

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}

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

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

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

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

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

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

⁷University 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 relevant variables, a task for which no general method exists. Here, we develop a systematic approach based on the information bottleneck to identify the relevant variables, defined as those most predictive of the future. We elucidate analytically the relation between these relevant variables and the eigenfunctions of the transfer operator describing the dynamics. Further, we show that in the limit of high compression, the relevant variables are directly determined by the slowest-decaying eigenfunctions. Our information-based approach indicates when to optimally stop increasing the complexity of the reduced model. Furthermore, it provides a firm foundation to construct interpretable deep learning tools that perform model reduction. We illustrate how these tools work in practice by considering uncured videos of atmospheric flows from which our algorithms automatically extract the dominant slow collective variables, as well as 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 air in your office may be described by a 10^{27} -dimensional state vector containing the positions and momenta of every particle in the room. Yet, for most practical purposes, it can be effectively described using only a small number of quantities such as pressure and temperature. 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 [1–9], to data-driven methods such as independent component analysis [10], dynamic mode decomposition [11–13], diffusion maps [14], spectral submanifolds [15, 16], and deep encoder-decoder neural networks [17–24].

The success of these approaches is limited by a fundamental difficulty: before performing any reduction, one has to identify a decomposition of the full system into relevant and irrelevant variables. In the absence of prior knowledge and intuition (e.g. a clear separation of scales), identifying such a decomposition is an open problem [4]. It may not even be clear *a priori* when to stop increasing the complexity of a simplified model or, conversely, when to stop reducing the amount of information needed to represent the dynamical state of a complex system. In

both cases one must first determine the minimal number of relevant variables that are needed. The answer to this question depends in fact on how precisely and how far in the future you wish to forecast. Nonetheless, this answer should be compatible with fundamental constraints on forecasting set by external perturbations and finite measurement accuracy [25, 26].

In order to address the difficulty identified in the previous paragraph, 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 [27], model reduction is about keeping information that matters most to predict the future [28, 29]. Inspired by this simple insight, we formalize model reduction as a lossy compression problem known as the information bottleneck (IB) [30, 31]. This formal step allows us to give a precise answer to the question of how to identify relevant and irrelevant variables. We show how and under what conditions the standard operator-theoretic formalism of dynamical systems [19, 32], which underlies most methods of model reduction, naturally emerges from optimal compression. Crucially, our framework systematically answers the question of when to stop increasing the complexity of a minimal model. 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. We illustrate our approach on benchmark dynamical systems and demonstrate that it works even on uncured datasets, such as satellite movies of atmospheric flows downloaded directly from YouTube and biological datasets composed of microscopy videos of cyanobacteria colonies in which we discover an emergent synchronization order parameter.

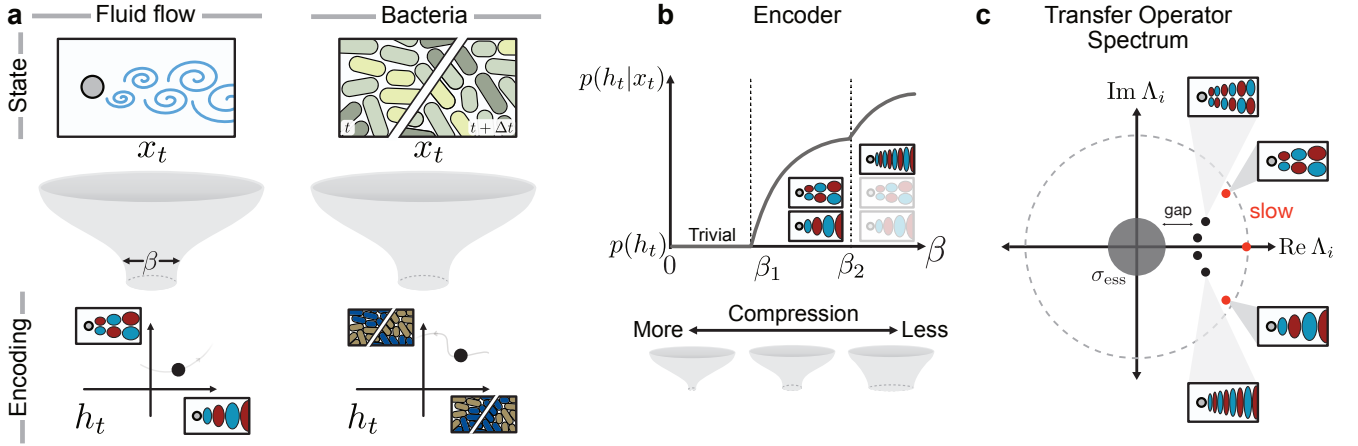


FIG. 1. **Interpretable dynamical variables in reduced models via the information bottleneck.** (a) The information bottleneck compresses high-dimensional state variables x_t , into simpler encoding variables h_t providing a controllable trade-off between the degree of compression and the predictive 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 (1)]. The resulting compressed, or encoded, variables h_t represent collective variables most predictive of the system’s future. (b) 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. (c) 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.

I. MODEL REDUCTION AS A COMPRESSION PROBLEM

We present here a method to extract collective variables most predictive of the system’s future evolution directly from data. This data is composed of a time sequence of measured states $x_1, x_2, \dots, x_T$. The system state x_t could 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). The full state can be very high dimensional, with a number of dimensions equal to the number of observed pixels in the case of imaging data. However, the variation of any individual pixel is often of limited interest to us, as noise (either inherent or due to measurement) induces uncertainty about its true value. Individual pixels are, due to this uncertainty, poor predictors of the future state of the system. We can say that they are *irrelevant* for predicting the future. On the other hand, certain spatially-averaged collective variables may evolve slowly in time, and the future state of the system may be reliably estimated from them.

We seek a way to “encode” each state x_t into a simple, lower-dimensional representation h_t in a way that isolates these relevant features of the input x_t . For instance, in Fig. 1a, both the velocity field of a fluid flow and the images of a dynamic cyanobacteria colony (upper row) may

be encoded as a point in a 2D space (lower row). The encoding is given by a probabilistic mapping $p(h_t|x_t)$ which can be thought of as a machine which takes a state x_t and assigns it to a value h_t . The fact that this mapping is probabilistic simply means we may have some uncertainty about the true value of h_t even given a measurement of the state. Whether or not some encoding is extracting “relevant” features of the state x_t is determined by the extent to which we can use it to predict the future. This predictive power can be quantified by the *mutual information* between the encoding and the future state, $I(H_t, X_{t+\Delta t})$, which tells us how much the knowledge of H_t reduces our uncertainty about the future $X_{t+\Delta t}$ (see Methods). (In our notation, upper-case X_t refers to the random variable, while x_t refers to a particular value taken by the random variable.) To find a good encoder that extracts relevant features, we might try to find an encoding which maximizes this information. However, an encoder obtained in this way would simply copy the original state, $h_t = x_t$, since x_t represents all the information we have about the system. In order to encourage the encoder to discard irrelevant features, we simultaneously seek an encoder which maximizes compression by minimizing the information about the original state, $I(H_t, X_t)$. This prescription for encoding relevant collective variables can be formalized by the information bottleneck (IB) method [28, 30, 31]. The information bottleneck objective function combines both of our stated goals – compression and predictive

power – into one mathematical expression:

$$\mathcal{L}_{\text{IB}}[p(h_t|x_t)] = I(X_t, H_t) - \beta I(X_{t+\Delta t}, H_t). \quad (1)$$

Crucially, the parameter β allows one to tune how much weight to assign to compression versus 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 captured by H_t , so that features of X_t must “compete” to pass through to the encoding variable (Fig. 1b). These features are reflected in the form of the encoder

$$p^*(h_t|x_t) = \arg \min \mathcal{L}_{\text{IB}} \quad (2)$$

which provides the optimal trade-off between compression and predictability [29]. Our goal is to connect the dynamical properties of the system to the features learned by the encoder.

In any realistic experimental setting, the presence of noise or uncertainty means we cannot predict precisely the future state of a system but instead can only predict a likely *distribution* of possible 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 . This conditional probability distribution completely characterizes the dynamics of the system, and determines how probability distributions evolve in time:

$$p(x_{t+\Delta t}) = \int p(x_{t+\Delta t}|x_t)p(x_t)dx_t. \quad (3)$$

For Markovian, or “memoryless” dynamics, such an evolution can be understood as the action of a (linear) transfer operator U which acts on probability distributions. U can be decomposed into its right and left eigenvectors as

$$U = \sum_n |\rho_n\rangle e^{\lambda_n \Delta t} \langle \phi_n| + U_{\text{ess}} \quad (4)$$

where $|\rho_n\rangle$ are right eigenvectors with eigenvalue $\Lambda_n \equiv e^{\lambda_n \Delta t}$ and $\langle \phi_n|$ are the corresponding left eigenvectors. λ_n are the eigenvalues of the infinitesimal generator of U , known as the Fokker-Planck operator (Fig. 1c). 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 [33, 34]. The eigenfunctions ϕ_n in (4) are in some sense “natural” features of the dynamics, as they evolve independently in time.

Our key observation is that the optimal encoder in (2) can be expressed in terms of the eigenvalues λ_n and left eigenfunctions ϕ_n of (the generator of) U ,

$$p^*(h_t|x_t) = \frac{p_\beta^*(h_t)}{\mathcal{N}(x_t)} \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x_t) f_n(h_t) \right] \quad (5)$$

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 the SI. These factors effectively determine what 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 extract. However, things become simple in the limit of small β , or high compression. When β is small the encoder is trivial: $p(h_t|x_t) = p(h_t)$. In this case the value h_t is assigned at random with no regard to the state x_t 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. 1b) [35–38]. The first transition happens at a finite value of β_1 when the first most predictive feature is learnt.

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

$$p_\beta^*(h_t|x_t) \approx \frac{p_\beta^*(h_t)}{\mathcal{N}(x_t)} \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x_t) f_1(h_t) \right) \quad (6)$$

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. 6 with clearly specified technical assumptions may be found in the Methods and SI.

The above statement 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. 6 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 non-Gaussian variables. The question of maximally informative features has additionally been explored in the context of animal vision, where one seeks to understand what features of the field of vision are encoded by retinal neurons [39, 40].

We further observe numerically that this picture holds true more generally: also at successive IB transitions, the learned features correspond to successive modes of the transfer operator. This picture is consistent with the exact results known for Gaussian IB, where the encoder learns eigenvectors of a matrix (related to the covariance of the joint $X_t, X_{t+\Delta t}$ distribution) in a step-wise fashion at each IB transition [37]. Together, this shows that the most informative features extracted by IB, an agnostic information-theoretic approach, correspond to physically-interpretable quantities – namely transfer operator eigenfunctions. As we show later, the insight above can be leveraged to systematically learn these slow variables directly from data with neural networks [41].

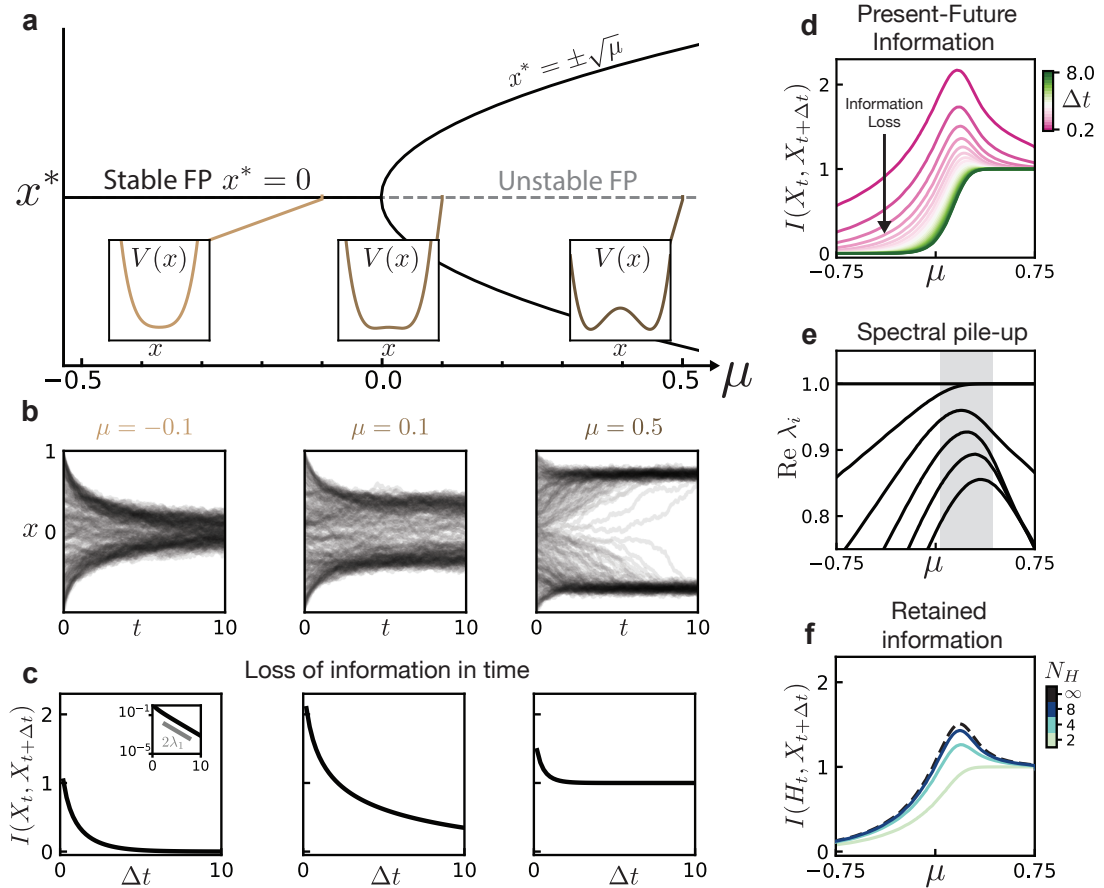


FIG. 2. **Information loss of a Brownian particle in a double well potential.** (a) Fixed point (FP) diagram of the dynamics given by Eq. 7 for zero noise. There is a bifurcation at $\mu = 0$ where the stable FP at $x = 0$ becomes unstable and two new stable FPs appear at $\pm\sqrt{\mu}$. Insets show the evolution of the corresponding potential $V(x)$, with the emergence of a double-well structure for $\mu > 0$. (b) Dynamics of the system Eq. 7 for varying values of μ corresponding to the potential insets in (a), with uniformly-distributed initial conditions. (c) Loss of information between the initial condition and the future state. Inset shows scaling given by the first eigenvalue of the transfer operator. (d) Mutual information between the present and future state for varying time delay Δt and bifurcation parameter μ . (e) 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 $\Lambda_i = e^{\lambda_i \Delta t}$. (f) Maximal mutual information which can be encoded into a discrete variable of N_H values, for a fixed time delay $\Delta t = 1.0$. Black dashed line shows $I(X_t, X_{t+\Delta t})$ for reference. Information is provided in units of bits.

