

A Analytical KEF derivation in 1D bistable system

We would like to find an analytical Koopman eigenfunction for the scalar dynamical system:

$$\dot{x} = x - x^3 \quad (17)$$

In the 1D case, the Koopman PDE (3) reduces to a first-order ordinary differential equation

$$\frac{d\psi}{dx} f(x) = \lambda \psi(x). \quad (18)$$

With $f(x) = x - x^3$ and $\lambda = 1$ we have:

$$\psi'(x)(x - x^3) = \psi(x) \quad (19)$$

$$\Rightarrow \frac{\psi'(x)}{\psi(x)} = \frac{1}{x - x^3} \quad (20)$$

To solve this integral we first simplify the integrand.

$$x - x^3 = x(1 - x^2) = x(1 - x)(1 + x) \quad (21)$$

So,

$$\frac{1}{x(1 - x)(1 + x)} = \frac{A}{x} + \frac{B}{1 - x} + \frac{C}{1 + x} \quad (22)$$

Solving for A, B, C yields $A = 1, B = \frac{1}{2}, C = -\frac{1}{2}$.

Now we can integrate,

$$\int \frac{1}{x - x^3} dx = \int \left(\frac{1}{x} + \frac{1}{2(1 - x)} - \frac{1}{2(1 + x)} \right) dx \quad (23)$$

$$= \log|x| - \frac{1}{2} \log|1 - x| - \frac{1}{2} \log|1 + x| + C \quad (24)$$

$$\log \psi(x) = \log|x| - \frac{1}{2} \log|1 - x| - \frac{1}{2} \log|1 + x| + C \quad (25)$$

$$\Rightarrow \psi(x) = C' \cdot \frac{|x|}{\sqrt{|1 - x^2|}} \quad (26)$$

To bring it into the form in the main text we use the product composition rule (9). We can multiply our solution by the $\text{sign}(x)$ function which is a $\lambda = 0$ eigenfunction because it remains constant in each basin (see Figure 3A). In other words, we flip the sign of our solution $\psi(x) \rightarrow -\psi(x)$ for $x < 0$.

$$\psi(x) = C' \frac{x}{\sqrt{|1 - x^2|}} \quad (27)$$

and this remains a KEF with $\lambda = 1$.

B Eigenfunction Degeneracy in higher dimensions

Consider a separable 2D dynamical system:

$$\dot{x} = f_x(x), \quad (28)$$

$$\dot{y} = f_y(y), \quad (29)$$

which we write compactly as:

$$\dot{\mathbf{x}} = \mathbf{f}(x, y) = \begin{bmatrix} f_x(x) \\ f_y(y) \end{bmatrix}. \quad (30)$$

We seek a Koopman eigenfunction $\psi(x, y)$ satisfying:

$$\nabla\psi \cdot \mathbf{f}(x, y) = \lambda\psi(x, y). \quad (31)$$

Assume $\lambda = 1$ and a separable form $\psi(x, y) = X(x)Y(y)$. Then:

$$\frac{\partial\psi}{\partial x} = X'(x)Y(y), \quad (32)$$

$$\frac{\partial\psi}{\partial y} = X(x)Y'(y), \quad (33)$$

$$\nabla\psi \cdot \mathbf{f} = X'(x)Y(y)f_x(x) + X(x)Y'(y)f_y(y) = X(x)Y(y). \quad (34)$$

Dividing both sides by $X(x)Y(y)$ gives:

$$\frac{X'(x)}{X(x)}f_x(x) + \frac{Y'(y)}{Y(y)}f_y(y) = 1. \quad (35)$$

The above equation requires that the sum of the above two terms, which each depend on different variables must be 1 for all x, y . It follows that each term is also a constant function.

$$\frac{X'(x)}{X(x)}f_x(x) = \mu, \quad (36)$$

$$\frac{Y'(y)}{Y(y)}f_y(y) = 1 - \mu, \quad (37)$$

for an arbitrary constant $\mu \in \mathbb{R}$.

