

Learning beyond experience: Generalizing to unseen state space with reservoir computing

Cite as: Chaos 35, 103146 (2025); doi: 10.1063/5.0283421

Submitted: 31 May 2025 · Accepted: 1 October 2025 ·

Published Online: 28 October 2025



View Online



Export Citation



CrossMark

Declan A. Norton,^{1,2} Yuanzhao Zhang,³ and Michelle Girvan^{1,2,3,4,a)}

AFFILIATIONS

¹Department of Physics, University of Maryland, College Park, Maryland 20742, USA

²Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, Maryland 20742, USA

³Santa Fe Institute, Santa Fe, New Mexico 87501, USA

⁴Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA

Note: This paper is part of the Special Topic on Nonlinear Dynamics of Reservoir Computing: Theory, Application and Realization.

a) Author to whom correspondence should be addressed: girvan@umd.edu

ABSTRACT

Machine learning techniques offer an effective approach to modeling dynamical systems solely from observed data. However, without explicit structural priors—built-in assumptions about the underlying dynamics—these techniques typically struggle to generalize to aspects of the dynamics that are poorly represented in the training data. Here, we demonstrate that reservoir computing—a simple, efficient, and versatile machine learning framework often used for data-driven modeling of dynamical systems—can generalize to unexplored regions of state space without explicit structural priors. First, we describe a multiple-trajectory training scheme for reservoir computers that supports training across a collection of disjoint time series, enabling the effective use of available training data. Then, applying this training scheme to multistable dynamical systems, we show that reservoir computers trained on trajectories from a single basin of attraction can achieve out-of-domain generalization by capturing system behavior in entirely unobserved basins.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0283421>

A plethora of machine learning (ML) techniques have shown impressive success in modeling complex dynamical systems from observed data alone—a challenge of broad practical importance, e.g., in climate science, public health, economics, ecology, and neuroscience. However, these so-called “black-box” models, which do not incorporate guidance based on known or suspected features of the system, often require extensive training data that cover the range of possible system behaviors. When the training domain is incomplete, they often fail outside of it, whereas models that incorporate knowledge of the system’s properties may still succeed. Here, we show that reservoir computing—a black-box ML approach commonly used to model dynamical systems—can overcome this limitation in important settings. By studying systems that can evolve toward multiple distinct long-term behaviors, we show that reservoir computers can make accurate predictions about behaviors they have never seen before, even when trained on limited data.

I. INTRODUCTION

Data-driven methods for modeling dynamical systems are essential in applications where a system of interest is complex or poorly understood and no sufficient knowledge-based (i.e., mathematical or physical) model is available. A number of machine learning (ML) techniques have proven effective for this purpose.^{1,2} When choosing among these ML approaches, however, one typically faces a trade-off between data-efficiency and model flexibility.

Unlike black-box approaches, methods that exploit partial knowledge of the system of interest through a structural prior, i.e., an explicit assumption or constraint about the system’s form, are often capable of generalizing to regions of the system’s state space not sampled in the training data (out-of-domain generalization),^{3–6} making them data-efficient and robust. Some such methods constrain the functional form of the ML model, e.g., sparse identification of nonlinear dynamics^{7,8} (SINDy) and next generation reservoir computing⁹ (NGRC), while others combine ML models with imperfect knowledge-based components in hybrid configurations.^{10–12} If

the inductive bias conferred to the model by its structural prior is inconsistent with the system of interest, however, the model performance can deteriorate substantially.³

On the other hand, black-box models that incorporate no explicit priors (but often have implicit inductive biases, which may be subtle^{13,14}) can be expressive enough to model diverse systems of interest, making them highly flexible. When presented with data outside of their training context, however, these models cannot rely on system-informed constraints to help them generalize. Thus, they typically perform poorly in regions of state space not well sampled by their training data.^{3–6,15,16} Black-box models that fall into this class include a broad set of techniques based on artificial neural networks, from recurrent architectures—e.g., reservoir computers (RCs), long short-term memory networks (LSTMs), and gated recurrent units (GRUs)—to feedforward methods such as neural ODEs.

Here, contrary to widely held assumptions, we demonstrate that reservoir computers (RCs)—a simple and efficient ML framework commonly used to learn and predict dynamical systems from observed time series^{17–20}—can generalize to unexplored regions of state space in many relevant settings, even without explicit structural priors to guide their behavior.

The simplicity of RCs makes them versatile, and they have been employed for a wide range of purposes: inferring unmeasured system variables from time series data,²¹ forecasting dynamics of extended networks²² or spatiotemporal systems,²³ separation of chaotic signals,²⁴ inferring network links,²⁵ and more.^{20,26,27} As with other black-box forecasting approaches, however, previous studies using RCs have focused mostly on monostable systems—those that exhibit a single stable long-term behavior, which is confined to a particular region of state space. For these systems, it suffices to train an RC on a single long time series that samples the relevant region well. Here, to test RCs' out-of-domain generalization ability, we apply reservoir computing to the challenging problem of basin prediction in “multistable” systems—that is, systems in which each trajectory evolves toward one of multiple distinct long-term behaviors, i.e., “attractors,” each confined to a different region of state space and having a corresponding “basin of attraction” containing all initial conditions whose trajectories converge to that attractor.

Because basins of attraction are non-overlapping, multistable systems present a natural setting to test RCs' ability to generalize to unexplored regions of state space. Such systems also arise frequently in important scenarios²⁸ (e.g., neuroscience,²⁹ gene regulatory networks,³⁰ cell differentiation and pattern formation,³¹ electrical grids,^{16,32} and financial markets³³) and are often too complex to confidently construct ML models with suitable structural priors. They remain underexplored, however, using black-box ML approaches.

In this paper, we describe a scheme to train RCs on a collection of disjoint time series, allowing for more flexible and exhaustive use of available data. This multiple-trajectory training has previously been applied to multi-task learning, where the goal is to train a single RC across multiple dynamical systems, each of which exhibits different dynamics.^{34–39} Here, we leverage the scheme's flexibility to improve sampling of the state space in multistable dynamical systems with short-lived transients. Then, we utilize the multistability of these systems to test when RCs can generalize from their training data to capture system dynamics in unexplored regions of

state space. Specifically, we show that an RC trained on trajectories from a single basin of attraction can recover the dynamics in other unexplored basins, capturing even fractal-like basin structures and unseen chaotic attractors.

II. CHALLENGES IN PREDICTING MULTISTABLE SYSTEMS

While basin prediction from a short initial time series is fundamentally a challenge related to “climate replication” (predicting the long-term statistics) of dynamical systems, which has been well studied with RCs in monostable dynamical systems,^{34,36,40–42} it differs from the traditionally studied monostable scenario in ways that make it substantially more challenging. Here, we highlight these challenges in the context of reservoir computing.

Reservoir computers (RCs) predict the evolution of a system with state \mathbf{x} whose dynamics are governed by

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) \quad (1)$$

by constructing an auxiliary dynamical system—the “reservoir system”—with “reservoir state” \mathbf{r} that evolves according to its own dynamical equation,

$$\frac{d\mathbf{r}}{dt} = \mathbf{F}[\mathbf{r}(t), \mathbf{u}(t)], \quad (2)$$

where $\mathbf{u}(t)$ is a driving signal. To train an RC for forecasting, we drive the reservoir system with an observed time series that is a function of the state of the true system, $\mathbf{u}(t) = \mathbf{g}(\mathbf{x}(t))$, in the “open-loop mode” (Fig. 1). Then, we choose a linear readout matrix or “output layer,” W_{out} , such that $\mathbf{u}(t)$ can be approximated through a linear projection of the auxiliary state

$$\hat{\mathbf{u}}(t) = W_{out}\mathbf{r}(t) \approx \mathbf{u}(t). \quad (3)$$

Once we have a suitable output layer, the RC can evolve as an autonomous dynamical system, using its own output as the driving signal in the “closed-loop mode” (Fig. 1), to mimic the system of interest

$$\frac{d\mathbf{r}}{dt} = \mathbf{F}[\mathbf{r}(t), \hat{\mathbf{u}}(t)] = \mathbf{F}[\mathbf{r}(t), W_{out}\mathbf{r}(t)]. \quad (4)$$

Typical approaches for predicting monostable dynamical systems with RCs assume that a single long training series, $\mathbf{u}(t)$, that evolves along the system's single stable attractor—a manifold M_{sys} —provides data that are well sampled on M_{sys} . So long as certain conditions are satisfied,^{43–47} the reservoir system, when driven by the training series $\mathbf{u}(t)$, will evolve along some corresponding manifold, M_{res} , in the state space of the reservoir once a transient response of the reservoir has passed.^{40,46,47} A well-trained output layer, thus, represents a mapping from the manifold M_{res} to the manifold M_{sys} and encodes the dynamics of the true system on this attracting manifold. The output layer may not, however, accurately represent the dynamics of the true system in regions of state space that were not explored in training, i.e., regions that are separated from the manifolds M_{res} and M_{sys} .