II. 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 trapped in a confining double-well potential. This might represent, for example, a molecule with a single degree of freedom that transitions between two metastable configurations [42]. 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

$$\dot{x}_t = -\partial_x V(x_t) + \sigma \eta_t. \quad (7a)$$

$$V(x) = \frac{1}{4}(\mu - x^2)^2 \quad (7b)$$

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). Sample trajectories, with noise, for a uniform initial distribution of particles are shown in Fig. 2b. For negative μ , the trajectories all converge to a fixed point at $x = 0$, while for $\mu > 0$ they fluctuate around the fixed points at $x = \pm\sqrt{\mu}$.

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 (Fig. 2c; 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 [43]

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

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

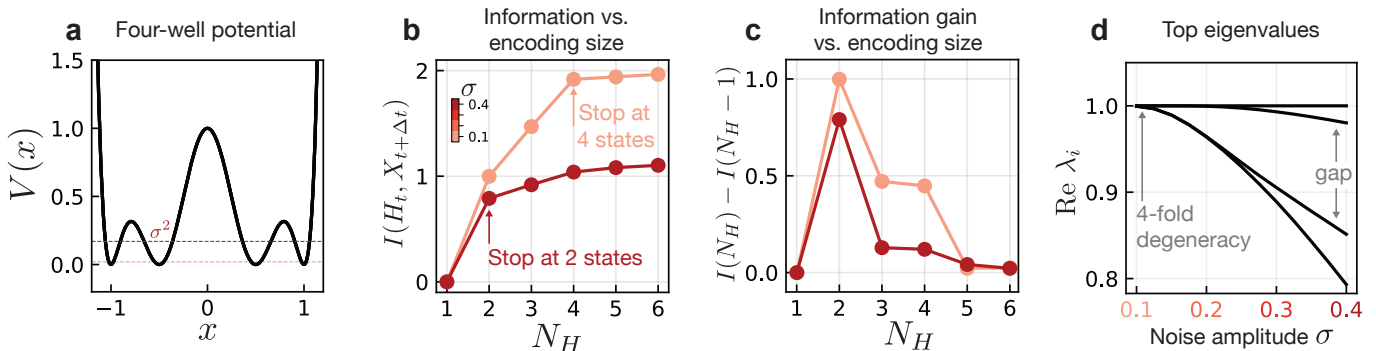


FIG. 3. “Knowing when to stop”. The spectral properties of the transfer operator determine the necessary complexity (i.e. “when to stop” [25]) of the reduced model, which we show is also visible in information theoretic metrics. (a) Four-well potential in which a Brownian particle fluctuates. The magnitude 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.

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

$$I(X_t, X_{t+\Delta t}) = e^{2\lambda_1 \Delta t} \langle \phi_1^2 \rangle \langle \rho_1^2 / \rho_0^2 \rangle + \mathcal{O}(e^{2\lambda_2 \Delta t}) \quad (8)$$

where expectations are taken over the steady state distribution (see SI). 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 probability of hopping between the wells, though this rate of hopping is exponentially small [33].

The loss of information in time depends on the bifurcation parameter μ as summarized in Fig. 2d. 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 $S(X_{t+\Delta t}|X_t) \approx 0$ (see Methods). 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 [33]. In the presence of noise, although the continuous spectrum becomes discrete [33, 34] there is still a pile-up of eigenvalues characterized by several eigenvalues becoming close to 1 (Fig. 2e). This pile-up gives rise to the information peak seen in Fig. 2d. 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. S4).

III. 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 [44–50]. Several works have approached this partition problem from a dynamical systems perspective, linking “optimal” partitions to eigenfunctions of the (adjoint) transfer operator [25, 51]. In this setting, a central question is “when to stop” [25, 26, 48, 49]: 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 [52–54], which results in effective model reduction due to the “sloppiness”, or irrelevance, of certain system variables [55, 56]. 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 $I(H_t; X_{t+\Delta t}) = S(H_t) - S(H_t|X_{t+\Delta t})$ 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.

Fig. 2f shows that the number of states necessary to describe the system depends strongly on the value of μ . For $|\mu| \gg 0$, a two-state discrete variable $h_t \in \{0, 1\}$ suffices to describe the system’s future. Increasing the number of reduced variables N_H does not allow more information to be captured. Near the information peak at $\mu \approx 0.2$ this changes: predicting the future state of the

system requires a more complex hidden variable of up to $N_H \approx 10$ values. Above, we saw that this peak arises due to the pile-up of eigenvalues at $\mu \approx 0.2$. The content of the transfer operator spectrum is thus directly reflected in the number of encoding variables needed to capture the system's statistics.

Noise can have a similarly dramatic impact on the reduced model complexity. Indeed, noise in some form, either inherent to the dynamics or due to measurement error, is necessary for a model to be reducible. In purely deterministic systems where the future state is a bijective function of the present state, information does not decay and complete knowledge of the state is required to predict the future.

Consider a fluctuating Brownian particle as 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 standard deviation of the fluctuations σ corresponds to 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 $x \lesssim 0$. 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 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 four-fold degenerate which indicates 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.

IV. TRANSFER OPERATOR EIGENFUNCTIONS ARE MOST INFORMATIVE FEATURES

Until now we have concerned ourselves with encodings whose capacity is limited only by the number of variables, rather than by the compression imposed by a small value of β . 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 . By studying the behavior of the encoder in this regime, in particular 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 return to the simple example of a particle in a double well with dynamics given by (7) which we map to a discrete variable $H_t \in \{0, \dots, N_H - 1\}$. In this system the IB loss function (1) can be optimized directly, as shown in Ref. [30], using an iterative scheme known as the Blahut-Arimoto algorithm [43] (see SI).

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 which are sketched in Fig. 1b and shown quantitatively in Fig. 4a. 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 . 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. 4a). The dependence of $p(h_t | x_t)$ on x can be explained by a stability analysis of the IB Lagrangian (see Methods and SI). Stability is governed by the eigenvalues η_i of the Hessian of the IB Lagrangian with respect to the parameters $f_n(h_t)$ in Eq. 5. 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. 4b 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. Our numerics confirm that these weights are dominated by f_1 as expected from our analytical result (Fig. 4c, top). By taking the logarithm of the encoder after the transition, we can in-

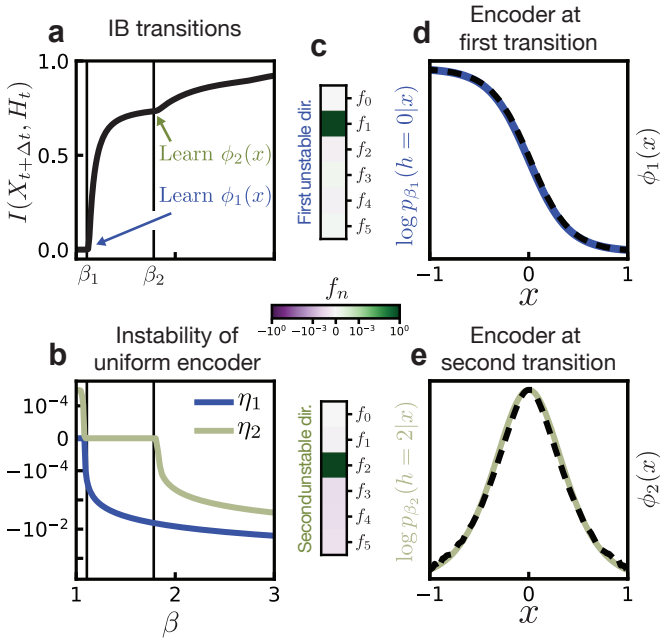


FIG. 4. **IB learns eigenfunctions of the adjoint transfer operator.** (a) When the relative weight β between both constraints in Eq. 1 is changed, more and more information can go through the encoder. This occurs in steps, where the spectral content of the transfer operator is included starting from eigenvalues with largest magnitude (i.e., the slowest ones). (b) 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. (c) The unstable directions are dominated by single components (note the color scale is logarithmic). (d) 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). (e) Likewise, at the second transition the encoder is given by $\phi_2(x)$.

independently confirm that the encoder depends only on $\phi_1(x)$ (Fig. 4d).

Our stability analysis predicts that a second mode becomes unstable at the second IB transition $\beta \approx \beta_2$ (Fig. 4b). Here we see that this unstable mode selects f_2 , and that the encoder correspondingly gains dependence on $\phi_2(x)$ (Fig. 4d). 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 the SI, we perform the same analysis for a triple-well potential where this difference is more apparent.

V. DATA-DRIVEN DISCOVERY OF SLOW VARIABLES

IB finds transfer operator eigenfunctions 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. The utility of exact IB for this purpose 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. We refer to this as variational IB. In the remainder of this paper, we show how to implement these networks for the discovery of slow variables directly from data.

First we show numerically that the results of the previous sections remain valid for high-dimensional systems by considering a simulated data of fluid flow past a disk [57]. 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 $v_0 \hat{e}_x$ 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.

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,

$$\phi_n[\mathbf{v}] = \langle \mathbf{v}(\mathbf{x}), \mathbf{m}^{(n)}(\mathbf{x}) \rangle, \quad (9)$$

where $\mathbf{m}^{(n)}$ is the n -th mode (often referred to as a *Koopman* mode [19]) and angled brackets denote integration over space. The true eigenfunction and corresponding modes can be computed via dynamic mode decomposition (DMD) [11, 12], 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. 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