Define the antiderivatives:

$$A(x) = \int \frac{1}{f_x(x)} dx, \quad (38)$$

$$B(y) = \int \frac{1}{f_y(y)} dy. \quad (39)$$

Then the logarithms of the separated components are:

$$\log X(x) = \mu A(x) \quad \Rightarrow \quad X(x) = \left(e^{A(x)}\right)^\mu, \quad (40)$$

$$\log Y(y) = (1 - \mu)B(y) \quad \Rightarrow \quad Y(y) = \left(e^{B(y)}\right)^{1-\mu}. \quad (41)$$

Thus, the general separable Koopman eigenfunction is:

$$\psi(x, y) = \left(e^{A(x)}\right)^\mu \cdot \left(e^{B(y)}\right)^{1-\mu}. \quad (42)$$

C Relation of our definition to the Koopman Operator

In the main text, we introduced Koopman eigenfunctions as scalar functions $\psi : \mathcal{X} \rightarrow \mathbb{R}$ that evolve exponentially along trajectories $\mathbf{x}(t) \in \mathcal{X}$ of a dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$:

$$\frac{d}{dt}\psi(\mathbf{x}(t)) = \lambda\psi(\mathbf{x}(t)). \quad (43)$$

Here, we clarify the origin of this equation by defining the Koopman operator, linking our approach to the broader theory.

Let $g : \mathcal{X} \rightarrow \mathbb{R}$ be a real-valued function of the system state—commonly referred to as an observable. The collection of such observables forms an infinite-dimensional function space, typically a Hilbert space once equipped with an inner product $\langle g, g' \rangle$. The Koopman operator acts linearly on this space.

For a continuous-time system, the Koopman operator \mathcal{K}_τ evolves observables according to the flow map $F_\tau : \mathcal{X} \rightarrow \mathcal{X}$, which advances the state forward by time τ :

$$(\mathcal{K}_\tau g)(\mathbf{x}(t)) = g(F_\tau(\mathbf{x}(t))) = g(\mathbf{x}(t + \tau)). \quad (44)$$

The infinitesimal generator of the Koopman semigroup $\{\mathcal{K}_\tau\}_{\tau \geq 0}$, often denoted simply as \mathcal{K} , is defined as:

$$\mathcal{K}g := \lim_{\tau \rightarrow 0} \frac{\mathcal{K}_\tau g - g}{\tau} = \lim_{\tau \rightarrow 0} \frac{g(F_\tau(\mathbf{x})) - g(\mathbf{x})}{\tau}. \quad (45)$$

When evaluated along a trajectory $\mathbf{x}(t)$, this yields:

$$\mathcal{K}g(\mathbf{x}(t)) = \lim_{\tau \rightarrow 0} \frac{g(\mathbf{x}(t + \tau)) - g(\mathbf{x}(t))}{\tau} = \frac{d}{dt}g(\mathbf{x}(t)) = \nabla g \cdot \dot{\mathbf{x}}(t) = \nabla g \cdot \mathbf{f}(\mathbf{x}(t)). \quad (46)$$

This operator is also known as the Lie derivative of g along the vector field \mathbf{f} .

Thus, an eigenfunction ψ of \mathcal{K} satisfying

$$\mathcal{K}\psi = \lambda\psi \quad (47)$$

recovers the Koopman eigenfunction equation (3) used in the main text.

D KEF degeneracy in randomly initialised DNN solutions

Main text Figure 3 illustrates challenges arising due to the degeneracy of the Koopman PDE (3). In Figure 8, we train several DNNs on a 2-unit GRU trained on the 2BFF. Each DNN is independently initialised and trained on a single distribution without the balance regularisation term \mathcal{L}_{bal} , i.e., $\gamma_{\text{bal}} = 0$. The resulting KEF approximations exhibit the same modes of degeneracy - zero on certain basins as well as vertical and horizontal variants.

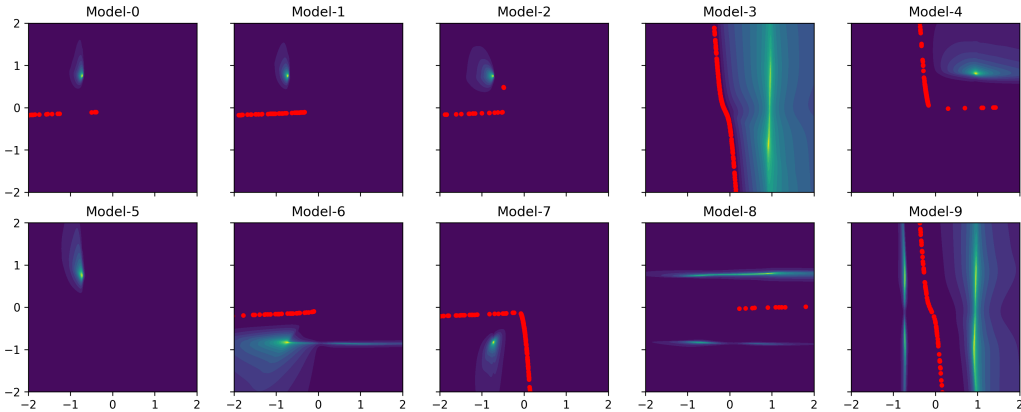


Figure 8: Many KEFs of to for 2 bit flip flop in 2D

E Curve-based validation approach

In high dimension, we cannot visualize the entire phase space to check whether zeros of the KEF coincide with the separatrix. Instead, we generate a family of smooth curves that connect two attractors, and hence must pass through a separatrix. In the 64D GRU flip flop example, the two attractors are the two stable fixed points $x, y \in \mathbb{R}^N$. We use *cubic Hermite interpolation* with randomized tangent vectors at the endpoints. Each curve is defined by:

