7e-f) and found that while the large-radius sample showed clear, nearly uniform oscillations in intensity, the small-radius samples appeared much more heterogeneous.

To quantify this we consider each pixel to be an independent oscillator, akin to a spatial Kuramoto model [81–83], and compute a global synchronization order parameter $r(t)$ (see SI Section 7). For each colony we calculate the time-averaged synchronization $\langle r(t) \rangle_t$ and find that two clusters emerge corresponding to high and low synchronization (Fig. 7e). These clusters are precisely those representing trajectories of large and small latent radius, suggesting that variational IB learns to encode the synchronization of the colony in the latent variable radius. As a check, we perform IB on a simulated locally-coupled Kuramoto model as a system sharing many features of the experimental system. Here we also learn an encoding in which the latent radius corresponds to the synchronization order parameter (SI Fig. S11).

In the SI we compare the performance of variational IB

to several other model reduction methods and find that IB delivers more interpretable and well-behaved features. This is likely due to the fact that many standard methods for data-driven model reduction rely on assumptions about the dynamics which may not be appropriate in the case at hand, such as linearity. Even among deep learning methods free of such assumptions, such as time-lagged autoencoders, the variables learned by IB appear more interpretable. This increased interpretability is likely due to the compression term which effectively regularizes the latent space by encouraging the network to learn slow transfer operator eigenfunctions. While there are many specific variants of DMD [17, 84–87] or autoencoders for dynamics [28–30, 88] that may outperform variational IB in some cases, we find that in this real-world example it yields the smoothest and most interpretable latent variables without a tailored pre-processing (SI Fig. S17).

By using variational IB we could reduce a complex system with multiple dynamical components – cell growth, division, and gene expression fluctuations – into a low dimensional form that retains only the most relevant information for the future. In addition to the insight that the

dynamics are dominated by oscillations in two dimensions, the latent variables clearly distinguished trajectories into two groups that were not apparent *a priori*. We were provided this data as a “blind” test with no knowledge of the underlying system. After we performed our analysis, it was revealed to us that these bacterial colonies have been engineered to control the translational efficiency of the Kai proteins by varying theophylline concentration [80]. The synchronization order parameter discovered by variational IB corresponds to differing experimental concentrations of theophylline, which is in agreement with the findings in Ref. [80]. IB can thus serve as a way to connect experimental control parameters to effective changes in dynamics.

VIII. CONCLUSION

We have related information-theoretical properties of dynamical systems to the spectrum of the transfer operator. We illustrate our findings on several simple and analytically tractable systems, and turn them into a practical tool using variational IB, which learns an encoding variable with a neural network. The latent variables of these networks can be interpreted as transfer operator eigenfunctions even though the network was not explicitly constructed to learn these: it optimizes a purely information-theoretic objective that contains no knowledge of a transfer operator or dynamics. This allows one to harness the power of neural networks to learn physically-relevant latent variables and, combined with methods such as SINDY, even complete reduced models with their governing equations. Biological systems are an ideal setting for such methods: despite their apparent complexity, they can often be captured by low-dimensional descriptions which are difficult to identify by physical considerations alone [3, 62, 89, 90]. We have shown that variational IB is a potentially powerful tool for these cases, and can discover slow variables even directly from image data without significant preprocessing. In the SI we showcase further applications, including to chaotic systems, where they may yield computationally efficient methods to calculate *e.g.* fractal dimensions.

IX. METHODS

Information processing inequality

Here we show how to obtain (2) from (1). The variables we are considering form the Markov chain

$$X_{t+\Delta t} - X_t - h_{\text{enc}}(X_t) - \mathcal{S}^{\Delta t} h_{\text{enc}}(X_t) - h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t))$$

where $X - Y - Z$ means that Z is conditionally independent of X , given Y : $p(z, x|y) = p(z|y)p(x|y)$. As a consequence of the data processing inequality, for any

transformation $Z = g(Y)$ one has

$$I(X, g(Y)) \leq I(X, Y).$$

where equality is obtained if g is a bijection [58]. Consequently, we have

$$\begin{aligned} I(h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) &\leq I(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t), X_{t+\Delta t}) \\ &\leq I(h_{\text{enc}}(X_t), X_{t+\Delta t}), \end{aligned}$$

where the inequalities are maximized if h_{dec}^{-1} and $\mathcal{S}^{\Delta t}$ are both bijective functions. Notably, $\mathcal{S}^{\Delta t} = \text{identity}$ is a solution. However, this should not be taken to mean that H_t has no dynamics, as it will still have dynamics induced by the evolution of X_t .

The information bottleneck

The information bottleneck [38] is an example of a rate-distortion problem which seeks to find an optimal compression which minimizes some distortion measure with the original signal [58]. Concretely, we call X the source signal, and let H denote the compressed signal. In IB, rather than using an *a priori* unknown distortion function, one seeks to ensure that the compression retains information about an additional relevance variable Y . As noted in the main text, the IB optimization objective is given by the Lagrangian

$$\mathcal{L}_{\text{IB}}[p(h|x)] = I(X, H) - \beta I(Y, H), \quad (12)$$

where in our case the source signal X is the state of the system X_t at time t , and the relevance variable is the state of the system $X_{t+\Delta t}$ at a future time $t + \Delta t$. The encoder which optimizes this objective can be solved for exactly and is given by [38]

$$p(h|x) = \frac{p(h)}{N(x)} \exp \left[-\beta D_{KL}(p(y|x) \| p(y|h)) \right]. \quad (13)$$

Encoder in terms of transfer operator eigenfunctions

To connect the optimal encoder to the transfer operator, we first rewrite (13) in terms of the transition probabilities,

$$p(h_t|x_t) = \frac{\tilde{p}(h_t)}{\tilde{N}(x_t)} \exp \left[\beta \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|h_t) \right]. \quad (14)$$

where we have absorbed terms in the exponent which depend only on h_t or x_t into the normalization factors. Into the above equation, we replace the transition probability with the spectral decomposition

$$p(x_{t+\Delta t}|x_t) = \sum_n e^{\lambda_n \Delta t} \rho_n(x_{t+\Delta t}) \phi_n(x_t). \quad (15)$$

From this, the (7) of the main text immediately follows, where

$$f_n(h_t) = \int dx_{t+\Delta t} \rho_n(x_{t+\Delta t}) \log p(x_{t+\Delta t}|h_t) \quad (16)$$

which may be interpreted as a sort of cross entropy (ρ_n is generally not a probability distribution) between each right eigenfunction and the decoding of h_t into the future state $x_{t+\Delta t}$.

To study the behavior of the encoder in the limit of high compression, we consider a transfer operator U with infinitesimal generator \mathcal{L}_U . For \mathcal{L}_U with a discrete spectrum with eigenvalues satisfying $0 = \lambda_0 > \lambda_1 > \lambda_2 \gg \lambda_3 \dots$ and for β just above the first IB transition β_1 , we show that the optimal encoder is given approximately by

$$p_\beta^*(h|x) = \frac{1}{\mathcal{N}(x)} p_\beta^*(h) \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x) f_1(h) \right) \quad (17)$$

with corrections due to the second eigenfunction given by $f_2(h) \approx f_1(h) e^{-\Gamma \Delta t} + \mathcal{O}(e^{-2\Gamma \Delta t})$ where $\Gamma = \lambda_1 - \lambda_2 > 0$ denotes the spectral gap. To see this, we compute the Hessian of the IB Lagrangian

$$H_{(\mu,n),(\nu,m)} = \frac{\partial^2}{\partial f_n(h_\mu) \partial f_m(h_\nu)} \mathcal{L}_{\text{IB}}.$$

Here we assume a finite alphabet of size N_H , i.e. h_μ with $\mu \in \{1, \dots, N_H\}$. At the uniform encoder, i.e. $f_n(h) = 0$, the Hessian decomposes into a tensor product

$$H_{(\mu,n),(\nu,m)} = A_{nm}^\beta \otimes G_{\mu\nu}$$

where A^β depends only on the indices of the coefficients and G captures the dependence on h_ν . The only part which depends on β is A^β .

We are concerned with the sign of the eigenvalues of H . A negative eigenvalue indicates that \mathcal{L}_{IB} is unstable to a perturbation in f , which means the loss can be lowered by changing f away from the trivial encoder at $f_n = 0$. Because the eigenvalues of a tensor product of matrices are products of the eigenvalues of the component matrices, the eigenvalues of H change sign when those of A^β do. A^β and its spectrum can be computed, which we do in the SI. The result of this calculation is that the first eigenvalue to become negative is associated with the eigenvector $\mathbf{v} = (1, 0, 0, \dots)$. This computation is exact for equilibrium systems, which are those in which the steady-state flux vanishes, but in nonequilibrium systems there may generally be a correction proportional to the flux. For details, please refer to the SI.

Variational IB compared to other dimensionality reduction techniques

Variational IB (VIB) is by no means the only numerical method for performing data-driven model reduction. Here

we provide a brief overview of the benefits and shortcomings of VIB with respect to other methods; an extended discussion can be found in the SI.

One class of methods is based on linear projections, such as principal component analysis (PCA), dynamic mode decomposition (DMD) [15, 16], or (time-lagged) independent component analysis (TICA) [24] (which is equivalent to DMD [91]). These methods can be extended to take into account non-linearity by introducing a library of non-linear terms on which one then applies the above methods, such as in kernel PCA [92] or extended DMD (eDMD) [17]. These methods have the advantages, relative to VIB, that their optimization (even for the extended algorithms) relies only on linear projections which are fast and interpretable. However, the success of these methods depends on the choice of an appropriate library of functions so that the projection onto this space is closed under the dynamics. Choosing an appropriate library is not always possible [93, 94].

A second category of non-linear dimensionality reduction techniques are graph-based or similarity-based methods, which typically assume that the data is distributed on a low-dimensional manifold embedded in a higher-dimensional space [95, 96]. One prominent example is diffusion maps [21], which starts from a set of data snapshots and, assuming the system evolves diffusively on short times, constructs an approximate transition matrix from which one can compute eigenfunctions to parameterize the data manifold. The assumption of diffusive dynamics can be violated when data is not sampled sufficiently frequently. This likely explains our finding that VIB produced more well-behaved low-dimensional embeddings on the cyanobacteria dataset (SI Fig. S12). VIB has the additional advantage, relative to this and similar methods, that it explicitly takes dynamics into account without the strong assumptions required by diffusion maps.

Finally, deep neural networks can be used for model reduction through encoder-decoder architectures that attempt to reconstruct the data from a low-dimensional latent space; VIB falls into this class of methods. Some standard neural network architectures from this class include autoencoders (AEs) and variational autoencoders (VAEs). For dynamical systems in particular, extensions to these methods have been proposed which impose constraints on the latent dynamics, such as linearity [28–30, 88]. Autoencoders often produce poorly-behaved latent spaces that distribute the latent variables on a narrow manifold with sharp features, see for example [88]. By regularizing the latent embedding to encourage smoothness, variational autoencoders can remedy some of these issues. We note that the VIB loss is very similar to a VAE loss with the contrastive InfoNCE loss replacing the reconstruction loss, so we expect that for many problems these should perform similarly. Other dynamically-constrained architectures such as in [28–30] work well for deterministic systems but it is unclear what effect stochasticity has on their performance. In our examples we have seen that VIB works well on noisy data.

In general when investigating a new system it is good practice to start by attempting to perform dimensionality reduction with linear methods such as PCA or DMD because they are fast, straightforward to implement, and easy to interpret. In situations where linear techniques are not sufficient, VIB may be preferable to other methods because it is guaranteed to find dynamically relevant variables (in contrast to diffusion maps, t-SNE, AEs, VAEs, etc.) and it does not require that one performs the carefully tailored preprocessing steps that are required by eDMD or kernel PCA, or other variants of DMD [84–87]. Additionally, it works well even when the dynamics are highly stochastic as shown in the cyanobacteria dataset.

ACKNOWLEDGEMENTS

The authors would like to thank Z. Ringel, M. Han and D.E. Gökmen for helpful discussions. M.S.S. acknowledges support from a MRSEC-funded Graduate Research Fellowship (DMR-2011854). M.K.-J. gratefully acknowledges financial support from the European Union’s Horizon 2020 programme under Marie Skłodowska-Curie Grant Agreement No. 896004 (COMPLEX ML). D.S.S. acknowledges support from a MRSEC-funded Kadanoff–Rice fellowship and the University of Chicago Materials Research Science and Engineering Center (DMR-2011854). V.V. acknowledges support from the Simons Foundation, the Complex Dynamics and Systems Program of the Army Research Office under grant W911NF19-1-0268, the National Science Foundation under grant DMR-2118415 and the University of Chicago Materials Research Science and Engineering Center, which is funded by the National Science Foundation under award no. DMR-2011854. All authors acknowledge support from the UChicago Research Computing Center which provided the computing resources for this work.

-
- [1] S. J. Cook, T. A. Jarrell, C. A. Brittin, Y. Wang, A. E. Bloniarz, M. A. Yakovlev, K. C. Q. Nguyen, L. T.-H. Tang, E. A. Bayer, J. S. Duerr, H. E. Bülow, O. Hobert, D. H. Hall, and S. W. Emmons, *Nature* **571**, 63 (2019).
 - [2] S. Dorkenwald, A. Matsliah, A. R. Sterling, P. Schlegel, S.-c. Yu, C. E. McKellar, A. Lin, M. Costa, K. Eichler, Y. Yin, W. Silversmith, C. Schneider-Mizell, C. S. Jordan, D. Brittain, A. Halageri, K. Kuehner, O. Ogedengbe, R. Morey, J. Gager, K. Kruk, E. Perlman, R. Yang, D. Deutsch, D. Bland, M. Sorek, R. Lu, T. Macrina, K. Lee, J. A. Bae, S. Mu, B. Nehoran, E. Mitchell, S. Popovych, J. Wu, Z. Jia, M. A. Castro, N. Kemnitz, D. Ih, A. S. Bates, N. Eckstein, J. Funke, F. Collman, D. D. Bock, G. S. X. E. Jefferis, H. S. Seung, M. Murthy, and T. F. Consortium, *Nature* **634**, 124 (2024).
 - [3] G. J. Stephens, B. Johnson-Kerner, W. Bialek, and W. S. Ryu, *PLOS Computational Biology* **4**, 1 (2008).
 - [4] T. Ahamed, A. C. Costa, and G. J. Stephens, *Nature Physics* **17**, 275 (2021).
 - [5] G. J. Berman, D. M. Choi, W. Bialek, and J. W. Shaevitz, *Journal of The Royal Society Interface* **11**, 20140672 (2014).
 - [6] Y. Kuramoto, *Chemical Oscillations, Waves, And Turbulence* (Springer, 1984).
 - [7] H. Haken, *Synergetics: An Introduction*, Springer Series in Synergetics (Springer Berlin Heidelberg, 1983).
 - [8] G. Pavliotis and A. Stuart, *Multiscale Methods: Averaging and Homogenization*, Texts in Applied Mathematics (Springer New York, 2010).
 - [9] D. Givon, R. Kupferman, and A. Stuart, *Nonlinearity* **17**, R55–R127 (2004).
 - [10] Y. Kuramoto and H. Nakao, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **377**, 20190041 (2019).
 - [11] T. Kunihiro, Y. Kikuchi, and K. Tsumura, *Geometrical Formulation of Renormalization-Group Method as an Asymptotic Analysis: With Applications to Derivation of Causal Fluid Dynamics*, Fundamental Theories of Physics (Springer Nature Singapore, 2022).
 - [12] Y. Kuramoto, *Progress of Theoretical Physics Supplement* **99**, 244–262 (1989).
 - [13] H. Mori and Y. Kuramoto, *Dissipative Structures and Chaos* , 93–117 (1998).
 - [14] L.-Y. Chen, N. Goldenfeld, and Y. Oono, *Physical Review E* **54**, 376–394 (1996).
 - [15] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Journal of Fluid Mechanics* **641**, 115–127 (2009).
 - [16] P. J. Schmid, *Journal of Fluid Mechanics* **656**, 5–28 (2010).
 - [17] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *Journal of Nonlinear Science* **25**, 1307 (2015).
 - [18] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, Computational Science and Engineering (Society for Industrial and Applied Mathematics, 2017).
 - [19] R. Geelen, L. Balzano, S. Wright, and K. Willcox, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **34**, 033122 (2024).
 - [20] R. Geelen, S. Wright, and K. Willcox, *Computer Methods in Applied Mechanics and Engineering* **403**, 115717 (2023).
 - [21] R. R. Coifman and S. Lafon, *Applied and Computational Harmonic Analysis* **21**, 5 (2006), special Issue: Diffusion Maps and Wavelets.
 - [22] M. Cenedese, J. Axås, B. Bäuerlein, K. Avila, and G. Haller, *Nature Communications* **13**, 10.1038/s41467-022-28518-y (2022).

- [23] S. Jain and G. Haller, *Nonlinear Dynamics* **107**, 1417–1450 (2021).
- [24] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis* (John Wiley & Sons, Ltd, 2001).
- [25] S. E. Otto and C. W. Rowley, *Annual Review of Control, Robotics, and Autonomous Systems* **4**, 59–87 (2021).
- [26] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, *Annual Review of Fluid Mechanics* **52**, 477 (2020), <https://doi.org/10.1146/annurev-fluid-010719-060214>.
- [27] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *SIAM Review* **64**, 229 (2022).
- [28] N. Takeishi, Y. Kawahara, and T. Yairi, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).
- [29] B. Lusch, J. N. Kutz, and S. L. Brunton, *Nature Communications* **9**, 4950 (2018).
- [30] J. Bakarji, K. Champion, J. Nathan Kutz, and S. L. Brunton, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **479**, 20230422 (2023), <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2023.0422>.
- [31] P. R. Vlachas, G. Arampatzis, C. Uhler, and P. Koumoutsakos, *Nature Machine Intelligence* **4**, 359 (2022).
- [32] T. Wu, T. Maruyama, and J. Leskovec, Learning to accelerate partial differential equations via latent global evolution (2022), [arXiv:2206.07681 \[cs.LG\]](https://arxiv.org/abs/2206.07681).
- [33] P. Cvitanović and D. Lippolis, *AIP Conference Proceedings* **10.1063/1.4745574** (2012).
- [34] P. Gaspard and X.-J. Wang, *Physics Reports* **235**, 291 (1993).
- [35] N. Jayant, J. Johnston, and R. Safranek, *Proceedings of the IEEE* **81**, 1385 (1993).
- [36] F. Creutzig and H. Sprikeler, *Neural Computation* **20**, 1026 (2008).
- [37] F. Creutzig, A. Globerson, and N. Tishby, *Phys. Rev. E* **79**, 041925 (2009).
- [38] N. Tishby, F. C. Pereira, and W. Bialek, The information bottleneck method (2000), [arXiv:physics/0004057 \[physics.data-an\]](https://arxiv.org/abs/physics/0004057).
- [39] W. Bialek, *Biophysics: searching for principles* (Princeton University Press, 2012).
- [40] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise* (Springer, 1998) p. 492.
- [41] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, in *International Conference on Learning Representations* (2017).
- [42] N. Slonim and N. Tishby, in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR00 (ACM, 2000) pp. 208–215.
- [43] D. Strouse and D. J. Schwab, *The deterministic information bottleneck* (2016), [arXiv:1604.00268 \[q-bio.NC\]](https://arxiv.org/abs/1604.00268).
- [44] A. Dimitrov and J. Miller, *Network: Computation in Neural Systems* **12**, 441 (2001).
- [45] V. Sachdeva, T. Mora, A. M. Walczak, and S. E. Palmer, *PLOS Computational Biology* **17**, 1 (2021).
- [46] S. E. Palmer, O. Marre, M. J. Berry, and W. Bialek, *Proceedings of the National Academy of Sciences* **112**, 6908 (2015).
- [47] E. Abdelaleem, I. Nemenman, and K. M. Martini, Deep variational multivariate information bottleneck – a framework for variational losses (2023), [arXiv:2310.03311 \[cs.LG\]](https://arxiv.org/abs/2310.03311).
- [48] E. Abdelaleem, I. Nemenman, and K. M. Martini, *Journal of Machine Learning Research* **26**, 1 (2025).
- [49] P. Gaspard, G. Nicolis, A. Provata, and S. Tasaki, *Phys. Rev. E* **51**, 74 (1995).
- [50] G. Froyland, *Physica D: Nonlinear Phenomena* **250**, 1 (2013).
- [51] A. Parker, T. Gedeon, and A. Dimitrov, in *Advances in Neural Information Processing Systems*, Vol. 15, edited by S. Becker, S. Thrun, and K. Obermayer (MIT Press, 2002).
- [52] T. Wu, I. Fischer, I. L. Chuang, and M. Tegmark, in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, Proceedings of Machine Learning Research, Vol. 115, edited by R. P. Adams and V. Gogate (PMLR, 2020) pp. 1050–1060.
- [53] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, *Journal of Machine Learning Research* **6**, 165 (2005).
- [54] V. Ngampruetikorn and D. J. Schwab, in *Advances in Neural Information Processing Systems*, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (2021).
- [55] L. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators* (Princeton University Press).
- [56] C. M. Bishop, *Pattern recognition and machine learning* (Springer, 2006).
- [57] J.-W. Chu and G. A. Voth, *Biophysical Journal* **93**, 3860 (2007).
- [58] T. Cover and J. Thomas, *Elements of Information Theory* (Wiley, 2012).
- [59] S. Ulam, *Problems in Modern Mathematics* (Science Editions, 1960).
- [60] T.-Y. Li, *Journal of Approximation Theory* **17**, 177 (1976).
- [61] J. Crutchfield and N. Packard, *Physica D: Nonlinear Phenomena* **7**, 201–223 (1983).
- [62] G. J. Berman, W. Bialek, and J. W. Shaevitz, *Proceedings of the National Academy of Sciences* **113**, 11943 (2016), <https://www.pnas.org/doi/pdf/10.1073/pnas.1607601113>.
- [63] A. C. Costa, T. Ahamed, D. Jordan, and G. J. Stephens, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **33**, 023136 (2023), https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/5.0129398/16745944/023136_1_online.pdf.
- [64] K. A. Murphy and D. S. Bassett, Optimized measurements of chaotic dynamical systems via the information bottleneck (2023), [arXiv:2311.04896 \[cs.LG\]](https://arxiv.org/abs/2311.04896).
- [65] G. Froyland, *Physica D: Nonlinear Phenomena* **200**, 205 (2005).
- [66] D. E. Gökmən, Z. Ringel, S. D. Huber, and M. Koch-Janusz, *Phys. Rev. Lett.* **127**, 240603 (2021).
- [67] D. E. Gökmən, Z. Ringel, S. D. Huber, and M. Koch-Janusz, *Phys. Rev. E* **104**, 064106 (2021).
- [68] D. E. Gökmən, S. Biswas, S. D. Huber, Z. Ringel, F. Flicker, and M. Koch-Janusz, Compression theory for inhomogeneous systems (2023), [arXiv:2301.11934 \[cond-mat.stat-mech\]](https://arxiv.org/abs/2301.11934).
- [69] B. B. Machta, R. Chachra, M. K. Transtrum, and J. P. Sethna, *Science* **342**, 604 (2013), <https://www.science.org/doi/pdf/10.1126/science.1238723>.
- [70] M. K. Transtrum, B. B. Machta, K. S. Brown, B. C. Daniels, C. R. Myers, and J. P. Sethna, *The Journal of Chemical Physics* **143**, 010901 (2015), <https://pubs.aip.org/aip/jcp/article-abstract/143/1/010901/27333>.

- pdf/doi/10.1063/1.4923066/15497236/010901_1_online.pdf.
- [71] W. Bialek, I. Nemenman, and N. Tishby, *Neural Computation* **13**, 2409 (2001).
 - [72] M. Tchernookov and I. Nemenman, *Journal of Statistical Physics* **153**, 442 (2013).
 - [73] J. Albright, Flow Visualization in a Water Channel (2017), @jacobalbright3585.
 - [74] J. Tencer and K. Potter, A tailored convolutional neural network for nonlinear manifold learning of computational physics data using unstructured spatial discretizations (2020), arXiv:2006.06154 [physics.comp-ph].
 - [75] @NOAASatellites, Earth from Orbit: von Kármán Vortices, <https://www.youtube.com/watch?v=SawKLWT1bDA> (2021), [Accessed 03-Sept-2023].
 - [76] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Proceedings of the National Academy of Sciences* **113**, 3932 (2016), <https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113>.
 - [77] A. A. Kaptanoglu, B. M. de Silva, U. Fasel, K. Kaheman, A. J. Goldschmidt, J. Callaham, C. B. Delahunt, Z. G. Nicolaou, K. Champion, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *Journal of Open Source Software* **7**, 3994 (2022).
 - [78] A. Bishop, M. Forest, D. McLaughlin, and E. Overman, *Physica D: Nonlinear Phenomena* **23**, 293 (1986).
 - [79] M. Ishiura, S. Kutsuna, S. Aoki, H. Iwasaki, C. R. Andersson, A. Tanabe, S. S. Golden, C. H. Johnson, and T. Kondo, *Science* **281**, 1519 (1998).
 - [80] J. Chew, E. Leypunskiy, J. Lin, A. Murugan, and M. J. Rust, *Nature Communications* **9**, 3004 (2018).
 - [81] L. Ren and G. B. Ermentrout, *SIAM Journal on Mathematical Analysis* **29**, 208 (1998).
 - [82] H. Hong, H. Park, and M. Y. Choi, *Physical Review E* **72**, 036217 (2005).
 - [83] C. R. Laing, *Physica D: Nonlinear Phenomena* **240**, 1960 (2011).
 - [84] K. K. Chen, J. H. Tu, and C. W. Rowley, *Journal of Nonlinear Science* **22**, 887 (2012).
 - [85] J. N. Kutz, X. Fu, and S. L. Brunton, Multi-resolution dynamic mode decomposition (2015), arXiv:1506.00564 [math.DS].
 - [86] A. Wynn, D. S. Pearson, B. Ganapathisubramani, and P. J. Goulart, *Journal of Fluid Mechanics* **733**, 473–503 (2013).
 - [87] M. J. Colbrook, *SIAM Journal on Numerical Analysis* **61**, 1585 (2023), <https://doi.org/10.1137/22M1521407>.
 - [88] C. Wehmeyer and F. Noé, *The Journal of Chemical Physics* **148**, 241703 (2018), https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5011399/16653374/241703_1_online.pdf.
 - [89] R. Seyboldt, J. Lavoie, A. Henry, J. Vanaret, M. D. Petkova, T. Gregor, and P. François, *Proceedings of the National Academy of Sciences* **119**, e2113651119 (2022), <https://www.pnas.org/doi/pdf/10.1073/pnas.2113651119>.
 - [90] S. R. Achar, F. X. P. Bourassa, T. J. Rademaker, A. Lee, T. Kondo, E. Salazar-Cavazos, J. S. Davies, N. Taylor, P. François, and G. Altan-Bonnet, *Science* **376**, 880 (2022), <https://www.science.org/doi/pdf/10.1126/science.abl5311>.
 - [91] S. Klus, F. Nüske, P. Kolta, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, *Journal of Nonlinear Science* **28**, 985 (2018).
 - [92] B. Schölkopf, A. Smola, and K.-R. Müller, *Neural Computation* **10**, 1299 (1998), <https://direct.mit.edu/neco/article-pdf/10/5/1299/813905/089976698300017467.pdf>.
 - [93] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Machine Learning: Science and Technology* **2**, 035023 (2021).
 - [94] J. Page and R. R. Kerswell, *Journal of Fluid Mechanics* **879**, 1–27 (2019).
 - [95] L. van der Maaten and G. Hinton, *Journal of Machine Learning Research* **9**, 2579 (2008).
 - [96] L. McInnes, J. Healy, and J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction (2020), arXiv:1802.03426 [stat.ML].
 - [97] N. Slonim, *The Information Bottleneck: Theory and Applications*, Ph.D. thesis, Hebrew University of Jerusalem (2002), retrieved from https://www.cs.huji.ac.il/labs/learning/Theeses/Slonim_PhD.pdf.
 - [98] R. Jiang, P. Y. Lu, E. Orlova, and R. Willett, *Advances in Neural Information Processing Systems* **36** (2024).
 - [99] S. B., *Network: Computation in Neural Systems* **7**, 7 (1996).
 - [100] H. HOTELLING, *Biometrika* **28**, 321 (1936), <https://academic.oup.com/biomet/article-pdf/28/3-4/321/586830/28-3-4-321.pdf>.
 - [101] K. Deng, P. G. Mehta, and S. P. Meyn, *IEEE Transactions on Automatic Control* **56**, 2793 (2011).
 - [102] N. Slonim, N. Friedman, and N. Tishby, *Neural Computation* **18**, 1739 (2006).
 - [103] Y. Polyanskiy and Y. Wu, *Information Theory: From Coding to Learning* (Cambridge University Press, 2025).
 - [104] F. Takens, in *Dynamical Systems and Turbulence, Warwick 1980*, edited by D. Rand and L.-S. Young (Springer Berlin Heidelberg, Berlin, Heidelberg, 1981) pp. 366–381.
 - [105] The notation $A - B - C$ means that C is conditionally independent of A , given B , or equivalently $p(c|b,a) = p(c|a)$.
 - [106] E. N. Lorenz, *Journal of the Atmospheric Sciences* **20**, 130–141 (1963).
 - [107] P. Ronceray, *Learning dynamical models from stochastic trajectories* (2024), arXiv:2406.02363 [cond-mat.soft].
 - [108] É. Roldán, J. Barral, P. Martín, J. M. Parrondo, and F. Jülicher (2020), 16 pages (including Supplement), 12 figures.
 - [109] A. van den Oord, Y. Li, and O. Vinyals, Representation learning with contrastive predictive coding (2019), arXiv:1807.03748 [cs.LG].
 - [110] A. Gordon, A. Banerjee, M. Koch-Janusz, and Z. Ringel, *Phys. Rev. Lett.* **126**, 240601 (2021).
 - [111] A. van den Oord, Y. Li, and O. Vinyals, Representation learning with contrastive predictive coding (2019), arXiv:1807.03748 [cs.LG].
 - [112] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, in *International Conference on Learning Representations* (2017).
 - [113] J. Mihaiel, Z. Monfared, and D. Durstewitz, in *Advances in Neural Information Processing Systems*, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., 2022) pp. 11297–11312.
 - [114] G. Koppe, H. Toutounji, P. Kirsch, S. Lis, and D. Durstewitz, *PLOS Computational Biology* **15**, 1 (2019).
 - [115] W. Gilpin, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 204–216.
 - [116] H. Eivazi, S. Le Clainche, S. Hoyas, and R. Vinuesa, *Expert Systems with Applications* **202**, 117038 (2022).