$$\frac{\partial \phi_n}{\partial v_j} = m_j^{(n)}. \quad (10)$$

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

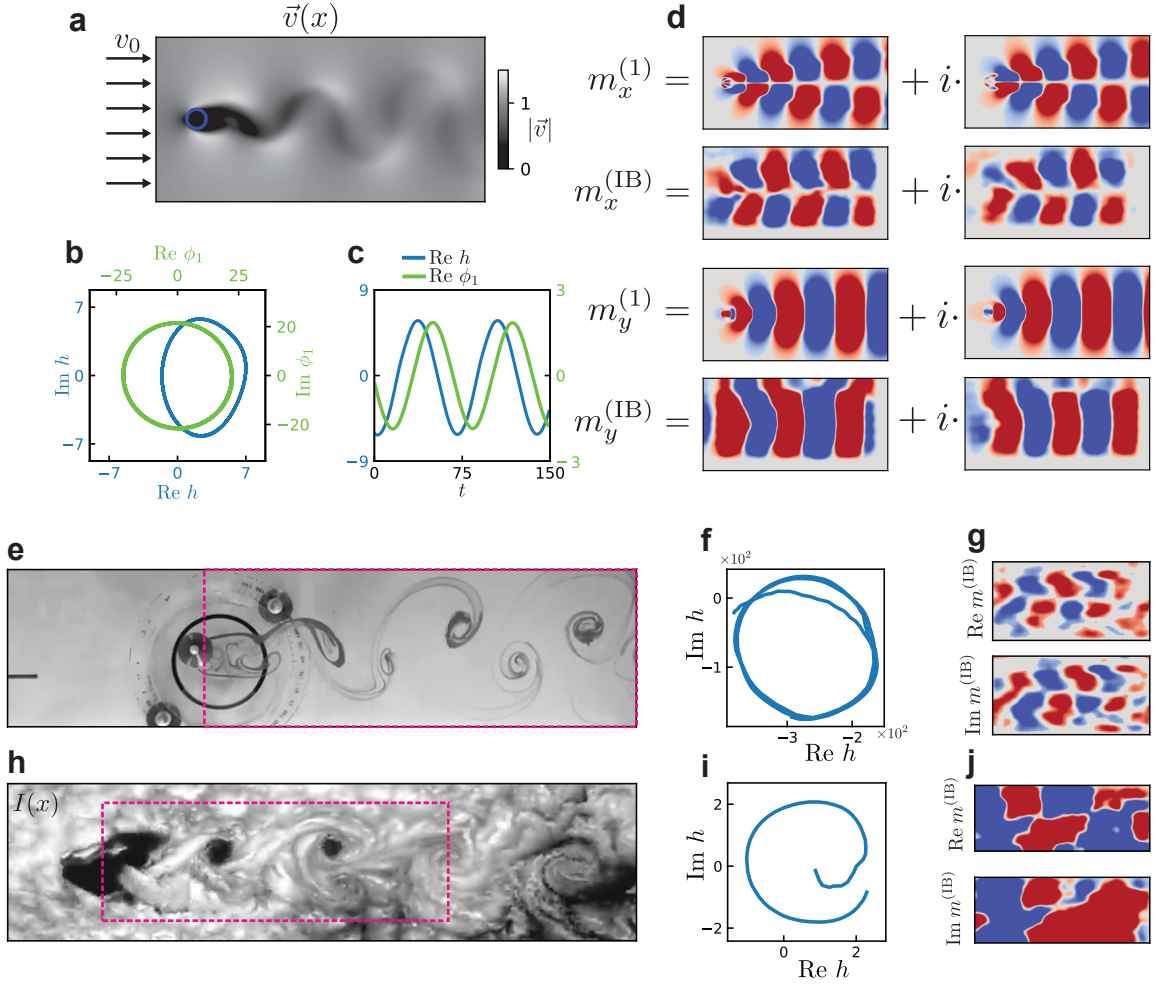


FIG. 5. **Variational IB for high-dimensional simulated fluid flow.** (a) A fluid flows into the system with uniform velocity v_0 in the x -direction and passes a disk-shaped obstacle, which perturbs the fluid and causes vortex shedding behind the object in a so-called von Kármán street. The state of the system is given by a spatially varying two-component vector field $\mathbf{v}(\mathbf{x})$. (b) The dynamics in latent space (blue) are very regular, traversing 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) Time evolution of one component of the latent variable (h_1 , blue) as well as the DMD mode amplitude (green). (d) 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.

to the input field

$$\frac{\partial h}{\partial v_j} = m_j^{(\text{IB})} + g_{\text{res},j}(\mathbf{v}(\mathbf{x})), \quad (11)$$

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 $\mathbf{g}_{\text{res}}$ is small. We indeed find this to be the case; Fig. 5d 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 $\mathbf{g}_{\text{res}}$ 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.

VI. RELEVANT VARIABLE IDENTIFICATION IN LABORATORY-GENERATED AND ATMOSPHERIC FLOWS

The scenario above is characterized by high-dimensional data and few samples; training was performed with only ~ 400 samples. We now demonstrate that our framework continues to hold approximately and yield interpretable latent spaces even for real-world fluid flow datasets scraped directly from videos on Youtube [58, 59] (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 [58]. We take a background-subtracted grayscale image of the flow field as our input (Fig. 5e) and task VIB with learning a two-dimensional latent variable as above. Also here, variational IB learns oscillatory dynamics of the latent variables (Fig. 5f). 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. 5g). This is expected, as the x -component of the velocity field has similar glide reflection symmetry as the intensity image.

Next, we apply variational IB to a von Kármán street arising due to flow around Guadalupe Island, which was imaged by a National Oceanic and Atmospheric Administration (NOAA) satellite [59] (Fig. 5h). The video consists of only 62 frames, and the von Kármán street undergoes a single oscillation. Even with this small amount of data, the variational IB neural network learns latent variables which capture this oscillation and have the expected dependence on the input variables (Fig. 5i-j). 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. 5d. This symmetry is less clear in the component $\frac{\partial h_1}{\partial I(x)}$, which is likely due to the fact that the von Kármán street is not as fully formed in this data as in our previous examples.

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. 6a). These oscillations are controlled by three Kai proteins [60] and depend in particular on the ratios of the copy number of these proteins which can be tuned experimentally [61].

We were provided with videos of 10 cyanobacteria colonies that were grown under various conditions that impact 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. S9). This allows us to isolate the motion of individual bacteria and fluorescence oscillations (Fig. 6b).

Our input to the variational IB neural network are these cropped images augmented with a time-lagged image of the same region (Fig. 6c). The purpose of this time lag is to make the dynamics Markovian: due to the oscillatory intensity field, if one observes only a single time point it is unclear whether the intensity is currently increasing or decreasing. These time lagged pairs comprise our system state, $X_t = \{I(\mathbf{x}, t), I(\mathbf{x}, t + \tau)\}$, where τ is the duration of the time lag. Here we take $\tau = 3$ hr and a prediction time horizon $\Delta t = 8$ hr, but find that choosing different Δt or τ does not change our results (SI Fig. S9).

With variational IB we compress the state X_t into a latent variable h of variable dimension (Fig. 6d-f). 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, $I(X_{t+\Delta t}, H_t)$. We see that 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” (Fig. 6f). 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 therefore restrict our focus to the $\dim H = 2$ case.

We noticed that there were notable differences in the radius of latent space oscillations from colony to colony, two of which are highlighted in (Fig. 6g). To understand this difference, we examined the original microscopy time series corresponding to both large and small latent radius (Fig. 6k) 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 [62–64], and compute a global synchronization order parameter $r(t)$ (see SI). 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. 6i). These clusters are precisely those representing trajectories of large and small latent radius (Fig. 6j), 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 which shares 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).

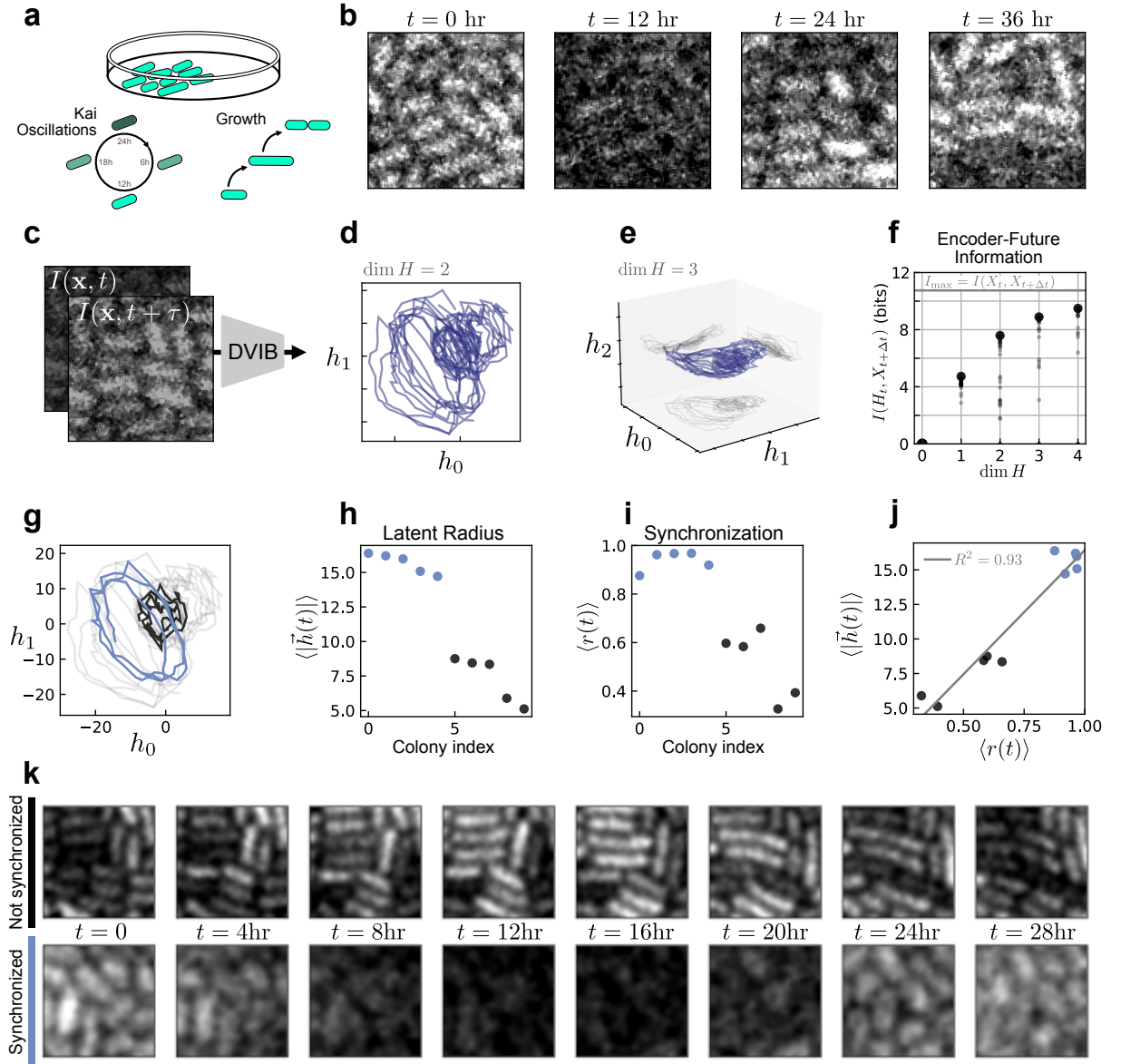


FIG. 6. Discovering slow collective variables in cyanobacteria populations (a) Fluorescent images of cyanobacteria colonies labeled with EYFP driven by the *kaiBC* promoter, allowing the visualization of Kai protein transcription. The colonies are imaged as they undergo cell growth and oscillations in 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.

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 such as time-lagged autoencoders that are free of such assumptions, 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 use variants of DMD [13, 65–68] or autoencoders for dynamics [20–22, 69] 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 requiring tailored pre-processing steps (SI Fig. S12).

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 [61]. The synchronization order parameter discovered by variational IB corresponds to differing experimental concentrations of theophylline, which is in agreement with the findings in Ref. [61]. 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. 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 [47, 70–72]. 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.

METHODS

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 [43]

$$S(X) = - \int dx p(x) \log p(x).$$

Given another random variable Y , such that X and Y have a joint distribution $p(x, y)$, we can ask how much information is shared between these two variables. This is given by the mutual information

$$I(X, Y) = S(X) - S(X|Y) \quad (12)$$

$$= D_{\text{KL}}(p(x, y) \| p(x)p(y)) \quad (13)$$

This can be interpreted as quantifying how much (on average) a measurement of Y can reduce our uncertainty about the value of X ((12)).

The information bottleneck

The information bottleneck [30] is an example of a rate-distortion problem which seeks to find an optimal compression which minimizes some distortion measure with the original signal [43]. 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

$$\mathcal{L}_{\text{IB}}[p(h|x)] = I(X, H) - \beta I(Y, H), \quad (14)$$

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 [30]

$$p(h|x) = \frac{p(h)}{N(x)} \exp [-\beta D_{\text{KL}}(p(y|x) \| p(y|h))]. \quad (15)$$

Encoder in terms of transfer operator eigenfunctions

To connect the optimal encoder to the transfer operator, we first rewrite (15) in terms of the transition

probabilities,

$$p(h_t|x_t) = \frac{\tilde{p}(h_t)}{\tilde{N}(x_t)} \exp \left[\beta \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|h_t) \right] \quad (16)$$

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

$$p(x_{t+\Delta t}|x_t) = \sum_n e^{\lambda_n \Delta t} \rho_n(x_{t+\Delta t}) \phi_n(x_t). \quad (17)$$

From this, the (5) of the main text immediately follows, where

$$f_n(h_t) = \int dx_{t+\Delta t} \rho_n(x_{t+\Delta t}) \log p(x_{t+\Delta t}|h_t) \quad (18)$$

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

$$p_\beta^*(h|x) = \frac{1}{\mathcal{N}(x)} p_\beta^*(h) \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x) f_1(h) \right) \quad (19)$$

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 $\mathcal{L}_{\text{IB}}$ 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. In summary, this means that in the limit of high-compression only the first component f_1 becomes non-zero, hence the encoder has the form given in (19).

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) [11, 12], or (time-lagged) independent component analysis (TICA) [10] (which is equivalent to DMD [73]). 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 [74] or extended DMD (eDMD) [13]. 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 [75, 76].

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 [77, 78]. One prominent example is diffusion maps [14], 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 [20–22, 69]. Autoencoders often produce poorly-behaved latent spaces that distribute the latent variables on a narrow manifold with sharp features, see for example [69]. 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 [20–22] 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 [65–68]. 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.