$$H(\alpha) = h_{00}(\alpha)x + h_{10}(\alpha)m_x + h_{01}(\alpha)y + h_{11}(\alpha)m_y, \quad \alpha \in [0, 1] \quad (48)$$

where the Hermite basis functions are:

$$h_{00}(\alpha) = 2\alpha^3 - 3\alpha^2 + 1, \quad (49)$$

$$h_{01}(\alpha) = -2\alpha^3 + 3\alpha^2, \quad (50)$$

$$h_{10}(\alpha) = \alpha^3 - 2\alpha^2 + \alpha, \quad (51)$$

$$h_{11}(\alpha) = \alpha^3 - \alpha^2. \quad (52)$$

Notice that $H(0) = x$ and $H(1) = y$.

The tangent vectors m_x and m_y are initialized as $y - x$ and perturbed with Gaussian noise:

$$m_x = (y - x) + \epsilon_x, \quad m_y = (y - x) + \epsilon_y, \quad \epsilon_x, \epsilon_y \sim \mathcal{N}(0, \sigma^2 I). \quad (53)$$

We sample multiple such curves with independently drawn perturbations. This produces a family of curves that interpolate between x and y , while varying in geometry, enabling randomized exploration of intermediate regions in state space. Crucially, the curves are not limited to the manifold spanned by the attractors, but extend to all dimensions (controlled by σ). Optional constraints (e.g., non-negativity) can be imposed by rejecting any curve that violates them. For each such curve, we both evaluate the KEF and simulate the ODE to determine the position of the separatrix.

F Neural network architectures

In most of the demonstrations we use a ResNet architecture [1] with a tanh activation function.

Let the input to the network be $\mathbf{x}_{\text{in}} \in \mathbb{R}^{d_{\text{in}}}$ and define the hidden layer activations $\mathbf{x}_{\text{hid}} \in \mathbb{R}^{d_{\text{hid}}}$, output dimension $\mathbf{x}_{\text{out}} \in \mathbb{R}^{d_{\text{out}}}$, and number of layers L .

The network receives inputs at the first layer $\mathbf{x}^{(0)} = \text{Pad}(\mathbf{x}_{\text{in}})$, where Pad appends zeroes to the input as we always choose $d_{\text{hid}} > d_{\text{in}}$. The network then transforms the input at each layer l ,

$$\mathbf{x}^{(\ell+1)} = \mathbf{x}^{(\ell)} + \tanh\left(W^{(\ell)}\mathbf{x}^{(\ell)} + \mathbf{b}^{(\ell)}\right), \quad \ell = 1, \dots, L-1, \quad (54)$$

where $W^{(l)} \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{hid}}}$ and $\mathbf{b} \in \mathbb{R}^{d_{\text{hid}}}$

The output is the final Residual layer activation after applying another linear layer to match the desired d_{out} :

$$\mathbf{x}_{\text{out}} = W^{\text{out}}\mathbf{x}^{(L)} + \mathbf{b}^{\text{out}}, \quad (55)$$

where $W^{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ and $\mathbf{b}^{\text{out}} \in \mathbb{R}^{d_{\text{out}}}$. During optimization, gradients $\nabla\theta\mathcal{L}_{\text{total}}$ are computed for all parameters $\theta = (W^{(1:L-1)}, \mathbf{b}^{(1:L-1)}, W^{\text{out}}, \mathbf{b}^{\text{out}})$.

Choices for each system

For the results in Figures 2, 4, 7, we use $L = 20$, $d_{\text{hid}} = 400$, $d_{\text{out}} = 1$ and $d_{\text{in}} = N$ the dimension of the dynamical system. For the gLV system in Figure 5 we use $L = 25$ and $d_{\text{hid}} = 1000$.

Radial Basis Function (RBF) Layer

For the limit cycles example in Figure 6 we use a single Radial Basis Function layer [4].