- [117] M. Rolinek, D. Zietlow, and G. Martius, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [118] S. H. Strogatz, *Nonlinear Dynamics and Chaos* (CRC Press, 2018) p. 532.
- [119] M. Dellnitz, G. Froyland, and O. Junge, in *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems*, edited by B. Fiedler (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001) pp. 145–174.
- [120] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *Journal of Computational Dynamics* **1**, 391 (2014).
- [121] A. T. Azar and S. Vaidyanathan, *Chaos modeling and control systems design*, Vol. 581 (Springer, 2015).
- [122] Y. Kuramoto, *Progress of Theoretical Physics Supplement* **64**, 346 (1978), <https://academic.oup.com/ptps/article-pdf/doi/10.1143/PTPS.64.346/5293041/64-346.pdf>.
- [123] G. Sivashinsky, *Acta Astronautica* **4**, 1177 (1977).
- [124] J. Theiler, *JOSA a* **7**, 1055 (1990).
- [125] G. Datseris, I. Kottlitz, A. P. Braun, and U. Parlitz, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **33** (2023).
- [126] P. Grassberger, *Journal of Statistical Physics* **26**, 173 (1981).
- [127] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, in *Database Theory — ICDT 2001*, edited by J. Van den Bussche and V. Vianu (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001) pp. 420–434.
- [128] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, in *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings* **7** (Springer, 1999) pp. 217–235.
- [129] G. Nolte, A. Ziehe, and K.-R. Müller, *Phys. Rev. E* **64**, 016112 (2001).
- [130] A. Lasota and M. C. Mackey, *Chaos, fractals, and noise: stochastic aspects of dynamics*, Vol. 97 (Springer Science & Business Media, 2013).
- [131] M. J. Colbrook and A. Townsend, *Communications on Pure and Applied Mathematics* **77**, 221 (2024).
- [132] G. Froyland, *Communications in Mathematical Physics* **189**, 237 (1997).
- [133] J. Mikhalev, Z. Monfared, and D. Durstewitz, *Advances in Neural Information Processing Systems* **35**, 11297 (2022).
- [134] J. Park, N. Yang, and N. Chandramoorthy, in *Advances in Neural Information Processing Systems*, Vol. 37, edited by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Curran Associates, Inc., 2024) pp. 43975–44008.
- [135] Y. Schiff, Z. Y. Wan, J. B. Parker, S. Hoyer, V. Kuleshov, F. Sha, and L. Zepeda-Núñez, arXiv preprint arXiv:2402.04467 (2024).
- [136] Y. Lan and P. Cvitanović, *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* **78**, 026208 (2008).
- [137] P. Cvitanović, *Physical Review Letters* **61**, 2729 (1988).
- [138] J.-P. Eckmann and D. Ruelle, *Reviews of modern physics* **57**, 617 (1985).
- [139] I. Procaccia, *Physica Scripta* **1985**, 40 (1985).
- [140] J. L. Kaplan and J. A. Yorke, in *Functional Differential Equations and Approximation of Fixed Points*, edited by H.-O. Peitgen and H.-O. Walther (Springer Berlin Heidelberg, Berlin, Heidelberg, 1979) pp. 204–227.
- [141] H.-l. Yang and G. Radons, *Phys. Rev. Lett.* **108**, 154101 (2012).
- [142] J. Tencer, *Cylinder in Crossflow* (2020).
- [143] G. Berkooz, P. Holmes, and J. L. Lumley, *Annual Review of Fluid Mechanics* **25**, 539 (1993).
- [144] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, *Nature Communications* **8**, 19 (2017).
- [145] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler, *Multiscale Modeling & Simulation* **7**, 842 (2008), <https://doi.org/10.1137/070696325>.
- [146] P. Y. Lu, S. Kim, and M. Soljačić, *Phys. Rev. X* **10**, 031056 (2020).
- [147] J. Colen, M. Han, R. Zhang, S. A. Redford, L. M. Lemma, L. Morgan, P. V. Ruijgrok, R. Adkins, Z. Bryant, Z. Dogic, M. L. Gardel, J. J. de Pablo, and V. Vitelli, *Proceedings of the National Academy of Sciences* **118**, e2016708118 (2021), <https://www.pnas.org/doi/pdf/10.1073/pnas.2016708118>.
- [148] D. P. Kingma and M. Welling, Auto-encoding variational bayes (2022), [arXiv:1312.6114 \[stat.ML\]](https://arxiv.org/abs/1312.6114).
- [149] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, Understanding disentangling in β -vae (2018), [arXiv:1804.03599 \[stat.ML\]](https://arxiv.org/abs/1804.03599).

Supplementary Information

I. BACKGROUND MATERIAL

Mutual information and entropy

Let X be a random variable which takes values x that are observed with probability $p(x)$. The entropy of this distribution measures the predictability of the outcome of a measurement of X and is given by [58]

$$\mathcal{H}(X) = - \int dx p(x) \log p(x).$$

High entropy means high uncertainty. The conditional entropy $\mathcal{H}(X|Y)$ between two variables is the average entropy of $p(x|y)$, where the average is taken over $p(y)$:

$$\mathcal{H}(X|Y) = - \int dy p(y) \int dx p(x|y) \log p(x|y).$$

The mutual information between X and Y corresponds to the (average) reduction of uncertainty about X , given knowledge of Y :

$$\begin{aligned} I(X, Y) &= \mathcal{H}(X) - \mathcal{H}(X|Y) \\ &= D_{\text{KL}}(p(x, y) \| p(x)p(y)) \\ &= \int dx dy p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \end{aligned} \quad (\text{S1})$$

The three formulations of the mutual information above are equivalent and useful in different settings. For example, formulation as a Kullback-Leibler divergence immediately tells us that mutual information is non-negative.

Information bottleneck

The information bottleneck was originally formulated in Ref. [38] as a rate distortion problem. Rate distortion theory describes how to maximally compress a signal (i.e. minimize its communication rate) such that it remains minimally distorted, which, however, presupposes the knowledge of a distortion function specifying which features need to be preserved[58]. In Ref. [38], Tishby et al. introduce a variant of the problem where the a priori unknown distortion function is not used, but rather one seeks to ensure that the compression retains information about an auxiliary variable Y correlated with the signal. As the correlations with Y define implicitly the relevant features of the signal to be preserved, Y is called the relevance variable. Concretely, we call X the source signal, and H denotes the compressed signal. The random variables form a Markov chain $H \leftrightarrow X \leftrightarrow Y$, meaning that H and Y are conditionally independent given X :

$$\begin{aligned} p(y|h, x) &= p(y|x)p(h, x) \\ p(h|x, y) &= p(h|x)p(y, x) \end{aligned}$$

As noted in the main text, the IB optimization objective is given by the Lagrangian

$$\mathcal{L}_{\text{IB}}[p(h|x)] = I(X, H) - \beta I(Y, H). \quad (\text{S2})$$

To enforce normalization of the $p(h|x)$ one introduces a Lagrange multiplier $\lambda(x)$ so that the full optimization function is

$$\mathcal{L}_{\text{IB}}[p(h|x)] = \sum_{x,h} p(h|x)p(x) \log \frac{p(h|x)}{p(h)} - \beta \sum_{y,h} p(y,h) \log \frac{p(y,h)}{p(h)p(y)} + \sum_x \lambda(x) \left(1 - \sum_h p(h|x) \right). \quad (\text{S3})$$

The encoder which optimizes this objective can be solved for exactly. Using the following functional derivatives,

$$\begin{aligned} \frac{\delta}{\delta p(h|x)} p(h') &= \delta(h - h')p(x) \\ \frac{\delta}{\delta p(h|x)} p(h', y') &= \delta(h - h')p(x, y'), \end{aligned}$$

one can compute the derivative of the Lagrangian. By evaluating the derivative and setting to zero, one finds

$$\log p(h|x) = \log p(h) - \beta \sum_y p(y|x) \log \frac{1}{p(y|h)} - \lambda(x)$$

which can be rearranged to give the optimal encoder

$$p(h|x) = \frac{p(h)}{N(x)} \exp \left[-\beta D_{KL}(p(y|x) \| p(y|h)) \right]. \quad (\text{S4})$$

By absorbing terms which only depend on h and x into $p(h)$ and $N(x)$, respectively, the encoder can be expressed as

$$p(h|x) = \frac{p(h)}{N(x)} \exp \left[\beta \int dy p(y|x) \log p(y|h) \right]. \quad (\text{S5})$$

When $\beta < 1$, it follows from the data processing inequality $I(X, H) \geq I(Y, H)$ that (S2) is minimized by a trivial encoder $p(h|x) = p(h)$. In this case, $\mathcal{L}_{IB} = 0$ and no information passes through the bottleneck. As β is increased, more information is allowed through the bottleneck until the relevant variables begin to gain a dependence on the state x . This occurs suddenly for a certain value of $\beta = \beta_1 > 1$ at which the encoder becomes non-uniform and $I(H, X)$ becomes non-zero. This is referred to as an IB transition; for increasing β , there may be a sequence of transitions at β_2, β_3, \dots etc.

Transfer operators

The transfer operator of a Markovian dynamical system describes the evolution of probability distributions

$$p_{X_{t+\Delta t}}(x) = U^{\Delta t}[p_{X_t}](x) = \int dx' p(X_{t+\Delta t} = x | X_t = x') p(X_t = x'). \quad (\text{S6})$$

The operator may be decomposed as

$$U^{\Delta t} = \sum_n |\rho_n\rangle e^{\lambda_n \Delta t} \langle \phi_n| + U_{\text{ess}}. \quad (\text{S7})$$

For noisy dynamical systems the operator $U^{\Delta t}$ is nearly compact, meaning that its essential spectral radius will be small [33, 49, 50]. We therefore neglect it in the following.

Above we adopted notation standard from quantum mechanics: $|\rho_n\rangle$ is treated as a vector in the function space on which $U^{\Delta t}$ acts (for example L^1), while $\langle \phi_n|$ is a vector in the dual space (for example L^∞). The right eigenfunctions can be thought of as densities, and ρ_0 is the steady state distribution and satisfies $U^{\Delta t}\rho_0 = \rho_0$. These vectors act on each other via

$$\langle \phi_n | \rho_m \rangle = \langle \phi_n, \rho_m \rangle = \int dx \phi_n(x) \rho_m(x) \quad (\text{S8})$$

and they satisfy the biorthogonality condition

$$\langle \phi_n, \rho_m \rangle = \delta_{nm}.$$

One may use these functions to decompose the conditional probability distribution as

$$p(x_{t+\Delta t}|x_t) = \sum_n e^{\lambda_n \Delta t} \rho_n(x_{t+\Delta t}) \phi_n(x_t). \quad (\text{S9})$$

In this work we consider dynamical systems given by a Langevin equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sqrt{2D(\mathbf{x})} \xi(t) \quad (\text{S10})$$

where the first term is the deterministic part of the dynamics, and the second term corresponds to noise, where

$$\langle \xi_i(t) \xi_j(t') \rangle = C_{ij} \delta(t - t').$$

This framework captures purely deterministic dynamics, which are obtained by setting $D = 0$. For dynamics given by a Langevin equation, probability distributions evolve according to the corresponding Fokker-Planck equation

$$[\mathcal{L}\rho](\mathbf{x}) = \partial_t \rho(\mathbf{x}) = -\partial_i(f_i(\mathbf{x})\rho(\mathbf{x})) + \partial_i \partial_j(D(\mathbf{x})C_{ij}\rho(\mathbf{x})).$$

Here we recognize \mathcal{L} as the infinitesimal generator of the transfer operator U : evolution for a time t is given by $U = e^{t\mathcal{L}}$. The adjoint operator is given by the so-called backward Kolmogorov equation, and it describes the evolution of functions ϕ .

$$\mathcal{L}^\dagger \phi(x) = f_i(\mathbf{x})\partial_i(\phi(\mathbf{x})) + D(\mathbf{x})C_{ij}\partial_i\partial_j(\phi(\mathbf{x})). \quad (\text{S11})$$

For $D = 0$, \mathcal{L} is the generator of the so-called Perron-Frobenius operator, while its adjoint is the Koopman operator [27].

The Fokker-Planck equation may be written

Numerically solving “exact” IB and Ulam approximations of the transfer operator

The optimal IB encoder (S4) can be found using the Blahut-Arimoto (BA) algorithm [58]. As described in detail in Refs. [38, 97] and sketched in Fig. S4, the algorithm is an iterative procedure, where the encoder at iteration $k + 1$ is updated according to

$$\begin{cases} p_{k+1}(h|x_t) = \frac{p_k(h)}{N_{k+1}(x)} \exp(-\beta D_{\text{KL}}(p(x_{t+\Delta t}|x_t)\|p_k(x_{t+\Delta t}|h))) \\ p_{k+1}(h) = \sum_{x_t} p(x_t)p_{k+1}(h|x_t) \\ p_{k+1}(x_{t+\Delta t}|h) = \sum_{x_t} p(x_{t+\Delta t}|x_t)p_{k+1}(x_t|h). \end{cases}$$

The first line simply plugs the previous estimate of the encoder in to Eq. S48 (and normalizes the distribution), while the following two lines update marginal and conditional distributions using the new estimate of the encoder. In Refs. [38, 97] it is shown that this algorithm converges.

The BA algorithm requires access to the conditional distribution $p(x_{t+\Delta t}|x_t)$ for each x_t , as well as the steady state $p(x_t)$. To solve the IB optimization problem we therefore need a numerical approximation of the transfer operator, which we obtain by an Ulam approximation [59, 60]. In brief, one divides space into bins and computes a finite-dimensional approximation to the conditional distribution as

$$P_{ij} = P(X_{t+\Delta t} = x_j | X_t = x_i) = N_{i \rightarrow j}/N_i, \quad (\text{S12})$$

where N_i is the number of trajectories starting in bin i and $N_{i \rightarrow j}$ the number of observed transitions from bin i to bin j . The transfer operator is then approximately given by

$$Up(x_j) \approx \sum_{x_i} P_{ij}p(x_i), \quad (\text{S13})$$

and eigenvalues and eigenvectors of U can be computed by diagonalizing P .

II. MODEL REDUCTION IN GENERAL

Consider a dynamical system which can be described by the evolution of the random variable X_t . In the case where X_t is high-dimensional but undergoes “stereotyped” or regular behavior, it often suffices to describe the system with a reduced dynamical variable H_t , where H_t is lower dimensional than X_t . Schematically, a reduced model would be used in the following way to predict the system’s evolution over a time Δt , for any choice of starting time t :

A reduced description is useful only if it can be used to simulate the system forward in time and make predictions about the future. In other words, one aims to find a reduction $h(X)$ and an evolution rule \mathcal{S} so that the above diagram commutes. Reducing the initial state and evolving it should be equivalent to first evolving the full system, and then reducing the final state.

The requirements of a reduced model may then be enumerated as follows:

1. Reduction

One hopes that the reduced state is more “compact” than the full state. One way to achieve this is to require, for example, that

$$\dim H_t \ll \dim X_t. \quad (\text{S14})$$

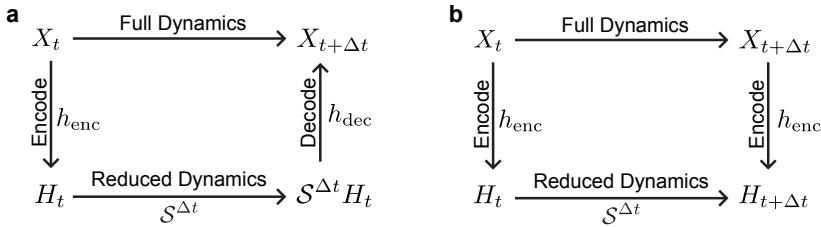


FIG. S1. Two views of model reduction. (a) In one view, the reduced model should act as a useful surrogate of the full system. After reducing the system via an encoder h_{enc} , the reduced state is evolved forward in time via $\mathcal{S}^{\Delta t}$ and subsequently decoded by h_{dec} to obtain an estimate of the full system $X_{t+\Delta t}$. In general, if significant information has been lost during encoding, the decoding will be probabilistic to produce a distribution of possible future states. (b) Alternatively, one may insist only that the encoding is “consistent” with respect to the dynamics (and is not necessarily informative about the full system). In this picture, it should be equivalent to either encode the future state, or to first encode and then evolve the encoded state.

2. Predictiveness

There are two ways to enforce predictiveness, which are shown in Fig. S1. On the one hand is the view that the reduced variables should be useful as a surrogate for the full system, which are simple enough to evolve forward for long times. The hope is that the forecasted latent variable may still contain sufficient information to approximately predict the future full state of the system [31, 98]. On the other hand, in physics often one instead aims to find variables which are simple to predict on their own. Instead of using them to decode into a prediction of the full state, one hopes that the evolved latent variable is equivalent to first evolving the full state, and then reducing it to its latent representation. Mathematically, the first view leads to the requirement that the quantity

$$d(h_{\text{dec}}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) \quad (\text{S15})$$

is small, where d is some choice of distance that measures how the two predictions differ from each other. The second view leads to the requirement that

$$d(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \quad (\text{S16})$$

is small.

In general, one may determine the functions h_{enc} , h_{dec} and the operator $\mathcal{S}^{\Delta t}$ as the solution to an optimization problem

$$\min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \dim h_{\text{enc}}(X_t) + \beta_1 d(h_{\text{dec}}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) + \beta_2 d(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \quad (\text{S17})$$

The parameters β_1 and β_2 must be chosen to reflect the desired properties of the reduced model. The choice of $\beta_2 = 0$ or $\beta_1 = 0$ correspond to the two views of model reduction described above, but in principle both terms may be present. In both the main text and here, we focus on the cases where only one term is present, but comment briefly on the case $\beta_1, \beta_2 \neq 0$ below. It turns out that the case $\beta_1 = 0$ is uninteresting (though see “Relation to other work” below). The reason is that the objective function may then be trivially solved by $h(X_t) = \text{const}$ and $\mathcal{S}^{\Delta t} = \text{id}$. For any choice of dimension, this will minimize the second term of the objective. Consequently, we take our model reduction objective function to be

$$\min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \mathcal{L}_{\text{MR}} \equiv \min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \dim h_{\text{enc}}(X_t) + \beta_1 d(h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}). \quad (\text{S18})$$

In the following, we will take the perspective that every step in the diagram above is, or may be, stochastic. That is, the stochastic time evolution of X_t and H_t are described by stochastic functions \mathcal{S} , \mathcal{T} , or equivalently by an evolution operator that acts on distributions

$$p_{X_{t+\Delta t}}(x) = [U^{\Delta t} p_{X_t}](x) \quad (\text{S19})$$

$$p_{H_{t+\Delta t}}(h) = [V^{\Delta t} p_{H_t}](h). \quad (\text{S20})$$

The encoding map $h(X_t)$ may also be a stochastic function, whose output is described by a distribution $p(h|x) = p(H_t = h|X_t = x)$.

In this work, we replace measures of distance $d(X, Y)$ with the mutual information $I(X, Y)$. The distance $d(X, Y)$ measures how “close” X is to Y , which depends inherently on the system’s geometry. In contrast, the mutual information $I(X, Y)$ measures how precisely Y can be determined given knowledge of X and abstracts away the underlying geometry of the system. With this choice, the objective is given by

$$\min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \dim h_{\text{enc}}(X_t) - \beta_1 I(h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) \quad (\text{S21})$$

where we have changed the sign of the second term; instead of minimizing distance, we want to maximize information.

As stated in the main text, this choice of “distance” has dramatic consequences for our optimization problem. To see this, we note that the latent evolution operator \mathcal{S} and decoder h_{dec}^{-1} only occurs in the second term, so we may write

$$\min_{h_{\text{enc}}, \mathcal{S}, h_{\text{dec}}^{-1}} \mathcal{L}_{\text{MR}} = \min_{h_{\text{enc}}} \left[\dim h_{\text{enc}}(X_t) - \min_{\mathcal{S}, h_{\text{dec}}^{-1}} \beta_1 I(h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) \right] \quad (\text{S22})$$

The variables appearing in the above equation the Markov chain

$$X_{t+\Delta t} — X_t — h_{\text{enc}}(X_t) — \mathcal{S}^{\Delta t} h_{\text{enc}}(X_t) — h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t))$$

where $X — Y — Z$ means that Z is conditionally independent of X , given Y : $p(z, x|y) = p(z|y)p(x|y)$. The data processing inequality states that, given a Markov chain $X — Y — Z$, then

$$I(X, Z) \leq I(X, Y).$$

In particular, if $Z = g(Y)$ then $I(X, g(Y)) \leq I(X, Y)$, where equality is obtained if g is a bijection [58]. Consequently, we have

$$\begin{aligned} I(h_{\text{dec}}^{-1}(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t)), X_{t+\Delta t}) &\leq I(\mathcal{S}^{\Delta t} h_{\text{enc}}(X_t), X_{t+\Delta t}) \\ &\leq I(h_{\text{enc}}(X_t), X_{t+\Delta t}), \end{aligned}$$

where the inequalities are maximized if h_{dec}^{-1} and $\mathcal{S}^{\Delta t}$ are both bijective functions. Notably the bound is saturated if $\mathcal{S}^{\Delta t} = \text{id}$ and $h_{\text{dec}}^{-1} = \text{id}$, so that the objective function takes the form

$$\min_{h_{\text{enc}}} \mathcal{L}_{\text{MR}} = \min_{h_{\text{enc}}} \left[\dim h_{\text{enc}}(X_t) - \beta_1 I(h_{\text{enc}}(X_t), X_{t+\Delta t}) \right] \quad (\text{S23})$$

Conceptually, this means that the purely information-theoretic framework we have taken does not tell us how to evolve the reduced variable forward in time, only how to choose the optimal reduced variables. However, the fact that any bijective $\mathcal{S}^{\Delta t}$ is a solution should not be taken to mean that H_t has no dynamics, or that its dynamics are undefined. Instead, it simply means that the mutual information is insensitive to the choice of $\mathcal{S}^{\Delta t}$, because this transformation does not impact the information contained about $X_{t+\Delta t}$. The dynamics of H_t may be obtained by encoding X_t at each time t . To determine which $\mathcal{S}^{\Delta t}$ is optimal among all choices of bijective transformations, one must introduce and solve another objective function, for example $\min_{\mathcal{S}} \|H_{t+\Delta t} - \mathcal{S}^{\Delta t} H_t\|^2$, which again requires a choice of distance function.

In practice, one may directly solve (S23) by fixing $\dim h_{\text{enc}}$ to various different values, and for each solving for the h_{enc} which optimizes the mutual information with $X_{t+\Delta t}$. To gain an analytical understanding of what this objective is solving, we instead replace the dimensionality reduction term with a term that similarly restricts the capacity of the reduced variable; namely, the mutual information $I(X_t, h_{\text{enc}}(X_t))$. As a result of this choice, the objective becomes

$$\min_{h_{\text{enc}}} \mathcal{L}_{\text{IB}} = \min_{h_{\text{enc}}} \left[I(X_t, h_{\text{enc}}(X_t)) - \beta_1 I(h_{\text{enc}}(X_t), X_{t+\Delta t}) \right]. \quad (\text{S24})$$

This is the information bottleneck objective [36, 38] (see Section I). The benefit of working with this objective is that it is analytically tractable, allowing us gain theoretical insight into the form of solutions of (S24) which we expect to be similar to solutions of (S23). Indeed in practice we see that, for fixed $\dim h_{\text{enc}}$, solutions to S24 (which may be found by setting $\beta_1 \gg 1$) are qualitatively similar to the solution to (S24).

Other variants of the objective function

InfoMax

We argued above that the $\beta_1 = 0$ case for the objective function (S17) is uninteresting because of the existence of a trivial solution. However, the situation changes when distances are replaced by mutual information. In that case, the objective becomes

$$\min_{h_{\text{enc}}} \mathcal{L}_{\text{InfoMax}} = \min_{h_{\text{enc}}} \left[\dim h_{\text{enc}}(X_t) - \beta_2 I(h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \right] \quad (\text{S25})$$

This objective is known as the InfoMax objective [99] and generally it does not have a trivial solution because mutual information is not a true metric function. For linear encodings h_{enc} it reduces to the so-called canonical correlation analysis objective (CCA) [100]. Interestingly, for discrete-state discrete-time Markov chains being partitioned into a binary reduced state $h_{\text{enc}}(X_t) \in \{0, 1\}$ it has been shown that the optimal solution is one which assigns classes based on the first subleading left eigenvector of the transfer operator [101].

The symmetric information bottleneck

We consider here the situation where $\beta_1 = 0$, as in the case of InfoMax, but with the dimensionality reduction term replaced by a term encouraging compression, $I(X_t, h_{\text{enc}}(X_t))$ as in the main text. For simplicity let us assume now that the dynamics are stationary. Then $p(X_t = x) = p(X_{t+\Delta t} = x)$ and $I(X_t, h_{\text{enc}}(X_t)) = I(X_{t+\Delta t}, h_{\text{enc}}(X_{t+\Delta t}))$. The objective function can consequently be written

$$\min_{h_{\text{enc}}} \mathcal{L}_{\text{SymIB}} = \min_{h_{\text{enc}}} \left[I(X_t, h_{\text{enc}}(X_t)) + \beta_2 I(h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \right] \quad (\text{S26})$$

$$= \min_{h_{\text{enc}}} \frac{1}{2} \left[I(X_t, h_{\text{enc}}(X_t)) + I(X_{t+\Delta t}, h_{\text{enc}}(X_{t+\Delta t})) - 2\beta_2 I(h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \right]. \quad (\text{S27})$$

This objective is known as the symmetric information bottleneck [102]. In settings where $p(X_t, X_{t+\Delta t})$ may be estimated, a solution to this objective may be found using a Blahut-Arimoto-like algorithm just as the regular IB problem.

The deterministic information bottleneck

The compression term we choose as a proxy for $\dim h_{\text{enc}}(X_t)$ is not unique. In particular, simply minimizing the Shannon entropy $S(h_{\text{enc}}(X_t))$ achieves a similar goal. This problem has been previously studied in Ref. [43]. Interestingly, this change leads the resulting encoder to be deterministic, $p(H_t|X_t) \sim \delta(H_t - f(X_t))$ where $f(X_t)$ is a deterministic function of X_t .

Other variants

The last variant we consider is the case $\beta_1 \neq 0$, $\beta_2 \neq 0$, but with the compression term $I(X_t, h_{\text{enc}}(X_t))$. The full objective function is

$$\min_{h_{\text{enc}}} \mathcal{L}_{\text{SymIB}} = \min_{h_{\text{enc}}} \left[I(X_t, h_{\text{enc}}(X_t)) - \beta_1 I(h_{\text{enc}}(X_t), X_{t+\Delta t}) - \beta_2 I(h_{\text{enc}}(X_t), h_{\text{enc}}(X_{t+\Delta t})) \right]. \quad (\text{S28})$$

This objective may be thought of as an information bottleneck, but with an additional term $I(H_t, H_{t+\Delta t})$ that encourages the encoding to learn a variable which is predictive of its own future. We do not consider this case further beyond noting that the addition of such a term does not affect our perturbative analysis below.

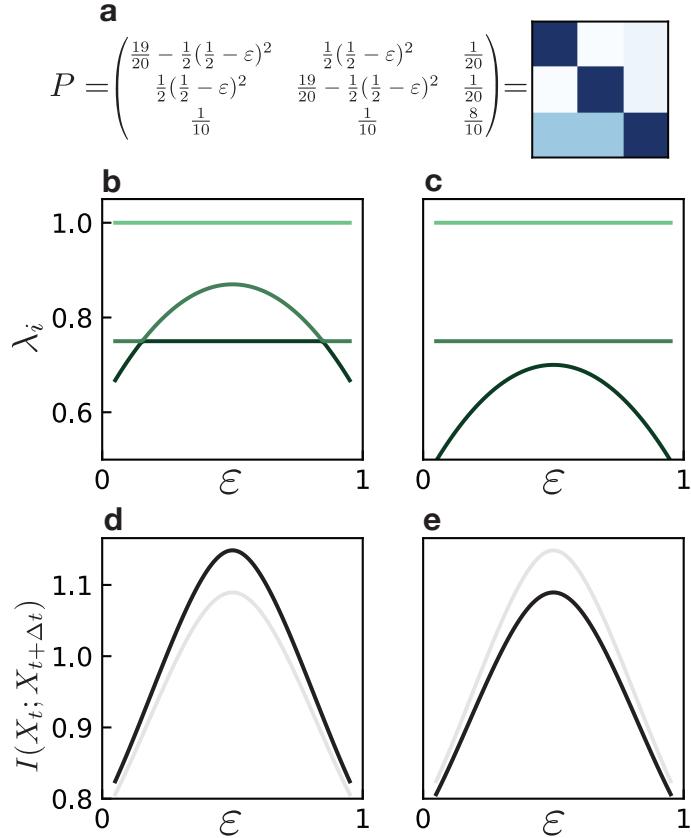


FIG. S2. Information gain due to an eigenvalue pile-up. (a) To study the role of a gap closing compared to the pile-up of eigenvalues (beyond the dominant one) in the mutual information $I(X_t; X_{t+\Delta t})$ we build a matrix P with which we can tune the eigenvalues via a parameter ε . (b) Eigenvalues for the matrix shown in (a) with varying parameter values ε ; for $\varepsilon \rightarrow 0.5$, the spectral gap $\lambda_0 - \lambda_1$ closes. (c) Eigenvalues for the matrix shown in (a), but with all $\frac{1}{2}(\frac{1}{2}-\varepsilon)^2$ terms subtracted with a constant (here $1/8$). Here, the spectral gap $\lambda_0 - \lambda_1$ does not close but there is an accumulation of eigenvalues at $\varepsilon \rightarrow 0.5$. (d-e) Mutual information for $\Delta t = 1$ for the matrices in (b-c), respectively, as a function of ε . The curve from the other scenario is shown in gray for comparison. Both exhibit a clear peak in information which differs only slightly in magnitude, showing that the information peak in Fig. 2d-f of the main text is not due to the closing gap alone, but rather due to contributions from subdominant eigenvalues.