Appendix A: Information bottleneck

The information bottleneck was originally formulated in Ref. [30] 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[43]. In Ref. [30], 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

$$\mathcal{L}_{\text{IB}}[p(h|x)] = I(X, H) - \beta I(Y, H). \quad (\text{A1})$$

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

$$\mathcal{L}_{\text{IB}}[p(h|x)] = \sum_{x,h} p(h|x)p(x) \log \frac{p(h|x)}{p(h)} - \beta \sum_{y,h} p(y, h) \log \frac{p(y, h)}{p(h)p(y)} + \sum_x \lambda(x) \left(1 - \sum_h p(h|x) \right). \quad (\text{A2})$$

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

$$p(h|x) = \frac{p(h)}{N(x)} \exp \left[-\beta D_{KL}(p(y|x) \| p(y|h)) \right]. \quad (\text{A3})$$

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

$$p(h|x) = \frac{p(h)}{N(x)} \exp \left[\beta \int dy p(y|x) \log p(y|h) \right]. \quad (\text{A4})$$

When $\beta < 1$, it follows from the data processing inequality $I(X, H) \geq I(Y, H)$ that (A1) is minimized by a trivial encoder $p(h|x) = p(h)$. In this case, $\mathcal{L}_{\text{IB}} = 0$ 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 $\beta_2, \beta_3, \dots$ etc.

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

The optimal IB encoder (A3) can be found using the Blahut-Arimoto (BA) algorithm [43]. As described in detail in Refs. [30, 79] and sketched in Fig. S1, the algorithm is an iterative procedure, where the encoder at iteration $k + 1$ is

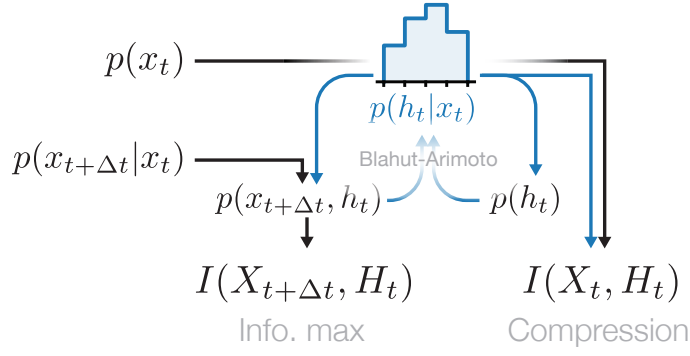


FIG. S1. **Exact IB.** Exact IB finds an optimal encoder using an iterative Blahut-Arimoto algorithm, described in the main text, for a rate-distortion problem with Kullback-Leibler distortion [30]. This process requires access to (an estimate of) the transfer matrix and the steady state distribution. Once an optimal encoder has been found, relevant quantities such as mutual information can be computed.

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. D2 (and normalizes the distribution), while the following two lines update marginal and conditional distributions using the new estimate of the encoder. In Refs. [30, 79] 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 [44, 45]. In brief, one divides space into bins and computes a finite-dimensional approximation to the conditional distribution as

$$P_{ij} = P(X_{t+\Delta t} = x_j | X_t = x_i) = N_{i \rightarrow j} / N_i, \quad (\text{B1})$$

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

$$Up(x_j) \approx \sum_{x_i} P_{ij} p(x_i), \quad (\text{B2})$$

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

Appendix C: Transfer operators

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sqrt{2D(\mathbf{x})}\xi(t) \quad (\text{C1})$$

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 $D = 0$. The transfer operator describes the evolution of probability distributions. 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 .

$$\mathcal{L}^\dagger \phi(x) = f_i(\mathbf{x}) \partial_i (\phi(\mathbf{x})) + D(\mathbf{x}) C_{ij} \partial_i \partial_j (\phi(\mathbf{x})). \quad (\text{C2})$$

For $D = 0$, $\mathcal{L}$ is the generator of the Perron-Frobenius operator, while its adjoint is the Koopman operator [19].

Appendix D: Derivations

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

$$p_\beta^*(h|x) \approx \frac{1}{\mathcal{N}(x)} p_\beta^*(h) \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x) f_1(h) \right) \quad (\text{D1})$$

In Ref. [30], it is shown that the encoding minimizing the IB objective satisfies the implicit equation

$$p_\beta^*(h|x) = \frac{p_\beta^*(h)}{\mathcal{N}(x)} \exp \left(-\beta D_{\text{KL}} \left[p_{X_{t+\Delta t}|X_t} \parallel p_{X_{t+\Delta t}|H_t} \right] \right) \quad (\text{D2})$$

where $\mathcal{N}$ is a normalization factor ensuring $\int dh p_\beta^*(h|x) = 1$, and

$$p_\beta^*(h) \equiv p_\beta^*(h) = \int dx p_\beta^*(h|x) p(x). \quad (\text{D3})$$

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

$$D_{\text{KL}}[p||q] = \int dx p(x) \log \frac{p(x)}{q(x)}; \quad (\text{D4})$$

in the D_{KL} in (D2), the integration is done with respect to the common random variable $X_{t+\Delta t}$ of the two conditional distributions.

Equation D2 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

$$p_{X_{t+\Delta t}|H_t} = \int_{X_t} p_{X_{t+\Delta t}|X_t} p_{H_t|X_t} \frac{p_{X_t}}{p_{H_t}}. \quad (\text{D5})$$

Note the appearance of the encoder $p_{H_t|X_t}$, making (D2) 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 [33, 34]. We can then write the conditional distribution in terms of right and left eigenvectors of U ,

$$p_{X_{t+\Delta t}|X_t}(x|x') = \sum_n e^{\lambda_n \Delta t} \rho_n(x) \phi_n(x'). \quad (\text{D6})$$

The Kullback-Leibler divergence in (D2) can be represented as a sum of two terms,

$$D_{\text{KL}}[\dots] = \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|x_t) - \int dx_{t+\Delta t} p(x_{t+\Delta t}|x_t) \log p(x_{t+\Delta t}|h_t). \quad (\text{D7})$$

The first term has no dependence on h and can hence be absorbed into the normalization $\mathcal{N}(x)$. Plugging the decomposition (D6) into the second term leads directly to

$$p_\beta^*(h|x) = \frac{1}{\mathcal{N}(x)} p_\beta^*(h) \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x) f_n(h) \right] \quad (\text{D8})$$

where

$$f_n(h) = \int dx_{t+\Delta t} \rho_n(x_{t+\Delta t}) \log p(x_{t+\Delta t}|h), \quad (\text{D9})$$

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 D8 depends only on the existence of the spectral decomposition; we have made no other assumptions on the dynamics (beyond Markovianity) up until this point.

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).

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

$$p_\beta^*(h|x) = \frac{1}{\mathcal{N}(x)} p_\beta^*(h) \exp \left(\beta e^{\lambda_1 \Delta t} \phi_1(x) f_1(h) \right) \quad (\text{D10})$$

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 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,

$$\frac{\partial^2}{\partial f_n(h_i) \partial f_m(h_j)} \mathcal{L}_{\text{IB}}. \quad (\text{D11})$$

Similar perturbative approaches have been studied in other contexts within IB in [35, 38, 80]. For concreteness, we consider a discrete state $x \in \{1, \dots, N_X\}$ and discrete encoding $h \in \{1, \dots, N_H\}$. We express the encoder as in Eq. D8 where we now expand the normalization terms

$$p_\beta^*(h_\mu|x_i) = \frac{f_0(h_\mu) \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x_i) f_n(h_\mu) \right]}{\sum_{h_\lambda} f_0(h_\lambda) \exp \left[\beta \sum_n e^{\lambda_n \Delta t} \phi_n(x_i) f_n(h_\lambda) \right]} \quad (\text{D12})$$

In the following, we simplify notation as

$$p(h_\mu|x_i) = \frac{f_0^\mu \exp \left[\sum_n \tilde{\phi}_n(x_i) f_n^\mu \right]}{\sum_{h_\lambda} f_0^\lambda \exp \left[\sum_n \tilde{\phi}_n(x_i) f_n^\lambda \right]} \quad (\text{D13})$$

where $\tilde{\phi}_n(x_i) = \beta e^{\lambda_n \Delta t} \phi_n(x_i)$. 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

$$\partial_n^\mu p(h_\nu) = \sum_{x_i} p(x_i) p(h_\nu|x_i) \tilde{\phi}_n(x_i) (\delta_{\mu\nu} - p(h_\mu|x_i)). \quad (\text{D14})$$

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 $p(h_\mu|x_i)|_{\beta=0} = p(h_\mu)$. We note in passing that we find the first derivative terms of $\mathcal{L}_{\text{IB}}$ vanish, see also [36, 38]. The compression term is given by

$$\begin{aligned} \partial_n^\mu \partial_m^\nu I(X, H) &= \left(\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) \\ &\quad \times (\delta_{\mu\nu} p(h_\mu) - p(h_\nu) p(h_\mu)) \end{aligned}$$

and the information maximization term is

$$\begin{aligned} \partial_n^\mu \partial_m^\nu I(Y, H) &= \left(\left\langle \frac{\langle \tilde{\phi}_n \rangle_{p(\cdot, y)} \langle \tilde{\phi}_m \rangle_{p(\cdot, y)}}{p(y)^2} \right\rangle - \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) \\ &\quad \times (\delta_{\mu\nu} p(h_\mu) - p(h_\nu) p(h_\mu)) \end{aligned}$$

Angled brackets with no subscript correspond to an average with respect to the steady state distribution, $\langle \cdot \rangle = \sum_{x_i} \cdot p(x_i)$. For the second term, we retain the integration variable y for clarity. In sum, the Hessian of the Lagrangian is given by

$$H_{(\mu,n),(\nu,m)} = \left(\langle \tilde{\phi}_n \tilde{\phi}_m \rangle - \beta \left\langle \frac{\langle \tilde{\phi}_n \rangle_{p(\cdot,y)} \langle \tilde{\phi}_m \rangle_{p(\cdot,y)}}{p(y)^2} \right\rangle - (1 - \beta) \langle \tilde{\phi}_n \rangle \langle \tilde{\phi}_m \rangle \right) (\delta_{\mu\nu} p(h_\mu) - p(h_\nu) p(h_\mu)) \quad (\text{D15})$$

which lives in $\mathbb{R}^{(N_X \times N_H) \times (N_X \times N_H)}$; we index H with a multi-index (μ, n) . From the form of Eq. D15, 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 . 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 quasi-equilibrium dynamics. In particular, we consider a Fokker-Planck operator $\mathcal{L}_{\text{FP}}$ of the form

$$\partial_t p(x) = \mathcal{L}_{\text{FP}} p(x) = -\partial_i (f_i(x) p(x)) + D \partial_i^2 p(x). \quad (\text{D16})$$

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

$$\mathcal{L}_{\text{FP}} \rho_0(x) = -\partial_i J_i(x) = 0 \quad (\text{D17})$$

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 $\mathcal{L}_{\text{FP}}$ become *right* eigenfunctions when multiplied by ρ_0 . To see this, note that

$$\begin{aligned} \mathcal{L}_{\text{FP}}(\rho_0 \phi_n) &= \phi_n \mathcal{L}_{\text{FP}} \rho_0 + \rho_0 \mathcal{L}_{\text{FP}}^\dagger \phi_n - 2 \partial_i \phi_n \underbrace{(f_i \rho_0 - D \partial_i \rho_0)}_{J_i} \\ &= \lambda_n \rho_0 \phi_n \end{aligned} \quad (\text{D18})$$

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

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

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

$$\beta = e^{-2\lambda_1 \Delta t}, e^{-2\lambda_2 \Delta t}, \dots \quad (\text{D19})$$

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 (D1). 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

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

where we have introduced the shorthand $B_{ij} = \left\langle \frac{\langle \tilde{\phi}_i \rangle_{p(\cdot, y)} \langle \tilde{\phi}_j \rangle_{p(\cdot, y)}}{p(y)^2} \right\rangle$ 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,

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

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

$$\frac{f_1}{f_2} = \frac{-e^{(\lambda_1 - \lambda_2) \Delta t}}{2\hat{a}_{12}} \left(-\hat{a}_{11} + e^{2(\lambda_2 - \lambda_1) \Delta t} \hat{a}_{22} - D \right) \quad (\text{D20})$$

$$= \frac{e^{\Gamma \Delta t}}{2\hat{a}_{12}} \left(2\hat{a}_{11} + \mathcal{O}(e^{-2\Gamma \Delta t}) \right) \quad (\text{D21})$$

$$= e^{\Gamma \Delta t} \frac{\hat{a}_{11}}{\hat{a}_{12}} + \mathcal{O}(e^{-2\Gamma \Delta t}). \quad (\text{D22})$$

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

$$\langle \phi_n \phi_m \rangle_{\rho_0} = -\frac{2}{\lambda_n - \lambda_m} \langle (\mathbf{J} \cdot \nabla \phi_n) \phi_m \rangle_{\rho_0}. \quad (\text{D23})$$

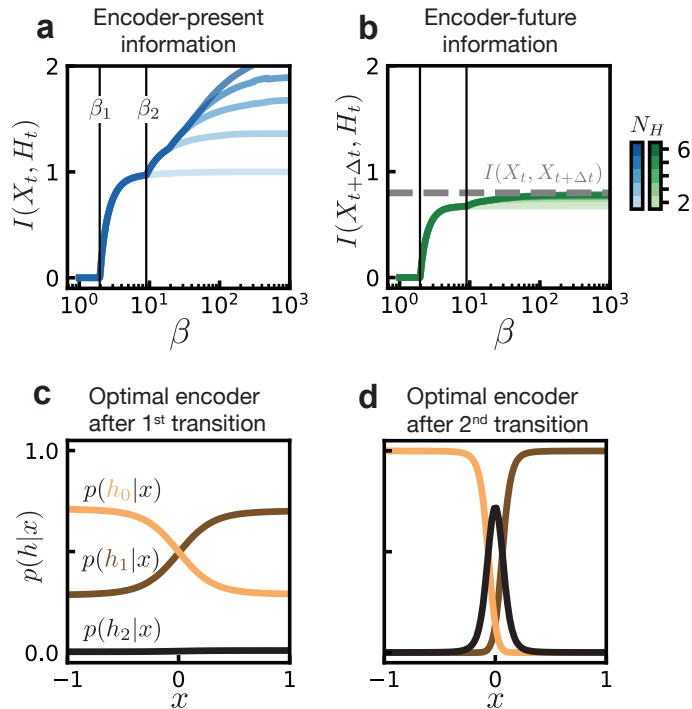


FIG. S2. **Information bottleneck for a Brownian particle in a double well.** (a) Mutual information between the current state X_t and its encoding H_t for various alphabet sizes N_H (color). The first two IB transitions are denoted with black lines and occur at β_1 and β_2 . (b) Mutual information between 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 $N_H = 3$.