Given an input $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$, the RBF layer maps it to an output $\mathbf{y} \in \mathbb{R}^{d_{\text{out}}}$ through a set of M radial basis functions, each centered at $\mathbf{c}_i \in \mathbb{R}^{d_{\text{in}}}$, with a shape parameter $\varepsilon_i > 0$ and linear combination weights $a_{ij} \in \mathbb{R}$.

To compute RBF activations for $i = 1, \dots, M$, we define the scaled radial distance:

$$s_i(\mathbf{x}) = \varepsilon_i \cdot \|\mathbf{x} - \mathbf{c}_i\|, \quad (56)$$

and then apply a gaussian radial basis function

$$\varphi_i(\mathbf{x}) = \exp(-s_i(\mathbf{x})^2). \quad (57)$$

The final output is a linear combination of the basis activations:

$$y_j(\mathbf{x}) = \sum_{i=1}^M a_{ji} \cdot \varphi_i(\mathbf{x}), \quad j = 1, \dots, d_{\text{out}} \quad (58)$$

We use $M = 300$, and $d_{\text{out}} = 1$. During optimization, gradients $\nabla \theta \mathcal{L}_{\text{total}}$ are computed for all parameters $\theta = (\{a_{ji}\}, \{\mathbf{c}_i\}, \{\varepsilon_i\})$.

G Optimisation

We minimise the total loss:

$$\mathcal{L}_{\text{total}} = \sum_{j=1}^J \mathcal{L}_{\text{ratio}}^j + \gamma_{\text{bal}} \mathcal{L}_{\text{bal}}^j. \quad (59)$$

where j corresponds to the j^{th} sampling distribution (see main text section ??). B N -dimensional points in the state space \mathcal{X} are sampled from each distribution $\mathbf{x}^j \sim p_j(\mathbf{x})$. The ratio loss is the Koopman PDE error, normalised by a sample-shuffled version:

$$\mathcal{L}_{\text{ratio}}^j = \frac{\sum_{i=1}^B (\text{LHS}_i^j - \text{RHS}_i^j)^2}{\sum_{i=1}^B (\text{LHS}_i^j - \text{RHS}_{\text{perm}(i)}^j)^2} \quad (60)$$

$$\text{LHS}_i^j = \nabla \psi(\mathbf{x}_i^j) \cdot f(\mathbf{x}_i^j) \quad \text{left-hand-side of the Koopman PDE (3)} \quad (61)$$

$$\text{RHS}_i^j = \lambda \psi(\mathbf{x}_i^j) \quad \text{right-hand-side of the Koopman PDE (3)} \quad (62)$$

where $\text{perm}(i)$ is a random permutation of the numbers $1, 2, \dots, B$ sampled during each training iteration.

The balance regularisation loss is the squared mean of the KEF values divided by their variance:

$$\mathcal{L}_{\text{bal}}^j = \frac{(\bar{\psi}^j)^2}{\frac{1}{B} \sum_{i=1}^B (\psi(\mathbf{x}_i^j) - \bar{\psi}^j)^2}, \quad (63)$$

$$\bar{\psi}^j = \frac{1}{B} \sum_{i=1}^B \psi(\mathbf{x}_i^j). \quad (64)$$

Table 1: Algorithm details and hyperparameters for various systems. System dimensionality N , Koopman eigenvalue λ , balance regularisation weight γ_{bal} , batch-size B , training iterations T , learning rate η , ResNet depth L and width d_{hid} , number of Radial Basis Functions M .

Dynamical System	N	λ	γ_{bal}	B	T	η	L	d_{hid}	M
Bistable 1D	1	1	5×10^{-2}	1000	1000	10^{-4}	20	400	–
Damped Duffing oscillator	2	1	5×10^{-2}	1000	1000	10^{-4}	20	400	–
1BFF, 2D GRU	2	1	5×10^{-2}	1000	1000	10^{-4}	20	400	–
2BFF, 3D GRU	3	0.2	5×10^{-2}	1000	1000	10^{-4}	20	400	–
1BFF, 64D	64	0.1	5×10^{-2}	1000	1000	10^{-4}	20	400	–
Two Limit Cycles	2	1	0	1000	1000	10^{-4}	–	–	300
Ecology gLV [5, 2]	11	0.1	5×10^{-2}	5000	5000	10^{-4}	25	1000	–

In general we set $\gamma_{\text{bal}} = 0.05$. For the limit cycles Figure 6 we set $\gamma_{\text{bal}} = 0$.

We compute $\nabla\psi(x)$ using Pytorch’s `torch.autograd.grad`, specifying `create_graph=True`, since we differentiate through this a second time to compute the gradients $\nabla_{\theta}\mathcal{L}_{\text{total}}$ with respect to the neural network parameters θ .