III. INFORMATION THEORETIC QUANTITIES FOR DYNAMICAL SYSTEMS

Rate of information decay

Here we show how the decay of information in the system may be related to the spectrum of its transfer operator. We consider the mutual information

$$I(X_t, X_{t+\Delta t}) = \sum_{x,y} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \quad (\text{S29})$$

where x denotes values of the random variable X_t and y the values of $X_{t+\Delta t}$. We further assume that the system's dynamics are given by a transfer operator U with integral kernel $p(y|x)$ that can be spectrally decomposed as

$$p(y|x) = \sum_n e^{\lambda_n t} \rho_n(y) \phi_n(x), \quad (\text{S30})$$

where $\lambda_0 = 0$, $\phi_0 = \text{const}$ and $\rho_0(y)$ is the steady state distribution. The mutual information can then be written

$$I(X_t, X_{t+\Delta t}) = \sum_{x,y} \sum_n e^{\lambda_n \Delta t} \rho_n(y) \phi_n(x) p(x) \log \left(1 + \sum_{m>0} e^{\lambda_m \Delta t} \frac{\rho_m(y)}{p(y)} \phi_m(x) \right) \quad (\text{S31})$$

$$= \sum_{n,m>0} e^{(\lambda_n + \lambda_m) \Delta t} \sum_{x,y} \phi_n(x) \phi_m(x) p(x) \frac{\rho_n(y) \rho_m(y)}{p(y)} \quad (\text{S32})$$

$$+ \sum_{n,m,\ell>0} e^{(\lambda_n + \lambda_m + \lambda_\ell) \Delta t} \sum_{x,y} \phi_n(x) \phi_m(x) \phi_\ell(x) p(x) \frac{\rho_n(y) \rho_m(y) \rho_\ell(y)}{p(y)} + \dots \quad (\text{S33})$$

For long times, the contribution of λ_1 dominates, with the remaining terms decaying as $e^{(\lambda_n - \lambda_1)\Delta t}$ for $n \geq 2$. Retaining only the first term, we see

$$I(X_t, X_{t+\Delta t}) = e^{2\lambda_1 \Delta t} \sum_x \phi_1^2(x) p(x) \sum_y \frac{\rho_1(y)^2}{p(y)}. \quad (\text{S34})$$

For short times, other eigenvalues will also contribute to the mutual information (Fig. S2).

Information-theoretic measures of Markovianity

Here we discuss measures of Markovianity, both in the context of the original state X_t as well as the encoded representation H_t . For concreteness we assume that our system is observed at regularly spaced intervals of size Δt and denote $x_n = x(t_0 + n\Delta t)$.

(Non-)Markovianity of the measured state X_t

A system is non-Markovian if the transition probability depends on the previous state, $p(x_{n+1}|x_n, x_{n-1}) \neq p(x_{n+1}|x_n)$. The deviation of a system from Markovianity can be quantified by measuring the difference between these distributions, given in terms of the Kullback-Leibler divergence

$$D_{\text{KL}}(p(x_{n+1}|x_n, x_{n-1}) \| p(x_{n+1}|x_n)). \quad (\text{S35})$$

If these distributions are (approximately) equal, this quantity will be (close to) zero. In general this divergence will depend on the precise values of x_n and x_{n-1} . For that reason, we instead consider the average over all possible values of x_n and x_{n-1} . This quantity coincides with the conditional mutual information,

$$I(X_{n+1}, X_{n-1}|X_n) = \sum_{\substack{x_{n+1} \\ x_n \\ x_{n-1}}} p(x_{n+1}, x_n, x_{n-1}) \log \frac{p(x_{n+1}, x_{n-1}|x_n)}{p(x_{n+1}|x_n)p(x_{n-1}|x_n)} \quad (\text{S36})$$

$$= \sum_{\substack{x_n \\ x_{n-1}}} p(x_n, x_{n-1}) \sum_{x_{n+1}} p(x_{n+1}|x_n, x_{n-1}) \log \frac{p(x_{n+1}|x_n, x_{n-1})}{p(x_{n+1}|x_n)} \quad (\text{S37})$$

$$= \mathbb{E}_{x_n, x_{n-1}} D_{\text{KL}}(p(x_{n+1}|x_n, x_{n-1}) \| p(x_{n+1}|x_n)). \quad (\text{S38})$$

Because the Kullback-Leibler divergence is non-negative, this conditional mutual information is zero only if $X_{n+1} - X_n - X_{n-1}$ forms a Markov chain [103]. Using the “chain rule” for conditional information, we can express this as a difference in mutual informations [58]

$$I(X_{n+1}, X_{n-1}|X_n) = I(X_{n+1}, \{X_n, X_{n-1}\}) - I(X_{n+1}, X_n). \quad (\text{S39})$$

A system which is non-Markovian can be made Markovian by introducing time-lagged variables, which follows from Taken’s embedding theorem [104]. One constructs a new dynamical variable which is a concatenation of the previous states of the system,

$$X_n^{(k)} = \{X_{n-k}, \dots, X_{n-1}, X_n\}.$$

However, a common problem in practice is that of finding the minimal sufficient k such that the dynamics are Markovian. This value of k can be identified by finding the value of k for which $\Delta I_k \approx 0$, where we define

$$\Delta I_k = I(X_{n+1}^{(k+1)}, X_n^{(k+1)}) - I(X_{n+1}^{(k)}, X_n^{(k)}).$$

To see this, note that

$$\begin{aligned} I(X_{n+1}^{(k)}, X_n^{(k)} |, X_{n-1}^{(k)}) &= I(X_{n+1}^{(k)}, X_n^{(k+1)}) - I(X_{n+1}^{(k)}, X_n^{(k)}) \\ &\leq I(X_{n+1}^{(k+1)}, X_n^{(k+1)}) - I(X_{n+1}^{(k)}, X_n^{(k)}) \\ &= \Delta I_k \end{aligned}$$

Thus a small ΔI_k guarantees a small conditional entropy, and hence Markovianity. We demonstrate the utility of this approach by considering a Lorenz system in which only the x -variable is observed. By increasing the number of time lags, we see that the mutual information $I(X_{n+1}^{(k)}, X_n^{(k)})$ plateaus at $k \approx 6$, implying Markovianity by the above argument (Fig. S3a). In Fig. S3b we compare our approach to that introduced in Ref. [63], which looks for a plateau in the transfer entropy of the system. A brief comparison of the benefits of each approach is worth mentioning. The approach of Ref. [63] relies on a binning procedure which can be very computationally costly for when the observed state variable is high dimensional. However, for low-dimensional systems this method can be very effective and may be more efficient than training a neural network to estimate mutual information. Our approach is particularly attractive in situations where the observed system is high dimensional, but the total information quantity is low. This is because the neural network which estimates the mutual information is free to reduce the full state to a lower-dimensional space before estimating the mutual information, which potentially resolves some of the problems of high-dimensionality faced by approaches based on binning.

(Non-)Markovianity of the encoding variable H_t

If the full system is non-Markovian its dynamics are no longer given by a global transfer operator. If Markovianity is violated so that $p(x_{t+\Delta t}|x_t, x_{t-\tau}) \neq p(x_{t+\Delta t}|x_t)$, the transfer operator U will no longer be independent of time and the eigenvectors corresponding to its dominant eigenvalues will not necessarily have any relation to the true long-time dynamics. What will IB learn in this case? The only quantity arising in the optimal encoder is the one-time conditional transition probability $p(x_{t+\Delta t}|x_t)$. If there is memory in the system, then this probability can be computed by integrating out the memory

$$p(x_{t+\Delta t}|x_t) = \int dx_{t-\tau} p(x_{t+\Delta t}, x_{t-\tau}|x_t) \quad (\text{S40})$$

$$= \int dx_{t-\tau} p(x_{t+\Delta t}|x_t, x_{t-\tau}) p(x_{t-\tau}|x_t) \quad (\text{S41})$$

$$= \int dx_{t-\tau} p(x_{t+\Delta t}|x_t, x_{t-\tau}) \frac{p(x_t|x_{t-\tau})}{p(x_t)} p(x_{t-\tau}) \quad (\text{S42})$$

Where we assume for simplicity that the memory manifests as a dependence on the state at a single fixed time τ in the past. This operator may also be decomposed into outer products of right and left eigenvectors, as in the Markovian case. However, the eigenvalues will no longer correspond to a rate of decay of correlations, and the dominant eigenvectors will not necessarily correspond to slow dynamics. While the study of the spectrum of this object may shed light on relevant processes in the system, we instead chose to deal with potential non-Markovianity by introducing time lags to make the dynamics approximately Markovian.

In general, the encoding $h(t)$ learned by IB will not exhibit Markovian dynamics. To see this, consider an encoder $p(h_t|x_t)$. The dynamics induced on the encoding will have transitions governed by the transition probabilities

$$p(h_{t+\Delta t}|h_t) = \int dx_{t+\Delta t} dx_t p(h_{t+\Delta t}, x_{t+\Delta t}, x_t|h_t) \quad (\text{S43})$$

$$= \int dx_{t+\Delta t} dx_t p(h_{t+\Delta t}|x_{t+\Delta t}) p(x_{t+\Delta t}|x_t) p(x_t|h_t) \quad (\text{S44})$$

where we used the fact that the variables form the Markov chain $H_t - X_t - X_{t+\Delta t} - H_{t+\Delta t}$ [105] For more than one

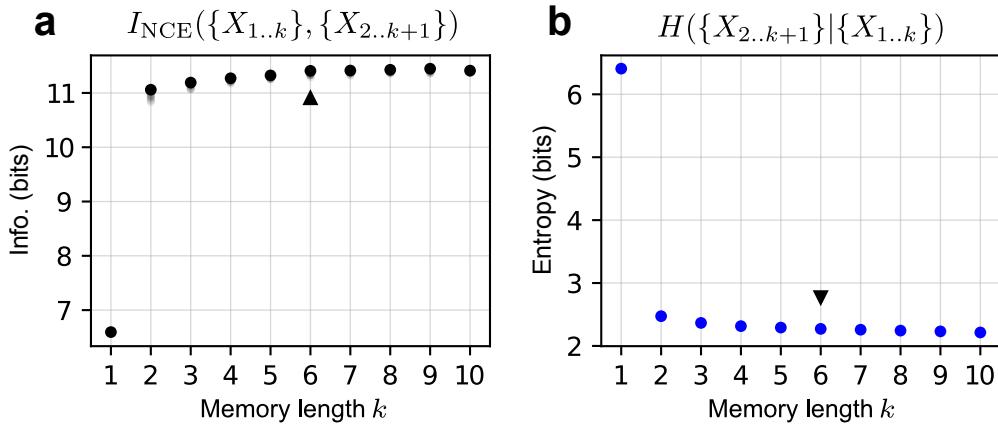


FIG. S3. **Testing for Markovianity in the data** Information can be used as a test of how far the input data deviates from Markovianity. Here we take data from the Lorenz system [106] which was simulated with a time step $\delta t = 0.02$, and we consider the state of our system as the x -coordinate, discarding the y and z measurements. (a) Mutual information between consecutive states as a function of memory length k . Mutual information is approximated with the noise contrastive estimate described in Methods. (b) Transfer entropy between states as a varying function of memory length. Entropy is approximated using the approach in Ref. [63].

step,

$$p(h_{t+2\Delta t}|h_{t+\Delta t}, h_t) = \int dx_{t+2\Delta t} dx_{t+\Delta t} dx_t p(h_{t+2\Delta t}|x_{t+2\Delta t}) p(x_{t+2\Delta t}|x_{t+\Delta t}) p(x_{t+\Delta t}|x_t, h_{t+\Delta t}) p(x_t|h_t) \quad (\text{S45})$$

$$\neq p(h_{t+2\Delta t}|h_{t+\Delta t}). \quad (\text{S46})$$

We show a concrete example of the difference between these two distributions for the encoding of a particle in a double-well in Fig. 4a,b of the main text. Under what conditions are the encoding dynamics also Markovian? A sufficient condition is $p(x_{t+\Delta t}|x_t, h_{t+\Delta t}) = p(x_{t+\Delta t}|h_{t+\Delta t})$. This is the case when the full state $x_{t+\Delta t}$ can be determined from the encoding $h_{t+\Delta t}$ as accurately as from the previous full state x_t . This is true, for example, when a (deterministic) high-dimensional system occupies a low-dimensional manifold. However, for many systems of interest this will not be strictly true. Consider the dynamics given by a noisy Hopf normal form as in the main text. Although the dynamics essentially take place on a circular limit cycle, the current state (r_t, θ_t) does contain information about the future radius $r_{t+\Delta t}$ which is not accounted for by the angular coordinate alone. Instead of hoping that Markovianity holds exactly, one instead hopes that the encoded dynamics undergo *approximately* Markovian dynamics.

A quantitative measure of Markovianity is given by $I(H_{t+2\Delta t}, H_t|H_{t+\Delta t})$. Asymptotically for small compression, this quantity will go to zero if the dynamics of X_t are Markovian because the encoding approaches a completely informative copy of X_t . However, the above expression offers no guarantees about how $I(H_{t+2\Delta t}, H_t|H_{t+\Delta t})$ approaches zero. In fact, as shown in Fig. 4 of the main text, the approach may be far from monotonic.

It is important to note that just because the encoded dynamics are Markovian does not mean that one has found a “correct” low dimensional description of the system. We illustrate this on driven sine-Gordon dynamics. This system evolves according to a one-dimensional PDE. For certain driving amplitudes it exhibits quasiperiodic dynamics, and hence its state can be described by a point on a torus. In Fig. 6 of the main text we show that the information shared between the encoding and the future state plateaus already after two dimensions, $h \in \mathbb{R}^2$. Indeed, this description does suffice to essentially completely recover the full state from the encoding. To do this, the torus on which the system lives must be “cut” so that it can be flattened out in two dimensions (Fig. 6b).

Estimating memory with mutual information

Markovianity is a crucial assumption underlying many approaches to data-driven inference and analysis of dynamical systems [17, 63, 107]. Deviations from Markovianity are commonly identified by studying the reduction in error of an inferred model when the system’s state is augmented with time-lagged measurements [17]. An alternative approach based on information theory is to instead consider how much information a measurement of the current state contains about the future state [63]. The benefit of such an approach is that it doesn’t require any assumptions about the underlying model governing a system’s evolution. The downside is that estimating entropy and related quantities

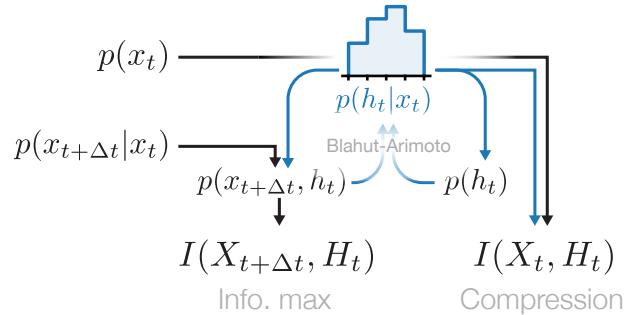


FIG. S4. **Exact IB.** Exact IB finds an optimal encoder using an iterative Blahut-Arimoto algorithm, described in the main text, for a rate-distortion problem with Kullback-Leibler distortion [38]. This process requires access to (an estimate of) the transfer matrix and the steady state distribution. Once an optimal encoder has been found, relevant quantities such as mutual information can be computed.

typically relies on binning, which for high-dimensional systems is rendered difficult by the curse of dimensionality [108].

Using the approach developed in Ref. [63] one can estimate the degree of non-Markovianity directly from data. However, this requires an approximation of the full transfer operator. This prompted us to ask whether direct measurements of the mutual information can identify the amount of Markovianity without explicitly constructing the transition probabilities. To do this, we consider the Lorenz system [106]. While the dynamics of the full state $X_t = (x_t, y_t, z_t)$ are Markovian, if one observes only one degree of freedom, for example $X_t = x_t$, the dynamics become non-Markovian. By constructing a new state using previous measurements, $X_t^{(k)} = (x_{t-k}, \dots, x_{t-1}, x_t)$, one can recover a Markovian system by taking sufficiently large k .

In Ref. [63], the memory length k is chosen to be the value for which the transfer entropy $H(X_{t+1}^{(k)}|X_t^{(k)})$ reaches its minimum value. This indicates a variable which is maximally predictive of its future. For the Lorenz system under consideration (simulated with a time step of $\delta t = 0.02$), we show the results of this approach in Fig. S3b, which indicates that a memory $k = 5$ is sufficient to obtain a Markovian system.

Instead of considering the entropy, we identify the necessary amount of memory by seeing when the mutual information $I(X_{t+1}^{(k)}, X_t^{(k)})$ plateaus as a function of k . One notable benefit of this approach is that it does not require an approximation of the full transfer operator, as above, but can be directly estimated from data using the noise contrastive estimate of the mutual information [109] discussed in the main text. We find that this metric returns the same critical value of $k = 6$ after which the state is maximally informative of the future.

IV. DERIVATIONS

Here we derive the form of the optimal encoder at the first IB transition,

$$p_\beta^*(h_t|x_t) \approx \frac{1}{N(x_t)} p_\beta^*(h_t) \exp\left(\beta e^{\lambda_1 \Delta t} \phi_1(x_t) f_1(h_t)\right) \quad (\text{S47})$$

In Ref. [38] (and reproduced above in (S5)), it is shown that the encoding minimizing the IB objective satisfies the implicit equation

$$p_\beta^*(h_t|x_t) = \frac{p_\beta^*(h_t)}{N(x_t)} \exp\left(-\beta D_{\text{KL}}[p(x_{t+\Delta t}|x_t) \| p(x_{t+\Delta t}|h_t)]\right) \quad (\text{S48})$$

where N is a normalization factor ensuring $\int dh p_\beta^*(h_t|x_t) = 1$, and

$$p_\beta^*(h_t) \equiv p_\beta^*(h_t) = \int dx_t p_\beta^*(h_t|x_t) p(x_t). \quad (\text{S49})$$

The Kullback-Leibler divergence between two probability distributions $p(x)$ and $q(x)$ is defined by

$$D_{\text{KL}}[p||q] = \int dx p(x) \log \frac{p(x)}{q(x)}. \quad (\text{S50})$$

In the D_{KL} in (S48), the integration is done with respect to the common random variable $X_{t+\Delta t}$ of the two conditional distributions.

Equation S48 contains the conditional distribution $p(x_{t+\Delta t}|h_t)$, which is the composition of two operations: decoding (going from H_t to X_t) and time evolution (going from X_t to $X_{t+\Delta t}$). The decoding is performed using Bayes' theorem $p(x_t|h_t)p(h_t) = p(h_t|x_t)p(x_t)$, so that we have

$$p(x_{t+\Delta t}|h_t) = \int dx_t p(x_{t+\Delta t}|x_t) p(h_t|x_t) \frac{p(x_t)}{p(h_t)}. \quad (\text{S51})$$

Note the appearance of the encoder $p(h_t|x_t)$, making (S48) only an implicit solution of the optimization problem.

The conditional distribution $p(x_{t+\Delta t}|x_t)$ can be written in terms of right and left eigenvectors of U . Here we neglect U_{ess} , which is the operator corresponding to the essential spectrum. The essential spectrum is the part of the spectrum that is not the discrete spectrum; by neglecting it, we are assuming that the essential radius ρ_{ess} (the maximum absolute value of eigenvalues in the essential spectrum) is small enough compared to the first few eigenvalues $|\lambda_n|$ in the point spectrum. While purely deterministic systems may exhibit a large essential radius, the introduction of noise causes the essential spectrum to shrink or disappear [49, 50]. We can then write the conditional distribution in terms of right and left eigenvectors of U ,

$$p(x_{t+\Delta t}|x_t) = \sum_n e^{\lambda_n \Delta t} \rho_n(x_{t+\Delta t}) \phi_n(x_t). \quad (\text{S52})$$

The Kullback-Leibler divergence in (S48) can be represented as a sum of two terms,

$$D_{\text{KL}}[\dots] = \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|x_t) - \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|h_t). \quad (\text{S53})$$

The first term has no dependence on h_t and can hence be absorbed into the normalization $N(x)$. Plugging the decomposition (S52) into the second term leads directly to

$$p_{\beta}^*(h_t|x_t) = \frac{1}{N(x_t)} p_{\beta}^*(h_t) \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x_t) f_n(h_t) \right] \quad (\text{S54})$$

where

$$f_n(h_t) = \int dx_{t+\Delta t} \rho_n(x_{t+\Delta t}) \log p(x_{t+\Delta t}|h_t), \quad (\text{S55})$$

which can be understood as a quasi-cross entropy between the right eigenvector ρ_n and the distribution on $x_{t+\Delta t}$ obtained by decoding h . Equation S54 depends only on the existence of the spectral decomposition; we have made no other assumptions on the dynamics (beyond Markovianity) up until this point.

A. Perturbation theory at the uniform encoder

In this section we derive our main result for the optimal encoder in the limit of high compression (small β). We begin by deriving the result for discrete latent states $h_t \in \{0, \dots, N_H - 1\}$, before discussing continuous latent variables below. In this section, for notational simplicity we will drop some subscripts and rename $x \equiv x_t$, $h \equiv h_t$, $y \equiv x_{t+\Delta t}$.

Mathematical result – Consider a transfer operator U with infinitesimal generator \mathcal{L}_U . Assume that \mathcal{L}_U has a discrete spectrum with eigenvalues satisfying $0 = \text{Re } \lambda_0 > \text{Re } \lambda_1 > \text{Re } \lambda_2 \gg \text{Re } \lambda_3, \dots$. Then, for β just above the first IB transition β_1 so that $f_n(h) \rightarrow 0$, the optimal encoder is given approximately by

$$p_{\beta}^*(h|x) = \frac{1}{N(x)} p_{\beta}^*(h) \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x) f_1(h) \right) \quad (\text{S56})$$

with corrections due to the second eigenfunction given by $f_2(h) \approx f_1(h) e^{-\Gamma \Delta t} + \mathcal{O}(e^{-2\Gamma \Delta t})$ where $\Gamma = \lambda_1 - \lambda_2 > 0$ denotes the spectral gap. For equilibrium systems exhibiting no probability fluxes in the steady state, this result is exact.

To show this we compute the $f_n(h)$ coefficients at the onset of instability. The instability at the first IB transition at $\beta = \beta_1$ corresponds to the emergence of negative eigenvalues in the Hessian of the IB loss,

$$\frac{\partial^2}{\partial f_n(h_i) \partial f_m(h_j)} \mathcal{L}_{\text{IB}}. \quad (\text{S57})$$

Similar perturbative approaches have been studied in other contexts within IB in [51, 54, 110]. For concreteness, we consider a discrete encoding $H_t \in \{0, \dots, N_H - 1\}$. We express the encoder as in Eq. S54 where we now expand the normalization terms

$$p_\beta^*(h_\mu|x) = \frac{f_0(h_\mu) \exp [\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x) f_n(h_\mu)]}{\sum_{h_\lambda} f_0(h_\lambda) \exp [\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x) f_n(h_\lambda)]}. \quad (\text{S58})$$

This conditional probability distribution is shorthand for $p_\beta^*(H_t = h_\mu | X_t = x)$. In the following, we simplify notation as

$$p(h_\mu|x) = \frac{f_0^\mu \exp [\sum_n \tilde{\phi}_n(x) f_n^\mu]}{\sum_{h_\lambda} f_0^\lambda \exp [\sum_n \tilde{\phi}_n(x) f_n^\lambda]} \quad (\text{S59})$$

where $\tilde{\phi}_n(x) = \beta e^{\lambda_n \Delta t} \phi_n(x)$. For the uniform encoder, $f_n^\mu = 0$ for all $n \neq 0$.

Hereafter we assign $\tilde{\phi}_0 = \frac{1}{f_0^\mu}$, taking care to track the correct Greek index which otherwise is not present for $\tilde{\phi}_n$ (in the following, n is always paired with μ and m always with ν). For example, a single derivative of the marginal is given by

$$\partial_n^\mu p(h_\nu) = \int dx p(x) p(h_\nu|x) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu|x)). \quad (\text{S60})$$

To compute the Hessian, we use the fact that $\sum_\lambda \partial_n^\mu p(h_\lambda) = 0$ (by exchanging sum and derivative), and that $p(h_\mu|x)|_{\mathbf{f}=0} = p(h_\mu)$. We note in passing that we find the first derivative terms of \mathcal{L}_{IB} vanish, see also [52, 54]. The compression term is given by

$$\partial_n^\mu \partial_m^\nu I(X, H) = (\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle) (\delta_{\mu\nu} p(h_\mu) - p(h_\nu)p(h_\mu))$$

and the information maximization term is

$$\partial_n^\mu \partial_m^\nu I(Y, H) = \left(\left\langle \langle \tilde{\phi}_n \rangle_{p(\cdot|y)} \langle \tilde{\phi}_m \rangle_{p(\cdot|y)} \right\rangle_y - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) (\delta_{\mu\nu} p(h_\mu) - p(h_\nu)p(h_\mu))$$

Details of the calculations leading to these results are provided in the appendix. Angled brackets with no subscript correspond to an average with respect to the steady state distribution, $\langle \cdot \rangle = \int dx \cdot p(x)$. Written out more explicitly, the second term is

$$\left\langle \langle \tilde{\phi}_n \rangle_{p(\cdot|y)} \langle \tilde{\phi}_m \rangle_{p(\cdot|y)} \right\rangle_y = \int dy p(y) \left(\int dx \tilde{\phi}_n(x) p(y|x) \right) \left(\int dx' \tilde{\phi}_m(x') p(y|x') \right) \quad (\text{S61})$$

where we retain the integration variable y as a subscript on the angled brackets for clarity. In sum, the Hessian of the Lagrangian is given by

$$H_{(\mu,n),(\nu,m)} = \left(\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \beta \left\langle \langle \tilde{\phi}_n \rangle_{p(\cdot|y)} \langle \tilde{\phi}_m \rangle_{p(\cdot|y)} \right\rangle_y - (1 - \beta) \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) (\delta_{\mu\nu} p(h_\mu) - p(h_\nu)p(h_\mu)) \quad (\text{S62})$$

where we index H with a multi-index (μ, n) . From the form of Eq. S62, we see that H is given by a Kronecker (tensor) product

$$H_{(\mu,n),(\nu,m)} = A_{nm}^\beta \otimes G_{\mu\nu}.$$

Eigenvalues of H are given by products of eigenvalues of A^β and G , while eigenvectors are given by the tensor product of eigenvectors of A^β and G . Note that G is nothing other than the covariance matrix for a multinomial distribution. The appearance of unstable directions of the Hessian corresponds to the appearance of negative eigenvalues in its spectrum. As G does not depend on β , zero-crossings of eigenvalues of H therefore correspond to zero-crossings of eigenvalues in the spectrum of A^β .