Similar to the equilibrium case, we see that at the first transition the encoder depends only on f_1 , giving (D1). 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. D10 will depend instead on $\phi_0(x)$, which identifies the independent sectors.

Appendix E: Rate of information decay

We consider the mutual information

$$I(X_t, X_{t+\Delta t}) = \sum_{x,y} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \quad (\text{E1})$$

where x denotes values of the random variable X_t and y the values of $X_{t+\Delta t}$. We further consider dynamics which can be given by a transfer operator U with integral kernel $p(y|x)$ that can be spectrally decomposed as

$$p(y|x) = \sum_n e^{\lambda_n t} \rho_n(y) \phi_n(x), \quad (\text{E2})$$

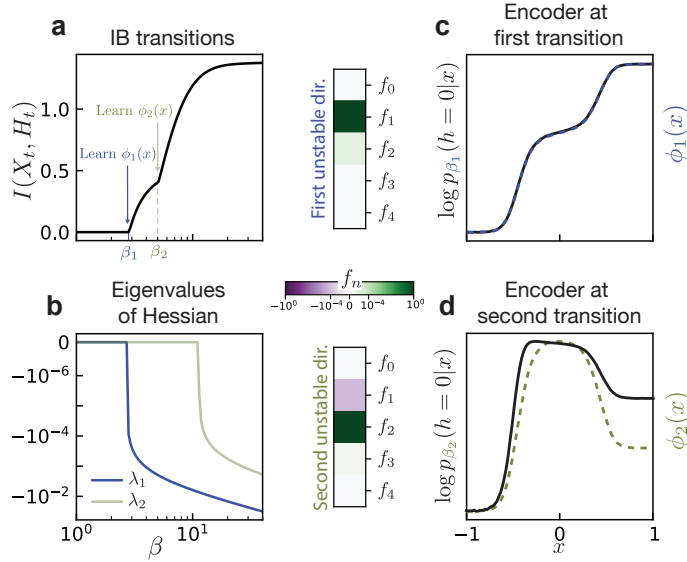


FIG. S3. **IB transitions for a Brownian particle in a triple-well potential.** (a) Information transitions with varying β for a particle in a triple-well potential. (b) Eigenvalues of the IB Hessian evaluated at the uniform encoder $p(h|x) = N_H^{-1}$. The appearance of an unstable direction at $\beta_1 \approx 3$ 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 $\phi_2(x)$.

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

$$I(X_t, X_{t+\Delta t}) = \sum_{x,y} \sum_n e^{\lambda_n \Delta t} \rho_n(y) \phi_n(x) p(x) \log \left(1 + \sum_{m>0} e^{\lambda_m \Delta t} \frac{\rho_m(y)}{p(y)} \phi_m(x) \right) \quad (\text{E3})$$

$$= \sum_{n,m>0} e^{(\lambda_n + \lambda_m) \Delta t} \sum_{x,y} \phi_n(x) \phi_m(x) p(x) \frac{\rho_n(y) \rho_m(y)}{p(y)} \quad (\text{E4})$$

$$+ \sum_{n,m,\ell>0} e^{(\lambda_n + \lambda_m + \lambda_\ell) \Delta t} \sum_{x,y} \phi_n(x) \phi_m(x) \phi_\ell(x) p(x) \frac{\rho_n(y) \rho_m(y) \rho_\ell(y)}{p(y)} + \dots \quad (\text{E5})$$

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

$$I(X_t, X_{t+\Delta t}) = e^{2\lambda_1 \Delta t} \sum_x \phi_1^2(x) p(x) \sum_y \frac{\rho_1(y)^2}{p(y)}. \quad (\text{E6})$$

For short times, other eigenvalues will also contribute to the mutual information (Fig. S4).

Appendix F: Variational IB

Exactly solving the IB problem in principle requires access to the full distribution $p(y, x)$, where x is the variable to be compressed and y is the relevance variable. One way around this is via so-called Deep Variational IB as introduced in Ref. [41]. The key result from [41] is an upper bound on the IB objective

$$\mathcal{L}_{\text{IB}} = I(X; H) - \beta I(Y; H). \quad (\text{F1})$$

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

$$\int dh p(h) \log p(h) \geq \int dh p(h) \log \hat{p}, \quad (\text{F2})$$

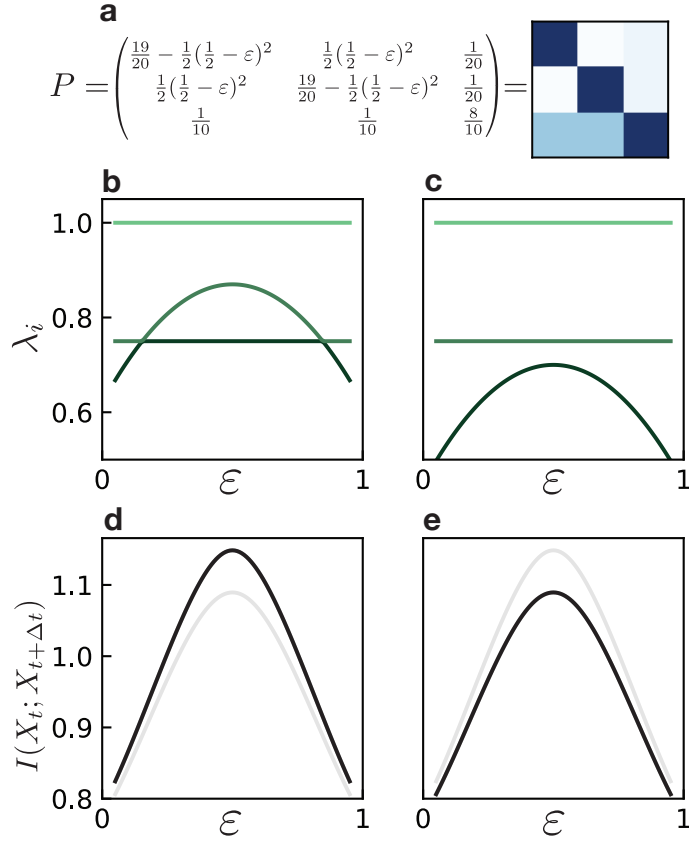


FIG. S4. **Information gain due to an eigenvalue pile-up.** (a) To study the role of a gap closing compared to the pile-up of eigenvalues (beyond the dominant one) in the mutual information $I(X_t; X_{t+\Delta t})$ we build a matrix P with which we can tune the eigenvalues via a parameter ε . (b) Eigenvalues 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.

and hence

$$I(X, H) \leq \int dx dh p(h|x) p(x) \log \frac{p(h|x)}{\hat{p}(h)} \quad (\text{F3})$$

$$= \mathbb{E}_x D_{\text{KL}}(p(h|x) \| \hat{p}(h)). \quad (\text{F4})$$

If $p(h|x)$ and $\hat{p}(h)$ 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 $p(h|x)$ and let the marginal $\hat{p}(h)$ be a spherical unit-variance Gaussian. More concretely, encoded variables H_t are sampled from $p(H_t|X_t)$ by computing

$$h_t = f_W(x_t) + \sigma_W(x_t)\eta, \quad (\text{F5})$$

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 $I(Y, H)$ from below. We do this using the noise-contrastive estimate of the mutual information introduced in Ref. [81]. This recasts the problem as one of distinguishing samples from the distributions $p(y|h)$ and $p(y)$. Given a batch of B pairs (y, h) and one particular value h_i , one asks what the probability is that a sample y_j is from $p(y|h_i)$ (is a *positive* sample) and not $p(y)$ (is a *negative* sample). This

probability is

$$p(y_j = \text{pos}|h_i) = \frac{\frac{p(y_i|h_i)}{p(y_i)}}{\sum_k^B \frac{p(y_k|h_i)}{p(y_k)}}, \quad (\text{F6})$$

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

$$\mathbb{E}[-\log p(y_j = \text{pos}|h_i)], \quad (\text{F7})$$

where the expectation is taken over indices i and j , is closely related to the mutual information $I(Y, H)$. In the limit of infinite samples $B \rightarrow \infty$, this quantity is given by

$$-\int dy dh p(h)p(y|h) \log p(y_j = \text{pos}|h_i) = \log B - I(Y, H). \quad (\text{F8})$$

If one had access to the probabilities $p(y_i = \text{pos}|h)$ appearing in Eq. F7, the mutual information could thus be easily determined. One attempts to estimate these probabilities by introducing a variational ansatz $f(y, h)$ to approximate the density ratio $\frac{p(y|h)}{p(y)}$. Typically this f is represented by a neural network. One can then obtain a bound on the mutual information by minimizing

$$\mathcal{L}_B = \mathbb{E} \left[-\log \frac{f(y_i|h_i)}{\sum_k^B f(y_k|h_i)} \right], \quad (\text{F9})$$

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

$$I_{\text{NCE}}(Y, H) = \log B - \mathcal{L}_B \leq I_{\text{true}}(Y, H). \quad (\text{F10})$$

The full objective to be minimized is given by

$$\mathcal{L}_{\text{VIB}} = D_{KL}(p(H_t|X_t) \|\hat{p}(H_t)) - \beta I_{\text{NCE}}(X_{t+\Delta t}; H_t), \quad (\text{F11})$$

an illustration can be seen in Fig. S5. 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 [82].

The VIB loss Eq. F11 is very similar to the loss function for a time-lagged β -variational autoencoder (β -VAE) [83], which have a form $\mathcal{L} = D_{KL} - \mathcal{L}_{\text{rec}}$, where the reconstruction term $\mathcal{L}_{\text{rec}}$ 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 [84, 85]. Other metrics which penalize the number of “false neighbors” in latent space have also shown to improve performance for chaotic systems [86].

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. S13, 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) [18, 87, 88].

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

$$\begin{aligned} \dot{r} &= r(\mu - r^2), \\ \dot{\theta} &= \omega. \end{aligned} \quad (\text{F12})$$

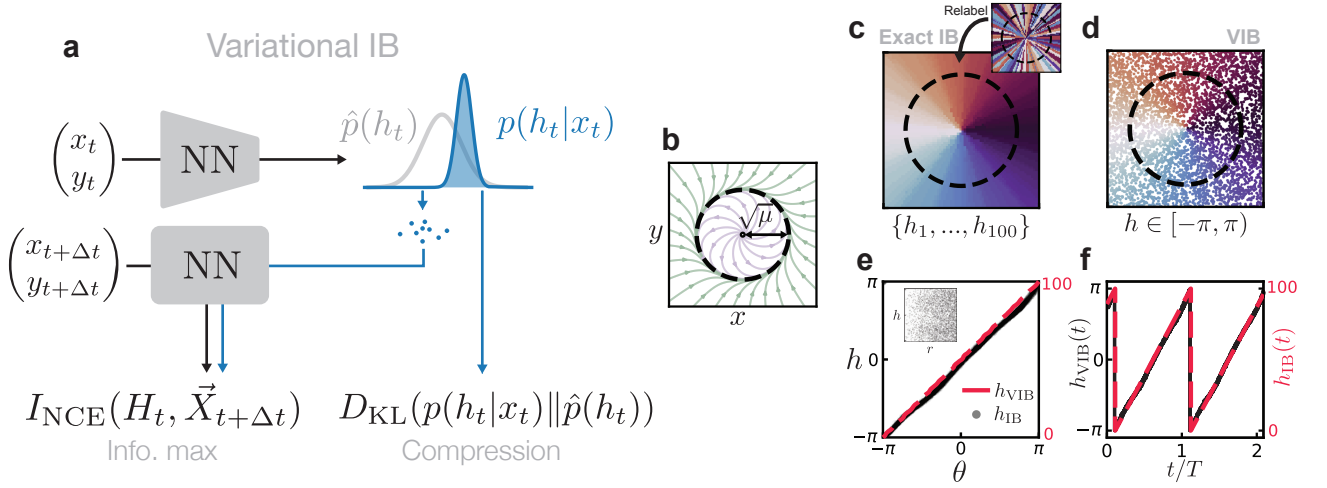


FIG. S5. **Variational IB extends exact IB.** (a) Variational IB optimizes the variational objective directly from samples, in contrast to exact IB which requires an estimation of the full conditional distribution $p(x_{t+\Delta t}|x_t)$. Values of h_t are drawn from a “latent” distribution $p(h|x)$, 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 .