Multistable dynamical systems, which have more than one attracting manifold, thus, present two challenges to forecasting

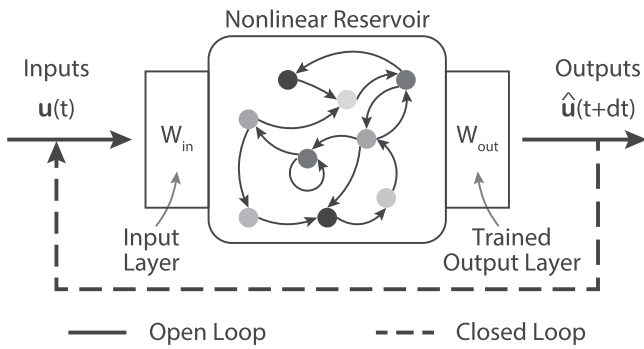


FIG. 1. A reservoir computer for time series prediction.

with reservoir computers. (1) It is often difficult to obtain training time series that sufficiently sample the state space of multistable dynamical systems. Since basins of attraction are necessarily non-overlapping, a single training series cannot sample more than one of the attracting manifolds. Moreover, the transient behaviors of trajectories that have not yet reached their attractors are hard to sample—the transients do not lie on any of the attracting manifolds and are also frequently short-lived. (2) Basins of attraction often have complex, intertwined boundaries, so that a system’s final state depends sensitively on its initial condition.⁴⁸ In such cases, a relatively small prediction error at one time step can push a trajectory from the correct basin of attraction to an incorrect basin of attraction, making climate replication much more challenging.

In Sec. III, we provide a detailed description of our reservoir computing implementation and of the multi-trajectory training scheme we use to facilitate more exhaustive use of disjoint training time series, allowing for better sampling of transient dynamics.

III. TRAINING A RESERVOIR COMPUTER ON MULTIPLE TRAJECTORIES

To construct and train reservoir computers (RCs) for our experiments, we use some of the same implementations as used in previous work by Norton and collaborators³⁴ built on the *rescompy* python package.⁴⁹ Accordingly, portions of our RC implementation and its description are adapted from the earlier work.³⁴

The central component of an RC is a recurrent neural network, “the reservoir,” whose nodes, indexed by i , have associated continuous-valued, time-dependent activation levels $r_i(t)$. The activations of all N_r nodes in the reservoir constitute the reservoir state, $\mathbf{r}(t) = [r_1(t), \dots, r_{N_r}(t)]^T$, which evolves in response to an input signal according to a dynamical equation with a fixed discrete time step, Δt ,

$$\mathbf{r}(t + \Delta t) = (1 - \lambda)\mathbf{r}(t) + \lambda \tanh(W_r \mathbf{r}(t) + W_{in} \mathbf{u}(t) + \mathbf{b}), \quad (5)$$

where the $\tanh()$ function is applied element-wise. The input weight matrix, W_{in} , couples the N_{in} -dimensional input $\mathbf{u}(t)$ to the reservoir nodes. The directed and weighted adjacency matrix, W_r , specifies the strength and sign of interactions between each pair of nodes, and a

random vector of biases, \mathbf{b} , breaks symmetries in the nodes’ dynamics. We say that the reservoir has “memory” if its state $\mathbf{r}(t + \Delta t)$ depends not only on the most recent input, $\mathbf{u}(t)$, but also (recursively) on previous inputs, $\mathbf{u}(t - m\Delta t)$ for $m > 0$ —i.e., if $(1 - \lambda)$ and/or the matrix W_r are nonzero. The leakage rate, λ , thus, influences the time-scale on which the reservoir state evolves, and, consequently, the duration of its memory.

The adjacency matrix, W_r , of each reservoir is a sparse, random, directed network with mean degree $\langle d \rangle$ and probability of connection between each pair of nodes given by $\langle d \rangle / N_r$. We assign non-zero elements of W_r random values from a uniform distribution $\mathcal{U}[-1, 1]$ and then normalize this randomly generated matrix such that its spectral radius (eigenvalue of largest absolute value) has some desired value, ρ . To generate the dense input matrix, W_{in} , and the bias vector, \mathbf{b} , we choose each entry from the uniform distributions $\mathcal{U}[-\sigma, \sigma]$ and $\mathcal{U}[-\psi, \psi]$, respectively. We call σ the input strength range and ψ the bias strength range.

To train an RC for a forecasting task, we choose its output layer, of dimension $N_{in} \times N_r$, so that at every time step over a set of N_{train} training signals, $\mathbf{u}^{(i)}$, $i = 1, \dots, N_{train}$, which we standardize such that each component has mean zero and maximum absolute value one (as measured across the union of all training signals), the RC’s output closely matches its input at the next time step

$$\mathbf{u}^{(i)}(t + \Delta t) \approx W_{out} \mathbf{r}^{(i)}(t + \Delta t). \quad (6)$$

The internal parameters of the reservoir (W_r , W_{in} , \mathbf{b} , and λ) are set prior to training and remain unaltered thereafter. To calculate the output layer, we add white noise to the input time series in order to promote stable predictions⁵⁰

$$\tilde{u}_j^{(i)}(t) = u_j^{(i)}(t) + \mathcal{N}(0, \eta \text{RMS}[u_j]), \quad (7)$$

where $\text{RMS}[u_j]$ is the root mean square amplitude of the j th component of the inputs calculated over all training time series, $\mathcal{N}(0, \xi)$ draws a random sample from a Gaussian distribution with mean zero and standard deviation ξ , and η is a small constant—the “noise amplitude.” We then drive the reservoir with the noisy training signals in the open-loop mode [Eqs. (2) and (5) and Fig. 1] and minimize the ridge-regression cost function

$$\sum_{i=1}^{N_{train}} \sum_{n=N_{trans}}^{N_i-1} \frac{\|W_{out} \mathbf{r}^{(i)}(n\Delta t) - \tilde{\mathbf{u}}^{(i)}(n\Delta t)\|^2}{N_{fit}} + \alpha \|W_{out}\|^2, \quad (8)$$

where N_i is the number of (evenly-spaced) data points in the i th signal (i.e., it has duration $(N_i - 1)\Delta t$), the scalar α is a (Tikhonov⁵¹) regularization parameter, which prevents over-fitting, $\| \cdot \|$ denotes the Euclidean (L^2) norm, and $N_{fit} = \sum_{i=1}^{N_{train}} N_i - N_{train}(N_{trans} + 1)$ is the number of input/output pairs used for fitting. Importantly, we discard the first N_{trans} reservoir states and target outputs of each training signal as a transient to allow the reservoir state to synchronize to each signal before fitting over the remaining time steps. [Note that Eq. (8) reduces to the usual cost function for single-trajectory training when $N_{train} = 1$.] The minimization problem, Eq. (8), has solution

$$W_{out} = YR^T (RR^T + \alpha N_{fit} I)^{-1}, \quad (9)$$

where I is the identity matrix and Y ($N_{in} \times N_{fit}$) and R ($N_r \times N_{fit}$), respectively, are the target and reservoir state trajectories over the fitting periods.

We highlight that the dimensions of the matrices YR^T and RR^T are independent of the number of training signals, N_{train} . This fact is useful when N_{fit} is large (either because we wish to train across a large number of input time series or because the time series are long) and storing the reservoir states in computer memory becomes a challenge. In such cases, we generate batches, b_i , $i = 1, \dots, N_b$, of reservoir states, each of which is small enough to store and calculate the total feature matrix RR^T as the sum of the feature matrices of the batches, $RR^T = \sum_{i=1}^{N_b} (RR^T)_{b_i}$. Once we have calculated the feature matrix $(RR^T)_{b_i}$ for batch b_i , we can discard the reservoir states for that batch and move on to the next batch. Thus, we need to store only one batch of reservoir states at a time. We can calculate YR^T similarly.

Once the RC has been trained as described above, we use it to obtain predictions, $\hat{\mathbf{u}}(t)$, of the system of interest. During the prediction phase, the RC operates in the closed-loop mode: at each time step, its input is set to its own output from the previous step, allowing it to evolve as an autonomous dynamical system. Used in this way, the RC is intended to mimic the behavior of the system of interest as in Eq. (4),

$$\mathbf{r}(t + \Delta t) = (1 - \lambda)\mathbf{r}(t) + \lambda \tanh(W_r \mathbf{r}(t) + W_{in} \hat{\mathbf{u}}(t) + \mathbf{b}), \quad (10a)$$

$$\hat{\mathbf{u}}(t + \Delta t) = W_{out} \mathbf{r}(t + \Delta t). \quad (10b)$$

In typical applications, we wish to predict how the system of interest will evolve, having observed its recent behavior over some period. In this case, we naturally use these recent observations to initialize the forecast. Namely, we drive the reservoir in open-loop mode [Eq. (5)] with a short “test signal,” \mathbf{u}_{test} , consisting of the available recent observations and then switch to the closed-loop mode [Eq. (10b)] to forecast from the end of \mathbf{u}_{test} . The test signal enables the state of the auxiliary reservoir system to synchronize to the state of the underlying system of interest. In general, an appropriate test signal can be substantially shorter than would be sufficient to train an RC accurately. Hence, an RC that has been trained to accurately capture the dynamics of $\mathbf{u}(t)$ can be used to predict from a different initial condition by starting from a comparatively short test signal. The test signal should, however, be at least as long as the RC’s memory to ensure that the memory is appropriately initialized. (A few recent studies have also proposed methods to “cold-start” forecasts with test signals that are even shorter than the RC’s memory.^{34,52})

IV. RESULTS

We evaluate the ability of our RC setup to generalize to unexplored regions of state space using simulated data from four multistable dynamical systems: the Duffing system,⁵³ a multi-well system with segregated basins, a magnetic pendulum system with fractal-like basins,⁵⁴ and a multistable Lorenz-like system with coexisting chaotic attractors.⁵⁵ Previous studies^{3,4} have shown that learning the dynamics of even low-dimensional multistable systems can be challenging for typical RC approaches.