Equilibrium systems

We first study the stability of the matrix A_{nm}^β in the case of dynamics which satisfy detailed balance. In particular, we consider a Fokker-Planck operator \mathcal{L}_{FP} of the form

$$\partial_t p(x) = \mathcal{L}_{\text{FP}} p(x) = -\partial_i(f_i(x)p(x)) + D\partial_i^2 p(x). \quad (\text{S63})$$

The steady state distribution $\rho_0(x)$ satisfies

$$\mathcal{L}_{\text{FP}}\rho_0(x) = -\partial_i J_i(x) = 0 \quad (\text{S64})$$

where $J_i = f_i\rho_0 - D\partial_i\rho_0$ is a flux. Our main assumption in this subsection is that in the steady state, fluxes vanish: $J_i = 0$. In this case, left eigenfunctions ϕ_n of \mathcal{L}_{FP} become *right* eigenfunctions when multiplied by ρ_0 . To see this, note that

$$\begin{aligned} \mathcal{L}_{\text{FP}}(\rho_0\phi_n) &= \phi_n\mathcal{L}_{\text{FP}}\rho_0 + \rho_0\mathcal{L}_{\text{FP}}^\dagger\phi_n - 2\partial_i\phi_n\underbrace{(f_i\rho_0 - D\partial_i\rho_0)}_{J_i} \\ &= \lambda_n\rho_0\phi_n \end{aligned} \quad (\text{S65})$$

where the first term disappears because ρ_0 is the steady-state distribution, and the third term disappears because of our above assumption on disappearing fluxes, $J_i = 0$. The following inner product then satisfies

$$\begin{aligned} \langle \phi_n\rho_0, \mathcal{L}_{\text{FP}}^\dagger\phi_m \rangle &= \langle \mathcal{L}_{\text{FP}}\phi_n\rho_0, \phi_m \rangle \\ \lambda_m\langle \phi_n\rho_0, \phi_m \rangle &= \lambda_n\langle \phi_n\rho_0, \phi_m \rangle \\ \rightarrow (\lambda_m - \lambda_n)\langle \phi_n\rho_0, \phi_m \rangle &= 0. \end{aligned}$$

This shows that if $\lambda_n \neq \lambda_m$, the inner product vanishes. Similarly, one can show that the same must be true for $\langle \frac{\rho_n}{\rho_0}, \frac{\rho_m}{\rho_0} \rangle$. These are precisely the types of terms appearing in the Hessian. Consequently, for equilibrium (no flux) systems the Hessian is diagonal. The first term follows directly from the above, while the second is given by

$$\begin{aligned} &\sum_y \frac{1}{p(y)} \left(\sum_x p(y, x)\tilde{\phi}_m(x) \right) \left(\sum_{x'} p(y, x')p_\lambda\tilde{\phi}_n(x') \right) \\ &= \sum_{ij} \sum_y \frac{1}{p(y)} \rho_i(y)\rho_j(y)e^{(\lambda_i + \lambda_n)\Delta t} \langle \phi_i\rho_0, \phi_n \rangle \\ &\quad \times e^{(\lambda_j + \lambda_m)\Delta t} \langle \phi_j\rho_0, \phi_m \rangle \\ &= \sum_{ij} e^{(\lambda_i + \lambda_n + \lambda_j + \lambda_m)\Delta t} \delta_{ij}\delta_{ni}\delta_{mj} \\ &= \delta_{nm}e^{(2\lambda_n + 2\lambda_m)\Delta t} \end{aligned}$$

The full matrix A appearing in the Hessian then takes the form

$$A_{nm} = \delta_{nm}e^{2\lambda_n\Delta t} \left(1 - \beta e^{2\lambda_n\Delta t} \right) \quad (\text{S66})$$

except for the $n = m = 0$ term, which is $A_{00} = 0$. The eigenvalues are given directly by these diagonal elements. For small β these are all positive, and become unstable one after the other at

$$\beta = e^{-2\lambda_1\Delta t}, e^{-2\lambda_2\Delta t}, \dots$$

which are increasing in order (remember $\lambda_i \leq 0$). It follows that at the first transition, only f_1 becomes non-zero, and hence the encoder takes the form given by (S47). In the equilibrium case, the encoder learns *exclusively* the first eigenfunction, with no correction due to the second eigenfunction.

General Case

We now show that the first component f_1 is selected at the first IB transition even when the flux \mathbf{J} is non-zero. From our assumption on the spectrum of \mathcal{L}_U , namely $0 > \text{Re}\lambda_1 > \text{Re}\lambda_2 \gg \text{Re}\lambda_3\dots$ it follows that near the first IB

transition the IB loss is given by

$$\mathcal{L}_{\text{IB}} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}^T \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} + \mathcal{O}(f_n^3, e^{\lambda_3 \Delta t})$$

with

$$\begin{aligned} a_{11} &= e^{2\lambda_1 \Delta t} (\langle \phi_1^2 \rangle - \beta B_{11}) = e^{2\lambda_1 \Delta t} \hat{a}_{11} \\ a_{12} &= e^{(\lambda_1 + \lambda_2) \Delta t} (\langle \phi_1 \phi_2 \rangle - \beta B_{12}) = e^{(\lambda_1 + \lambda_2) \Delta t} \hat{a}_{12} \\ a_{22} &= e^{2\lambda_2 \Delta t} (\langle \phi_2^2 \rangle - \beta B_{22}) = e^{2\lambda_2 \Delta t} \hat{a}_{22}, \end{aligned}$$

where we have introduced the shorthand $B_{ij} = \left\langle \langle \tilde{\phi}_i \rangle_{p(\cdot|y)} \langle \tilde{\phi}_j \rangle_{p(\cdot|y)} \right\rangle_y$ and all angled brackets denote averaging with respect to the steady state distribution. The stability of the uniform encoder at $f_n = 0$ is given by the stability of the 2×2 matrix above. The eigenvalues ω_i and eigenvectors \mathbf{v}_i of this matrix can be computed explicitly,

$$\begin{aligned} \eta_{\pm} &= \frac{1}{2} \left(e^{2\lambda_1 \Delta t} \hat{a}_{11} + e^{2\lambda_2 \Delta t} \hat{a}_{22} \pm D \right) \\ \mathbf{v}_{\pm} &= \begin{pmatrix} \frac{-1}{2e^{(\lambda_1 + \lambda_2) \Delta t} \hat{a}_{12}} (-e^{2\lambda_1 \Delta t} \hat{a}_{11} + e^{2\lambda_2 \Delta t} \hat{a}_{22} \mp D) \\ 1 \end{pmatrix} \end{aligned}$$

where

$$D = (e^{4\lambda_1 \Delta t} \hat{a}_{11}^2 + 2e^{2(\lambda_1 + \lambda_2) \Delta t} (2\hat{a}_{12}^2 - \hat{a}_{11}\hat{a}_{22}) + e^{4\lambda_2 \Delta t} \hat{a}_{22}^2)^{1/2}.$$

In what follows, we will express quantities in terms of the spectral gap $\Gamma = \lambda_1 - \lambda_2 > 0$. For example, the expression above can be written

$$\begin{aligned} D &= e^{2\lambda_1 \Delta t} \left(\hat{a}_{11}^2 + 2e^{-2\Gamma \Delta t} (2\hat{a}_{12}^2 - \hat{a}_{11}\hat{a}_{22}) + e^{-4\Gamma \Delta t} \hat{a}_{22}^2 \right)^{1/2} \\ &= e^{2\lambda_1 \Delta t} \left(\hat{a}_{11} + \mathcal{O}(e^{-2\Gamma \Delta t}) \right). \end{aligned}$$

The first eigenvalue to become negative is η_+ . We are interested in the ratio of the corresponding eigenvector components, $f_1/f_2 = \mathbf{v}_{+,1}/\mathbf{v}_{+,2}$, which tells us how much the encoder will depend on the first eigenfunction $\phi_1(x)$ compared to the second. This ratio is given by

$$\begin{aligned} \frac{f_1}{f_2} &= \frac{-e^{(\lambda_1 - \lambda_2) \Delta t}}{2\hat{a}_{12}} \left(-\hat{a}_{11} + e^{2(\lambda_2 - \lambda_1) \Delta t} \hat{a}_{22} - D \right) \\ &= \frac{e^{\Gamma \Delta t}}{2\hat{a}_{12}} \left(2\hat{a}_{11} + \mathcal{O}(e^{-2\Gamma \Delta t}) \right) \\ &= e^{\Gamma \Delta t} \frac{\hat{a}_{11}}{\hat{a}_{12}} + \mathcal{O}(e^{-2\Gamma \Delta t}). \end{aligned} \tag{S67}$$

This suggests that we must make one additional assumption, which is that the factor $\hat{a}_{11}/\hat{a}_{12}$ is not small. This is true whenever the flux is small, as

$$\langle \phi_n \phi_m \rangle_{\rho_0} = -\frac{2}{\lambda_n - \lambda_m} \langle (\mathbf{J} \cdot \nabla \phi_n) \phi_m \rangle_{\rho_0}. \tag{S68}$$

Similar to the equilibrium case, we see that at the first transition the encoder depends only on f_1 , giving (S47). In contrast to the equilibrium case, there may be a small correction due to the second eigenfunction, however this becomes exponentially small for long times Δt .

Note that in this calculation, $\phi_0(x)$ is constant (following from the assumption of a non-degenerate eigenvalue at 0) so that the $f_0(h)$ factors can be absorbed into $p(h)$. This changes in the case of a degenerate ground state, which corresponds a situation where there are decoupled sectors in which the dynamics evolve independently. Then, each eigenfunction corresponding to the zero eigenvalue is piecewise constant on one of the independent sectors. The optimal encoder Eq. S56 will depend instead on $\phi_0(x)$, which identifies the independent sectors.

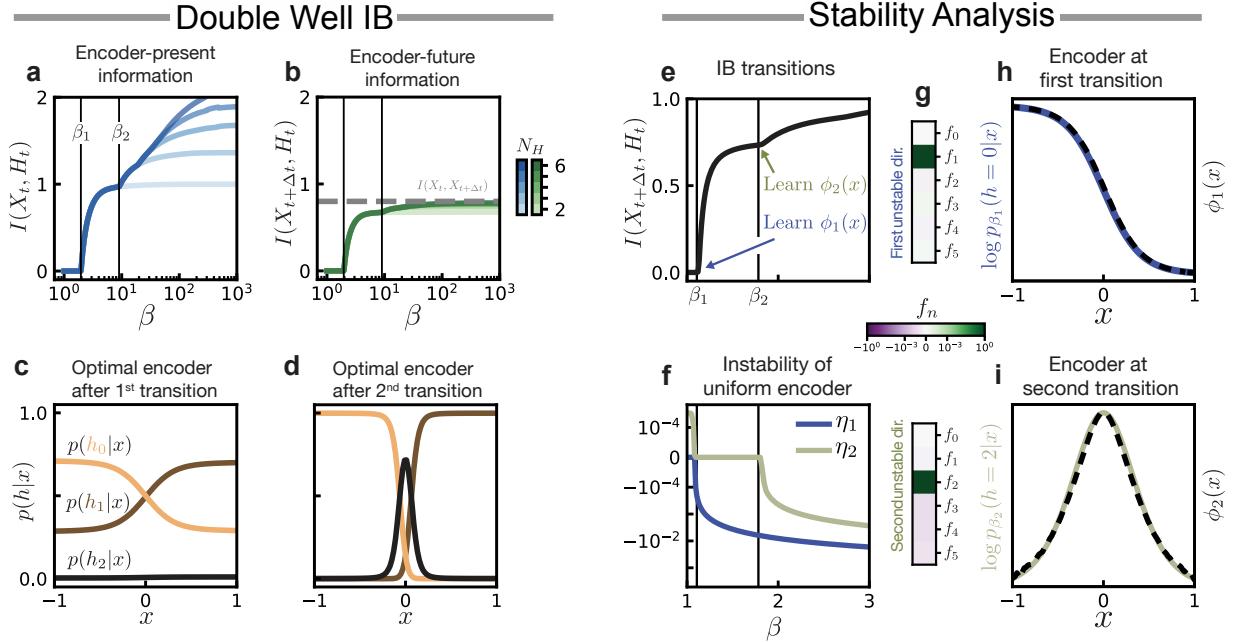


FIG. S5. Information bottleneck for a Brownian particle in a double well. (a) Mutual information between the current state X_t and its encoding H_t for various alphabet sizes N_H (color). The first two IB transitions are denoted with black lines and occur at β_1 and β_2 . (b) Mutual information between the current state's encoding H_t and the future $X_{t+\Delta t}$. The gray dashed line denotes the maximum attainable information, which is the mutual information $I(X_t, X_{t+\Delta t})$. Color denotes encoding alphabet size. Black lines showing IB transitions are shown for reference. (c) Optimal encoder after the first transition ($\beta \gtrsim \beta_1$) with alphabet size $N_H = 3$. (d) Optimal encoder after the second transition ($\beta \gtrsim \beta_2$) with alphabet size $N_H = 3$. (e) When β is increased, the IB objective favors less compression so that more and more information can go through the encoder. This occurs in steps, where the spectral content of the transfer operator is included starting from eigenvalues with largest magnitude (i.e., the slowest ones). (f) Transitions are characterized by the appearance of negative eigenvalues in the spectrum of the Hessian of the IB loss function. Here we consider the Hessian evaluated at the uniform encoder $p(h|x) = N_H^{-1}$. The IB transitions $\beta_1 \approx 1.1$ and $\beta_2 \approx 1.8$ correspond to the appearance of negative eigenvalues of the Hessian. (g) The unstable directions are dominated by single components (note the color scale is logarithmic). (h) At the first transition, the logarithm of the encoder is given by the eigenfunction $\phi_1(x)$, up to rescaling (y-axis is shown in arbitrary units). (i) Likewise, at the second transition the encoder is given by $\phi_2(x)$.

B. Numerical experiments

Here we numerically study the behavior of the encoder in the high-compression regime. By isolating its dependence on x_t , we may identify the most relevant features of the state variable and show that they coincide with left eigenfunctions of the transfer operator.

We consider the simple example of a particle in a double well considered in the main text, which we map to a discrete variable $H_t \in \{0, \dots, N_H - 1\}$. In this system the IB loss function can be optimized directly, as shown in Ref. [38], using the Blahut-Arimoto algorithm [58] described above.

To focus on the properties of encodings for varying degrees of compression β , we consider a fixed set of dynamical parameters μ and σ . Increasing β reduces the amount of compression, i.e. “widens” the bottleneck, allowing more information to pass into the encoder. This leads to a series of IB transitions (Fig. S5a,b). The form of the optimal encoder changes qualitatively at these transitions. Before β_1 , the optimal encoder has no dependence on x so that $p(H_t = h_i | x_t) = \text{const}$ for all h_i . After the first transition, the encoder begins to associate regions of x to particular values of h (Fig. S5c,d). We are interested in the form of the encoder at $\beta \gtrsim \beta_1$, just above the first IB transition, as this reflects the *most informative* features of the full state variable x (Fig. S5e). The dependence of $p(h_t | x_t)$ on x can be explained by our stability analysis above. Stability is governed by the eigenvalues η_i of the Hessian of the IB Lagrangian with respect to the parameters $f_n(h_t)$. These parameters tell us how much the encoder “weights” each transfer operator eigenfunction; $f_n(h_t) = 0$ (for $n > 0$) corresponds to the uniform, or trivial encoder $p(h_t | x_t) = p(h_t)$.

For small β all eigenvalues η_i are positive, indicating that the uniform encoder is a stable minimum of the IB Lagrangian. In Fig. S5f we show the smallest two eigenvalues of the IB Hessian when evaluated at the uniform encoder. At the first transition one eigenvalue becomes negative, so that the uniform encoder is unstable. The eigenvector

corresponding to the unstable eigenvalue η_1 indicates how the weights $f_n(h_t)$ should be adjusted to lower the value of the IB Lagrangian. As expected from our analysis, these are given by a vector which isolates f_1 (Fig. S5g, top). By taking the logarithm of the encoder after the transition, we can independently confirm that the encoder depends only on $\phi_1(x)$ (Fig. S5h).

Our stability analysis predicts that a second mode becomes unstable at the second IB transition $\beta \approx \beta_2$. Here we see that this unstable mode selects f_2 , and that the encoder correspondingly gains dependence on $\phi_2(x)$ (Fig. S5i). Note that in general, η_2 must not necessarily become negative precisely at β_2 because the stability analysis is performed at the uniform encoder while the true optimal encoder has already deviated from uniformity. In Figure S6, we perform the same analysis for a triple-well potential where this difference is more apparent.

C. Stability analysis for continuous encodings

Here we compute the stability of the uniform encoder for continuous latent variables h , where we constrain $p(h|x)$ to be Gaussian. This is motivated by the fact that variational IB is performed with precisely such an Ansatz. For simplicity, we assume a fixed and isotropic standard deviation σ :

$$p(h|x) = N \exp \left[-\frac{1}{2\sigma^2} (h - \mu(x))^2 \right] \quad (\text{S69})$$

We further consider the following expansion for μ

$$\mu(x) = \sum_i m_i \phi_i(x). \quad (\text{S70})$$

Our goal is to study the stability of the IB objective as a function of the coefficients m_i , analogous to the section above where we consider variations of the coefficients f_i . The trivial encoder again corresponds to $m_i = 0$ for all i .

We start by computing

$$\frac{\partial}{\partial m_i} p(h|x) = \frac{-1}{\sigma^2} \phi_i(x) (h - \mu(x)) p(h|x) \quad (\text{S71})$$

which, at the uniform encoder, evaluates to

$$\frac{\partial}{\partial m_i} p(h|x) = \frac{-1}{\sigma^2} h \phi_i(x) p(h) \quad (\text{S72})$$

The compression term is given by

$$\begin{aligned} \partial_n \partial_m I(X, H) &= \int dx' dh \frac{p(x')}{p(h)} \partial_m p(h|x') \partial_n p(h|x') - \underbrace{\int dh \frac{1}{p(h)} \partial_m p(h) \partial_n p(h)}_{\partial_m \partial_n 1=0} \\ &= \frac{1}{\sigma^4} \int dx' dh p(x') \phi_n(x') \phi_m(x') h^2 p(h) \\ &= \frac{1}{\sigma^2} \langle \phi_n \phi_m \rangle. \end{aligned}$$

The Hessian of the prediction term is, similarly to the above calculation for discrete h , given by

$$\begin{aligned} \partial_n \partial_m I(Y, H) &= \int dy dh \frac{1}{p(y)p(h)} \left(\int dx p(y, x) \partial_m p(h|x) \right) \left(\int dx' p(y, x') \partial_n p(h|x') \right) \\ &= \frac{1}{\sigma^4} \int dy p(y) \left(\int dx \phi_m(x) p(x|y) \right) \left(\int dx' \phi_n(x') p(x'|y) \right) \int dh p(h) h^2 \\ &= \frac{1}{\sigma^2} \left\langle \langle \phi_m \rangle_{p(\cdot|y)} \langle \phi_n \rangle_{p(\cdot|y)} \right\rangle_y \end{aligned}$$

The Hessian of the full Lagrangian, given this Ansatz, is then

$$\partial_i \partial_j \mathcal{L}_{\text{IB}} = \frac{1}{\sigma^2} \left(\langle \phi_n \phi_m \rangle - \beta \left\langle \langle \phi_m \rangle_{p(\cdot|y)} \langle \phi_n \rangle_{p(\cdot|y)} \right\rangle_y \right) \quad (\text{S73})$$

This is exactly the form that A_{ij}^β takes above, without the $n, m = 0$ terms that come from the normalization term (with our Gaussian Ansatz the normalization term is unaffected and hence these terms do not appear). Note that for discrete encodings, all information about h was contained in the matrix $G_{\mu\nu}$, and this matrix did not affect the stability of the uniform encoder. Similarly, the prefactor here is nothing other than the inverse of the variance of the trivial encoder $p(h)$.

Encoder after the first transition

We now build on our result from above showing that at the first transition the m_1 direction becomes unstable. Assuming an Ansatz

$$p(h|x) = N \exp \left(\frac{-1}{2\sigma^2} (h - m_1 \phi_1(x))^2 \right), \quad (\text{S74})$$

we ask at what value of m_1 the IB objective obtains its minimum. To do so, we perform an asymptotic expansion of the objective in m_1 up to fourth order. The conditional distribution can be expanded as

$$p(h|x) = p_0(h) \left[1 + \frac{h}{\sigma^2} \phi_1(x) m_1 + \frac{(h^2 - \sigma^2)}{2\sigma^2} \phi_1^2(x) m_1^2 + \frac{h(h^2 - 3\sigma^2)}{6\sigma^2} \phi_1^3(x) m_1^3 + \frac{(h^4 - 6h^2\sigma^2 + 3\sigma^4)}{24\sigma^8} \phi_1^4(x) m_1^4 + \mathcal{O}(m_1^5) \right]$$

We compute the integral to check normalization:

$$\int p(h|x) dh = 1 + \frac{(\sigma^2 - \sigma^2)}{2\sigma^2} \phi_1^2(x) m_1^2 + \frac{(3\sigma^4 - 6\sigma^4 + 3\sigma^4)}{24\sigma^8} \phi_1^4(x) m_1^4 = 1$$

The compression term is

$$\begin{aligned} I(X, H) &= \int dx p(x) \frac{1}{2\sigma^2} (\phi_1(x) - \langle \phi_1 \rangle)^2 m_1^2 + \frac{1}{4\sigma^4} (2(\phi_1(x) - \langle \phi_1 \rangle)^2 + \langle \phi_1 \rangle^2 - \langle \phi_1^2 \rangle) (\langle \phi_1 \rangle^2 - \langle \phi_1^2 \rangle) m_1^4 \\ &= \frac{1}{2\sigma^2} (\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2) m_1^2 + \frac{1}{4\sigma^4} (\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2) (\langle \phi_1 \rangle^2 - \langle \phi_1^2 \rangle) m_1^4 \\ &= \left(\frac{\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2}{2\sigma^2} \right) m_1^2 - \left(\frac{\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2}{2\sigma^2} \right)^2 m_1^4 \end{aligned}$$

Note that the prefactors may be written

$$\left(\frac{\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2}{2\sigma^2} \right) = \frac{\text{Var}(\phi_1)}{2\sigma^2}$$

The fourth order term in the prediction term is

$$\begin{aligned} &= -\frac{1}{4\sigma^4} \int dy \left((\langle \phi \rangle_{p(\cdot|y)}^2 - \langle \phi^2 \rangle_{p(\cdot|y)})^2 - (\langle \phi \rangle^2 - \langle \phi^2 \rangle_{p(\cdot|y)})^2 + (\langle \phi^2 \rangle_{p(\cdot|y)} - \langle \phi^2 \rangle)^2 \right) m_1^4 \\ &= \frac{1}{4\sigma^4} \left(\left\langle (\langle \phi \rangle_{p(\cdot|y)}^2 - \langle \phi^2 \rangle_{p(\cdot|y)})^2 \right\rangle_y - (\langle \phi \rangle^2 - \langle \phi^2 \rangle)^2 \right) m_1^4 \\ &= \left(\left\langle \left(\frac{\langle \phi \rangle_{p(\cdot|y)}^2 - \langle \phi^2 \rangle_{p(\cdot|y)}}{2\sigma^2} \right)^2 \right\rangle_y - \left(\frac{\langle \phi \rangle^2 - \langle \phi^2 \rangle}{2\sigma^2} \right)^2 \right) m_1^4 \end{aligned}$$

While the second order term is of the same form as in $I(X, H)$

$$\left\langle \left(\frac{\langle \phi_1 \rangle_{p(\cdot|y)} - \langle \phi_1 \rangle}{2\sigma^2} \right)^2 \right\rangle_y m_1^2 \quad (\text{S75})$$

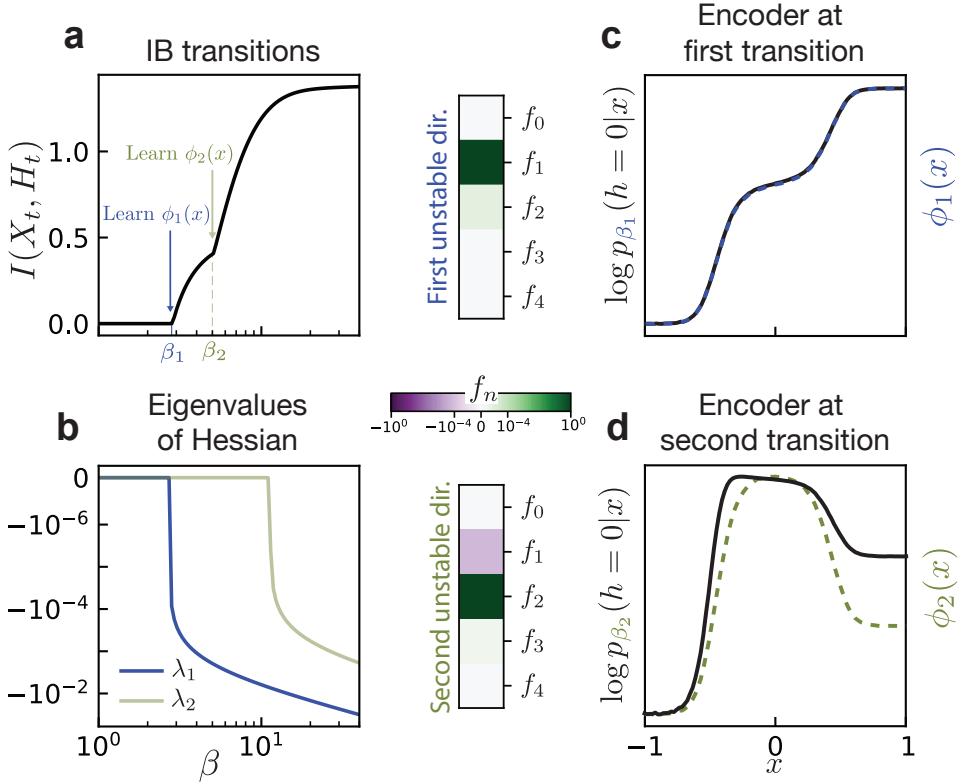


FIG. S6. **IB transitions for a Brownian particle in a triple-well potential.** (a) Information transitions with varying β for a particle in a triple-well potential. (b) Eigenvalues of the IB Hessian evaluated at the uniform encoder $p(h|x) = N_H^{-1}$. The appearance of an unstable direction at $\beta_1 \approx 3$ coincides with the first IB transition. The emergence of a second unstable direction doesn't correspond precisely to the IB transition at β_2 because stability is evaluated at the uniform encoder, but the true optimal encoder at β_2 is not uniform. (c) The unstable directions are dominated by single components (note the color scale is logarithmic). At the first transition, the logarithm of the encoder is given by the eigenfunction $\phi_1(x)$, up to rescaling (y -axis is shown in arbitrary units). (d) Likewise, at the second transition the encoder is given primarily by $\phi_2(x)$.

Put together, we find

$$\begin{aligned} \mathcal{L}_{\text{IB}} = & \frac{1}{2\sigma^2} \left[\left(\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2 \right) - \beta \left\langle \langle \phi_1 \rangle_{p(\cdot|y)}^2 - \langle \phi_1 \rangle^2 \right\rangle_y \right] m_1^2 \\ & - \frac{1}{(2\sigma^2)^2} \left[(1-\beta) \left(\langle \phi_1^2 \rangle - \langle \phi_1 \rangle^2 \right)^2 - \beta \left\langle \left(\langle \phi_1 \rangle_{p(\cdot|y)}^2 - \langle \phi_1 \rangle_{p(\cdot|y)}^2 \right)^2 \right\rangle_y \right] m_1^4 \end{aligned}$$

This can be neatly summarized as

$$\mathcal{L}_{\text{IB}} \propto 2\sigma^2 (\text{Var}(\phi_1) - \beta \text{Var}(\phi_1|y)) m_1^2 - \left((1-\beta) \text{Var}^2(\phi_1) - \beta \text{Var}^2(\phi_1|y) \right) m_1^4 \quad (\text{S76})$$

which has minima at

$$m_1 = \begin{cases} 0 & \text{if } \text{Var}(\phi_1) \geq \beta \text{Var}(\phi_1|y) \\ \sqrt{\frac{\sigma^2(\text{Var}(\phi_1) - \beta \text{Var}(\phi_1|y))}{(1-\beta)\text{Var}^2(\phi_1) - \beta \text{Var}^2(\phi_1|y)}} & \text{otherwise.} \end{cases} \quad (\text{S77})$$

After the first transition, both the numerator and denominator in the radical will become negative so that the entire quantity is real.

D. Limitations of the above results

As mentioned above, our result is exact only for systems (such as equilibrium systems) that satisfy

$$\langle \phi_n, \rho_0 \phi_m \rangle = \delta_{nm}.$$

Although our computation of the Hessian is still correct for non-equilibrium systems, it may become very far from a diagonal matrix in the presence of transient dynamics. Unless otherwise stated, we assume that the systems in this work are close to a steady state, so that the initial state x_t is sampled from the stationary distribution.

What happens when this assumption is violated, and we only seek to describe our system during some transient behavior? One such setting is when the system is prepared in a state far from the equilibrium so that it undergoes strong transient dynamics. In such regimes, pseudospectra may be more relevant than the operator's spectrum, and transfer operator eigenfunctions may cease to be a useful descriptor of the system.

In our theory, this deviation from a description in terms of the transfer operator spectrum is reflected in a change in structure in the Hessian, which is clearly illustrated in the case of a Brownian particle in a potential well in Fig. S7. This means that the features learned by IB will not coincide exactly with the operator's eigenfunctions. The Hessian is unstable with respect to deviations exactly in the direction of $\phi_1(x)$ only in the case that the overlap between eigenfunctions $\langle \phi_1(x), p(x)\phi_n(x) \rangle$ is small for $n \neq 1$. For equilibrium systems, this quantity is zero. However, if $p(x)$ is not the steady state but some other arbitrary distribution, this quantity may be far from zero. That is particularly clear in the potential well, where the eigenfunctions are nearly constant away from the center of the well. By choosing a distribution that is only supported in these regions where $\phi_1 \approx C_1$ and $\phi_n \approx C_n$, one can get large values of the overlap. The encoder learned by IB for transient dynamics is shown in Fig. S7, where we see that it strongly depends on x in the region of the transient. This shows that IB learns a relevant feature even in the case of transient dynamics.

E. “Distillation” of the optimal encoder

Here we elaborate on how to interpret the encoding variable h and its relation to the transfer operator eigenfunctions. As mentioned in the main text, in the limit of high compression (small β), the encoder is given by a small deviation δp from the trivial encoder $p(h)$,

$$p(h_t|x_t) = p(h_t)(1 + \delta p(h_t|x_t)). \quad (\text{S78})$$

In this regime the dynamics of h_t are dominated by noise. One can see this by considering

$$p(h_{t+\Delta t}, h_t) = \int dx_{t+\Delta t} dx_t p(h_{t+\Delta t}, x_{t+\Delta t}, x_t|h_t)p(h_t) \quad (\text{S79})$$

$$= \int dx_{t+\Delta t} dx_t p(h_{t+\Delta t}|x_{t+\Delta t})p(x_{t+\Delta t}|x_t)p(x_t|h_t)p(h_t) \quad (\text{S80})$$

$$= p(h_{t+\Delta t})p(h_t)(1 + \mathcal{O}(\delta p^2)), \quad (\text{S81})$$

which shows that the variables $h_{t+\Delta t}$ and h_t are nearly statistically independent of one another. This means that h may still be a nearly completely random variable; it is the small deviations from randomness that interest us.