Equation (F12) is the normal form (in polar coordinates) for dynamics near a Hopf bifurcation [89]. For $\mu > 0$ this system exhibits a circular limit cycle of radius $\sqrt{\mu}$ (Fig. S5b). In Section G we show that this system has infinite purely imaginary eigenvalues $\lambda_n = in\omega$ 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 the those expected for the deterministic case (see Section G). 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. S5c).

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. S5e-f) and is nearly uncorrelated with r (inset). This shows that our mathematical results illustrated for exact IB in Sec. D hold also in the approximate framework of VIB.

VIB for deterministic dynamics

IB is an inherently probabilistic framework. To handle deterministic dynamics, we make them effectively stochastic by introducing a stochastic sampling scheme. 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 (D1), 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 [90]. 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. S6a). 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) (Fig. S6b). 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. S6c-e). The dynamics of h also become notably less “slow”, and are instead nearly constant with

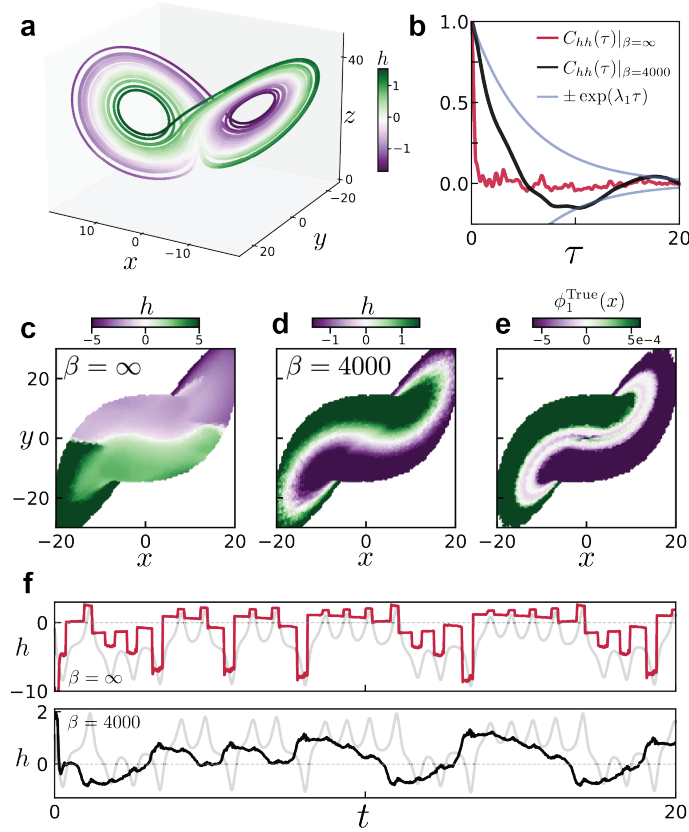


FIG. S6. **Variational IB discovers slow variables.** (a) Variational IB applied to the Lorenz system. Color corresponds the encoding value h_t . (b) Time correlation functions of the encoded variable $C_{hh}(\tau) = \langle (X(\tau) - \bar{X})(Y(0) - \bar{Y}) \rangle$ for high compression (black) and no compression (red). Blue shows the decay of the subleading Koopman eigenfunction using numerically-obtained eigenvalues. (c-d) Lorenz attractor projected onto the x - y plane, colored by encoding variable. (e) Same projection, colored by the value of the true subleading Koopman eigenfunction. (f) Dynamics of h_t obtained by encoding the state at every time during the true trajectory. Top corresponds to no compression ($\beta = \infty$), bottom corresponds to high compression ($\beta = 4000$).

occasional large jumps (Fig. S6f). This shows that in order for h to be a valid slow variable, the compression term in Eq. (F11) 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).

Appendix G: Optimal encoding for Hopf normal form dynamics

In Fig. S5 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

$$\dot{x} = (\mu - x^2 - y^2)x - \omega y \quad (G1)$$

$$\dot{y} = (\mu - x^2 - y^2)y - \omega x \quad (G2)$$

Expressed in polar coordinates, we have

$$f_x \hat{e}_x = f_x \cos \theta \hat{e}_r - f_x \sin \theta \hat{e}_\theta \quad (G3)$$

$$f_y \hat{e}_y = f_y \sin \theta \hat{e}_r + f_y \cos \theta \hat{e}_\theta \quad (G4)$$

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

$$f_i \partial_i \phi = (\partial_r \phi)(\mu - r^2)r + \frac{1}{r}(\partial_\theta \phi)\omega r \quad (G5)$$

$$= r(\mu - r^2)(\partial_r \phi) + \omega \partial_\theta \phi \quad (G6)$$

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 [19]. The eigenvalue equation for the radial coordinate is solved in Ref. [33], and in Ref. [76] 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 $p(h|x)$.

$$p(h|r, \theta) \propto \exp\left\{\sum_n^\infty e^{\lambda_n \Delta t} \phi_n(r, \theta) \int r' dr' d\theta' \rho_n(r', \theta') \log p(r', \theta'|h)\right\} \quad (\text{G7})$$

$$= \exp\left\{\sum_n^\infty e^{in\omega \Delta t} \int r' dr' d\theta' e^{in(\theta - \theta')} \log p(r', \theta'|h)\right\} + \mathcal{O}(e^{\text{Re}\lambda_k \Delta t}) \quad (\text{G8})$$

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

$$p(h|r, \theta) \propto \exp\left\{\int r' dr' \log p(r', (\theta + \omega \Delta t)|h)\right\}. \quad (\text{G9})$$

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 Eqn. G9 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

$$p(r_{t+\Delta t}, \theta_{t+\Delta t}|h) = \int r dr d\theta p(r_{t+\Delta t}, \theta_{t+\Delta t}|r_t, \theta_t) p(r, \theta_t|h) \quad (\text{G10})$$

$$\propto \delta((\theta_{t+\Delta t} - \omega \Delta t) - h) \quad (\text{G11})$$

where we used that the two terms in the integrand of the first equation are both delta functions. Plugging this into (G9) 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. S7).

Appendix H: VIB for fluid flow around a cylinder

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) [11, 12, 91], also described below in Section K. 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.

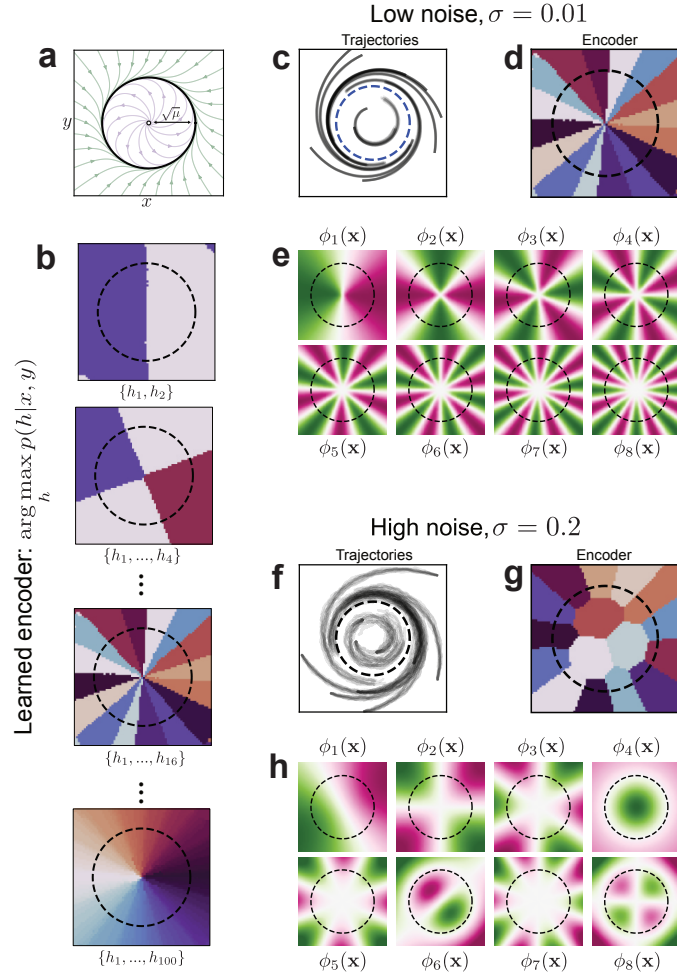


FIG. S7. **Eigenfunctions and IB partitions for the Hopf Oscillator.** (a) Phase portrait for the deterministic Hopf oscillator. (b) Partitions found by IB for high beta but a restricted encoding alphabet N_H . (c) Example trajectories of the nearly deterministic Hopf oscillator, with small noise amplitude $\sigma = 0.01$. (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 .

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,

$$\phi_n[\mathbf{v}] = \langle \mathbf{v}(\mathbf{x}), \mathbf{m}^{(n)}(\mathbf{x}) \rangle, \quad (\text{H1})$$

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 H) [11, 12].

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. (H1). This can be done by examining gradients of the network with respect to the input field

$$\frac{\partial h}{\partial v_j} = m_j^{(\text{IB})} + g_{\text{res},j}(\mathbf{v}(\mathbf{x})) \quad (\text{H2})$$

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

$$\frac{\partial \phi_n}{\partial v_j} = m_j^{(n)}. \quad (\text{H3})$$

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 $\mathbf{g}_{\text{res}}$ 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. S8 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 $\mathbf{m}$.

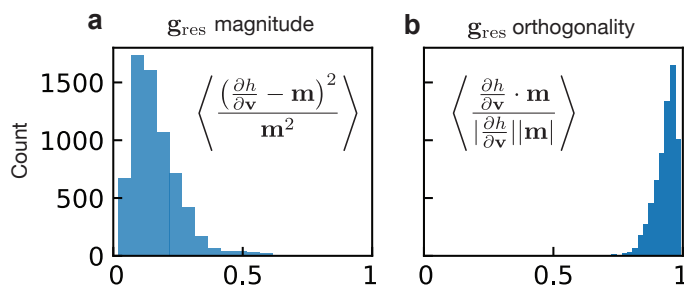


FIG. S8. **VIB gradients for fluid flow are nearly linear.** We compare the linear part of the gradients of the VIB network to the residual which depends on the input field $\mathbf{v}(\mathbf{x})$. (a) The residual parts are smaller in magnitude than the linear part $\mathbf{m}$. Angle brackets denote 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.

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. [52]. 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. S13b.

Appendix I: Cyanobacteria experiments

The experimental parameters used in the cyanobacteria experiments are described in detail in [61]. In brief, the authors in [61] 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. S9, 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 colony growth, which otherwise dominates the VIB results. Each latent trajectory shown in Fig. S9 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 S9 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