We use the Adam optimiser [3] with learning rate 10^{-4} and l2 normalisation 10^{-5} . We use $B = 1000$ and train for 1000 iterations.

Only in the case of the 11D gLV, Figure 5 we use $B = 5000$ and train for 5000 iterations.

A summary of all hyperparameters is provided in Table 1.

H Computational Resources and run time

We ran all experiments on a system with four GeForce GTX 1080 GPUs with 10 Gbps of memory each.

All the 2D systems take 1-5 minutes to train the KEFs. The 11D gLV takes up to 20 minutes. The 64D RNN performing 1BFF takes 5 minutes to train the KEF.

I Choice of eigenvalue λ for numerics

In the main text we look for approximations to the Koopman PDE (3) for a real positive eigenvalue λ . What should the value of λ be? It is known that products of KEFs are KEFs themselves with different eigenvalues. In particular, for a KEF ψ with eigenvalue λ , we see that:

$$\nabla[\psi(x)^{\alpha}] \cdot f(x) = \alpha\psi(x)^{\alpha-1}\nabla\psi(x) \cdot f(x) \quad (65)$$

$$= \alpha\lambda\psi(x)^{\alpha} \quad (66)$$

Therefore, $\psi(x)^{\alpha}$ is also a Koopman eigenfunction, with eigenvalue $\alpha\lambda$. This translates to changes in the shape of the KEF, i.e., the sharpness of the peaks, while maintaining the position of the zeroes.

In practice the choice of λ affects training convergence, and it is therefore an important hyperparameter in the optimisation procedure (see Figure 9). We attribute this to the time scale of interest in the system $\dot{\mathbf{x}} = f(\mathbf{x})$, and differences in the propagation of gradients for different λ .

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern*

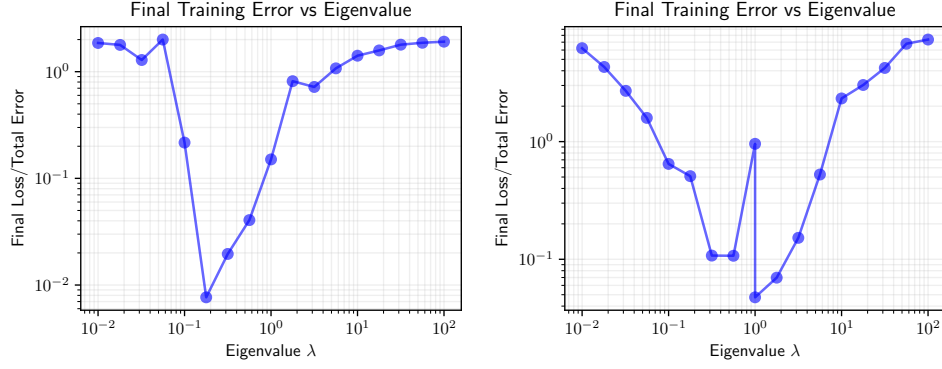


Figure 9: Training convergence as a function of eigenvalue λ , evaluated by normalised PDE error $\mathcal{L}_{\text{ratio}}$ 6 for two systems: LEFT, 1D bistable system $\dot{x} = x - x^3$ (see Figure 2) and 2BFF GRU 3D (see Figure 4).

- Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL <http://ieeexplore.ieee.org/document/7780459/>.
- [2] Eric W. Jones and Jean M. Carlson. Steady-state reduction of generalized Lotka-Volterra systems in the microbiome. *Physical Review E*, 99(3):032403, March 2019. doi: 10.1103/PhysRevE.99.032403. URL <https://link.aps.org/doi/10.1103/PhysRevE.99.032403>. Publisher: American Physical Society.
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL <http://arxiv.org/abs/1412.6980>. arXiv:1412.6980 [cs].
- [4] M. J. D. Powell. Radial basis functions for multivariable interpolation: a review. In *Algorithms for approximation*, pages 143–167. Clarendon Press, USA, January 1987. ISBN 978-0-19-853612-3.
- [5] Richard R. Stein, Vanni Bucci, Nora C. Toussaint, Charlie G. Buffie, Gunnar Rättsch, Eric G. Pamer, Chris Sander, and João B. Xavier. Ecological Modeling from Time-Series Inference: Insight into Dynamics and Stability of Intestinal Microbiota. *PLOS Computational Biology*, 9(12):e1003388, December 2013. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1003388. URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003388>. Publisher: Public Library of Science.