To evaluate the performance of our RC implementation in the context of dissipative systems with fixed-point attractors, we adopt

a working definition for when a time series $\mathbf{u}(t)$ is said to converge to a fixed point A^i , $A(\mathbf{u}) = A^i$. Specifically, if A^i is the nearest stable fixed point to the final point of the series, $\mathbf{u}(t_f)$, our convergence criteria require that $\mathbf{u}(t)$ satisfies one of two additional conditions, depending on whether the system state is fully or partially measured. When $\mathbf{u}(t)$ contains full system state information at every time step, we require that the energy of the system at t_f , $E[\mathbf{u}(t_f)]$, is below the potential barrier, E_0 , between the system’s stable attractors: $E[\mathbf{u}(t_f)] < E_0$. When the full system state is not available, and we cannot calculate the system energy, we instead require that the final 25 points of $\mathbf{u}(t)$ are all within a threshold distance, ε_c , of the attracting fixed point A^i .

Given a set of true system trajectories, $\{\mathbf{u}^k : k = 1, \dots, N\}$, and corresponding predicted trajectories $\{\hat{\mathbf{u}}^k : k = 1, \dots, N\}$, we, thus, approximate the true basin of attraction for the attractor A^i as the set of initial conditions whose trajectories converge to A^i ,

$$B(A^i) = \{\mathbf{u}^k(0) : A(\mathbf{u}^k) = A^i, k = 1, \dots, N\}. \quad (11)$$

Similarly, we estimate the predicted basin of attraction of A^i as

$$\hat{B}(A^i) = \{\hat{\mathbf{u}}^k(0) : A(\hat{\mathbf{u}}^k) = A^i, k = 1, \dots, N\}. \quad (12)$$

A. Intertwined basins in the Duffing system

The Duffing system⁵³ models an oscillator moving under a nonlinear elastic force with linear damping. It is governed by a pair of coupled ordinary differential equations,

$$\dot{x} = y, \quad (13a)$$

$$\dot{y} = F_0 + ay - bx - cx^3. \quad (13b)$$

With $a = -1/2$, $b = -1$, and $c = 1/10$, the Duffing system is dissipative and multistable, with two attracting fixed points. When the system is unforced ($F_0 = 0$), the positions of these attractors are

$$A^\pm = \left(\pm \sqrt{-b/c}, 0 \right) \approx (\pm 3.16, 0). \quad (14)$$

There is also an unstable fixed point at the origin. Trajectories starting from almost any initial condition (x_0, y_0) will, thus, converge to one of the attractors, A^\pm . The initial conditions from which trajectories converge to A^+ form the basin of attraction $B(A^+)$ and the initial conditions from which trajectories converge to A^- form the basin of attraction $B(A^-)$. Only trajectories that move along the stable manifold of the unstable fixed point do not converge to one of the two attractors. The corresponding set of initial conditions, which is of measure zero, forms the boundary between $B(A^+)$ and $B(A^-)$.

We show in Fig. 2 that a reservoir computer (RC) trained on trajectories from only one basin of attraction, $B(A^-)$, can capture the unforced Duffing system’s dynamics in both basins of attraction and infer the existence of the other fixed-point attractor, which is unseen in the training data. Specifically, we train the RC on $N_{train} = 10$ different trajectories, each of which converges to the fixed point A^- [Fig. 2(a)]. We then evaluate the trained RC on short test signals with initial conditions sampled from both basins of attraction [Fig. 2(b)]. Each test signal consists of the first $N_{test} = 10$ observations of the true system’s evolution from a new initial condition, representing the kind of partial trajectory typically available in a prediction task.

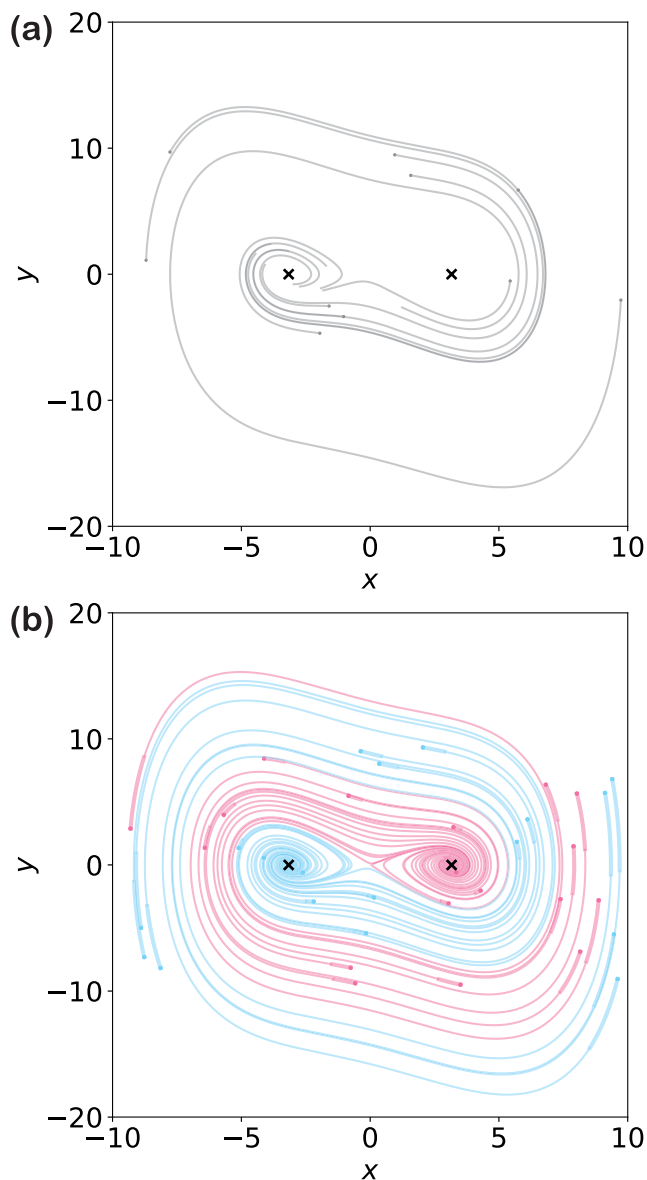


FIG. 2. RC identification of an unseen attractor in the Duffing system. (a) We train an RC with $N_r = 200$ nodes on $N_{train} = 10$ fully observed trajectories (gray lines) from one of the Duffing system's basins of attraction. (b) Then, we forecast the system's evolution from 36 initial conditions (dots) and corresponding short test signals (thick lines, $N_{test} = 10$ observations). Predictions are illustrated by thin lines. We color each trajectory according to the true basin of its initial condition. The RC recovers system behavior in both the seen (blue) and unseen (pink) basins and predicts the correct fixed point for all sample test signals.

These test signals serve both to initialize the forecast and to allow the reservoir to synchronize to the new system state before entering the autonomous closed-loop mode. Even though the RC's training data explore only the basin of attraction of the fixed point A^- , the RC infers the existence and location of the second attractor, A^+ , and

correctly predicts that A^+ is also a fixed point (and not, for example, a limit cycle).

In Figs. 3(a)–3(e), we likewise train an RC on $N_{train} = 10$ trajectories from the basin of attraction $B(A^-)$, but give the RC access to just the x -component of each trajectory, so that it receives only partial information of the system state at every time step. Then, we again forecast from test signals in both basins of attraction, each consisting of $N_{test} = 10$ observations of the true system's evolution. The sample forecasts in Figs. 3(d) and 3(e) highlight the challenging nature of the basin prediction problem. The trajectories of the true system in both cases are almost identical until the system approaches the unstable fixed point at the origin around time step 500. In Fig. 3(d), the RC correctly predicts that the trajectory converges to the unseen fixed point, A^+ . In Fig. 3(e), however, the true trajectory converges to the seen fixed point, A^- , while the RC predicts that it converges to the unseen fixed point. As the system approaches the stable manifold of the unstable fixed point at the origin (in this case, it approaches the origin itself), the seemingly small prediction error is sufficient to push the autonomous RC system into the incorrect basin of attraction, and the predicted and true trajectories then separate rapidly.

In Fig. 3(a) we make predictions from test signals with initial conditions (x_0, y_0) arranged in a 150×150 grid spanning $-10 \leq x_0 \leq 10$ and $-10 \leq y_0 \leq 10$ and plot the RC-predicted basin structure. As before, the RC has access to just the x -component of each test signal; we use y_0 for plotting purposes only. Test signals for which the RC correctly predicts that the system converges to the seen and the unseen attractors are colored blue and pink. Yellow indicates test signals for which the RC predicts the incorrect attractor. The $N_{train} = 10$ initial conditions of the training signals, all from the basin $B(A^-)$, are marked by black dots and the positions of the two attracting fixed points are marked by black crosses.

Figure 3(a) demonstrates clearly that an RC trained on data from only one basin of attraction is able to generalize to the unexplored basin. The RC not only infers the existence of the unseen fixed point, but it also achieves high accuracy in predicting whether any given initial condition belongs to the basin $B(A^-)$ or the basin $B(A^+)$. Only near the basin boundary, where a trajectory's final state is most sensitive to perturbations, does the RC occasionally predict the incorrect basin.