How can we extract the “interesting”, predictive part of the encoder? We are not interested in the variable h itself, but rather how it couples to the variable x . To isolate this, we disregard the noise by taking the weak coupling we discover with IB, and effectively letting the coupling become strong. In practice, this is achieved by simply taking the most likely state for a particular value of x_t :

$$h_t(x_t) = \arg \max_h p(h|x_t).$$

For variational (continuous) IB this is straightforward: rather than showing the latent variable after “reparameterization” (which amounts to adding Gaussian noise; see Section 4C and Section 5) we show the variable before noise has been added. The resulting h_t is given by the mean (and, because we assume Gaussianity, the mode) of the distribution. For exact IB this is also straightforward because we have access to the full encoder $p(h|x)$, so we simply take the most-likely discrete state for each x . We note that this is equivalent to fixing a (small) dimensionality and taking large β , which is what was done for the discrete problems in the main text.

For continuous variables learned in VIB this procedure directly delivers the transfer function eigenfunctions. For discrete IB this is not the case, however, because the learned variable and the eigenfunction live in different spaces:

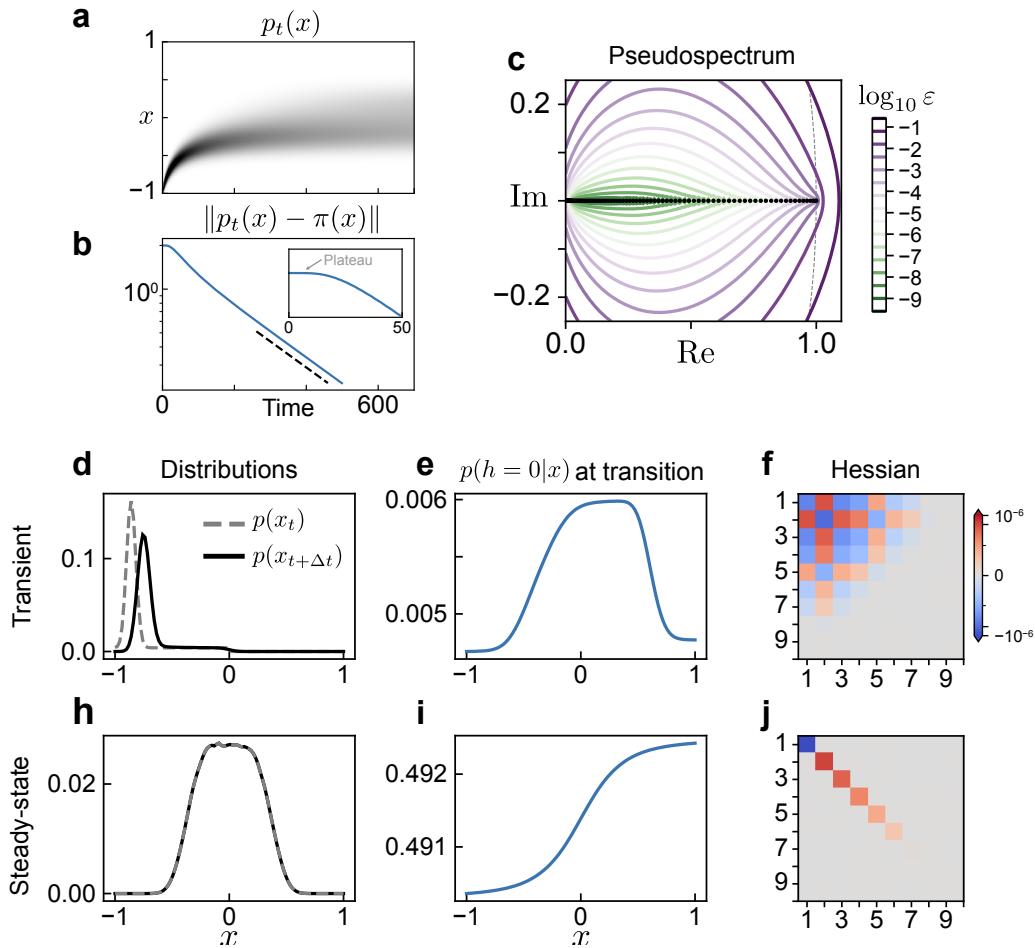


FIG. S7. **IB for transient dynamics** (a) Evolution of an initial distribution of particles prepared in a state highly concentrated state near $x \approx -1$ in a potential well centered around $x = 0$. (b) Relaxation of the distribution towards the steady state distribution $\pi(x)$. For long times, this relaxation occurs on a time scale given by the subleading eigenvalue λ_1 of the transfer operator (dashed line has slope λ_1). For short times however (inset), the dynamics has nothing to do with this eigenvalue and instead exhibits a “cut-off” phenomenon [55] where the distance from the steady state is constant until some time $t \approx 20$ where the distance begins to decay. (c) Pseudospectrum of the transfer operator in (a,b), shown as level sets of the resolvent. (d) Initial and final distribution of transient dynamics. (e) IB-optimal encoder at the first IB transition for the transient dynamics in (d). (f) Hessian of the IB objective function parameterized in terms of coefficients for transfer operator eigenfunctions (as in SI). (h-i) Corresponding results for IB performed on steady-state dynamics.

for example, $h_t \in \{1, \dots, N_H\}$ while $\phi(x_t) \in L^\infty(\mathbb{R}^n)$. In order to obtain the eigenfunctions from these encodings (as presented in e.g. Figs. S5 and S6), one must effectively extract the conditional probability. In the Blahut Arimoto algorithm we use, we have direct access to this quantity. If one instead only has samples of h_t it may be approximated as the mean $p(h_t|x_t) = \mathbb{E}[h_t|x_t]$.

The above distillation procedure is closely related to the deterministic information bottleneck [43], however the two procedures are not identical, as noted in Ref. [43]. In short, deterministic IB yields a delta-distributed encoder

$$p(h|x_t) = \delta(h - h^*(x_t))$$

with

$$h_{\text{DIB}}^*(x_t) = \underset{h}{\operatorname{argmax}} [\log p_{\text{DIB}}(h) - \beta D_{\text{KL}}(p(x_{t+\Delta t}|x) \| p_{\text{DIB}}(x_{t+\Delta t}|h))]. \quad (\text{S82})$$

where the distributions $p_{\text{DIB}}(h)$ and $p_{\text{DIB}}(x_{t+\Delta t}|h)$ are determined self-consistently with the definition of $h_{\text{DIB}}^*(x_t)$.

Our distillation procedure is similar, but uses the IB-optimal encoder:

$$h_{\text{IB}, \text{distilled}}^*(x_t) = \underset{h}{\operatorname{argmax}} \log \left(\frac{p_{\text{IB}}(h)}{\mathcal{N}(x)} \exp [-\beta D_{\text{KL}}(p(x_{t+\Delta t}|x) \| p_{\text{IB}}(x_{t+\Delta t}|h))] \right) \quad (\text{S83})$$

$$= \underset{h}{\operatorname{argmax}} [\log p_{\text{IB}}(h) - \beta D_{\text{KL}}(p(x_{t+\Delta t}|x) \| p_{\text{IB}}(x_{t+\Delta t}|h))] \quad (\text{S84})$$

Here, the distributions $p_{\text{IB}}(h)$ and $p_{\text{IB}}(x_{t+\Delta t}|h)$ are not determined self-consistently in relation to $h_{\text{IB}, \text{distilled}}^*(x_t)$, but instead from the IB objective. Hence, the solutions may differ in practice.

V. VARIATIONAL IB

Exactly solving the IB problem in principle requires access to the full distribution $p(x_t, x_{t+\Delta t})$. One way around this is via so-called Deep Variational IB as introduced in Ref. [41] which optimizes an upper bound on the IB objective

$$\mathcal{L}_{\text{IB}} = I(X_t; H_t) - \beta I(X_{t+\Delta t}; H_t). \quad (\text{S85})$$

To bound the first term, [41] introduces a variational ansatz for the marginal $\hat{p}(h_t)$. It follows from positivity of the Kullback-Leibler divergence $D_{\text{KL}}(p(h_t) \| \hat{p}(h_t))$ that

$$\int dh_t p(h_t) \log p(h_t) \geq \int dh_t p(h_t) \log \hat{p}(h_t), \quad (\text{S86})$$

and hence

$$I(X_t, H_t) \leq \int dx_t dh_t p(h_t|x_t) p(x_t) \log \frac{p(h_t|x_t)}{\hat{p}(h_t)} \quad (\text{S87})$$

$$= \mathbb{E}_{X_t} D_{\text{KL}}(p(h_t|x_t) \| \hat{p}(h_t)). \quad (\text{S88})$$

If $p(h_t|x_t)$ and $\hat{p}(h_t)$ are chosen properly, the Kullback-Leibler can be expressed analytically which enables gradients to be effectively computed. As in Ref. [41], we take a Gaussian ansatz for $p(h_t|x_t)$ and let the marginal $\hat{p}(h_t)$ be a spherical unit-variance Gaussian. More concretely, encoded variables H_t are sampled from $p(h_t|x_t)$ by computing

$$h_t = f_W(x_t) + \sigma_W(x_t)\eta, \quad (\text{S89})$$

where f_W and σ_W are deterministic functions modeled by neural networks with parameters (weights) W , and η is a Gaussian random variable with unit variance.

To bound the entire loss from above, we must bound $I(X_{t+\Delta t}, H_t)$ from below. We do this using the noise-contrastive estimate of the mutual information introduced in Ref. [111]. This recasts the problem as one of distinguishing samples from the distributions $p(x_{t+\Delta t}|h_t)$ and $p(x_{t+\Delta t})$. Given a batch of B pairs $(x_{t+\Delta t}, h_t)$ and one particular sampled value $h_t^{(i)}$, one asks what the probability is that a sampled value $x_{t+\Delta t}^{(j)}$ is from $p(x_{t+\Delta t}|h_t^{(i)})$ (is a *positive* sample) and not $p(x_{t+\Delta t})$ (is a *negative* sample). This probability is

$$p(x_{t+\Delta t}^{(j)} = \text{pos} | h_t^{(i)}) = \frac{\frac{p(x_{t+\Delta t}^{(j)} | h_t^{(i)})}{p(x_{t+\Delta t}^{(i)})}}{\sum_k^B \frac{p(x_{t+\Delta t}^{(k)} | h_t^{(i)})}{p(x_{t+\Delta t}^{(k)})}}, \quad (\text{S90})$$

where i is the index of the positive sample. The log likelihood of these probabilities,

$$\mathbb{E}[-\log p(x_{t+\Delta t}^{(j)} = \text{pos} | h_t^{(i)})], \quad (\text{S91})$$

where the expectation is taken over indices i and j , is closely related to the mutual information $I(X_{t+\Delta t}, H_t)$: in the limit of infinite samples $B \rightarrow \infty$, this quantity is given by

$$-\int dx_{t+\Delta t} dh_t p(h_t) p(x_{t+\Delta t}|h_t) \log p(x_{t+\Delta t} = \text{pos} | h_t) = \log B - I(X_{t+\Delta t}, H_t). \quad (\text{S92})$$

If one had access to the probabilities $p(x_{t+\Delta t}^{(j)} = \text{pos} | h_t^{(i)})$ appearing in Eq. S91, the mutual information could thus be easily determined. One attempts to estimate these probabilities by introducing a variational ansatz $f(x_{t+\Delta t}, h_t)$ to

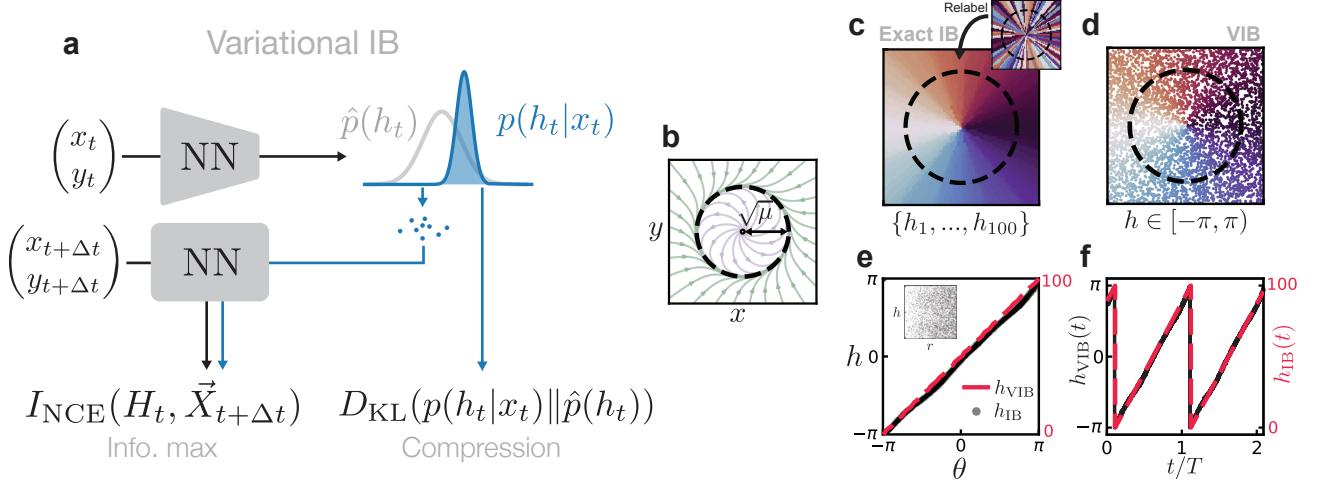


FIG. S8. **Variational IB extends exact IB.** (a) Variational IB optimizes the variational objective directly from samples, in contrast to exact IB which requires an estimation of the full conditional distribution $p(x_{t+\Delta t}|x_t)$. Values of h_t are drawn from a “latent” distribution $p(h|x)$, from which one can estimate the mutual information $I(X_{t+\Delta t}, H_t)$. The compression term is approximated by the Kullback-Leibler divergence between the learned $p(h|x)$ and a variational ansatz for the marginal $\hat{p}(h)$. (b) Phase portrait for a dynamical system above a Hopf bifurcation. (c) Exact IB learns (up to permutations) an encoding h which corresponds to the polar angle coordinate θ . (d) Variational IB similarly learns the correct encoding, directly from samples. (e) Correspondence between IB encodings and the angle coordinate θ . The encoding learned by VIB is independent of r (inset). (f) Time dependence of the encoded variable h_t .

approximate the density ratio $\frac{p(x_{t+\Delta t}|h_t)}{p(x_{t+\Delta t})}$. Typically this f is represented by a neural network. One can then obtain a bound on the mutual information by minimizing

$$\mathcal{L}_B = \mathbb{E} \left[-\log \frac{f(x_{t+\Delta t}^{(i)}|h_t^{(i)})}{\sum_k^B f(x_{t+\Delta t}^{(k)}|h_t^{(i)})} \right], \quad (\text{S93})$$

from which the InfoNCE estimate of the mutual information can be calculated as

$$I_{\text{NCE}}(X_{t+\Delta t}, H_t) = \log B - \mathcal{L}_B \leq I_{\text{true}}(X_{t+\Delta t}, H_t). \quad (\text{S94})$$

The full objective to be minimized is given by

$$\mathcal{L}_{\text{VIB}} = D_{KL}(p(H_t|X_t) \| \hat{p}(H_t)) - \beta I_{\text{NCE}}(X_{t+\Delta t}; H_t), \quad (\text{S95})$$

an illustration can be seen in Fig. S8. This loss is evaluated on batches of sample pairs $\{(x_t^{(1)}, x_{t+\Delta t}^{(1)}), (x_t^{(2)}, x_{t+\Delta t}^{(2)}), \dots\}$. A minimum is found via stochastic gradient descent. Note that other variational loss functions inspired by the information bottleneck have been derived, for example [47].

The VIB loss Eq. S95 is very similar to the loss function for a time-lagged β -variational autoencoder (β -VAE) [112], which have a form $\mathcal{L} = D_{KL} - \mathcal{L}_{\text{rec}}$, where the reconstruction term \mathcal{L}_{rec} measures the deviation from the true future state $X_{t+\Delta t}$ and the reconstruction from the latent variable H_t . In the VIB, this term is replaced by the mutual information I_{NCE} : rather than searching for a latent variable which can reconstruct the While a standard β -VAE with mean squared error (MSE) loss finds a latent variable that can reconstruct the full state at a time Δt in the future, the VIB merely tries to reconstruct the statistics of $X_{t+\Delta t}$ conditioned on X_t .

Replacing reconstruction losses with information-theoretical loss functions makes sense in some scenarios, such as chaotic systems. Recent work in this direction has used the Kullback-Leibler divergence as a loss function in place of a L_2 loss which generated well-behaved long-term dynamics [113, 114]. Other metrics which penalize the number of “false neighbors” in latent space have also shown to improve performance for chaotic systems [115].

The extent to which our findings might apply to time-lagged VAEs with L_2 reconstruction losses is an interesting question for future work. In Fig. S18, we observe that time-lagged VAEs learn a similar latent variable as VIB for the simulated fluids dataset. Indeed, several works have noted the apparent similarity between modes learned by variational and regular autoencoders with linear methods such as principal component analysis (PCA) [26, 116, 117].

VIB verification for noisy Hopf oscillator

We now show that VIB learns an encoding consistent with exact IB, i.e. one which depends only on the dominant transfer operator eigenfunctions. We consider the example of a Hopf oscillator given by the dynamical system

$$\begin{aligned}\dot{r} &= r(\mu - r^2), \\ \dot{\theta} &= \omega.\end{aligned}\tag{S96}$$

Equation (S96) is the normal form (in polar coordinates) for dynamics near a Hopf bifurcation [118]. For $\mu > 0$ this system exhibits a circular limit cycle of radius $\sqrt{\mu}$ (Fig. S8b). In Section B we show that this system has infinite purely imaginary eigenvalues $\lambda_n = i\omega$ and that as a result, encoders which exactly encode the angle coordinate, $p(h|r, \theta) \propto \delta(h - \theta)$, are solutions of the IB optimization problem.

The exact IB calculation breaks down for perfectly deterministic dynamics, hence we add a small amount of white noise to the dynamics. While this slightly perturbs the spectral content of the transfer operator, we still find our results to be consistent with those expected for the deterministic case (see Section B). As we increase the size of the encoding alphabet N_H , the encoder partitions space by finer and finer angular wedges. The learned encoding is invariant under permutations of the encoded symbols. Upon reordering, we find that, for large alphabets $N_H \gg 1$, the encoder indeed approximates $p(h|r, \theta) \propto \delta(h - \theta)$ as expected (Fig. S8c).

Using VIB, we learn a continuous h_t which can be computed directly from samples of the state variable. The encoding h_t learned by VIB closely approximates the angle coordinate (Fig. S8e-f) and is nearly uncorrelated with r (inset). This shows that our mathematical results illustrated for exact IB in Sec. IV hold also in the approximate framework of VIB.

VIB for deterministic dynamics

IB is an inherently probabilistic framework. To handle deterministic dynamics, we inject noise by introducing a stochastic sampling scheme. This step is necessary only for deterministically simulated dynamics; in our experience experimental data is sufficiently noisy on its own. Concretely, rather than taking as our IB relevance variable $Y = X_{t+\Delta t}$, we take $Y = X_{t+\Delta t+\eta}$ where η is a random uniformly-distributed time shift. Despite using a different relevance variable, this yields essentially the same optimal encoding as (S47), where the eigenvalue $e^{\lambda_n \Delta t}$ is replaced by $\int d\eta p(\eta) \exp(\lambda_n(\Delta t + \eta))$. Crucially, the encoder retains its dependence on the transfer operator eigenfunctions $\phi_n(x)$ as before. In general, this need not be the case: selecting a new relevance variable changes the IB objective and will generically lead to a different encoder. Our choice of stochasticity which we introduce to the dynamics is chosen in such a way to preserve the form of the encoder.

We illustrate this with a prototypical example of deterministic chaotic dynamics, the Lorenz system [106]. In the steady state, the state variable of the Lorenz system resides on a chaotic attractor consisting of two “lobes” which encircle unstable fixed points (Fig. S9a). Eigenvalues λ_n as well as right eigenvectors $\phi_n(x)$ of the Perron-Frobenius operator were computed numerically using the Ulam method. The correspondence between $\phi_1(x)$ and h disappears as β is increased (Fig. S9b-d). The dynamics of h also become notably less “slow”, and are instead nearly constant with occasional large jumps (Fig. S9e). The encoding variable h learned by VIB is the slowest varying function of the state \mathbf{x} , decaying as $\exp(\lambda_1^{\text{Re}} t)$ where λ_1^{Re} is the real part of the first subleading eigenvalue of the transfer operator (here Perron-Frobenius operator), which we show below in Fig. S11. This shows that in order for h to be a valid slow variable, the compression term in Eq. (S95) is crucial. These results are consistent with those obtained for exact IB, where we showed that the encoder incorporates the slow modes at low beta (high compression).

VI. VIB FOR FLUID FLOW AROUND A CYLINDER

Form of the eigenfunctions

What do the true eigenfunctions look like in this system? Because it is well approximated by linear dynamics, eigenfunctions of the adjoint transfer operator are linear functions of the state variable,

$$\phi_n[\mathbf{v}] = \langle \mathbf{v}(\mathbf{x}), \mathbf{m}^{(n)}(\mathbf{x}) \rangle,\tag{S97}$$

where $\mathbf{m}^{(n)}$ is the n -th mode (often referred to as a *Koopman* mode [27]) and angled brackets denote integration over space. The true eigenfunction and corresponding modes can be computed via dynamic mode decomposition (DMD)

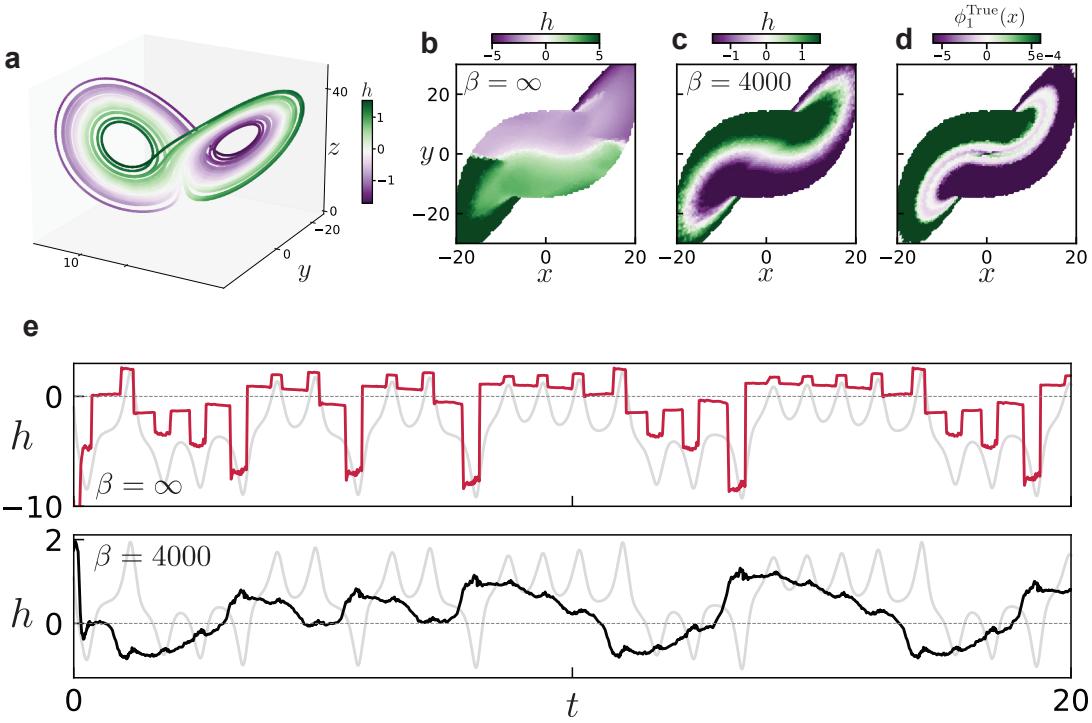


FIG. S9. VIB applied to the Lorenz-63 system (a) Illustration of the function found by VIB when allowed to learn a one-dimensional encoding. (b-c) Learned functions shown projected onto $x - y$, left is in the limit of low compression, while right is for high compression. (d) True eigenfunction, computed using the GAIO library [119]. (e) Time series of h in the high- and low-compression limits.

[15, 16], as described in the SI. The eigenfunctions for this system are in general complex, and come in conjugate pairs: $\phi_2(x) = \phi_1^*(x)$. In this situation any linear combination of ϕ_1 and ϕ_2 will decay at the same rate, and hence we expect to learn some arbitrary combination of the two dominant eigenfunctions, or equivalently a combination of the real and imaginary parts of ϕ_1 . We therefore take a two-dimensional encoding variable $[h_0, h_1]$, so that it can represent the full complex eigenfunction rather than only the real or imaginary part.

Our learned latent variables are oscillatory with the correct frequency as shown in Fig. 5b-c of the main text. A more stringent test is whether we are also learning the correct mode $\mathbf{m}^{(1)}$. From the true eigenfunctions, the modes can be extracted by computing the gradient

$$\frac{\partial \phi_n}{\partial v_j} = m_j^{(n)}. \quad (\text{S98})$$

As the learned function $h[\mathbf{v}]$ is a neural network, we can efficiently compute gradients of the network with respect to the input field

$$\frac{\partial h}{\partial v_j} = m_j^{(\text{IB})} + g_{\text{res},j}(\mathbf{v}(\mathbf{x})), \quad (\text{S99})$$

where we have separated the part of the gradient which is independent of \mathbf{v} from a residual part which is dependent on \mathbf{v} . If h corresponds to the true eigenfunction, we expect that $\mathbf{m}^{(\text{IB})}$ is approximately equal to the Koopman mode $\mathbf{m}^{(1)}$, and that \mathbf{g}_{res} is small. We indeed find this to be the case; Fig. 5d of the main text shows these gradients averaged over several instantiations of the neural network, which corresponds strongly to the true mode. Details concerning both the averaging procedure and the residuals \mathbf{g}_{res} can be found in the SI. This shows that variational IB not only recovers the essential oscillatory nature of the dynamics, but does so by learning the correct slowly varying functions of the state variable given by the adjoint transfer operator eigenfunctions.

Eigenfunction computation via dynamic mode decomposition

Here we provide some additional details on the computation used to generate components of Fig. 5 of the main text. The true Koopman modes were computed using dynamical mode decomposition (DMD) [15, 16, 120], also described below in Section X. DMD attempts to find a finite dimensional approximation of the Koopman operator using n snapshots of the system's state $x \in \mathbb{R}^d$ which are assembled into a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. One then attempts to find a linear evolution operator K which propagates the state forward in time

$$\mathbf{X}_{t+\Delta t} = K\mathbf{X}_t.$$

The approximate Koopman operator is given by the least squares solution $K = \mathbf{X}_{t+\Delta t}\mathbf{X}_t^+$ where \mathbf{X}_t^+ denotes the pseudo-inverse of \mathbf{X}_t . Approximate Koopman eigenfunctions are given by $\phi_n(x) = x \cdot w_n$ where w_n denotes the n th eigenvector of the matrix K and is known as the n th DMD mode, denoted by $m^{(n)}$ in the main text.

Details on gradient analysis

The fluid flow of a von Karmen vortex street is well approximated by linear dynamics, which can be understood by recognizing that the system is poised just after a Hopf bifurcation so that there is an angle coordinate which rotates with constant angular velocity. This constant rotation can be described by a linear dynamical system. It follows from linearity that eigenfunctions of the adjoint transfer operator are given by linear functions of the state variable,

$$\phi_n[\mathbf{v}] = \langle \mathbf{v}(\mathbf{x}), \mathbf{m}^{(n)}(\mathbf{x}) \rangle, \quad (\text{S100})$$

where $\mathbf{m}^{(n)}$ is the n -th “Koopman mode” and angled brackets denote integration over space. To compute these modes and the corresponding transfer operator spectrum, we use dynamic mode decomposition (DMD; see SI Section VI) [15, 16].

We allow VIB to learn a two-dimensional encoding variable $[h_0, h_1]$, so that it can learn the complete first eigenfunction rather than only the real or imaginary part. For the purposes of comparing our learned variable with ϕ_1 , we construct a complex $h = h_0 + ih_1$ out of the two learned components.

To understand which function has been learned by the neural network, we check whether the learned functions $h_i[\mathbf{v}]$ are of the form Eq. (S100). This can be done by examining gradients of the network with respect to the input field

$$\frac{\partial h}{\partial v_j} = m_j^{(\text{IB})} + g_{\text{res},j}(\mathbf{v}(\mathbf{x})) \quad (\text{S101})$$

where we have separated the part of the gradient which is independent of \mathbf{v} from a residual part which is dependent on \mathbf{v} . Gradients of the true eigenfunctions are given simply by

$$\frac{\partial \phi_n}{\partial v_j} = m_j^{(n)}. \quad (\text{S102})$$

We can then directly compare our latent variables with the true eigenfunctions by comparing their derivatives. If h corresponds to the true eigenfunction, we expect that $\mathbf{m}^{(\text{IB})}$ is approximately equal to the Koopman mode $\mathbf{m}^{(1)}$, and that \mathbf{g}_{res} is small. While it is unclear how to perform this decomposition in a general setting, we assume that the residual component $g_{\text{res},j}$ averages to zero over an oscillation period of the flow field \mathbf{v} . Then, $m_j^{(\text{IB})} \approx \langle \frac{\partial h}{\partial v_j} \rangle_t$ and $g_{\text{res},j}$ is given by variations about the mean. We see in Fig. S10 that these variations are much smaller than the mean in magnitude, and that they are essentially orthogonal to the mean vector. From this, we conclude that the gradients are given primarily by the constant part \mathbf{m} .