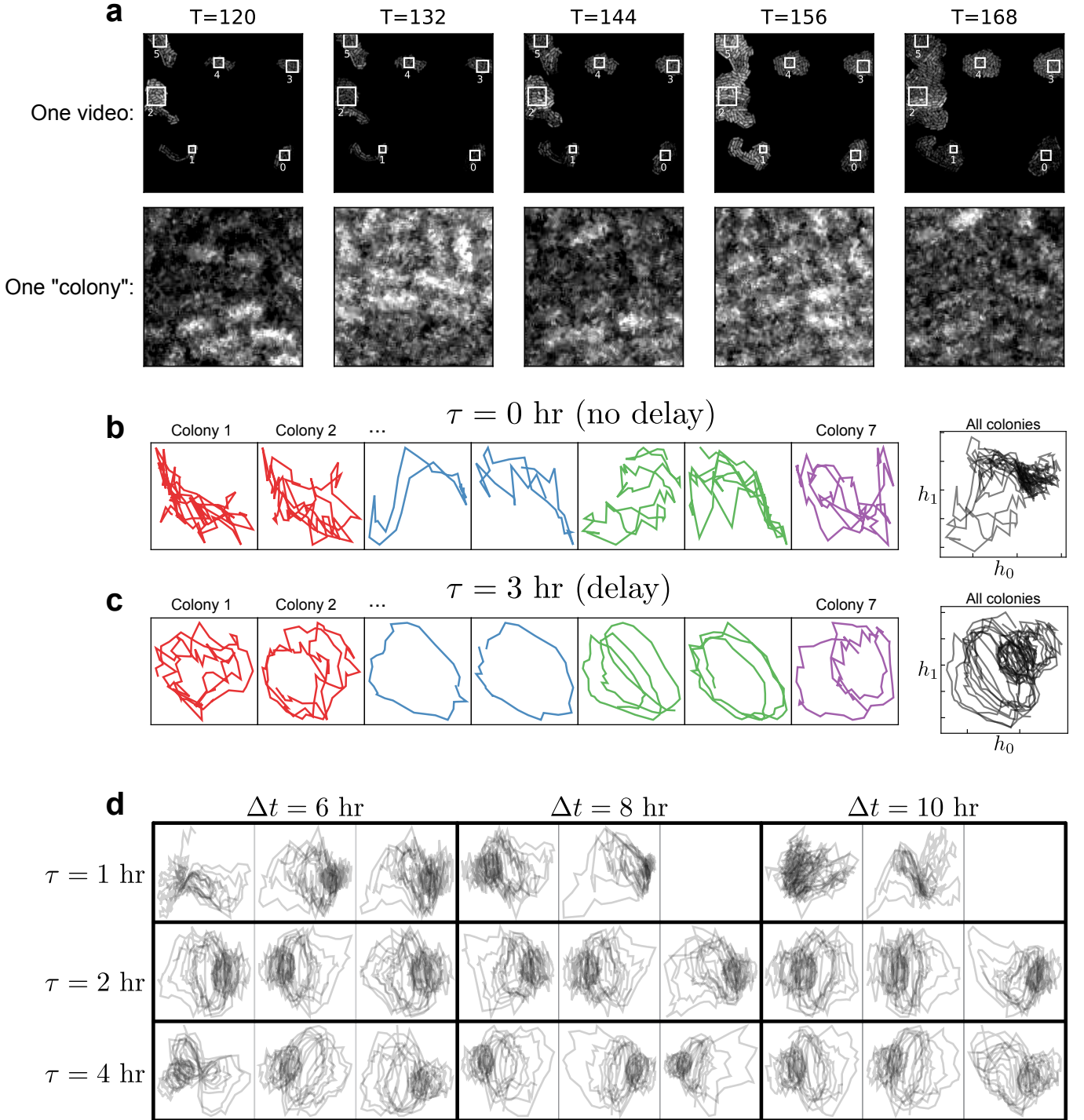


FIG. S9. Cyanobacteria data and choice of time delay (a) The cyanobacteria dataset consists of 5 videos, where each has multiple (up to 3) colonies. (Top) time evolution of one video, where the interiors of the colonies are outlined with white boxes. The dataset is composed of all colonies which are 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.

field, an order parameter can be computed as

$$r(t)e^{i\psi(t)} = \frac{1}{V} \left| \int d\mathbf{x} e^{i\theta(\mathbf{x},t)} \right| \quad (\text{I1})$$

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.

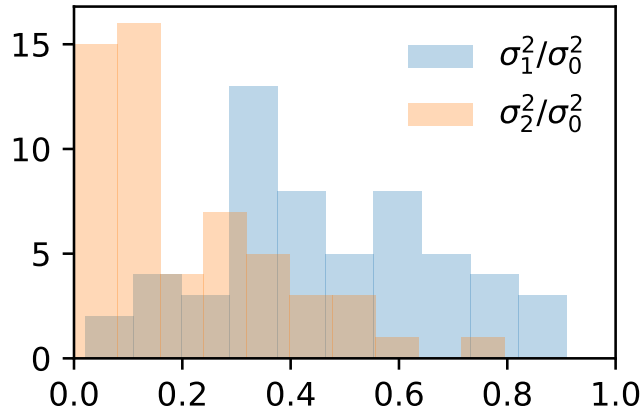


FIG. S10. **Variance along principal component directions** For each DVIB model trained on the cyanobacteria dataset, we compute the principal component decomposition of the resulting point cloud in latent space. Here we show the variance along each principal component for 60 instantiations of the DVIB model. Variances are normalized by the variance in the first principal component direction.

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. S11 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.

Appendix J: 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

$$dx_t = \frac{F(x_t)}{\gamma} dt + \sigma \sqrt{dt} \eta_t \quad (\text{J1})$$

where η_t is white noise with unit variance. The noise magnitude σ is related to the diffusion constant in Eq. C1 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.

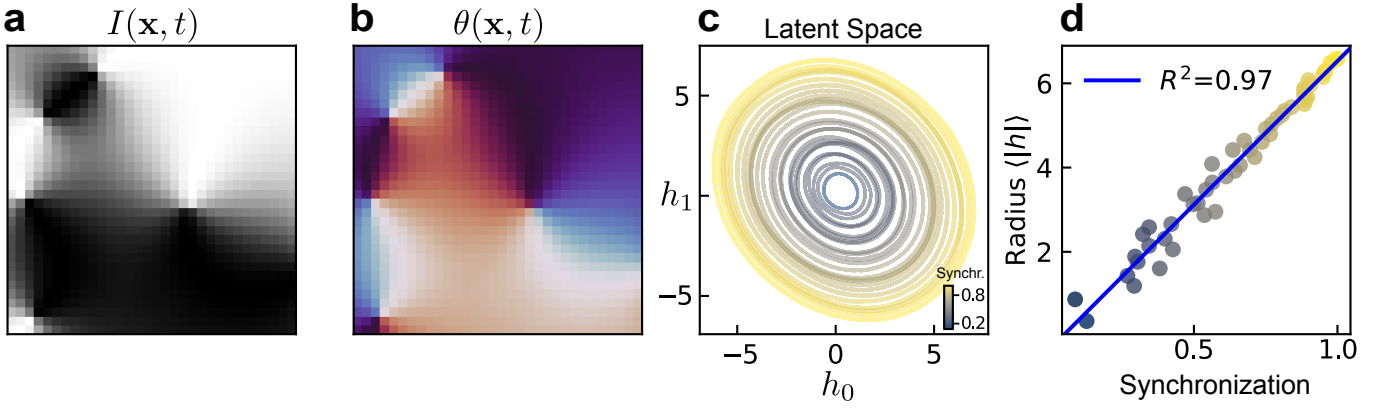


FIG. S11. **VIB applied to simulated Kuramoto model** Inputs are made to mimic the intensity field of the cyanobacteria data. The intensity field (a) is derived from the phase field (b) according to $I(\mathbf{x}, t) = \frac{1}{2} \cos \theta(\mathbf{x}, t) - 1$. (c) DVIB trained with two latent variables learns oscillations. Here each trajectory is colored by its synchronization order parameter. (d) Latent oscillations are very highly correlated with synchronization order parameter.

Pitchfork bifurcation simulations The system was evolved according to the stochastic differential equation Eq. J1 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. J1 with the force given by Eq. F12 in the main text, 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 $dt = 2 \cdot 10^{-3}$.

Lorenz system simulations The system was evolved according to

$$\dot{x} = \sigma(y - x) \quad (J2)$$

$$\dot{y} = x(\rho - z) - y \quad (J3)$$

$$\dot{z} = xy - \beta z \quad (J4)$$

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 [92].

Fluid flow simulations The fluid flow simulations are contained in the “Cylinder in Crossflow” dataset, downloaded from Ref. [93]. 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 F).

Locally-coupled Kuramoto model In SI Fig. S11 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 $J = 1.0$.

Appendix K: Comparison of various dimensionality reduction methods

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. S12 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 [94]. In the dynamical systems literature, PCA is also known as a proper orthogonal decomposition (POD) [95]. 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 $\mathbf{C} = \mathbf{X}^T \mathbf{X}$. Instead, a more computationally efficient way is to compute the singular value decomposition (SVD) of the data matrix,

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

The principal components are given by columns of $\mathbf{V}$, and the singular values in the matrix $\mathbf{\Sigma}$ 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. [73]. 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 [11, 12]. 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 [19].

To find the matrix $\mathbf{A}_{\text{DMD}}$, 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 $\mathbf{A}_{\text{DMD}}$ 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 $\mathbf{A}_{\text{DMD}}$ 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 [13]. 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 [96]. 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” [10]. 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

$$\mathbf{x} = \mathbf{A}_{\text{ICA}} \mathbf{s}. \quad (\text{K1})$$

Here, $\mathbf{x}$ denotes the recorded signal, $\mathbf{s}$ denotes the independent sources (conversations), and $\mathbf{A}_{\text{ICA}}$ 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 [10].

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 $\mathcal{J}_{\text{neg}}$ 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 $\mathcal{J}_{\text{neg}}$, see Ref. [10] 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 [10]. 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 $\mathbf{A}_{\text{ICA}}$ cannot be estimated if

the components $\mathbf{s}$ are normal Gaussian variables: any orthogonal matrix will satisfy (K1). 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 $\mathbf{z}_t$ [10]. 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 $\mathbf{A}_{\text{ICA}}$ 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 $\mathbf{A}_{\text{DMD}}$ [73]. Thus, the two methods are equivalent.

Diffusion maps

Diffusion maps are a technique which attempts to approximate the Perron-Frobenius operator, or rather the integral kernel $p(x_{t+\Delta t} | x_t)$ [14, 97]. 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 [14, 97]. 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 [98, 99]. 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 [100]. This is done by assuming that the latent variable is low dimensional, and optimizing the objective

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

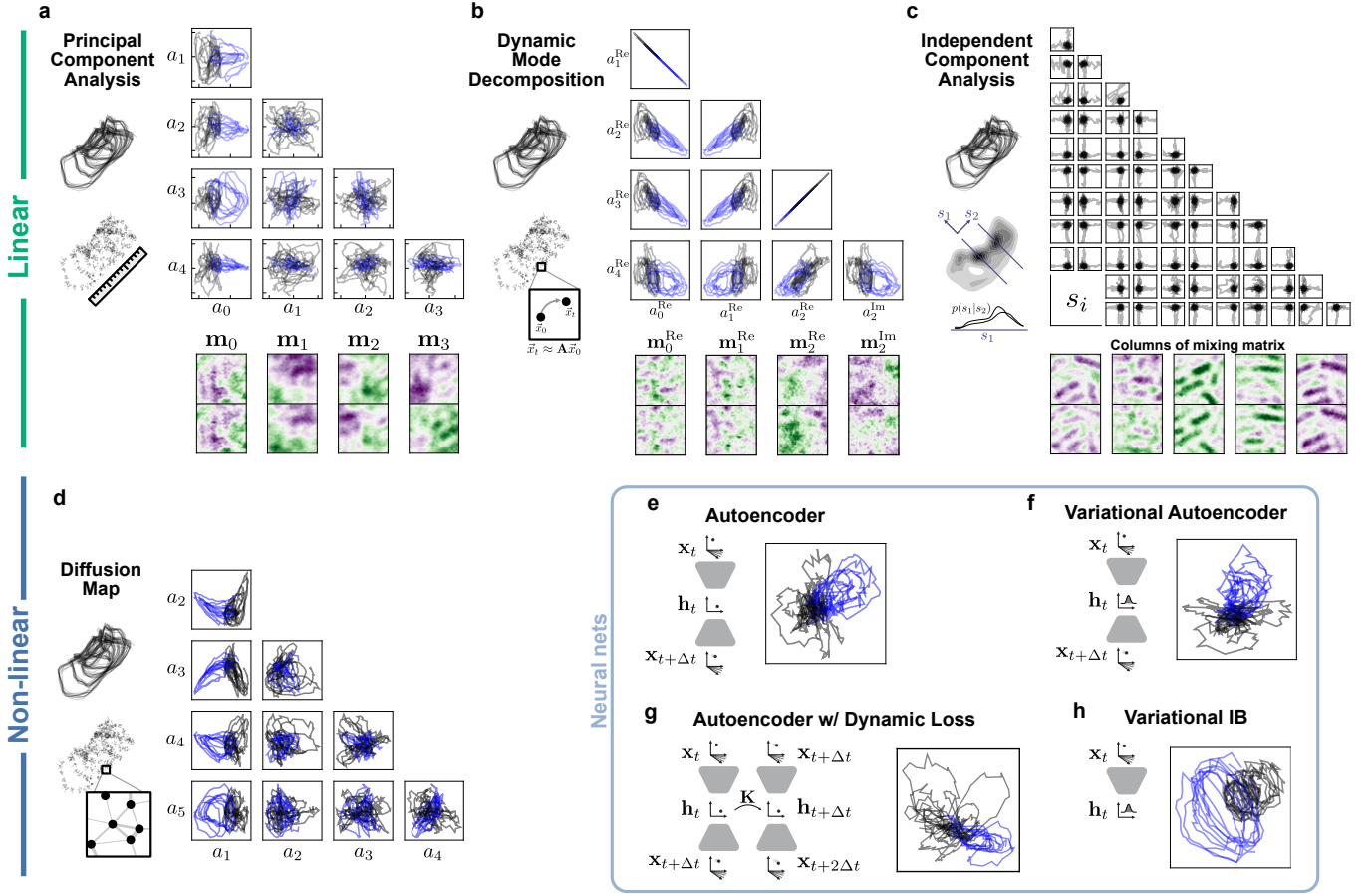


FIG. S12. Comparison of various dimensionality reduction techniques on cyanobacteria dynamics Top row illustrates linear methods, bottom row shows non-linear methods. For each method, we plot the trajectories of all cyanobacteria colonies when projected onto the dominant modes (whose definition depends 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 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. [20] 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).