To further explore the ability of RCs to generalize, we investigate the effects of training data diversity, e.g., in terms of the range of initial conditions sampled. In Figs. 3(b) and 3(c), we reduce the half-width, Δ_0^{train} , of the range from which we select training initial conditions, while holding fixed the half-width of the test initial condition range, $\Delta_0^{test} = 10$. Here, we see that for the RC to reliably predict system behavior in both basins, the initial conditions of the training trajectories must be sufficiently far from the fixed-point attractors. For $\Delta_0^{train} = 7$ [Fig. 3(b)], the RC is still able to infer the existence of the unseen attractor and accurately reconstruct a large part of the unseen basin. However, for some initial conditions in the outer regions of the test grid, it predicts that the corresponding trajectories converge to the incorrect attractor (yellow). When the training range is reduced to $\Delta_0^{train} = 4$ [Fig. 3(c)], the RC fails to identify the unseen fixed-point attractor $A^+ \approx (3.16, 0)$. Trajectories starting in the white regions of Fig. 3(c) instead converge to spurious fixed-point attractors near $x \approx -8.4$ and $x \approx 1.75$ (not shown)

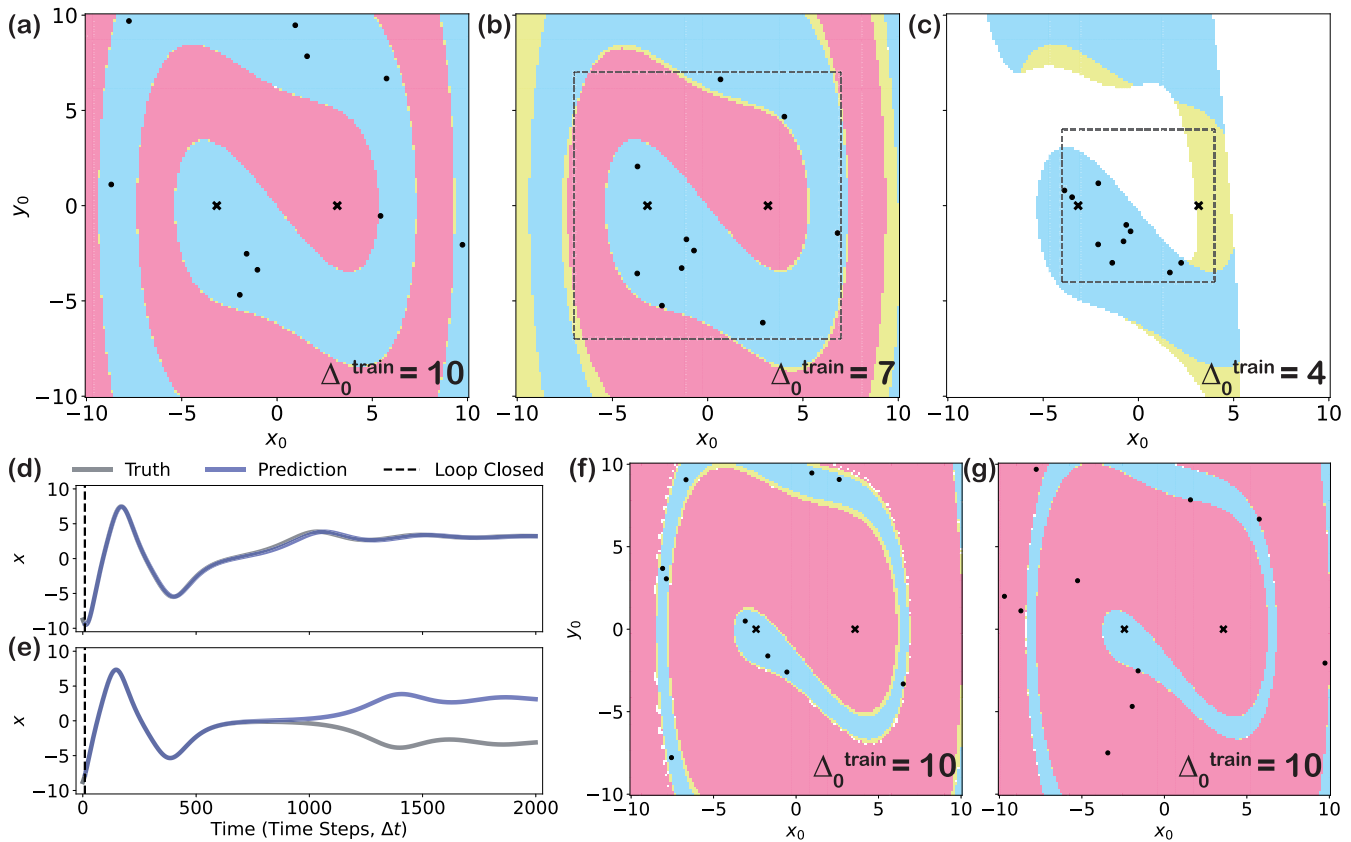


FIG. 3. RCs can generalize to unseen basins even with sparse and restricted training data. (a)–(e) We train an RC of $N_r = 200$ nodes on $N_{train} = 10$ partially observed trajectories (x only) of the unforced Duffing system ($F_0 = 0$) from the basin of attraction $B(A^-)$ (blue). Then, we make predictions from short test signals ($N_{test} = 10$ observations each) from both basins of attraction. We draw the initial conditions, (x_0, y_0) , of the training signals from a random distribution that uniformly covers the intersection of $B(A^-)$ with a square of side $2\Delta_0^{train}$ centered at the origin (dashed boxes), i.e., $x_0 \sim \mathcal{U}[-\Delta_0^{train}, \Delta_0^{train}]$ and $y_0 \sim \mathcal{U}[-\Delta_0^{train}, \Delta_0^{train}]$, subject to $(x_0, y_0) \in B(A^-)$. (a)–(c) The RC-predicted basins for different values of Δ_0^{train} , from $\Delta_0^{train} = 10$ (a) to $\Delta_0^{train} = 4$ (c). The blue/pink initial conditions are those that the RC correctly predicts belong to the seen/unseen basin. The initial conditions of test signals for which the RC-predicted trajectory converges to the incorrect attractor are yellow. White initial conditions indicate that the predicted trajectory converges to a spurious attractor. Crosses mark the true fixed point attractors, and dots mark the training initial conditions. We use y_0 for plotting purposes only; the RC has access to and then predicts only the x -component of the Duffing system. (d) and (e) Example predictions for $\Delta_0^{train} = 10$. The test signal ends at the vertical dashed line. (f) and (g) The RC still generalizes to unseen basins when a constant external forcing ($F_0 = 1$) breaks the Duffing system’s rotational symmetry, whether we train the RC on the smaller basin (f) or the larger basin (g).

that are inconsistent with the Duffing system’s true dynamics. In the [supplementary material](#) (Fig. S1), we present a similar result with RCs trained on fully observed states of the Duffing system, and a similar spurious attractor is clearly visualized.

We demonstrate in [Figs. 3\(f\) and 3\(g\)](#) that the RC does not rely on the Duffing system’s rotational symmetry to generalize to the unexplored basin. Specifically, we break the symmetry by adding a constant external forcing, $F_0 = 1$, to Eq. (13b). This change also moves the attracting fixed points to $A_F^- \approx (-2.42, 0)$ and $A_F^+ \approx (3.58, 0)$. We then train the RC on $N_{train} = 10$ partially observed trajectories from a single basin of attraction and make predictions from short test signals ($N_{test} = 10$) in both basins of attraction, as before. The RC still recovers the system behavior in

both basins, whether we train it on data from the smaller blue basin (panel f) or from the larger pink basin (panel g).

In [Fig. 4](#), we compare the performance of RCs trained on trajectories from only one basin of attraction of the unforced Duffing system, $B(A^-)$, to that of RCs trained on trajectories from both basins. To quantify performance, we measure the fraction of trajectories for which the RC predicts the correct attractor

$$f_c = \frac{\text{Number of correctly predicted basins}}{\text{Total number of predictions}}. \tag{15}$$

At every grid point of the heatmaps in [Fig. 4](#), we plot the mean fraction correct, f_c , averaged over ten independent random draws of the

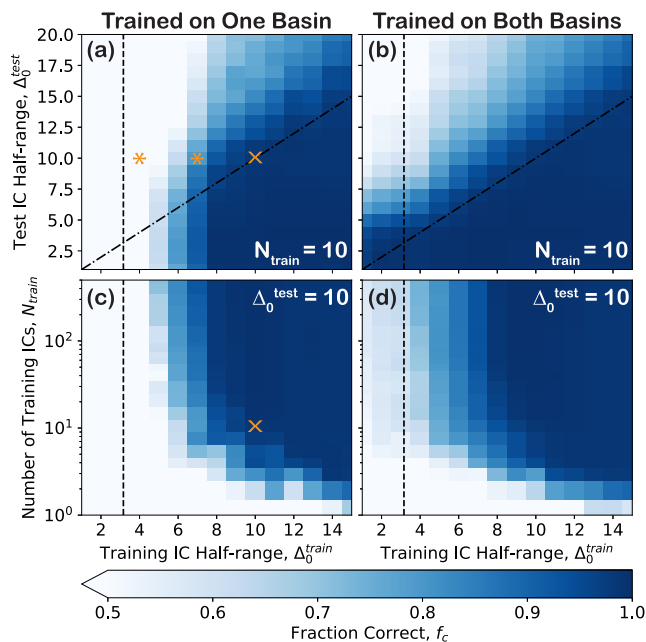


FIG. 4. In many scenarios, RC performance in unseen basins is comparable to performance in the training basin. We train an RC of $N_r = 200$ nodes to forecast partially observed states of the Duffing system (x only) and plot the fraction of short test signals for which the RC predicts the correct basin of attraction as we vary the number of training initial conditions, N_{train} , and the half-widths of the ranges of the training and test initial conditions, Δ_0^{train} and Δ_0^{test} . (a) and (c) All training signals are from $B(A^-)$. (b) and (d) Training signals are from both basins, $B(A^\pm)$. In all cases, the training initial conditions are random, but the initial conditions of the test signals, belonging to both basins, form a 50×50 uniform grid. Each test signal consists of $N_{test} = 10$ consecutive data points from the true system, starting from the specified initial condition. For all grid points, we plot the mean fraction correct calculated over ten random realizations of the RC's internal structure and of the training initial conditions. (a) and (b) $N_{train} = 10$. (c) and (d) $\Delta_0^{test} = 10$. Vertical lines mark the distance of the fixed points from the origin, $\|A^\pm\|$, and diagonal lines mark $\Delta_0^{test} = \Delta_0^{train}$. Crosses and asterisks indicate the values of N_{train} , Δ_0^{train} , and Δ_0^{test} that we use in Figs. 3(a), 3(d), and 3(e) and in Figs. 3(b) and 3(c), respectively.