The learned latent functions vary for different training instances of the neural network. To extract the *average* gradient, we use PCA in an approach similar to that in Ref. [66]. The learned functions can have arbitrary sign structure; $h = h_0 + ih_1$ is just as likely to be learned as $h = -h_0 + ih_1$, for example. While in principle the network could learn arbitrary rotations, rather than simply changes in sign, we observe this is not the case. The distribution of gradients forms clusters in the high-dimensional gradient space corresponding to the four possible permutations of sign. PCA picks out the directions separating these clusters, as these are precisely the directions along which the data varies the most. This procedure gives an average gradient, while taking the varying sign structure into account. For reference, the gradient of a single instantiation of the VIB network can be seen in Fig. S18b.

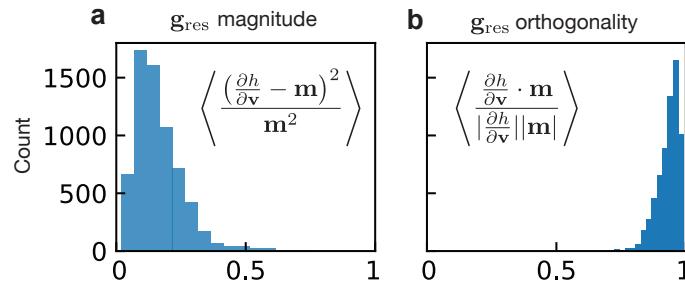


FIG. S10. **VIB gradients for fluid flow are nearly linear.** We compare the linear part of the gradients of the VIB network to the residual which depends on the input field $\mathbf{v}(\mathbf{x})$. (a) The residual parts are smaller in magnitude than the linear part \mathbf{m} . Angle brackets denote average over space. (b) The residual parts are nearly orthogonal to the mean \mathbf{m} , as we see that the projection of the full gradient onto the mean is nearly 1.

VII. CHAOTIC SYSTEMS

Chaotic systems exhibit many features which make them “difficult” to forecast. For example, they reside on strange attractors with fractal structure; even infinitesimal shifts in initial conditions lead to large deviations in a short time; and they are “mixing”, meaning that correlations decay to zero in time which is typically characteristic of stochastic systems. Although the goal of IB is compression (and not forecasting) we ask how the IB-optimal encoding reflects these properties. We do this in the contexts of the Lorenz-63 system [106], a hyperchaotic system [121], and the Kuramoto-Sivashinsky system [122, 123].

Fractal attractors

A unique feature of strange attractors is their fractal structure. While this is an interesting theoretical property, it is difficult to estimate from data in practice. The dominant methods for estimating fractal dimension from a time-series are little changed from those introduced nearly 50 years ago [124]. It is not known how to estimate the Hausdorff dimension of an attractor [125]; one typically relies on an estimate of the correlation dimension instead, which in many cases is close to the Hausdorff dimension [126].

One difficulty in the estimation of correlation dimension is that it requires the computation of distances between states of the system. In high-dimensional systems the effect of noise on measures of distance can become large: the ratio of distances between a point to its nearest and farthest neighbors approaches 1 [127, 128], which can lead to large corrections to the correlation dimension [129]. We will show that the encoding learned by VIB preserves the fractal dimension of the full system. This opens the door to using it in conjunction with denoising strategies to reliably determine the fractal dimension of very high-dimensional systems.

We consider two systems, a driven one-dimensional sine-Gordon (sG) equation and a one-dimensional Kuramoto-Sivashinsky (KS) equation. Both systems are described by time-evolving partial differential equations, and are hence high-dimensional. However, they exhibit low-dimensional dynamics. The sine-Gordon system is quasiperiodic and can thus be described by dynamics on a torus, while the KS system is chaotic and evolves on a low-dimensional strange attractor. We show in Fig. S12 and Fig. 6 of the main text that the fractal dimension of the respective *embedded* attractors is consistent with estimates of the full attractors. We find that the fractal dimension of attractor embedded in latent space coincides with that of the full system, which is much more computationally-expensive to estimate. This suggests that in the case where the true fractal dimension is unknown, it can be recovered by looking only at the dynamics in the encoding space.

Mixing dynamics

In terms of the transfer operator spectrum, a system is mixing if P has only $\lambda = 1$ in its point spectrum [130, 131], and otherwise consists only of a continuous spectrum (please see our comment below on “continuous spectra”). A more experimentally-relevant property of mixing systems is that correlations decay to zero $\langle A(t)B(0) \rangle - \langle A \rangle \langle B \rangle \xrightarrow{t \rightarrow \infty} 0$. In this context, the primary quantity of interest is the subleading eigenvalue of the transfer operator spectrum, as this bounds the rate of decay [132]. Here we ask whether the latent variables learned by VIB, which approximate the

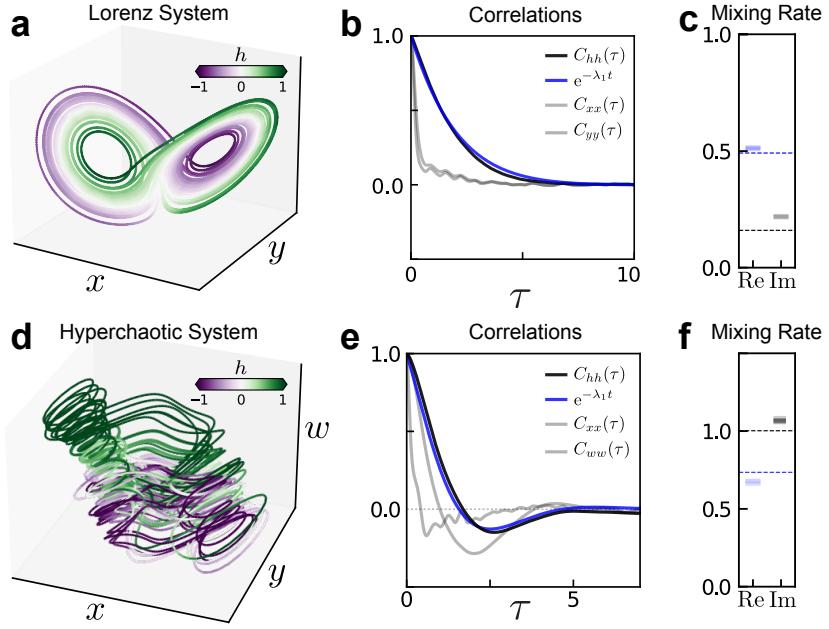


FIG. S11. **Mixing in chaotic systems**

By finding the dominant eigenfunction of the transfer function, one can estimate the mixing rate of dynamical systems which we demonstrate on (a-c) the Lorenz system and (d-f) a 4-dimensional hyperchaotic system. (a) Attractor of the Lorenz system colored by the learned encoding value. (b) Time autocorrelations for the encoding $C_{hh}(\tau) = \langle h(t)h(t+\tau) \rangle_t - \langle h(t) \rangle_t^2$ (black) compared to the mixing rate computed by estimating the full transfer operator (blue; see text). The encoding variable decays over a much slower time scale than the state variable x (gray). (c) Mixing rate estimated by fitting the blue curve in (b) to an exponential. We show the fitted real and imaginary parts of the decay rate separately. The true values are given by dashed lines of the respective color. (d-f) Show the analogous results for the hyperchaotic system mentioned in the text. Note that the full state is four-dimensional, so that in (d) we are only showing a projection onto three selected components.

subleading eigenfunction of the transfer operator, enable us to estimate mixing rates. As a demonstration, we consider the chaotic Lorenz-63 system as well as a hyperchaotic system governed by the equations of motion [121]:

$$\begin{aligned}\dot{x} &= \alpha(y - x) + w \\ \dot{y} &= x(\beta - z) - y \\ \dot{z} &= xy - \gamma z \\ \dot{w} &= \delta w + \frac{1}{2}y^2 - xz\end{aligned}$$

where $(\alpha, \beta, \gamma, \delta) = (10, 28, 2.7, 2.2)$.

To ensure that our latent variable isolates the first eigenfunction and does not include further subleading eigenfunctions, we restrict h to be one-dimensional and take a small compression parameter β . The results of the learned encoding are shown in Fig. S11a,d. The correlations of the latent variable decay on a slower time-scale than single components of the system Fig. S11b,e. For these relatively low-dimensional systems, we can independently approximate the dominant eigenvalue λ_1 using an Ulam approach, which we do using the software package GAIO [119]. We find good agreement between these eigenvalues and the decay rate estimated from a fit to the correlation function $C(\tau) = \langle h(t+\tau)h(t) \rangle - \langle h(t) \rangle^2$ where averages are taken over time Fig. S11c,f.

Exponential divergence of nearby trajectories

Exponential divergence of nearby trajectories is a fundamental issue when attempting to forecast chaotic systems [133]. For that reason, modern methods typically avoid training neural networks to forecast dynamics using mean-square error alone, but rather include higher-order information about the dynamics, such as Jacobians [134] or place emphasis instead on preserving long-term statistics of the system [98, 135]. However, the goal of our work is not to predict the full dynamics: neither at the level of the trajectories, nor at the statistical level. Instead our goal is to find an appropriate dimensionally-reduced representation of the state of the system.

In Fig. S12a we consider the Kuramoto-Sivashinsky model, which governs the chaotic evolution of a one-dimensional continuous field [122, 123]. Though the system itself is high-dimensional (depending on the coarseness of discretization of the continuous field) it lives on a low-dimensional attractor (Fig. S12b) [136, 137]. We see that the top Lyapunov exponent, which characterizes the exponential divergence of nearby trajectories, is preserved by VIB when the dimensionality of the latent space is sufficient to fully encode the system’s attractor Fig. S12c. This fact is not necessarily surprising; as long as the encoding is a smooth transformation of points on the system’s attractor, this will generally be true. To see this, recall the definition of the top Lyapunov exponent [138]

$$\lambda_x = \lim_{t \rightarrow \infty} \frac{1}{t} \log \|\delta x(t)\|$$

where the small perturbation evolves according to $\dot{\delta x}_i = \frac{\partial f_i}{\partial x_j} \delta x_j$, with f governing the system’s evolution $\dot{x} = f(x)$. Under a transformation $y = g(x)$, a small perturbation δx yields the perturbation $\delta y_i = \frac{\partial g_i}{\partial x_j} \delta x_j$, or more concisely, $\delta y = D_x g \cdot \delta x$. Hence, the Lyapunov exponent estimated using the y variables is given by

$$\lambda_y = \lim_{t \rightarrow \infty} \frac{1}{t} \log \|\delta y(t)\| = \lim_{t \rightarrow \infty} \frac{1}{t} \log \|(D_x g) \delta x(t)\|$$

We assume that the attractor in x is bounded, and that the transformation g is sufficiently smooth and invertible so that the attractor in y is also bounded and $D_x g \leq C$ for some constant C . In addition, we will use the fact that $\|Ax\| \geq \sigma_{\min} \|x\|$ where σ_{\min} is the minimal singular value of A . The Lyapunov exponent can consequently be bounded

$$\lim_{t \rightarrow \infty} \frac{1}{t} \log \sigma_{\min} \leq \lambda_y - \lambda_x \leq \lim_{t \rightarrow \infty} \frac{1}{t} \log C$$

Assuming σ_{\min} and C are both finite, in the limit this implies $\lambda_y = \lambda_x$.

This result breaks down when the transformation is non-invertible, in which case we have $\sigma_{\min} \rightarrow 0$. So, while we expect the Lyapunov exponents to be captured as long as the encoding recovers the full attractor, for smaller dimensions the exponents may differ. This effect is demonstrated in Fig. S12d. The inset shows the evolution of a small perturbation $\langle |\delta h|(t) \rangle$ over time, where the average is taken over $\mathcal{O}(10^4)$ initial conditions. Different lines of the same color denote different instantiations of the same neural network (cross validation). For small dimensions, points which are distinct in the full system get mapped close together in the embedding space. In the case that they come from different trajectories, they may move apart very quickly (and could in principle even move in the opposite direction). This explains the rapid growth observed for small dimensions. For sufficiently high dimensional embeddings the exponent approaches the true value $\lambda \approx 0.048$, suggesting that the network has learned the full attractor. This demonstrates that the latent representation found by VIB retains the essential dynamical features of the full system.

Continuous spectra

In our work we take the perspective that noise is an unavoidable and essential feature of the system we aim to describe, which is reflected in our choice to invoke information theory and our focus on stochastic dynamical systems. While continuous spectra are relevant for the study of *deterministic* chaotic systems [27], in practice a system’s essential spectrum will shrink due to the presence of noise [33, 50]. This does not mean that continuous spectra are meaningless, just that the accompanying singularities are “smoothed.” Consider for example a Brownian particle in a double-well potential, as shown in the main text. In the deterministic limit, the system exhibits a continuous spectrum at the bifurcation $\mu = 0$ which is associated with critical slowing down and the algebraic decay of correlations [49]. However, the presence of noise renders the spectrum discrete for all values of μ . While the system still “slows down” near the deterministic bifurcation point, the massive qualitative change in the system’s dynamics is smoothed over [49].

VIII. CYANOBACTERIA EXPERIMENTS

The experimental parameters used in the cyanobacteria experiments are described in detail in [80]. In brief, the authors in [80] control the translation of the KaiA protein with a theophylline riboswitch, allowing them to tune the copy number of KaiA proteins in the bacteria by modulating the concentration of theophylline. The clock state of each individual bacteria is visualized with a fluorescent marker EYFP driven by the *kaiBC* promoter. Colonies are imaged once per hour. The full dataset consists of 5 videos, such as the one shown in Fig. S13, which each contains several colonies. For each video we isolate regions which are filled by bacteria at all times to eliminate the effect of exponential

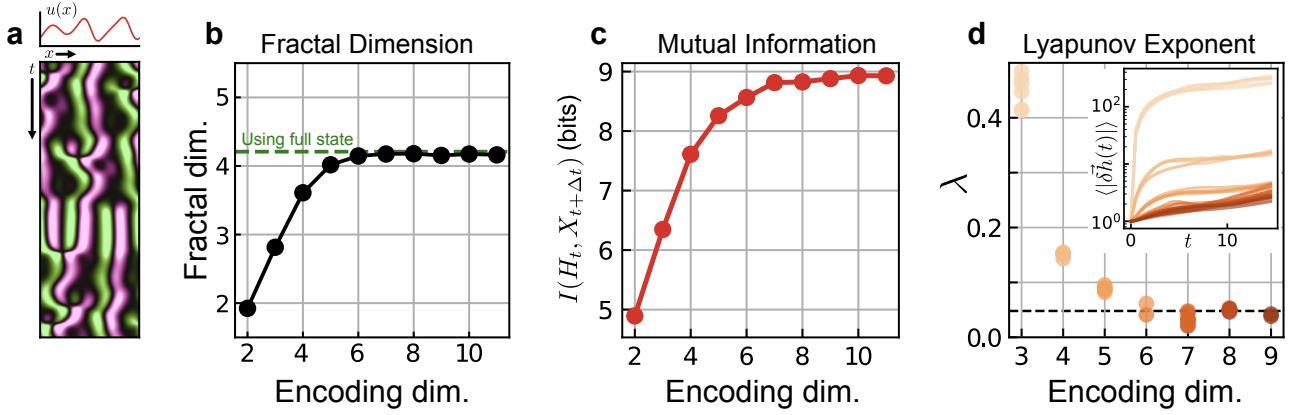


FIG. S12. **Fractal dimension of VIB encodings**

(a) Fractal dimension of the embedded Kuramoto-Sivashinsky attractor with varying encoding dimension. We estimate fractal dimension here by computing the correlation dimension [125, 139]. The gray dashed line shows the upper bound on the fractal dimension given by the Kaplan-Yorke conjecture [140, 141]. Green dashed line denotes the dimension of the attractor approximated for the full system, which lives in an \mathbb{R}^N , $N = 128$ discretization of the system's domain. **b:** Fractal dimension of the embedded system as a function of encoding dimension. Dashed line shows the fractal dimension of the system measured using the full state. **c:** Mutual information between the embedding and the future state, as a function of encoding dimension. **d:** At sufficiently high embedding dimension, encodings preserve the system's Lyapunov exponent. Dashed line shows the true exponent of $\lambda \approx 0.048$. Inset shows the distance $|\delta h|$ as a function of time for points which start close together (initial distance is normalized to one for comparison).

colony growth, which otherwise dominates the VIB results. Each latent trajectory shown in Fig. S13 as well as Fig. 6 in the main text corresponds to the trajectory of one single colony evolving under one of four theophylline conditions. There are 10 trajectories (colonies) in total, coming from 4 experimental conditions (colonies with radius smaller than 64 pixels are not considered). Figure S13 also shows the effect of the choice of time delay, which in the main text we take as $\tau = 3$ hr.

Synchronization measurements We measure the synchronization of the cyanobacterial oscillations using a metric inspired by a locally-coupled Kuramoto model which considers a spatially-varying phase field $\theta(\mathbf{x}, t)$. In terms of this field, an order parameter can be computed as

$$r(t)e^{i\psi(t)} = \frac{1}{V} \left| \int d\mathbf{x} e^{i\theta(\mathbf{x}, t)} \right| \quad (\text{S103})$$

where V refers to the volume being integrated over. The value of $r(t)$ is referred to as the synchronization order parameter, while $\psi(t)$ is the average phase.

To calculate the field $\theta(\mathbf{x}, t)$ from an intensity field $I(\mathbf{x}, t)$ we imagine that the intensity field represents one component of the complex phase, for example $I(\mathbf{x}, t) = \sin(\theta(\mathbf{x}, t))$. The other component can be accessed using a time-delay, $\cos(\theta(\mathbf{x}, t)) = \cos(\theta(\mathbf{x}, t + \tau)) = I(\mathbf{x}, t + \tau) =$, where τ here should be chosen so that the intensity field undergoes one quarter of a full oscillation. As we know the true period of the circadian cycle is 24 hours/frames, we choose a delay of $\tau = 8$ frames. Then, the phase can be computed as

$$\theta(\mathbf{x}, t + \tau) \approx \arctan \frac{I(\mathbf{x}, t)}{I(\mathbf{x}, t + \tau)}.$$

In Fig. S16 we show the results of VIB when applied to a locally-coupled Kuramoto model. Here we learn the same latent features which undergo oscillatory dynamics of varying radius, where the radius corresponds to the synchronization order parameter.

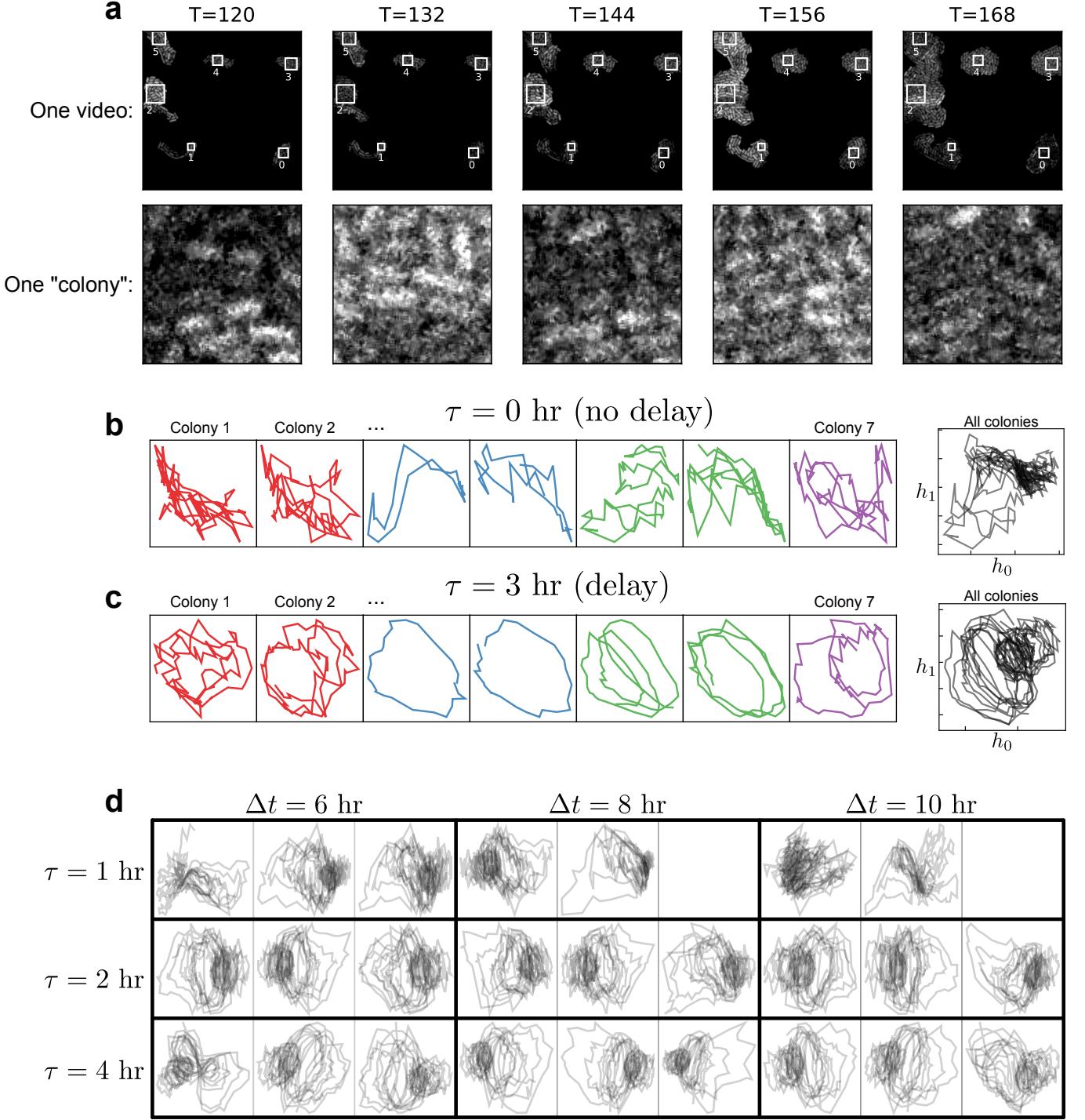


FIG. S13. Cyanobacteria data and choice of time delay (a) The cyanobacteria dataset consists of 5 videos, where each has multiple (up to 3) colonies. (Top) time evolution of one video, where the interiors of the colonies are outlined with white boxes. The dataset is composed of all colonies which are greater than 64px in height and width. (Bottom) evolution in time of one selected colony. (b) Learned VIB encodings when no time delay is used, i.e. $\tau = 0$. Left shows the evolution of (a subset of) individual colonies, colored by the original experimental video which they belong to. Note that the axes scale is adjusted for each trajectory so that it fills the plot. Right panel shows all colonies in latent space. (c) Same as above, but with the 3 hr time delay used in the main text. (d) Hyperparameter sweep over different choices of time delay τ and prediction horizon Δt . Here each set of three plots was trained with the same parameters but different neural network instantiations to understand how robust these latent spaces are.

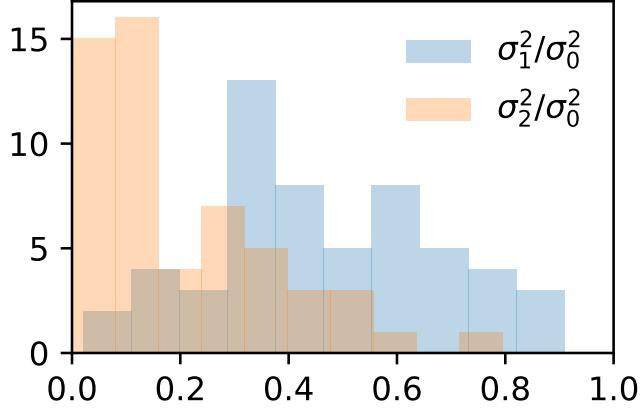


FIG. S14. **Variance along principal component directions** For each DVIB model trained on the cyanobacteria dataset, we compute the principal component decomposition of the resulting point cloud in latent space. Here we show the variance along each principal component for 60 instantiations of the DVIB model. Variances are normalized by the variance in the first principal component direction.

IX. SIMULATION PARAMETERS

Triple well simulations

For the dynamics of the triple well we work directly with the force $F = -\partial_x U$,

$$F(x) = \frac{-1}{200} (9375x^5 - 7500x^3 + 1100x - 20).$$

The evolution of the Brownian particle's position x_t is given by

$$dx_t = \frac{F(x_t)}{\gamma} dt + \sigma \sqrt{dt} \eta_t \quad (\text{S104})$$

where η_t is white noise with unit variance. The noise magnitude σ is related to the diffusion constant in Eq. S10 by $D = \frac{\sigma^2}{2}$. We use $\sigma = 1.0$ and $\gamma = 0.2$.

Simulations were performed with $N_{\text{init}} = 10^5$ initial conditions, with 300 trajectories generated from each initial condition. The state is evolved for 100 steps at $dt = 2 \cdot 10^{-3}$. The transfer matrix is approximated by binning the space $x \in [-1, 1]$ with $N_{\text{bins}} = 100$ bins. IB is performed using a time delay $\Delta t = 64$ steps.

Pitchfork bifurcation simulations

The system was evolved according to the stochastic differential equation Eq. S104 with $F(x) = -\mu x - x^3$, $\gamma = 1$ and $\sigma = 0.1$. For each value of μ , 10^5 initial conditions were simulated with 2000 trajectories starting at each initial condition. These were evolved for 100 time steps of $dt = 2 \cdot 10^{-3}$.

Hopf oscillator simulations

The system was evolved according to Eq. S104 with the force given by Eq. S96 above (transformed into cartesian coordinates), with $\gamma = 1$ and $\sigma = 10^{-2}$, $\omega = 4.0$ and $\mu = 0.25$. For each value of μ , 10^6 initial conditions were simulated with 1000 trajectories starting at each initial condition. These were evolved for 50 time steps of $dt = 2 \cdot 10^{-3}$.

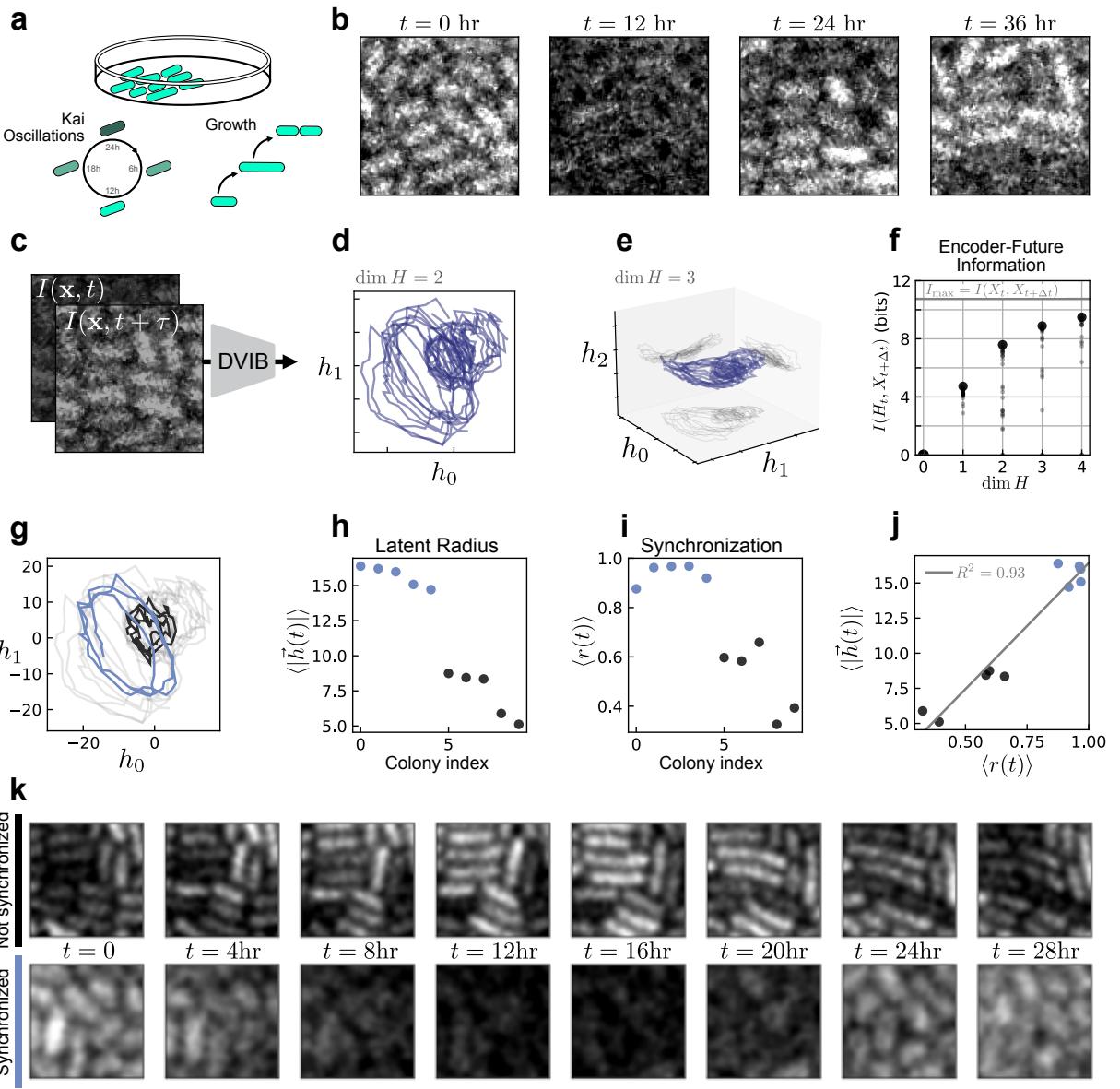


FIG. S15. Discovering slow collective variables in cyanobacteria populations (a) Fluorescent images of cyanobacteria colonies labeled with EYFP driven by the *kaiBC* promoter, allowing the visualization of Kai protein transcription. The colonies are imaged as they undergo cell growth and oscillations in Kai expression associated with the circadian rhythm. (b) Time series of one cropped section of an individual colony (see SI Fig. S9 for details). (c) The “state” used for variational IB is the time-lagged intensity field with lag time τ . (d) Variational IB embeddings of time-lagged images into two dimensions (see Supplementary Movie 2). Every line corresponds to one colony’s evolution in time. Note the apparent oscillations of different radii. (e) Embedding into three dimensions. We orient our axes to correspond to the three principal components of the data. Thus, the projection onto the $h_2 = 0$ plane corresponds to a projection along the dominant two principal components. In this subspace, we see a similar structure as in the 2D embedding. (f) Mutual information between the future state and the encoding, given by $I(X_{t+\Delta t}, H_t)$, for varying dimension of the latent space. Increasing the dimension of the embedding space beyond two leads only to marginal increases in $I(X_{t+\Delta t}, H_t)$; this tells us “when to stop.” Each small point represents one training instance of the variational IB model, while the large point shows the maximum estimated value. Because the InfoNCE estimator is a lower bound on the true information, we consider the maximum as the estimated mutual information. The value of $I_{\max} = I(X_t, X_{t+\Delta t})$ is the mutual information estimated for the true dynamics. (g) Selected trajectories in latent space with large and small radii. (h) Time-average latent radius of all colonies. (i) Mean synchronization order parameter of the intensity images; see SI for computation details. (j) Mean radius versus synchronization parameter for each colony. VIB identifies clusters of cells characterized by high (blue) or low (black) synchronization. As revealed to us after our analysis, this clustering corresponds to differing theophylline concentrations across experiments. Within each experimental movie (each of which contains 2-3 colonies) the radii are mostly constant. (k) Sample time series of weakly (black) and highly (blue) synchronized colonies. We apply a slight Gaussian blur to better visualize the bacteria boundaries.