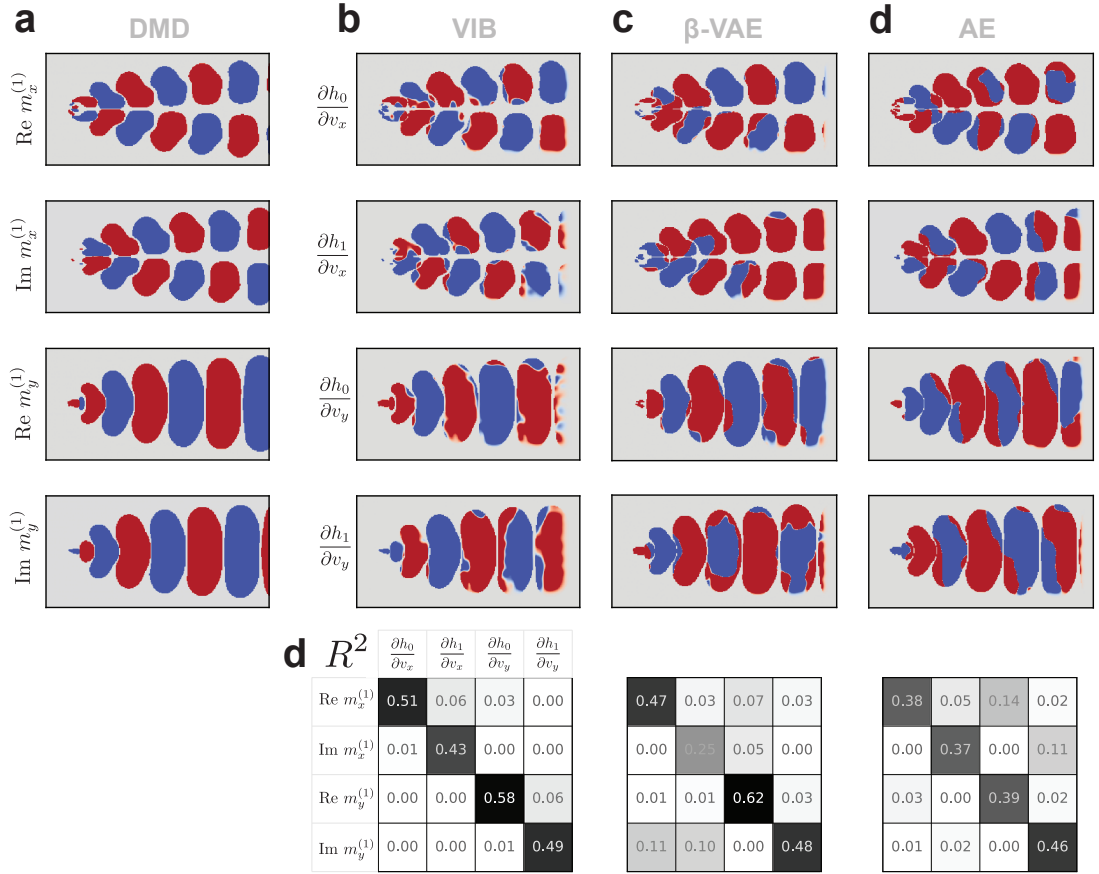


FIG. S13. **Learned latent variables for fluid flows with various deep neural networks.** To compare latent variables we compare their gradients as in Fig. 5 of the main text, which should correspond to Koopman modes. Here we show the gradients masked by regions in the DMD mode which have large amplitude. (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.

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 [83]. With $\beta = 1$, which is the case for the original VAE [100], 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 [101]. 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 β -VAEs learn the same thing, some preliminary results in Fig. S13 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. S12).

Other neural networks

Other deep architectures can also be used for model reduction. For example, Ref. [23] 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, [20] 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. [21, 22]. As an illustration, we trained one such network on the cyanobacteria which can be seen in Fig. S12. 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.

-
- [1] Y. Kuramoto, *Chemical Oscillations, Waves, And Turbulence* (Springer, 1984).
 - [2] H. Haken, *Synergetics: An Introduction*, Springer Series in Synergetics (Springer Berlin Heidelberg, 1983).
 - [3] G. Pavliotis and A. Stuart, *Multiscale Methods: Averaging and Homogenization*, Texts in Applied Mathematics (Springer New York, 2010).
 - [4] D. Givon, R. Kupferman, and A. Stuart, *Nonlinearity* **17**, R55–R127 (2004).
 - [5] Y. Kuramoto and H. Nakao, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **377**, 20190041 (2019).
 - [6] 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).
 - [7] Y. Kuramoto, *Progress of Theoretical Physics Supplement* **99**, 244–262 (1989).
 - [8] H. Mori and Y. Kuramoto, *Dissipative Structures and Chaos*, 93–117 (1998).
 - [9] L.-Y. Chen, N. Goldenfeld, and Y. Oono, *Physical Review E* **54**, 376–394 (1996).
 - [10] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis* (John Wiley & Sons, Ltd, 2001).
 - [11] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Journal of Fluid Mechanics* **641**, 115–127 (2009).
 - [12] P. J. Schmid, *Journal of Fluid Mechanics* **656**, 5–28 (2010).
 - [13] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *Journal of Nonlinear Science* **25**, 1307 (2015).
 - [14] R. R. Coifman and S. Lafon, *Applied and Computational Harmonic Analysis* **21**, 5 (2006), special Issue: Diffusion Maps and Wavelets.
 - [15] M. Cenedese, J. Axås, B. Bäuerlein, K. Avila, and G. Haller, *Nature Communications* **13** (2022), 10.1038/s41467-022-28518-y.
 - [16] S. Jain and G. Haller, *Nonlinear Dynamics* **107**, 1417–1450 (2021).
 - [17] S. E. Otto and C. W. Rowley, *Annual Review of Control, Robotics, and Autonomous Systems* **4**, 59–87 (2021).
 - [18] 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>.
 - [19] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *SIAM Review* **64**, 229 (2022).
 - [20] 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).
 - [21] B. Lusch, J. N. Kutz, and S. L. Brunton, *Nature Communications* **9**, 4950 (2018).
 - [22] 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>.
 - [23] P. R. Vlachas, G. Arampatzis, C. Uhler, and P. Koumoutsakos, *Nature Machine Intelligence* **4**, 359 (2022).
 - [24] 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).
 - [25] P. Cvitanović and D. Lippolis, *AIP Conference Proceedings* (2012), 10.1063/1.4745574.
 - [26] P. Gaspard and X.-J. Wang, *Physics Reports* **235**, 291 (1993).
 - [27] N. Jayant, J. Johnston, and R. Safranek, *Proceedings of the IEEE* **81**, 1385 (1993).
 - [28] F. Creutzig and H. Sprekeler, *Neural Computation* **20**, 1026 (2008).
 - [29] F. Creutzig, A. Globerson, and N. Tishby, *Phys. Rev. E* **79**, 041925 (2009).
 - [30] 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).
 - [31] W. Bialek, *Biophysics: searching for principles* (Princeton University Press, 2012).

- [32] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise* (Springer, 1998) p. 492.
- [33] P. Gaspard, G. Nicolis, A. Provata, and S. Tasaki, *Phys. Rev. E* **51**, 74 (1995).
- [34] G. Froyland, *Physica D: Nonlinear Phenomena* **250**, 1 (2013).
- [35] 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).
- [36] 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.
- [37] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, *Journal of Machine Learning Research* **6**, 165 (2005).
- [38] V. Ngampuetikorn and D. J. Schwab, in *Advances in Neural Information Processing Systems*, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (2021).
- [39] S. E. Palmer, O. Marre, M. J. Berry, and W. Bialek, *Proceedings of the National Academy of Sciences* **112**, 6908 (2015).
- [40] V. Sachdeva, T. Mora, A. M. Walczak, and S. E. Palmer, *PLOS Computational Biology* **17**, 1 (2021).
- [41] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, in *International Conference on Learning Representations* (2017).
- [42] J.-W. Chu and G. A. Voth, *Biophysical Journal* **93**, 3860 (2007).
- [43] T. Cover and J. Thomas, *Elements of Information Theory* (Wiley, 2012).
- [44] S. Ulam, *Problems in Modern Mathematics* (Science Editions, 1960).
- [45] T.-Y. Li, *Journal of Approximation Theory* **17**, 177 (1976).
- [46] J. Crutchfield and N. Packard, *Physica D: Nonlinear Phenomena* **7**, 201–223 (1983).
- [47] 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>.
- [48] T. Ahamed, A. C. Costa, and G. J. Stephens, *Nature Physics* **17**, 275 (2021).
- [49] 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.
- [50] 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).
- [51] G. Froyland, *Physica D: Nonlinear Phenomena* **200**, 205 (2005).
- [52] D. E. Gökmen, Z. Ringel, S. D. Huber, and M. Koch-Janusz, *Phys. Rev. Lett.* **127**, 240603 (2021).
- [53] D. E. Gökmen, Z. Ringel, S. D. Huber, and M. Koch-Janusz, *Phys. Rev. E* **104**, 064106 (2021).
- [54] D. E. Gökmen, 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).
- [55] 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>.
- [56] 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-pdf/doi/10.1063/1.4923066/15497236/010901_1_online.pdf.
- [57] 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 \[physics.comp-ph\]](https://arxiv.org/abs/2006.06154).
- [58] J. Albright, “Flow Visualization in a Water Channel,” (2017), @jacobalbright3585.
- [59] @NOAASatellites, “Earth from Orbit: von Kármán Vortices,” <https://www.youtube.com/watch?v=SawKLWT1bDA> (2021), [Accessed 03-Sept-2023].
- [60] 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).
- [61] J. Chew, E. Leypunskiy, J. Lin, A. Murugan, and M. J. Rust, *Nature Communications* **9**, 3004 (2018).
- [62] L. Ren and G. B. Ermentrout, *SIAM Journal on Mathematical Analysis* **29**, 208 (1998).
- [63] H. Hong, H. Park, and M. Y. Choi, *Physical Review E* **72**, 036217 (2005).
- [64] C. R. Laing, *Physica D: Nonlinear Phenomena* **240**, 1960 (2011).
- [65] K. K. Chen, J. H. Tu, and C. W. Rowley, *Journal of Nonlinear Science* **22**, 887 (2012).
- [66] J. N. Kutz, X. Fu, and S. L. Brunton, “Multi-resolution dynamic mode decomposition,” (2015), [arXiv:1506.00564 \[math.DS\]](https://arxiv.org/abs/1506.00564).
- [67] A. Wynn, D. S. Pearson, B. Ganapathisubramani, and P. J. Goulart, *Journal of Fluid Mechanics* **733**, 473–503 (2013).
- [68] M. J. Colbrook, *SIAM Journal on Numerical Analysis* **61**, 1585 (2023), <https://doi.org/10.1137/22M1521407>.
- [69] 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.
- [70] G. J. Stephens, B. Johnson-Kerner, W. Bialek, and W. S. Ryu, *PLOS Computational Biology* **4**, 1 (2008).
- [71] 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>.
- [72] 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>.
- [73] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, *Journal of Nonlinear Science* **28**, 985 (2018).
- [74] 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>.
- [75] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Machine Learning: Science and Technology* **2**, 035023 (2021).
- [76] J. Page and R. R. Kerswell, *Journal of Fluid Mechanics* **879**, 1–27 (2019).
- [77] L. van der Maaten and G. Hinton, *Journal of Machine Learning Research* **9**, 2579 (2008).

- [78] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” (2020), [arXiv:1802.03426 \[stat.ML\]](https://arxiv.org/abs/1802.03426).
- [79] 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/Theses/Slonim_PhD.pdf.
- [80] A. Gordon, A. Banerjee, M. Koch-Janusz, and Z. Ringel, *Phys. Rev. Lett.* **126**, 240601 (2021).
- [81] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” (2019), [arXiv:1807.03748 \[cs.LG\]](https://arxiv.org/abs/1807.03748).
- [82] 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).
- [83] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, in *International Conference on Learning Representations* (2017).
- [84] J. Mikhaeil, 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.
- [85] G. Koppe, H. Toutounji, P. Kirsch, S. Lis, and D. Durstewitz, *PLOS Computational Biology* **15**, 1 (2019).
- [86] 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.
- [87] H. Eivazi, S. Le Clainche, S. Hoyas, and R. Vinuesa, *Expert Systems with Applications* **202**, 117038 (2022).
- [88] M. Rolinek, D. Zietlow, and G. Martius, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [89] S. H. Strogatz, *Nonlinear Dynamics and Chaos* (CRC Press, 2018) p. 532.
- [90] E. N. Lorenz, *Journal of the Atmospheric Sciences* **20**, 130–141 (1963).
- [91] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *Journal of Computational Dynamics* **1**, 391 (2014).
- [92] 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.
- [93] J. Tencer, “Cylinder in Crossflow,” (2020).
- [94] C. M. Bishop, *Pattern recognition and machine learning* (Springer, 2006).
- [95] G. Berkooz, P. Holmes, and J. L. Lumley, *Annual Review of Fluid Mechanics* **25**, 539 (1993).
- [96] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, *Nature Communications* **8**, 19 (2017).
- [97] 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>.
- [98] P. Y. Lu, S. Kim, and M. Soljačić, *Phys. Rev. X* **10**, 031056 (2020).
- [99] 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>.
- [100] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” (2022), [arXiv:1312.6114 \[stat.ML\]](https://arxiv.org/abs/1312.6114).
- [101] 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).