RC's internal connections, W_{in} and W_r , and of the initial conditions of the training trajectories.

In Figs. 4(a) and 4(b), we vary the half-ranges, Δ_0^{train} and Δ_0^{test} , of both the training and test initial conditions. We note that when Δ_0^{train} is sufficiently large ($\Delta_0^{train} \gtrsim 7$) and the training time series sample the transient dynamics of the Duffing system far from its attractors, RCs trained on trajectories from only one basin of attraction capture the system's basin structure just as reliably as those trained on trajectories from both basins. The degradation in performance above the diagonal in both cases, however, demonstrates that RCs have difficulty extrapolating to regions of state space far from their training data—even as those trained on trajectories from only one basin reliably generalize to the unexplored basin. Finally, the white region to the left of Fig. 4(a), where the fraction correct is no better than chance, highlights again that RCs trained on only

one basin fail to generalize to the unexplored basin if their training signals do not sufficiently sample the transient dynamics of the Duffing system far from its attractors, as illustrated in Fig. 3. In contrast, RCs trained on data from both basins still perform well when Δ_0^{train} is small, so long as Δ_0^{test} is also small.

In Figs. 4(c) and 4(d), we vary the number of training trajectories, N_{train} , while holding the range of test initial conditions fixed at $\Delta_0^{test} = 10$ (for consistency with Fig. 3 and because it is large enough that the RC must capture the Duffing system's transient dynamics well to offer good basin prediction). Here, we see that there is little or no generalization gap. That is, there is no substantial difference in accuracy between RCs trained on data from both basins and those trained on data from only one. In both cases, basin prediction is challenging if the training initial conditions are restricted to a narrow range, even if a large number of training trajectories is available. On the other hand, if the training trajectories are drawn from a wide range of initial conditions and sample well the Duffing system's transient dynamics, RCs can offer useful basin predictions with very few training trajectories.

We demonstrate in the supplementary material (Fig. S2) that RCs can still reliably generalize to an unseen basin of the unforced Duffing system when moderate stochastic forcing (dynamical/process noise) influences the evolution of the training trajectories.

B. Segregated basins in a multi-well system

In the Duffing system, the basins of attraction for the two fixed-point attractors are substantially intertwined. When the training initial conditions span a wide range of phase space, some of the resulting trajectories from the observed basin spend time close to regions belonging to the unseen basin of attraction, providing useful information for generalization. In this section, we investigate how well our RC setup can generalize in a system with segregated basins of attraction, where training trajectories may start near basin boundaries but quickly move away from them, making the extrapolation task potentially more challenging.

Specifically, we employ a simple multi-well potential with four segregated basins of attraction. The dynamics are governed by two independent differential equations,

$$\dot{x} = \frac{1}{2}x(1 - x^2), \tag{16a}$$

$$\dot{y} = \frac{1}{2}y(1 - y^2), \tag{16b}$$

which yield four stable fixed-point attractors at the corners of the unit square and an unstable fixed point at the origin.

We demonstrate in Fig. 5 that our RC setup can generalize to unseen segregated basins in the multi-well system. However, the quality of this generalization is somewhat influenced by whether inputs to the reservoir are standardized based on the training distribution (a common practice in machine learning). Standardization schemes aim to ensure that each component of the inputs has a similar scale, which helps prevent a subset of inputs from dominating the dynamics and promotes more effective use of the system's computational capacity. However, standardization schemes that are based on the training distribution implicitly bias the model to perform

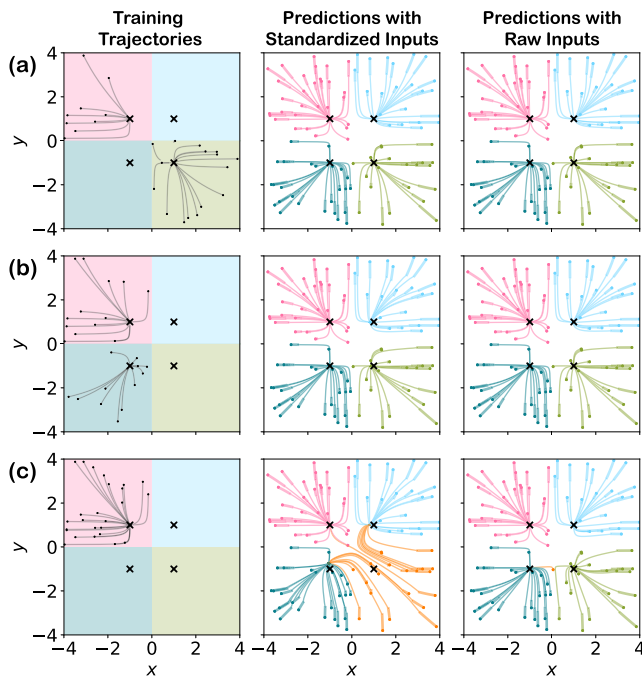


FIG. 5. RC identification of unseen attractors in a multi-well system with segregated basins. We train an RC of $N_r = 200$ nodes on $N_{train} = 25$ fully observed trajectories (gray lines) from two [(a) and (b)] or just one (c) of the multi-well system’s basins of attraction (left). Then, we forecast the system’s evolution from 100 initial conditions (dots) and corresponding short test signals (thick lines, $N_{test} = 5$ observations) in two cases: when the inputs to the RC are standardized (center), and when the inputs to the RC are not standardized (right). We color predictions (thin lines) orange if they go to an incorrect attractor, and according to their true basin otherwise.

well on data with similar distributions (i.e., prioritize in-distribution performance). This bias may not hinder generalization to unseen intertwined basins of attraction, but we show in Fig. 5 that it can slightly degrade performance in unseen segregated basins of attraction, where the mean and range of the test data differ substantially from the training data.

In the central column of Fig. 5, we use the input strength range $\sigma = 1.0$ (Table III), and we standardize all inputs such that the training inputs have component-wise mean zero and maximum absolute value one, as before. (It is also common to standardize inputs to have unit variance over the training data, but we use the maximum absolute value in this paper so that the standardization is not affected by long trajectories that are near a fixed point.) In the right-hand column, we do not standardize the training or test signal inputs and we scale the input strength range, $\sigma = 0.25$, accordingly. When the range of the training trajectories approximately spans the range of the test initial conditions, $-4 \lesssim x \lesssim 4$ and $-4 \lesssim y \lesssim 4$ (row a), the RC performs similarly with standardized and raw inputs. When the training trajectories sample only the left two basins of attraction, $x < 0$ (row b), the RC again generalizes well in both cases but captures the locations of the unseen fixed points at $x = 1$ slightly more

accurately when the input time series are not standardized. Finally, if the training data sample only one basin of attraction (row c), the RC identifies all of the unseen attractors when the inputs are not standardized (raw) but fails to infer one of the unseen fixed-point attractors when the inputs to the reservoir are standardized.

C. Complex basin structure in a magnetic pendulum system

We now demonstrate that RCs can achieve similar out-of-domain generalization in a multistable system with a more complex basin structure than those of the Duffing or multi-well systems. The magnetic pendulum system⁵⁴ consists of an iron bob suspended at the end of a pendulum above a plane that contains three magnetic point charges. The magnets sit at the vertices of an equilateral triangle and, when the pendulum hangs straight down, the bob is a height d above the center of this triangle. We choose our coordinate system such that the origin is the triangle’s center and the magnets’ positions are $(\tilde{x}_1, \tilde{y}_1) = (\frac{1}{\sqrt{3}}, 0)$, $(\tilde{x}_2, \tilde{y}_2) = (-\frac{1}{2\sqrt{3}}, -\frac{1}{2})$, and $(\tilde{x}_3, \tilde{y}_3) = (-\frac{1}{2\sqrt{3}}, \frac{1}{2})$. Taking the pendulum to be much longer than the distance between magnets (which is one), so that small angle approximations are applicable, the equations of motion are

$$\ddot{x} = -\omega_0^2 x - \gamma \dot{x} + \sum_{i=1}^3 \frac{\tilde{x}_i - x}{D_i(x, y)^3}, \tag{17a}$$

$$\ddot{y} = -\omega_0^2 y - \gamma \dot{y} + \sum_{i=1}^3 \frac{\tilde{y}_i - y}{D_i(x, y)^3}, \tag{17b}$$

where γ is a damping coefficient, ω_0 is the natural frequency of the pendulum, and

$$D_i(x, y) = \sqrt{(\tilde{x}_i - x)^2 + (\tilde{y}_i - y)^2 + d^2} \tag{18}$$

is the distance from the bob to the i th magnet. The pendulum has three stable fixed points, each corresponding to it hanging directly above one of the three magnets. We choose the frequency $\omega_0 = 0.5$, damping $\gamma = 0.2$, and pendulum height $d = 0.2$, so that the system is dissipative and all trajectories, except for those on the stable manifold of an unstable fixed point at the origin (a set of measure zero), converge to one of the three stable fixed points.

The basin structure of the magnetic pendulum, which we plot in the plane $\dot{x} = \dot{y} = 0$ in Fig. 6(a), is considerably more complex than that of the Duffing system. In fact, while not a true fractal, the basin boundary forms a fractal-like structure⁵⁴—a so-called “slim fractal.”⁵⁶ The resulting sensitivity of the pendulum’s final state to small perturbations makes basin prediction considerably more difficult. Despite the challenge posed by transient chaos, however, we demonstrate in Fig. 6(b) that an RC trained on trajectories from only one of the magnetic pendulum’s basins of attraction can nonetheless generalize to the other unexplored basins.