Lorenz system simulations

The system was evolved according to

$$\dot{x} = \sigma(y - x) \quad (\text{S105})$$

$$\dot{y} = x(\rho - z) - y \quad (\text{S106})$$

$$\dot{z} = xy - \beta z \quad (\text{S107})$$

with $\rho = 28$, $\beta = 8/3$ and $\sigma = 10$. We took 10^3 initial conditions from which trajectories were simulated for 10^5 steps with $dt = 2 \cdot 10^{-3}$. We compute the true eigenfunctions using the GAIO library [119].

Driven sine-Gordon simulations

The equation of motion of the continuous field $\phi(x)$ is [78]

$$\partial_t^2 \phi - \partial_x^2 \phi + \sin \phi = \varepsilon[-\alpha \partial_t \phi + \Gamma \cos(\omega t)]. \quad (\text{S108})$$

The system is periodic in space, and was simulated with initial conditions

$$\phi(x, t)|_{t=0} = 4 \arctan \left[\frac{\sqrt{1 - \omega_{\text{br}}^2}}{\omega_{\text{br}} \cosh \left(\sqrt{1 - \omega_{\text{br}}^2} (x - L/2) \right)} \right] \quad (\text{S109})$$

$$\partial_t \phi(x, t)|_{t=0} = 0 \quad (\text{S110})$$

where $\omega_{\text{br}} = 0.77$ is the frequency of the breather initial condition and L is the system size. Key simulation parameters are $dt = 10^{-2}$, $L = 24$ (discretized onto $N_{\text{bins}} = 256$ bins), $\varepsilon\alpha = 0.04$, $\varepsilon\Gamma = 0.105$, and $\omega = 0.87$ (see Ref. [78]).

Kuramoto-Sivashinsky simulations

The Kuramoto Sivashinsky model describes the evolution of a one-dimensional field $u(x)$ according to

$$\partial_t u = -\partial_x^4 u - \partial_x^2 u - u \partial_x u \quad (\text{S111})$$

Key simulation parameters are the system size $L = 22$, time resolution $dt = 2 \cdot 10^{-3}$, total simulation time $T = 300$, and spatial resolution $N_{\text{bins}} = 128$.

Hyperchaotic system simulations

As described above, the system was evolved according to [121]

$$\dot{x} = \alpha(y - x) + w$$

$$\dot{y} = x(\beta - z) - y$$

$$\dot{z} = xy - \gamma z$$

$$\dot{w} = \delta w + \frac{1}{2}y^2 - xz$$

where $(\alpha, \beta, \gamma, \delta) = (10, 28, 2.7, 2.2)$. We took 10^2 initial conditions, which were allowed a burn-in time of $t_{\text{burn-in}} = 1000$ steps to reach the attractor, after which they were simulated for 10^5 steps with $dt = 10^{-2}$.

Fluid flow simulations

The fluid flow simulations simulations are contained in the ‘‘Cylinder in Crossflow’’ dataset, downloaded from Ref. [142]. We take the dataset at Reynolds number 150, and interpolate the velocity field from the unstructured 6569-node mesh to a regular grid of 300×150 pixels. The VIB networks are trained with $\beta = 10^7$ and a time buffer Δt chosen randomly between [4, 24] (see our comment on time randomization in ‘‘VIB for deterministic dynamics’’ in Section V).

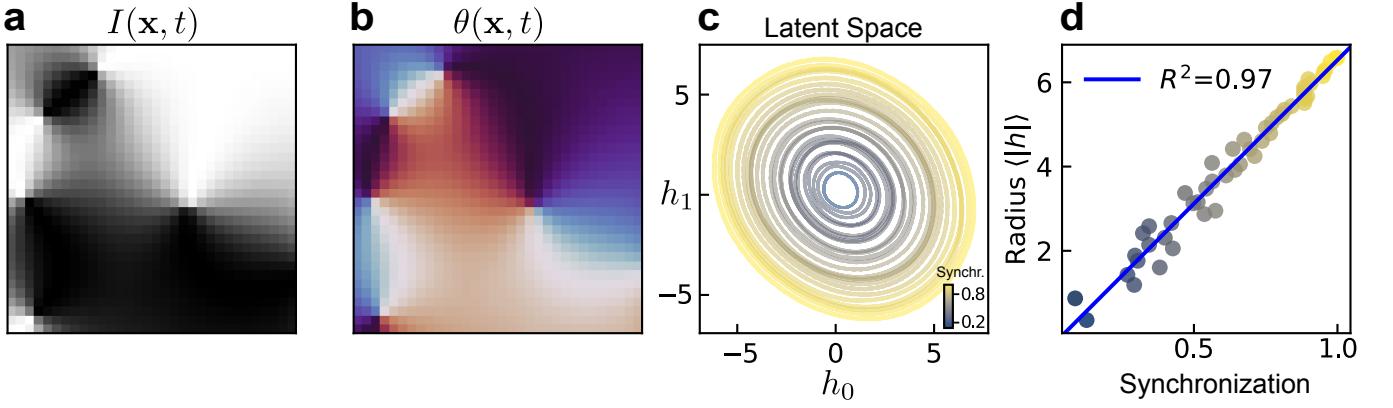


FIG. S16. VIB applied to simulated Kuramoto model Inputs are made to mimic the intensity field of the cyanobacteria data. The intensity field (a) is derived from the phase field (b) according to $I(\mathbf{x}, t) = \frac{1}{2} \cos \theta(\mathbf{x}, t) - 1$. (c) DVIB trained with two latent variables learns oscillations. Here each trajectory is colored by its synchronization order parameter. (d) Latent oscillations are very highly correlated with synchronization order parameter.

Locally-coupled Kuramoto model

In SI Fig. S16 we perform variational IB on a locally-coupled Kuramoto model and find the latent variables learn a synchronization order parameter as in the cyanobacteria data. This model considers a spatially-distributed field of phases which evolves according to

$$\partial_t \theta(\mathbf{x}_i, t) = \omega(\mathbf{x}_i) + J \sum_{j \in \mathcal{N}_i} \sin(\theta(\mathbf{x}_i, t) - \theta(\mathbf{x}_j, t))$$

where \mathbf{x}_i denotes the location of gridpoint i and \mathcal{N}_i denotes the sites neighboring i . These dynamics are integrated using the Euler method, where we take $\omega(\mathbf{x}) = \text{const} = 0.05$ for the natural frequencies, and $J = 1.0$.

X. COMPARISON OF VARIOUS DIMENSIONALITY REDUCTION METHODS

Variational IB (VIB) is by no means the only numerical method for performing data-driven model reduction. In this section we compare the performance of variational IB (VIB) to several common data-driven model reduction or inference methods. VIB uses a neural network to identify the relevant variables that are most predictive of the future. This construction learns *dynamically* relevant variables in contrast to methods such as principal component analysis or diffusion maps (in the absence of additional stringent assumptions), and it finds potentially non-linear variables in an agnostic way without requiring a tailored library of non-linearities (such as extended dynamic mode decomposition). In addition to these factors, VIB produces physically well-defined relevant variables – transfer operator eigenfunctions – in contrast to other deep methods such as (variational) autoencoders. In Fig. S17 we compare the low-dimensional latent space trajectories of the reduced models obtained using several of these standard methods to those output by VIB. In the remainder of this section we discuss these methods in detail.

Principal Component Analysis/Proper Orthogonal Decomposition

Principal component analysis, or PCA, is a *linear* method that projects the data onto a subspace which accounts for the most variance in the dataset [56]. In the dynamical systems literature, PCA is also known as a proper orthogonal decomposition (POD) [143]. The starting point for PCA is a collection of *samples* $\mathbf{x}^{(k)}$, where each sample has N_{feat} different *features*: $\mathbf{x}^{(k)} \in \mathbb{R}^{N_{\text{feat}}}$. Given these samples, one computes a correlation matrix between every pair of features

$$C_{ij} = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} (x_i^{(n)} - \bar{x}_i)(x_j^{(n)} - \bar{x}_j).$$

The principal components of the data are eigenvectors of the covariance matrix; the dominant principal component determines the direction of most variation in the dataset. In practice, if many samples are present the correlation

matrix is expensive to compute. Writing the dataset as a matrix, $\mathbf{X} \in \mathbb{R}^{N_{\text{samples}} \times N_{\text{feat}}}$, computing the correlation matrix requires evaluating the matrix product $\mathbf{C} \propto \mathbf{X}^T \mathbf{X}$. Instead, a more computationally efficient way is to compute the singular value decomposition (SVD) of the data matrix,

$$\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T.$$

The principal components are given by columns of \mathbf{V} , and the singular values in the matrix Σ are the square root of the variance along each principal component direction. PCA's relation to SVD also means that one can interpret a reconstruction of \mathbf{X} using only k principal components as the optimal rank- k approximation of the dataset, where optimality is here defined with respect to the Frobenius norm.

Compared to VIB, the linearity assumptions underlying PCA can be viewed either as a benefit or a drawback. The benefit is that a linear projection can be intuitive to understand in terms of the original data, and that one can compute this projection quickly. The potential downside is that a linear projection may not be sufficient or appropriate for systems that are highly non-linear. In addition, PCA makes no reference to the dynamics of the system, unlike VIB. An example of where PCA can fail is shown in Fig. 3 of Ref. [91]. In words, imagine a particle is hopping in a 2D double-well potential, where the wells are centered at $x \pm 1$. If the vertical extent of the wells is very large ($|y| \gg 0$), then the dominant principal component will be along the y -direction, even though the interesting dynamics are the hops between wells, which happens in the x -direction.

Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) starts from the assumption that the system evolves linearly [15, 16]. Concretely, if $\mathbf{x}_t \in \mathbb{R}^{n_{\text{feat}}}$ denotes the observed quantity at one point in time, then one assumes dynamics given by

$$\mathbf{x}_{t+1} = \mathbf{A}_{\text{DMD}} \mathbf{x}_t.$$

The matrix \mathbf{A} is a finite-dimensional approximation to the Koopman operator [27].

To find the matrix \mathbf{A}_{DMD} , we start by assembling a collection of samples into a data matrix $\mathbf{X}_t \in \mathbb{R}^{N_{\text{samples}} \times N_{\text{feat}}}$. In addition to this, we assemble a time-shifted matrix \mathbf{X}_{t+1} of the same shape as \mathbf{X}_t . In the scenario where we only have one trajectory of duration T , each sample may correspond to the system at a measured time point (excluding the final time), so that

$$\mathbf{X}_t = \begin{pmatrix} -\mathbf{x}^{(1)} - \\ -\mathbf{x}^{(2)} - \\ \vdots \\ -\mathbf{x}^{(T-1)} - \end{pmatrix}, \quad \mathbf{X}_{t+1} = \begin{pmatrix} -\mathbf{x}^{(2)} - \\ -\mathbf{x}^{(3)} - \\ \vdots \\ -\mathbf{x}^{(T)} - \end{pmatrix}$$

The matrix \mathbf{A}_{DMD} is given by the least squares solution

$$\mathbf{A}_{\text{DMD}}^T = \mathbf{X}_t^+ \mathbf{X}_{t+1}$$

where \mathbf{M}^+ denotes the pseudoinverse of the matrix \mathbf{M} . Eigenvectors \mathbf{w} of the matrix \mathbf{A}_{DMD} are Koopman modes, which correspond to left eigenfunctions of the transfer operator. The evolution of these Koopman modes in time is given by the time-dependent amplitude $a_i(t) = \mathbf{x}_t \cdot \mathbf{w}$, where we take \mathbf{x}_t here to be a single measurement at time t .

Similar to PCA, DMD reduces the dimensionality of the dataset by finding a linear projection of the data onto a subspace of the full features space. However, a key difference between the two approaches is that DMD uses the system's dynamics to identify an “optimal” subspace, whereas PCA identifies a subspace based only on the steady-state distribution of the data.

DMD in its original formulation assumes that the Koopman operator linearly evolves the observed state variable. However, the true Koopman operator evolves arbitrary non-linear functions of the state variable forward in time. DMD has therefore been extended to account for non-linear functions through “extended DMD”, or eDMD [17]. eDMD augments the state vector with non-linear transformations of the state. As an example, a state vector $[-\mathbf{x}-]$ may be replaced by monomials $[-\mathbf{x}-, -\mathbf{x}^2-, -\mathbf{x}^3-]$ (where here we understand exponentiation as element-wise). Other approaches also exist, such as augmenting the state vector by several time-delayed state vectors [144]. Rather than constructing the full $d \times d$ matrix, where d is the dimension of the (possibly augmented) state, one can directly construct a low-rank approximation of K by computing the reduced-rank singular value decomposition (SVD) of the matrix K from the SVDs of \mathbf{X}_t and $\mathbf{X}_{t+\Delta t}$.

Similar to PCA, DMD is primarily limited in its assumption of linear dynamics. In some cases this can be resolved with eDMD, which requires that one identifies a suitable set of non-linear terms to account for potential non-linear eigenfunctions of the Koopman operator. It is unclear how to choose this set of functions in a generic setting, which constitutes the biggest disadvantage to VIB. The features for eDMD must be hand-selected, which in some cases may defeat the purpose of using it as a feature-learning tool. VIB is not subject to this restriction, and can learn relevant features directly from the data.

Independent Component Analysis

Independent component analysis (ICA) was originally formulated in the context of blind source separation, where a useful picture is the “cocktail party problem” [24]. Imagine you place a set of microphones in a room at a cocktail party; these microphones will record the combination of all the conversations happening at once. The goal of blind source separation is to find a way to, from the recorded signals, isolate the original conversations. Mathematically, ICA finds a solution to the equation

$$\mathbf{x} = \mathbf{A}_{\text{ICA}} \mathbf{s}. \quad (\text{S112})$$

Here, \mathbf{x} denotes the recorded signal, \mathbf{s} denotes the independent sources (conversations), and \mathbf{A}_{ICA} is the mixing matrix. The only measured quantity is the vector \mathbf{x} ; both the mixing matrix and the independent sources must be learned.

To find one independent component, we start with an initial guess $y = \mathbf{b}^T \mathbf{x}$ which, for a correct choice of \mathbf{b} , should be equal to some component s_i . To identify the correct \mathbf{b} we will use the fact that sums of independent variables are more Gaussian than the original variables themselves, which follows from the central limit theorem. By assumption, the observed signals are a linear combination of independent components, so y is also: $y = \mathbf{q}^T \mathbf{s}$. If multiple q_i are non-zero, this will be more non-Gaussian than if only one is non-zero. Thus, $y = s_i$ for the choice of \mathbf{b} for which y is maximally non-Gaussian [24].

Numerically, one optimizes the deviation from Gaussianity using an approximation of the “negentropy” of the distribution of y : $\mathcal{J}_{\text{neg}} = S(y_{\text{Gaussian}}) - S(y)$, where y_{Gaussian} is a normally distribution random variable and S is the Shannon entropy. The intuition for this formulation is that the \mathcal{J}_{neg} is minimized if y has a unit-Gaussian distribution, so that it serves as a metric for non-Gaussianity. In practice one uses an approximation for \mathcal{J}_{neg} , see Ref. [24] for details. The optimal \mathbf{b} can then be found by doing gradient ascent on this objective function.

On some level, independent component analysis (ICA) is similar to PCA in that it searches for a linear projection of the data onto a subspace of the feature space. In its basic implementation ICA, like PCA, does not incorporate dynamics in contrast to DMD or VIB. The essential difference between PCA and ICA is the assumption of independence of the components. In PCA, unless the data distribution is actually multivariate Gaussian, the components are unlikely to be independent. IB makes no assumptions about the independence of learned encoding variables with each other, only about their level of independence with the original state. We note that ICA can be understood as minimizing the mutual information between encoding variables, $I(H_1; H_2)$, compared to IB’s minimization of the mutual information with the original state [24]. Which objective is more desirable depends on the system at hand, but in general we do not expect these to lead to the same encodings.

Time-lagged independent component analysis

The leveraging of non-Gaussianity in ICA is required due to the fact that, if using whitened data ($\mathbf{z} = \mathbf{V}\mathbf{x}$ so that \mathbf{z} is zero mean, $\mathbb{E}[\mathbf{z}] = \mathbf{0}$, and unit variance, $\mathbb{E}[\mathbf{z}\mathbf{z}^T] = \mathbf{I}$), then the mixing matrix \mathbf{A}_{ICA} cannot be estimated if the components \mathbf{s} are normal Gaussian variables: any orthogonal matrix will satisfy (S112). For dynamical systems, however, one can get around this requirement of non-Gaussianity. In particular, one can use a time-lagged variable $\mathbf{z}_{t-\tau} = \mathbf{A}_{\text{ICA}} \mathbf{s}_{t-\tau}$ to compute the mixing matrix from the time-correlation matrix of the signal \mathbf{z}_t [24]. In particular, we have

$$\begin{aligned} \mathbb{E}[\mathbf{z}_t \mathbf{z}_{t-\tau}^T] &= \mathbf{A}_{\text{ICA}} \mathbb{E}[\mathbf{s}_t \mathbf{s}_{t-\tau}^T] \mathbf{A}_{\text{ICA}}^T \\ &= \mathbf{A}_{\text{ICA}} \mathbf{D} \mathbf{A}_{\text{ICA}}^T, \end{aligned}$$

where \mathbf{D} is a diagonal matrix. In going to the bottom line, we used the assumption that the components s_i are independent not just instantaneously, but also for a time lag τ . From here we can read off that the matrix \mathbf{A}_{ICA} is composed of eigenvectors of the correlation matrix of the whitened signal. When written in terms of the original (unwhitened) data \mathbf{x}_t and $\mathbf{x}_{t-\tau}$ it can be shown these eigenvectors are nothing other than the eigenvectors of the transfer matrix \mathbf{A}_{DMD} [91]. Thus, the two methods are equivalent.

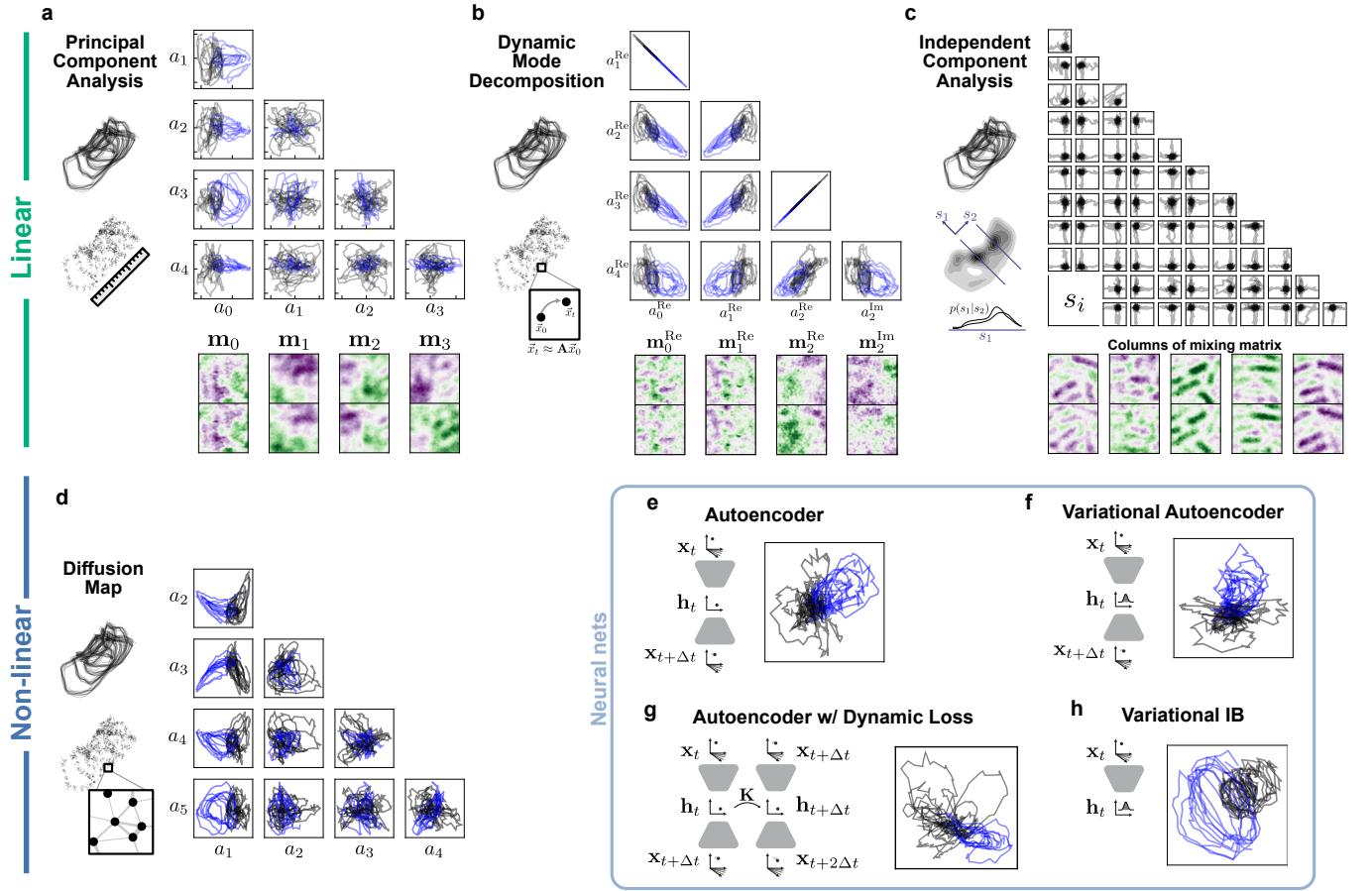


FIG. S17. Comparison of various dimensionality reduction techniques on cyanobacteria dynamics Top row illustrates linear methods, bottom row shows non-linear methods. For each method, we plot the trajectories of all cyanobacteria colonies when projected onto the dominant modes (whose definition depends on the particular method). Blue trajectories show the highly synchronized colonies corresponding to high theophylline concentration, black/gray trajectories show the unsynchronized colonies from low theophylline conditions. Unlike the first four methods, deep neural networks (bottom right) directly produce a two-dimensional latent variable and do not provide access to subleading modes. (a) Principal component analysis reduces the dimensionality of data by finding a projection onto the directions of most variance (sketch). Here we show the projection of the cyanobacteria trajectories onto several pairs of principal components, $a_i(t) = \mathbf{x}(t) \cdot \mathbf{m}_i$. Each curve corresponds to the trajectory of one single colony evolving under one of four theophylline conditions; there are 10 trajectories in total, coming from 4 experimental conditions. Below, the first four principal components are shown. Here, the state is composed of two time-lagged images of the cyanobacteria colony; the top half of \mathbf{m} corresponds to the earlier image, the bottom half corresponds to the lagged image. (b) Dynamic mode decomposition is performed on pairs of data points and aims to find a linear evolution operator \mathbf{A} that links them (sketch). (Right) Trajectories of the projection of the state onto several eigenvectors of \mathbf{A} . The eigenvectors themselves are shown below (the imaginary parts of the first two modes are uniformly zero). (c) Independent component analysis seeks a linear projection of the data onto statistically independent components \mathbf{s}_i . In other words, the distribution $p(s_i|s_j)$ is independent of s_j (sketch). (Right, top) Trajectories of the projection of the state onto several independent components. (Bottom) columns of the mixing matrix; the components s_i describe how the weighting of these columns evolves in time. (d) (Left) Diffusion maps build a graph of the observed state variables with edges weights determined by their distance from one another. (Right) Projection of the trajectories onto the first non-trivial eigenfunctions (because $a_0 = \text{const}$) of the inferred transfer operator on the graph. (e) A (time-lagged) autoencoder is a deep neural network architecture which encodes the state \mathbf{x}_t into a low-dimensional latent space. From the encoding one then tries to reconstruct the future state. (Right) Trajectories of the cyanobacterial colonies when encoded into a two-dimensional latent space. (f) A variational autoencoder is similar to a standard autoencoder, where one instead learns a distribution over possible encodings, $p(\mathbf{h}_t|\mathbf{x}_t)$. (Right) Trajectories of the cyanobacterial colonies when encoded into a two-dimensional latent space. (g) Implementation of the network in Ref. [28] which encodes the state using a time-lagged autoencoder with an additional loss term that aims to enforce linearity of the embedded dynamics, $\mathbf{h}_{t+\Delta t} \approx \mathbf{K}\mathbf{h}_t$. (h) Latent trajectories produced by variational IB (VIB).

Diffusion maps

Diffusion maps are a technique which attempts to approximate the Perron-Frobenius operator, or rather the integral kernel $p(x_{t+\Delta t}|x_t)$ [21, 145]. Given this approximation, one computes eigenfunctions and uses them as a low-dimensional parameterization of the data (“diffusion coordinates”).

This method takes data pairs $\{x_t^{(i)}, x_{t+\Delta t}^{(i)}\}_i$ and approximates the probability of observing these two points via a kernel $p(x_t^{(i)}, x_{t+\Delta t}^{(j)}) \approx k(x_t^{(i)}, x_{t+\Delta t}^{(j)})$, where one typically takes a Gaussian Ansatz

$$k_\epsilon(x, y) \propto \exp\left[\frac{-(x-y)^2}{\epsilon}\right].$$

From this, one can assemble the conditional probability distributions into a matrix

$$P_{ij} = p(x_{t+\Delta t}^{(j)}|x_t^{(i)}) = \frac{p(x_t^{(i)}, x_{t+\Delta t}^{(j)})}{p(x_t^{(i)})} = \frac{k(x_t^{(i)}, x_{t+\Delta t}^{(j)})}{\sum_j k(x_t^{(i)}, x_{t+\Delta t}^{(j)})}.$$

This matrix describes the evolution of probability distributions on a *graph* where each node is a data point. In practice, a symmetrized version of P is constructed and the learned eigenvectors are adjusted after the diagonalization [21, 145]. To compute the diffusion coordinates of an arbitrary point that wasn’t in the original dataset, one inverts the definition of the adjoint transfer operator eigenfunction

$$\phi_i(\mathbf{x}_{\text{new}}) \approx \frac{1}{\lambda_i} \sum_k P_{jk}^\dagger \phi_i(\mathbf{x}_k)$$

where λ_i is the i -th eigenvalue.

Diffusion maps have the advantage, relative to DMD, that they find the full Perron-Frobenius operator and not a linear approximation to it. However, while DMD can isolate the dominant eigenvectors of the operator using reduced-rank SVD, it is less clear how they can be extracted with diffusion maps without first computing the full matrix P . We note that VIB, like DMD, also directly learns the dominant modes and does not require estimation of the full transfer operator.