The complexity of the pendulum’s basin structure means that small forecast errors can easily push the closed-loop RC system into an incorrect basin of attraction. As a result, we find that to achieve good performance, compared with the Duffing and multi-well systems, our setup for the magnetic pendulum requires (1) more training trajectories and a more powerful (i.e., larger) reservoir

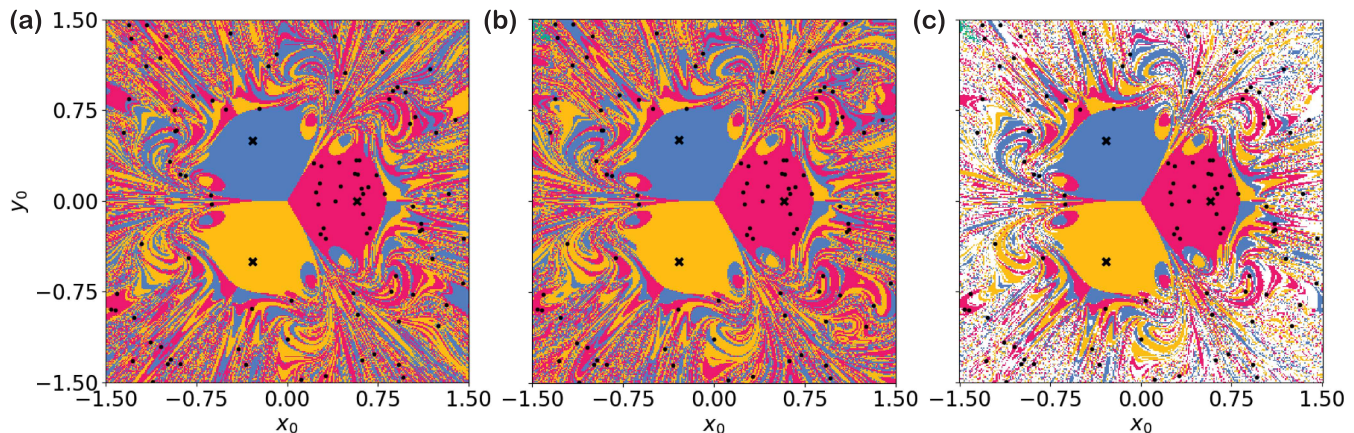


FIG. 6. An RC trained on data from only one basin can capture the magnetic pendulum's complex basin structure. (a) Basins of attraction of the magnetic pendulum in the plane $\dot{x} = \dot{y} = 0$. Black crosses mark the positions of the magnets. (b) RC-predicted basin structure using test signals of length $N_{test} = 100$ data points with initial conditions on a 300×300 grid. The training and test signals are partially observed (\dot{x} and \dot{y} withheld); the RC learns to predict only the position components of the pendulum state. All $N_{train} = 100$ training trajectories (black dots) are from the pink basin of attraction. Only $f_{spurious} \approx 8 \times 10^{-4}$ of the predicted trajectories converge to spurious attractors (green). (c) RC-predicted basins, with all initial conditions for which the RC predicts that the system converges to an incorrect fixed point (real or spurious) colored white.

to improve the output-layer accuracy and (2) longer test signals to improve the RC's initial synchronization. In Fig. 6, for example, we train a reservoir of size $N_r = 2500$ nodes on $N_{train} = 100$ trajectories and make predictions using test signals of length $N_{test} = 100$ data points. [We demonstrate in Fig. S3 of the [supplementary material](#) that these test signals are still short enough such that predicting the correct basin from the end of the test signal is nearly as difficult as predicting it from the initial condition (start of the test signal). We also show how the RC's performance varies with the length of the test signals.] Remarkably, despite the fractal-like basin boundaries, the RC is able to provide good predictions not only for the training (pink) basin but also for the two other unseen basins. Still, because of the difficulty of the problem, we do not expect that any RC—even one trained on trajectories from all basins—can reliably predict the correct attractor for initial conditions far from the fixed points. (Even a next-generation reservoir computer with a very strong structural prior requires training data with a high sampling rate to reliably predict the basins of the magnetic pendulum.³) Figure 6 illustrates this inherent challenge. The RC-predicted basin structure [Fig. 6(b)] matches the true basin structure [Fig. 6(a)] well qualitatively, even as the RC struggles to reliably predict the basins of individual test signals whose initial conditions are in the outer regions of the test grid, where the basin structure is most complex [Fig. 6(c)]. The fraction of test signals for which the RC predicts the correct attractor, $f_c \approx 0.66$, however, is still substantially higher than the baseline fraction of 0.47 achieved by an approach that simply guesses that the pendulum will converge to the magnet nearest to it at the end of the test signal ([supplementary material](#), Fig. S3).

Assessing the qualitative accuracy of the predicted basins in Fig. 6(b) is analogous to evaluating traditional climate replication in monostable dynamical systems, which focuses on capturing statistical properties of the system over time. Here, our goal is for the statistics of the predicted system behavior – collected over different

test signals—to accurately reflect the system's multistability. Indeed, we see that the RC-predicted trajectories rarely converge to spurious attractors ($f_{spurious} \approx 8 \times 10^{-4}$, i.e., 75 of the 90 000 sample predictions in Fig. 6) and the RC broadly captures where extended regions of state space belong to the same basin of attraction and where the basins are more intertwined.

At the per-basin level, the fraction correct (f_c), false negative rate ($1 - f_c$), and false positive rate are summarized in Table I. These results, coupled with Fig. S4 of the [supplementary material](#), indicate that RC performance in predicting basin membership is comparable for seen and unseen basins, and that, in many cases, an RC trained on trajectories from only one basin of the magnetic pendulum still recovers the overall basin structure almost as well as an RC trained on all three basins—consistent with what we find for the Duffing system (Fig. 4). However, even when the RC is successful at predicting basin membership, it may still miss some of the detailed dynamics of the unseen basins (see [supplementary material](#), Fig. S5). Overall, these findings suggest that an RC can learn the system dynamics in a way that enables generalization to unseen regions of state space in an operationally useful manner, even without strictly capturing system symmetries.

TABLE I. Per-basin RC performance in predicting the correct attractor in Fig. 6. Baseline: a predictor that always guesses the magnet nearest the pendulum at the end of the test signal.

Basin	Fraction correct	False neg. rate	False pos. rate
Pink	0.66	0.34	0.17
Blue	0.65	0.35	0.17
Yellow	0.66	0.34	0.18
Baseline	0.47

D. Coexisting chaotic attractors in a Lorenz-like system

The Duffing, multi-well, and magnetic pendulum systems are dissipative, with all trajectories converging to fixed-point attractors. In many multistable systems, however, the system state never reaches a fixed point and instead converges to a limit cycle or chaotic attractor, along which it evolves indefinitely. In this section, we highlight a case in which an RC captures the existence of an unseen chaotic attractor.

Following Lü *et al.*,⁵⁵ we study a multi-stable Lorenz-like system that is governed by three coupled ordinary differential equations,

$$\dot{x} = -\frac{ab}{a+b}x - yz + c, \tag{19a}$$

$$\dot{y} = ay + xz, \tag{19b}$$

$$\dot{z} = bz + xy. \tag{19c}$$

When $a = -10$, $b = -4$, and $c = 18.1$, the system has two chaotic attractors. In Fig. 7(a), we train an RC on a single long trajectory (black dashed line), which approaches and then evolves along one of the attractors. Then, we show that the RC captures the existence and structures of both the seen attractor (the blue lobe, A_1) and the unseen attractor (the pink lobe, A_2). To obtain a clear image of the predicted chaotic attractors in Fig. 7(a), we make predictions using test signals of $N_{test} = 50$ data points and omit the test signals themselves from the figure. These test signals are, however, long enough that each prediction starts when the system state has already reached one of the chaotic attractors.

In Fig. 7(b), we reduce the length of the test signals to $N_{test} = 5$ observations so that, for most initial conditions, the system is still

far from the attractors when prediction begins. Then, we plot the RC-predicted basin structure for test signals with initial conditions in the plane $x = 0$. The RC accurately captures the basin structure of the multistable Lorenz-like system. In fact, the RC predicts the correct attractor for almost all initial conditions in the 100×100 test grid, indicating that it can accurately capture the system’s transient behavior en route to the chaotic attractors (in addition to reconstructing the chaotic attractors themselves).

The convergence criteria used for the Duffing, multi-well, and magnetic pendulum systems in Secs. IV A–IV C do not apply to the Lorenz-like system because its attractors are not fixed points. Instead, to determine whether a trajectory $\mathbf{u}(t)$ converges to one of the system’s chaotic attractors, we compute the Kullback–Liebler (KL) divergence of the empirical state distribution, $P_u(\mathbf{v})$, over the final 500 time steps of $\mathbf{u}(t)$ relative to the distribution $P_A(\mathbf{v})$ from a reference trajectory on the attractor

$$\Delta_{KL}(P_A, P_u) = \int_{\mathbf{v} \in \mathbb{R}^3} P_A(\mathbf{v}) \log \frac{P_A(\mathbf{v})}{P_u(\mathbf{v})} d\mathbf{v}. \tag{20}$$

Following previous work,^{57,58} we use Gaussian mixture models to estimate the distributions $P_u(\mathbf{v})$ and $P_A(\mathbf{v})$ from the predicted and reference trajectories. If the KL divergence of the predicted distribution relative to the true distribution on the attractor A is less than a threshold value, $\Delta_{KL}(P_A, P_u) < \epsilon_c$, we say that the predicted trajectory converges to the attractor A .

In Fig. 7(c), we use the KL divergence to assess the quality of the reconstructed attractors in the seen and unseen basins. The RC achieves comparable accuracy on the unseen (pink) attractor

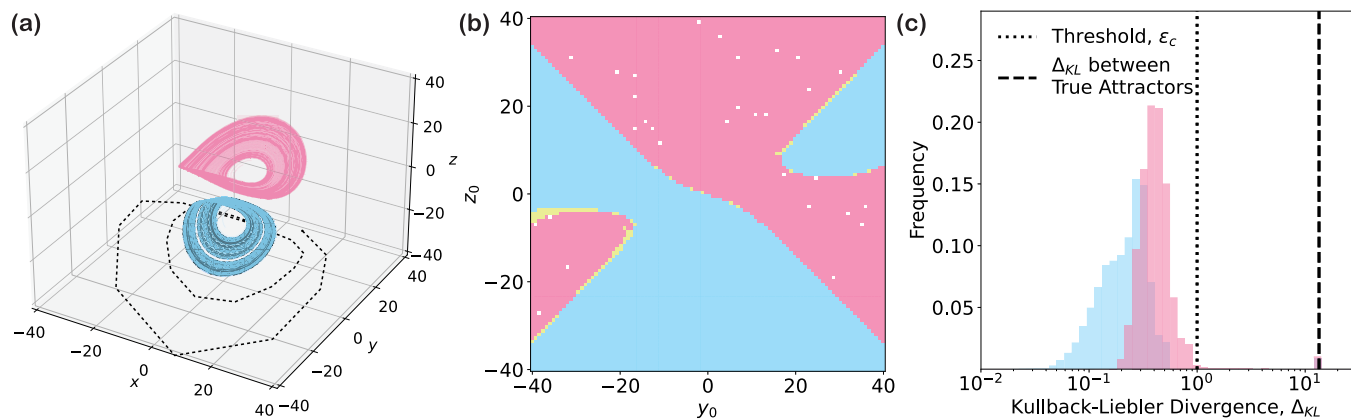


FIG. 7. RC reconstruction of an unseen chaotic attractor in a multistable Lorenz-like system. We train an RC with $N_r = 500$ nodes on a single fully observed trajectory (dashed line) of the Lorenz-like system. (a) We forecast the system’s evolution from 36 short test signals ($N_{test} = 50$ observations, test signal trajectories not shown for clarity), each starting from a random initial condition. Forecasts start from the endpoints of these test signals, and each predicted trajectory is colored according to the true basin of its initial condition. (b) We make predictions from shorter test signals ($N_{test} = 5$) with initial conditions forming a 100×100 grid in the $x = 0$ plane and plot the RC-predicted basin structure. Yellow and white, respectively, indicate initial conditions for which the prediction converges to the incorrect attractor and those for which the prediction does not accurately resemble either true attractor [$\Delta_{KL}(P_{A_i}, P_u) > \epsilon_c$ for $i = 1, 2$]. (c) Distribution of the KL divergence between the predicted and true attractors, computed over all predictions in panel (b). Each histogram is formed from predictions associated with test signals from the same true basin of attraction and is colored accordingly. The vertical dotted line indicates the convergence threshold, $\epsilon_c = 1$, and the vertical dashed line marks the KL divergence between the true attractors, estimated as $\frac{1}{2} (\Delta_{KL}(P_{A_1}, P_{A_2}) + \Delta_{KL}(P_{A_2}, P_{A_1}))$.

as on the seen (blue) attractor, with the KL divergence of the predicted attractor relative to the true attractor in both cases substantially lower than the KL divergence between the two true attractors (vertical dashed line).

V. DISCUSSION

Our results show that reservoir computers (RCs) can successfully generalize to entirely unobserved regions of state space in multistable dynamical systems. Unlike approaches that rely on explicit system knowledge or dense coverage of the training domain, our RC setup requires only a limited set of observed trajectories and makes no assumptions about the underlying dynamics—i.e., it operates without explicit structural priors, such as known equations, symmetries, or conservation laws—yet still learns representations that support strong out-of-domain generalization.

We make use of a training scheme that allows the RC to incorporate information from multiple disjoint time series. Importantly, we show that RCs trained on trajectories from a single basin of attraction can accurately predict system behavior in other basins. After training, the RC can generate predictions from a new initial condition and observed signal—one that only needs to be long enough for the reservoir to synchronize—without needing to retrain or reconfigure the model.

A strength of this approach is that it enables generalization from a relatively limited set of training trajectories—limited both in number and in the portion of state space sampled—notably more restricted than typically used in data-intensive machine learning frameworks. However, successful generalization still depends on whether the training data contain sufficient information about the system's dynamics. If the training trajectories fail to capture a sufficient range of the system's behaviors, the RC will not have enough information to represent behavior beyond the training domain, and generalization will break down.

These findings suggest that RCs can construct a generalizable internal representation of system dynamics that extends beyond the training data. In doing so, they can capture global structures such as basins of attraction and make reliable forecasts in regions of state space that were never directly observed. This kind of generalization—achieved from sparse, disjoint training data and without structural assumptions—positions RCs as a practical tool for modeling complex systems in settings where prior knowledge is unavailable and data are limited.

Moving forward, a promising direction for future work is to develop a mathematical understanding of what enables out-of-domain generalization in RCs. Generally speaking, vector fields in part of the state space do not uniquely determine vector fields in other parts of the state space, so some kind of inductive bias is needed to accurately predict dynamics far away from the training data. Unlike most neural networks trained via gradient descent, RC output weights are determined through regularized linear regression—a convex optimization problem with a closed-form solution. In the overparameterized regime, the Moore–Penrose pseudoinverse in the closed-form solution selects the solution with the smallest norm. This optimization process may introduce an implicit inductive bias that favors simpler, lower-complexity solutions. In many systems, such simplicity may be an effective inductive bias

that enables generalization beyond the training domain. Exploring how different regularization schemes influence this tendency and how they connect to broader ideas in machine learning—such as flat minima⁵⁹ and double descent^{14,60}—offers a promising direction for deepening our understanding of generalization in RCs.

SUPPLEMENTARY MATERIAL

See the [supplementary material](#) for Figs. S1–S5, which provide further information on the sensitivity of RC generalization to dynamical/process noise, test signal length, and training data diversity, as well as illustrations of predicted near-attractor behavior and spurious attractors.

ACKNOWLEDGMENTS

We thank Edward Ott and Brian Hunt for helpful conversations, insights, and suggestions. We also acknowledge the University of Maryland supercomputing resources (<http://hpcc.umd.edu>) made available for conducting the research reported in this paper. The contributions of D.A.N. and of M.G. were supported, respectively, by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1840340 and by ONR under Grant No. N000142212656. Y.Z. was supported by the Omidyar Fellowship and the National Science Foundation under Grant No. DMS 2436231. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Department of Defense, or the U.S. Government.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Declan A. Norton: Conceptualization (equal); Formal analysis (lead); Funding acquisition (equal); Investigation (lead); Methodology (equal); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead). **Yuanzhao Zhang:** Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Visualization (supporting); Writing – review & editing (equal). **Michelle Girvan:** Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Project administration (lead); Resources (lead); Supervision (lead); Visualization (supporting); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX: EXPERIMENTAL SETUP

We obtain trajectories of the Duffing, multi-well, and multi-stable Lorenz-like systems by integrating Eqs. (13) and (19) using

TABLE II. Experimental parameters.

		Duffing	Wells	Magnetic	Lorenz
RC time step	Δt	0.01	0.01	0.02	0.02
Training transient	N_{trans}	5	5	25	5
Training signal length	$N_i \forall i$	500	500	500	5000
Forecast horizon	N_f	2000	2000	2000	5000
Convergence threshold	ε_c	0.5	...	0.25	1.0

a fourth order Runge–Kutta integrator. We generate trajectories of the magnetic pendulum system by integrating Eq. (17) using the SCIPY `integrate.solve_ivp` implementation of the DOP853 eighth order Runge–Kutta integration scheme. For the magnetic pendulum, the integration time step is adaptive, and for the other systems, it is fixed ($\Delta t = 0.01$). In all cases, the time step between samples in the RCs' training and test signals is fixed: $\Delta t = 0.01$ for the Duffing and multi-well systems and $\Delta t = 0.02$ for the magnetic pendulum and multistable Lorenz-like systems, as in Table II. Because we intend for the RC to learn from and predict the transient dynamics of each system, we do not discard any portion of the integrated trajectories before forming training or test signals. (We do, however, discard a short transient response of the reservoir's internal state at the start of each training signal, as described in Sec. III.)

The parameters we use to construct and train RCs for our experiments are provided in Table III. The other parameters defining our experiments are in Table II. For simplicity, we use training trajectories that are of equal length, $N_i = 500 \forall i = 1, \dots, N_{train}$, in all of our experiments (except for the multistable Lorenz-like system, which we address separately in the next paragraph). Our multiple-trajectory training scheme (Sec. III) does not, however, require that all training signals are the same length. When all training trajectories must be from the same basin of attraction, we first generate a trajectory of length $4000\Delta t$ from each sample initial condition and

TABLE III. Reservoir hyperparameters.

		Duffing	Wells	Magnetic	Lorenz
Reservoir size	N_r	200	200	2500	500
Mean in-degree	$\langle d \rangle$	10	10	10	10
Input strength range	σ	1.0	1.0 ^a	5.0	0.5
Spectral radius	ρ	0.4	0.4	0.4	0.4
Bias strength range	ψ	0.5	0.5	0.5	0.5
Leakage rate	λ	1.0	1.0	1.0	1.0
Tikhonov regularization	α	10^{-12}	10^{-12}	10^{-10}	10^{-10}
Training noise amplitude	η	10^{-5}	10^{-5}	10^{-3}	10^{-3}

^aWe use $\sigma = 1.0$ when we standardize inputs to the reservoir and $\sigma = 0.25$ when we do not standardize the inputs.

check whether the trajectory converges to the corresponding desired attractor. If a trajectory converges to the right attractor, we include its first $N_i = 500$ data points in the RC's training data. We repeat this process with new sample initial conditions until we obtain the desired number of training trajectories, N_{train} , from the chosen basin. In all of our experiments (except the Lorenz-like system), we forecast from the end of each provided test signal, i.e., $t = (N_{test} - 1)\Delta t$, to time $t = N_f\Delta t = 2000\Delta t$.