Deep (Variational) Autoencoders

Autoencoders (AEs) belong to a class of deep learning methods used for model reduction. Such approaches have successfully been applied in various domains for forecasting the dynamics of complex systems in terms of simpler latent dynamics [146, 147]. Autoencoders are composed of an encoder which compresses the observable \mathbf{x} into a lower-dimensional *latent* variable \mathbf{z} , and a decoder which attempts to reconstruct the original state \mathbf{x} . Variational autoencoders aim to learn a probability distribution over observations $p_\theta(\mathbf{x}) \approx p(\mathbf{x})$ from which one can directly sample [148]. This is done by assuming that the latent variable is low dimensional, and optimizing the objective

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] - \beta D_{\text{KL}}(q(z|x)\|\hat{p}(z))$$

where $q_\phi(z|x)$ is the posterior on the latent variables z and is parameterized by a neural network (encoder) with parameters ϕ , and $p_\theta(x|z)$ denotes the decoding network with parameters θ . Strictly speaking, we present in the above objective function the β -VAE loss [112]. With $\beta = 1$, which is the case for the original VAE [148], the objective is an upper bound on the log likelihood $-\mathbb{E}[\log p_\theta(x)]$ which is minimized when $p_\theta(x)$ is equal to the true data distribution $p(x)$. The term β controls compression as in the IB objective, however it has the opposite effect: small β_{IB} corresponds to high compression, while small β_{VAE} corresponds to low compression.

In cases where one directly computes the probabilities $p_\theta(x|z)$ the first term can be evaluated directly, else it is typically replaced with an L_2 loss $\|x - g_\theta(z)\|^2$ (where g_θ is a deterministic neural network), which is equivalent to assuming a Gaussian Ansatz for p_θ with a fixed variance.

The VIB loss function is very similar to the β -VAE loss [149]. Rather than attempting to reconstruct the original state x , VIB replaces this term with an estimate of the mutual information between the latent variable z and some other relevance variable y (for us, $y = x_{t+\Delta t}$). In contrast to DMD and diffusion maps, neither VIB nor β -VAEs make any mention of transfer operators and are instead motivated by purely statistical considerations. As we show in the main text, the latent variables learned by VIB correspond to eigenfunctions of the transfer operator. While we cannot claim that that β -VAEs learn the same thing, some preliminary results in Fig. S18 suggest they may coincide to some degree. In the cyanobacteria dataset, we see that the latent variables learned by a β -VAE show the same qualitative structure as in VIB, but are less smooth (Fig. S17).

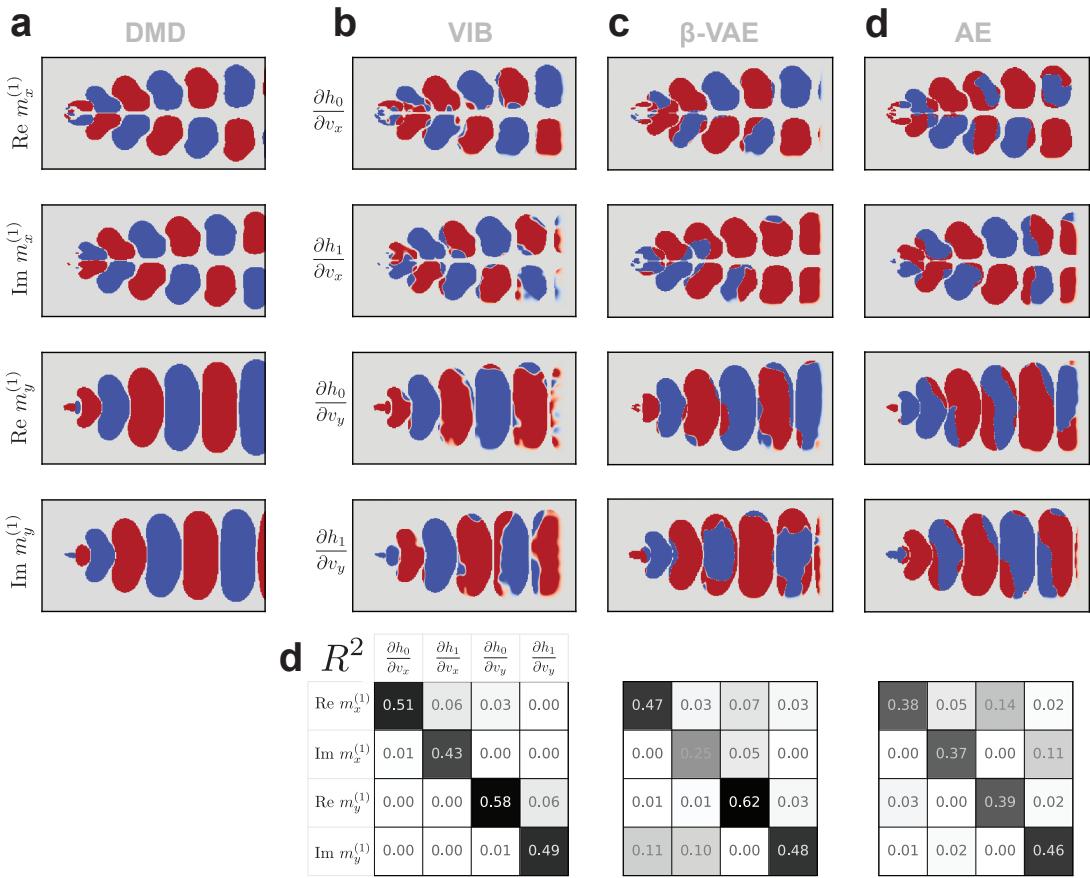


FIG. S18. **Learned latent variables for fluid flows with various deep neural networks.** To compare latent variables we compare their gradients as in Fig. 5 of the main text, which should correspond to Koopman modes. Here we show the gradients masked by regions in the DMD mode which have large amplitude. (a) True Koopman modes, with real and imaginary parts shown, obtained via DMD. (b) Corresponding modes for a VIB network. Matrix in the bottom row shows the R^2 values obtained by regressing the pixel values of the VIB gradients against the different components of the Koopman modes. (c) Corresponding modes for a β -variational autoencoder (VAE). The VAE has the same encoder structure as the VIB network, while the decoder has an inverted architecture to predict the future state $x_{t+\Delta t}$ from the latent variable h_t . Bottom row shows R^2 values, analogously to (b). (d) Corresponding modes for an autoencoder.

Other neural networks

Other deep architectures can also be used for model reduction. For example, Ref. [31] uses recurrent neural networks (RNNs) to learn the evolution of macroscopic variables. To ensure stability and fidelity, the macroscopic variables are periodically “lifted” to the full microscopic state, which is then evolved for several time steps to recalibrate the RNN’s hidden state. While interpretability of the latent variables has yet to be explored in such models, we expect the addition of VIB-like objective functions may aid interpretability without harming performance.

As another example, [28] attempts to learn the full Koopman operator using neural networks. This can be thought of as an extension of eDMD, where instead of prescribing the library of nonlinear terms by hand, they can be learned by a neural network. The latent dynamics are then encouraged to be linear by minimizing the deviation between the true future (latent) state $\mathbf{z}_{t+\Delta t}$ and its linear approximation found by least squares. Similar approaches have also been explored in Refs. [29, 30]. As an illustration, we trained one such network on the cyanobacteria which can be seen in Fig. S17. This approach was applied primarily to deterministic systems. We expect that combining their method of encouraging linear latent dynamics together with the VIB objective function may be fruitful and lead to more well-behaved latent variables.

Appendix A: Appendix: Details for the calculation of the Hessian of the IB Lagrangian

First order derivatives for $n \neq 0$ are given by

$$\partial_n^\mu p(h_\nu|x) \equiv \frac{\partial}{\partial f_n(h_\mu)} p(h_\nu|x) = p(h_\nu|x) \delta_{\mu\nu} \tilde{\phi}_n(x) - \frac{f_0^\nu \exp \left[\sum_n \tilde{\phi}_n(x) f_n^\nu \right]}{\left(\sum_{h_\lambda} f_0^\lambda \exp \left[\sum_n \tilde{\phi}_n(x) f_n^\lambda \right] \right)^2} \left[\sum_{h_\lambda} f_0^\lambda \exp \left[\sum_n \tilde{\phi}_n(x) f_n^\lambda \right] \tilde{\phi}_n \delta_{\mu\lambda} \right] \quad (\text{S1})$$

$$= p(h_\nu|x) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu|x)) \quad (\text{S2})$$

and for $n = 0$,

$$\partial_n^\mu p(h_\nu|x) = p(h_\nu|x) \frac{1}{f_0^\mu} (\delta_{\mu\nu} - p(h_\mu|x)). \quad (\text{S3})$$

We assign $\tilde{\phi}_0 = \frac{1}{f_0^\mu}$, taking care to track the correct Greek index which otherwise is not present for $\tilde{\phi}_n$. Subsequently,

$$\partial_n^\mu p(h_\nu) = \int dx p(x) p(h_\nu|x) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu|x)) \quad (\text{S4})$$

As it turns out, we do not need second-order derivatives of $p(h_\nu|x)$, as only first-order terms appear in the Hessian. In the following, for brevity we denote the marginals simply by $p(h_\lambda) = p_\lambda$.

We are now in a position to calculate the Hessian of the Lagrangian. We begin with the compression term,

$$\begin{aligned} \partial_n^\mu \partial_m^\nu I(X, H) &= \partial_n^\mu \partial_m^\nu \int dx \sum_{h_\lambda} p(h_\lambda|x) p(x) \log \left(\frac{p(h_\lambda|x)}{p(h_\lambda)} \right) \\ &= \int dx \sum_{h_\lambda} p(x) (\partial_n^\mu \partial_m^\nu p(h_\lambda|x)) \log \left(\frac{p(h_\lambda|x)}{p(h_\lambda)} \right) + p(x) \partial_m^\nu p(h_\lambda|x) \frac{\partial_n^\mu p(h_\lambda|x)}{p(h_\lambda|x)} - p(x) \partial_m^\nu p(h_\lambda|x) \frac{\partial_n^\mu p(h_\lambda)}{p(h_\lambda)} \\ &\quad + \int dx \sum_{h_\lambda} p(x) \partial_n^\mu p(h_\lambda|x) \frac{\partial_m^\nu p(h_\lambda|x)}{p(h_\lambda|x)} + p(x) p(h_\lambda|x) \frac{\partial_n^\mu \partial_m^\nu p(h_\lambda|x)}{p(h_\lambda|x)} - p(x) p(h_\lambda|x) \frac{\partial_n^\mu p(h_\lambda|x)}{p(h_\lambda|x)^2} \partial_m^\nu p(h_\lambda|x) \\ &\quad + \int dx \sum_{h_\lambda} p(x) \partial_n^\mu p(h_\lambda|x) \frac{\partial_m^\nu p_\lambda}{p_\lambda} + p(x) p(h_\lambda|x) \frac{\partial_n^\mu \partial_m^\nu p_\lambda}{p_\lambda} - p(x) p(h_\lambda|x) \frac{\partial_n^\mu p_\lambda}{p_\lambda^2} \partial_m^\nu p_\lambda \end{aligned}$$

by evaluating at the uniform encoder, many terms drop out

$$\begin{aligned} &= \int dx \sum_{h_\lambda} \frac{p(x)}{p_\lambda} [\partial_m^\nu p(h_\lambda|x) \partial_n^\mu p(h_\lambda|x) - \partial_m^\nu p(h_\lambda|x) \partial_n^\mu p(h_\lambda)] \\ &\quad + \int dx \sum_{h_\lambda} \frac{p(x)}{p_\lambda} [\partial_n^\mu p(h_\lambda|x) \partial_m^\nu p_\lambda - \partial_n^\mu p_\lambda \partial_m^\nu p_\lambda] \\ &= \int dx \sum_{h_\lambda} \frac{p(x)}{p_\lambda} \partial_m^\nu p(h_\lambda|x) \partial_n^\mu p(h_\lambda|x) - \sum_{h_\lambda} \frac{1}{p_\lambda} \partial_m^\nu p_\lambda \partial_n^\mu p_\lambda \end{aligned}$$

where for the vanishing second line we used that $\partial_n^\mu \sum_{h_\lambda} p(h_\lambda|x) = 0$ (and analogously for terms with a single p_λ), as well as the fact that $\sum_x p(x) \partial_n^\mu p(h_\lambda|x) = \partial_n^\mu p_\lambda$. We now substitute the expressions for the derivatives, evaluated at the uniform encoder,

$$\begin{aligned} \partial_n^\mu \partial_m^\nu I(X, H) &= \sum_{x, h_\lambda} \frac{p(x)}{p_\lambda} (p_\lambda \tilde{\phi}_m(x) (\delta_{\lambda\nu} - p_\nu)) (p_\lambda \tilde{\phi}_n(x) (\delta_{\lambda\mu} - p_\mu)) \\ &\quad - \sum_{h_\lambda} \frac{1}{p_\lambda} (p_\lambda \langle \tilde{\phi}_m(x) \rangle (\delta_{\lambda\nu} - p_\nu)) (p_\lambda \langle \tilde{\phi}_n(x) \rangle (\delta_{\lambda\mu} - p_\mu)) \\ &= (\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle) \sum_{h_\lambda} p_\lambda (\delta_{\lambda\nu} \delta_{\lambda\mu} - p_\nu \delta_{\lambda\mu} - p_\mu \delta_{\lambda\nu} + p_\nu p_\mu) \\ &= (\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle) (\delta_{\mu\nu} p_\mu - p_\nu p_\mu) \end{aligned} \quad (\text{S5})$$

We now perform the same calculation for the prediction term,

$$\begin{aligned}
\partial_n^\mu \partial_m^\nu I(Y, H) &= \partial_n^\mu \partial_m^\nu \sum_{y, h_\lambda} p(y, h_\lambda) \log \left(\frac{p(y, h_\lambda)}{p_\lambda p(y)} \right) \\
&= \int dy \sum_{h_\lambda} \partial_m^\nu p(y, h_\lambda) \left(\frac{\partial_n^\mu p(y, h_\lambda)}{p(y, h_\lambda)} - \frac{\partial_n^\mu p_\lambda}{p_\lambda} \right) + \partial_n^\mu \partial_m^\nu p(y, h_\lambda) \\
&= \int dy \sum_{h_\lambda} \frac{1}{p(y)p_\lambda} \partial_m^\nu p(y, h_\lambda) \partial_n^\mu p(y, h_\lambda) - \sum_{h_\lambda} \frac{1}{p_\lambda} \partial_m^\nu p_\lambda \partial_n^\mu p_\lambda \\
&= \int dy \sum_{h_\lambda} \frac{1}{p(y)p_\lambda} \left(\int dx p(y, x) \partial_m^\nu p(h_\lambda|x) \right) \left(\int dx' p(y, x') \partial_n^\mu p(h_\lambda|x') \right) - \sum_{h_\lambda} \frac{1}{p_\lambda} \partial_m^\nu p_\lambda \partial_n^\mu p_\lambda \\
&= \sum_{y, h_\lambda} \frac{1}{p(y)p_\lambda} \left(\int dx p(y, x) p_\lambda \tilde{\phi}_m(x) (\delta_{\lambda\nu} - p_\nu) \right) \left(\int dx' p(y, x') p_\lambda \tilde{\phi}_n(x') (\delta_{\lambda\mu} - p_\mu) \right) - \sum_{h_\lambda} \frac{1}{p_\lambda} \partial_m^\nu p_\lambda \partial_n^\mu p_\lambda \\
&= \left(\left\langle \frac{\langle \tilde{\phi}_n \rangle_{p(\cdot, y)} \langle \tilde{\phi}_m \rangle_{p(\cdot, y)}}{p(y)} \right\rangle_y - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) (\delta_{\mu\nu} p_\mu - p_\nu p_\mu)
\end{aligned}$$

Appendix B: Appendix: Optimal encoding for Hopf normal form dynamics

In Fig. S8 we show that IB learns the angle coordinate when applied to a dynamical system above a Hopf bifurcation. Here, we show that this is expected. We begin by deriving the adjoint transfer operator in polar coordinates, and then continue to solve for its eigenfunctions. In Cartesian coordinates, the equations of motion for the particle are

$$\dot{x} = (\mu - x^2 - y^2)x - \omega y \quad (\text{S1})$$

$$\dot{y} = (\mu - x^2 - y^2)y - \omega x \quad (\text{S2})$$

Expressed in polar coordinates, we have

$$f_x \hat{e}_x = f_x \cos \theta \hat{e}_r - f_x \sin \theta \hat{e}_\theta \quad (\text{S3})$$

$$f_y \hat{e}_y = f_y \sin \theta \hat{e}_r + f_y \cos \theta \hat{e}_\theta \quad (\text{S4})$$

From this we can compute $\mathcal{L}_K \phi = f_i \partial_i \phi$,

$$f_i \partial_i \phi = (\partial_r \phi)(\mu - r^2)r + \frac{1}{r} (\partial_\theta \phi) \omega r \quad (\text{S5})$$

$$= r(\mu - r^2)(\partial_r \phi) + \omega \partial_\theta \phi \quad (\text{S6})$$

Because this differential equation is separable, we can find eigenfunctions by looking for eigenfunctions that are a function of either r or θ . Recall that for deterministic dynamics, a product of adjoint transfer operator (or *Koopman* operator) eigenfunctions is again an eigenfunction [27]. The eigenvalue equation for the radial coordinate is solved in Ref. [49], and in Ref. [94] it is shown that for this system there is no globally valid Koopman decomposition. However we are only interested in a subset of eigenfunctions, in particular those with eigenvalue with real part $\text{Re}\lambda_i \approx 0$, which may not suffice to approximate arbitrary functions of the state variable.

For the angle coordinate alone the situation is much simpler,

$$\omega \partial_\theta \phi(\theta) = \lambda \phi(\theta)$$

which is solved by functions $\phi_n(\theta) = \exp\left(\frac{\lambda_n}{\omega}\theta\right)$. Periodicity requires $\frac{\lambda}{\omega}2\pi = i2\pi n$, which leads to $\lambda = in\omega$. The eigenfunctions and eigenvalues are then

$$\phi_n(\theta) = e^{in\theta}, \quad \lambda_n = in\omega.$$

The corresponding eigenfunctions of the Perron-Frobenius operator are given by $\rho = e^{-in\theta}$. We now consider an encoding $p(h|x)$.

$$p(h|r, \theta) \propto \exp \left\{ \sum_n^{\infty} e^{\lambda_n \Delta t} \phi_n(r, \theta) \int r' dr' d\theta' \rho_n(r', \theta') \log p(r', \theta' | h) \right\} \quad (\text{S7})$$

$$= \exp \left\{ \sum_n^{\infty} e^{in\omega \Delta t} \int r' dr' d\theta' e^{in(\theta - \theta')} \log p(r', \theta' | h) \right\} + \mathcal{O}(e^{\text{Re}\lambda_k \Delta t}) \quad (\text{S8})$$

where we retain only the eigenvalues with zero real part; the λ_k in the above refer to those eigenvalues with non-zero real part. After neglecting these terms, it can be seen directly that $p(h|r, \theta) = p(h|\theta)$. The coordinates (r', θ') are used to denote $(r_{t+\Delta t}, \theta_{t+\Delta t})$. The expression can be further simplified by replacing the sum over n with a delta function, which leads to

$$p(h|r, \theta) \propto \exp \left\{ \int r' dr' \log p(r', (\theta + \omega \Delta t) | h) \right\}. \quad (\text{S9})$$

We next ask whether an encoder of the form $p(h|\theta) \propto \delta(h - \theta)$ is a solution. Recall that the probability distribution appearing in (S9) is a distribution over *future* positions, $p(R_{t+\Delta t} = r', \Theta_{t+\Delta t} = \theta + \omega \Delta t | h)$. This distribution can be calculated as

$$\begin{aligned} p(r_{t+\Delta t}, \theta_{t+\Delta t} | h) &= \int r dr d\theta p(r_{t+\Delta t}, \theta_{t+\Delta t} | r_t, \theta_t) p(r, \theta_t | h) \\ &\propto \delta((\theta_{t+\Delta t} - \omega \Delta t) - h) \end{aligned}$$

where we used that the two terms in the integrand of the first equation are both delta functions. Plugging this into (S9) shows that $p(h|\theta) \propto \delta(h - \theta)$ is consistent, and hence is an optimal encoding. In the presence of noise, the eigenvectors are perturbed and may gain a dependence on r (see Fig. S19).

Appendix C: Appendix: Stability analysis of $I(H_t, H_{t+\Delta t})$

Here we compute the stability of the InfoMax objective. For notational simplicity we write $H_t \equiv H_0$, $H_{t+\Delta t} \equiv H_t$. The mutual information we will calculate the Hessian of is given by

$$I(H_0, H_t) = \int dh_0 dh_t p(h_0, h_t) \log \frac{p(h_0, h_t)}{p(h_0)p(h_t)}.$$

We want to compute the derivatives of this object with respect to f_n^μ , where

$$p(h_\mu | x) = \frac{p(h_\mu)}{N(x)} \exp \left[\sum_i \phi_i(x) f_i^\mu \right]$$

As a reminder, Greek indices correspond to indices of the discrete latent variable $h_\mu \in \{0, \dots, N_H - 1\}$ while latin indices correspond to eigenfunctions of the transfer operator, ordered by eigenvalue. From our computations above, we have

$$\begin{aligned} \partial_n^\mu p(h_\nu | x) &= p(h_\nu | x) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu | x)) \\ &= p(h_\nu) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu)) \end{aligned} \quad (\text{S1})$$

where for $n = 0$ we assign $\tilde{\phi}_0 = \frac{1}{f_0^\mu}$, taking care to track the correct Greek index which otherwise is not present for $\tilde{\phi}_0$. The second line is the derivative evaluated at the trivial encoder ($f_i = 0$ for all i). Similarly,

$$\begin{aligned} \partial_n^\mu p(h_\nu) &= \int dx p(x) p(h_\nu | x) \tilde{\phi}_n(x) (\delta_{\mu\nu} - p(h_\mu | x)) \\ &= p(h_\nu) (\delta_{\mu\nu} - p(h_\mu)) \delta_{n0} \end{aligned} \quad (\text{S2})$$

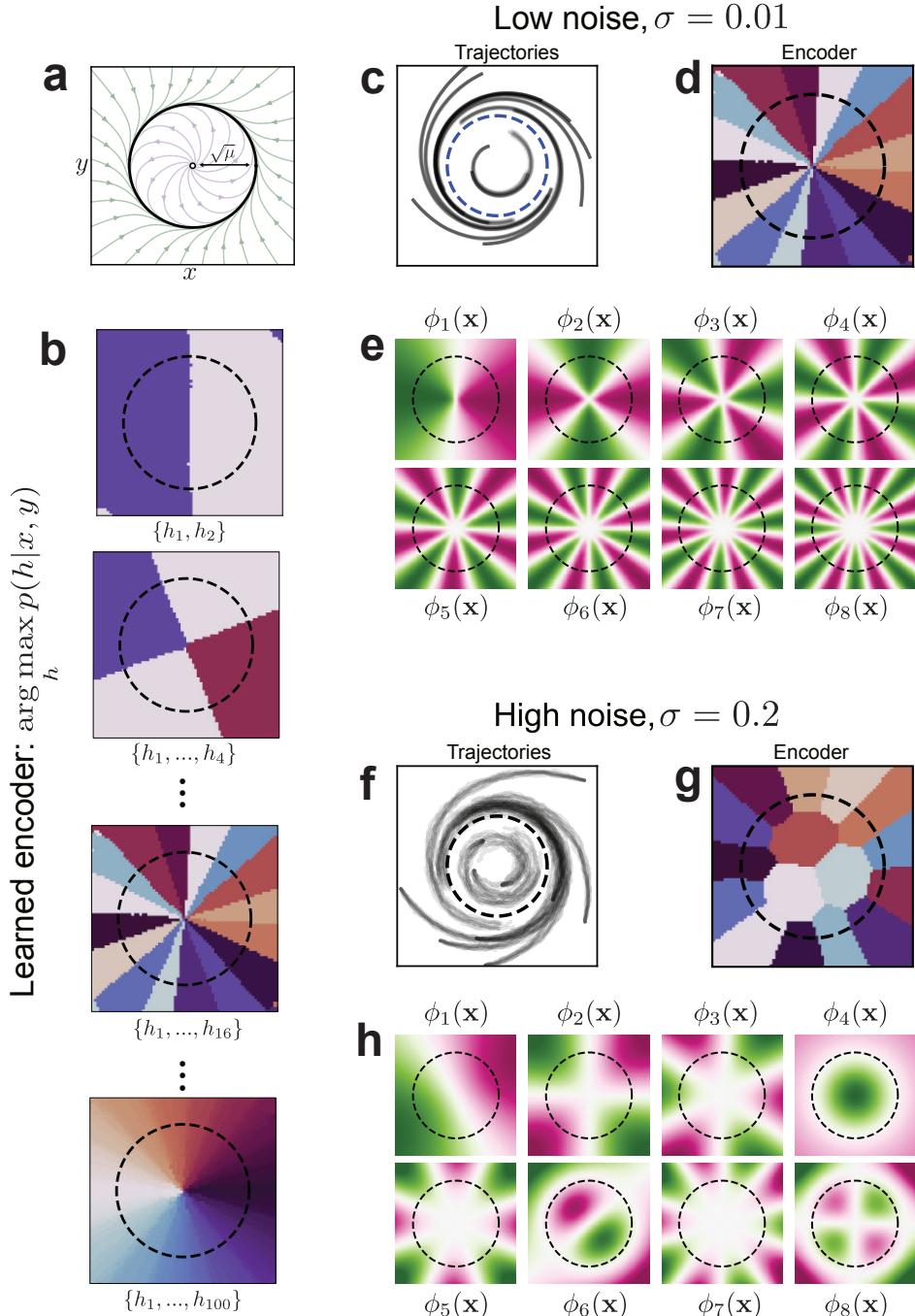


FIG. S19. Eigenfunctions and IB partitions for the Hopf Oscillator. (a) Phase portrait for the deterministic Hopf oscillator. (b) Partitions found by IB for high beta but a restricted encoding alphabet N_H . (c) Example trajectories of the nearly deterministic Hopf oscillator, with small noise amplitude $\sigma = 0.01$. (d) IB partition of the low noise dynamics for a $N_H = 16$ encoding alphabet (same as appears in (b)). (e) For small noise, the first several eigenfunctions obtained numerically approximate the numerically expected ones of the form $\cos n\theta$. (f) Simulated trajectories of Hopf oscillator with higher noise amplitude $\sigma = 0.2$. (g) IB partition of the low noise dynamics for a $N_H = 16$ encoding alphabet; note the dependence on r . (h) The presence of noise changes the eigenfunctions, in particular they depend on r .

also evaluated at the trivial encoder. The joint distribution may be written

$$\begin{aligned} p(h_0, h_t) &= \int dx_0 dx_t p(h_0, h_t | x_0, x_t) p(x_0, x_t) \\ &= \int dx_0 dx_t p(h_t | x_t) p(h_0 | x_0) p(x_0, x_t) \end{aligned}$$

where we used the fact that the variables form the Markov chain $H_0 - X_0 - X_t - H_t$. The derivative is thus

$$\begin{aligned} \partial_n^\mu p(h_0, h_t) &= \int dx_0 dx_t p(h_0, h_t | x_0, x_t) p(x_0, x_t) \\ &= \int dx_0 dx_t (\partial_n^\mu p(h_t | x_t)) p(h_0 | x_0) p(x_0, x_t) + p(h_t | x_t) (\partial_n^\mu p(h_0 | x_0)) p(x_0, x_t) \\ &= \int dx_0 dx_t p(h_\nu^t) (\delta_{\mu\nu} - p(h_\nu^t)) \tilde{\phi}_n(x_0) p(h_\lambda^0) p(x_0, x_t) + p(h_\nu^t) p(h_\lambda^0) (\delta_{\mu\lambda} - p(h_\lambda^0)) \tilde{\phi}_n(x_t) p(x_0, x_t) \\ &= \delta_{n0} p(h_\nu^t) p(h_\lambda^0) (\delta_{\mu\nu} - p(h_\nu^t) + \delta_{\mu\lambda} - p(h_\lambda^0)) \end{aligned}$$

To get to the last line, we evaluated everything at the trivial encoder and used the assumption that $p(x_0)$ and $p(x_t)$ are the same distribution (steady state assumption). Note that the derivative depends only trivially on n , so that it does not “see” subleading eigenfunctions. The Hessian of the mutual information is, denoting $h_i^0 = h_i$

$$\begin{aligned} \partial_n^\nu \partial_m^\mu I(H_0, H_t) &= \partial_n^\nu \sum_{h_i, h_j^t} \partial_m^\mu p(h_i, h_j^t) \left(\log \frac{p(h_i, h_j^t)}{p(h_i)p(h_j^t)} + 1 \right) - \underbrace{\frac{p(h_i, h_j^t)}{p(h_i)} \partial_m^\mu p(h_i)}_{\sim \sum_i \partial_n p(h_i) = 0} - \frac{p(h_i, h_j^t)}{p(h_j^t)} \partial_n^\nu p(h_i) \\ &= \sum_{h_i, h_j^t} \partial_n^\nu \partial_m^\mu p(h_i, h_j^t) \left(\log \frac{p(h_i, h_j^t)}{p(h_i)p(h_j^t)} + 1 \right) + \partial_n^\nu p(h_i, h_j^t) \partial_m^\mu p(h_i, h_j^t) \frac{1}{p(h_i, h_j^t)} \\ &\quad - \sum_{h_i, h_j^t} \partial_n^\nu p(h_i) \partial_m^\mu p(h_i, h_j^t) \frac{1}{p(h_i)} - \partial_n^\nu p(h_j^t) \partial_m^\mu p(h_i, h_j^t) \frac{1}{p(h_j^t)} \end{aligned} \tag{S3}$$

Evaluated at the uniform encoder, the first term disappears because we have the derivative of a constant. From the equations for the derivatives (S1) and (S2), we can conclude $\partial_n^\nu \partial_m^\mu I(H_0, H_t) \propto \delta_{n0} \delta_{m0}$, which shows that we lose all information about the eigenfunctions. The Hessian of the Lagrangian for IB modified by the term $I(H_0, H_t)$ will therefore have the same stability properties as the original IB problem.