For our experiments with the multistable Lorenz-like system, we use only one training trajectory, which is of length $N_1 = 5000$ to allow for adequate sampling of the on-attractor chaotic dynamics. Similarly, for each test signal, we forecast from $t = (N_{test} - 1)\Delta t$ to time $t = N_f\Delta t = 5000\Delta t$. To assess whether a prediction converges to one of the system's true attractors, we measure the Kullback–Liebler (KL) divergence of the empirical distribution of Lorenz states over the final 500 time steps of the predicted trajectory relative to that of a reference trajectory on the attractor, as described in Sec. IV D. We measure the KL divergences using only the final 500 time steps of the predicted trajectory to ensure that (1) the transient approach to the attractor does not obscure the on-attractor behavior and (2) the RC's prediction is stable and has remained on the attractor for a prolonged period (the previous $4500 - N_{test}$ predicted data points). To estimate the KL divergence, we use the `DYSTS`^{61,62} metrics.estimate_kl_divergence routine with its default arguments: `n_samples = 1000`, `sigma_scale = 1`, and `eps = 10-10`.

As shown in Table III, the reservoir hyperparameters that we use for our experiments with the Duffing, multi-well, magnetic pendulum, and Lorenz-like systems are identical except for the input strength range, σ , regularization strength, α , and noise amplitude, η . We chose these hyperparameters by coarse hand-tuning to allow for good but not necessarily optimal performance. We chose the other hyperparameters to have values that typically allow for reasonably accurate forecasting with reservoir computers and performed no experiment-specific tuning of these values. While more robust hyperparameter tuning may improve performance overall, our priority is to demonstrate that reservoir computers can generalize to unexplored regions of state space without system-specific structural constraints, rather than to obtain highly optimized forecasts.

REFERENCES

- ¹S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2019).
- ²Z. Han, J. Zhao, H. Leung, K. F. Ma, and W. Wang, "A review of deep learning models for time series prediction," *IEEE Sens. J.* **21**, 7833–7848 (2021).
- ³Y. Zhang and S. P. Cornelius, "Catch-22s of reservoir computing," *Phys. Rev. Res.* **5**, 033213 (2023).
- ⁴N. Göring, F. Hess, M. Brenner, Z. Monfared, and D. Durstewitz, "Out-of-domain generalization in dynamical systems reconstruction," [arXiv:2402.18377 \[cs.LG\]](https://arxiv.org/abs/2402.18377) (2024).
- ⁵D. J. Gauthier, I. Fischer, and A. Röhm, "Learning unseen coexisting attractors," *Chaos* **32**, 113107 (2022).
- ⁶R. Yu and R. Wang, "Learning dynamical systems from data: An introduction to physics-guided deep learning," *Proc. Natl. Acad. Sci. U.S.A.* **121**, e2311808121 (2024).
- ⁷S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U.S.A.* **113**, 3932–3937 (2016).
- ⁸S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.* **3**, e1602614 (2017).

- ⁹D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, “Next generation reservoir computing,” *Nat. Commun.* **12**, 5564 (2021).
- ¹⁰J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, “Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model,” *Chaos* **28**, 041101 (2018).
- ¹¹T. Arcomano, I. Szunyogh, A. Wikner, J. Pathak, B. R. Hunt, and E. Ott, “A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model,” *J. Adv. Model. Earth Syst.* **14**, e2021MS002712 (2022).
- ¹²R. Chepuri, D. Amzalag, T. M. Antonsen, and M. Girvan, “Hybridizing traditional and next-generation reservoir computing to accurately and efficiently forecast dynamical systems,” *Chaos* **34**, 063114 (2024).
- ¹³G. Vardi, “On the implicit bias in deep-learning algorithms,” *Commun. ACM* **66**, 86–93 (2023).
- ¹⁴A. H. Ribeiro, J. N. Hendriks, A. G. Wills, and T. B. Schön, “Beyond Occam’s razor in system identification: Double-descent when modeling dynamics,” *IFAC-PapersOnLine* **54**, 97–102 (2021), 19th IFAC Symposium on System Identification SYSID 2021.
- ¹⁵A. Röhm, D. J. Gauthier, and I. Fischer, “Model-free inference of unseen attractors: Reconstructing phase space features from a single noisy trajectory using reservoir computing,” *Chaos* **31**, 103127 (2021).
- ¹⁶Y. Du, Q. Li, H. Fan, M. Zhan, J. Xiao, and X. Wang, “Inferring attracting basins of power system with machine learning,” *Phys. Rev. Res.* **6**, 013181 (2024).
- ¹⁷H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *Science* **304**, 78–80 (2004).
- ¹⁸B. Schrauwen, D. Verstraeten, and J. Campenhout, “An overview of reservoir computing: Theory, applications and implementations,” in Proceedings of the 15th European Symposium on Artificial Neural Networks (2007), pp. 471–482.
- ¹⁹C. Sun, M. Song, D. Cai, B. Zhang, S. Hong, and H. Li, “A systematic review of echo state networks from design to application,” *IEEE Trans. Artif. Intell.* **5**, 23–37 (2024).
- ²⁰M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ²¹Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, “Reservoir observers: Model-free inference of unmeasured variables in chaotic systems,” *Chaos* **27**, 041102 (2017).
- ²²K. Srinivasan, N. Coble, J. Hamlin, T. Antonsen, E. Ott, and M. Girvan, “Parallel machine learning for forecasting the dynamics of complex networks,” *Phys. Rev. Lett.* **128**, 164101 (2022).
- ²³J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.* **120**, 024102 (2018).
- ²⁴S. Krishnagopal, M. Girvan, E. Ott, and B. R. Hunt, “Separation of chaotic signals by reservoir computing,” *Chaos* **30**, 023123 (2020).
- ²⁵A. Banerjee, J. D. Hart, R. Roy, and E. Ott, “Machine learning link inference of noisy delay-coupled networks with optoelectronic experimental tests,” *Phys. Rev. X* **11**, 031014 (2021).
- ²⁶G. Tanaka *et al.*, “Recent advances in physical reservoir computing: A review,” *Neural Networks* **115**, 100–123 (2019).
- ²⁷E. Bollt, “On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD,” *Chaos* **31**, 013108 (2021).
- ²⁸A. Wagemakers, “The basins zoo,” [arXiv:2504.01580](https://arxiv.org/abs/2504.01580) [nlin.CD] (2025).
- ²⁹E. M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting* (The MIT Press, 2006).
- ³⁰D. A. Rand, A. Raju, M. Sáez, F. Corson, and E. D. Siggia, “Geometry of gene regulatory dynamics,” *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2109729118 (2021).
- ³¹F. Corson, L. Couturier, H. Rouault, K. Mazouni, and F. Schweisguth, “Self-organized Notch dynamics generate stereotyped sensory organ patterns in *Drosophila*,” *Science* **356**, eaai7407 (2017).
- ³²P. J. Menck, J. Heitzig, J. Kurths, and H. Joachim Schellnhuber, “How dead ends undermine power grid stability,” *Nat. Commun.* **5**, 3969 (2014).
- ³³F. Cavalli and A. Naimzada, “Complex dynamics and multistability with increasing rationality in market games,” *Chaos, Solitons Fractals* **93**, 151–161 (2016).
- ³⁴D. A. Norton, E. Ott, A. Pomerance, B. Hunt, and M. Girvan, “Tailored forecasting from short time series via meta-learning,” [arXiv:2501.16325](https://arxiv.org/abs/2501.16325) [cs.LG] (2025).
- ³⁵L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, “Machine learning prediction of critical transition and system collapse,” *Phys. Rev. Res.* **3**, 013090 (2021).
- ³⁶S. Panahi and Y.-C. Lai, “Adaptable reservoir computing: A paradigm for model-free data-driven prediction of critical transitions in nonlinear dynamical systems,” *Chaos* **34**, 051501 (2024).
- ³⁷L.-W. Kong, G. A. Brewer, and Y.-C. Lai, “Reservoir-computing based associative memory and itinerancy for complex dynamical attractors,” *Nat. Commun.* **15**, 4840 (2024).
- ³⁸J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, “Teaching recurrent neural networks to modify chaotic memories by example,” [arXiv:2005.01186](https://arxiv.org/abs/2005.01186) [cond-mat.dis-nn] (2020).
- ³⁹Z. Lu and D. S. Bassett, “Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems,” *Chaos* **30**, 063133 (2020).
- ⁴⁰Z. Lu, B. R. Hunt, and E. Ott, “Attractor reconstruction by machine learning,” *Chaos* **28**, 061104 (2018).
- ⁴¹D. Patel, D. Canaday, M. Girvan, A. Pomerance, and E. Ott, “Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity,” *Chaos* **31**, 033149 (2021).
- ⁴²S. Panahi, L.-W. Kong, B. Glaz, M. Haile, and Y.-C. Lai, “Unsupervised learning for anticipating critical transitions,” [arXiv:2501.01579](https://arxiv.org/abs/2501.01579) [nlin.CD] (2025).
- ⁴³H. Jaeger, “The ‘echo state’ approach to analysing and training recurrent neural networks,” GMD Report 148, GMD - German National Research Institute for Computer Science, 2001.
- ⁴⁴M. Lukoševičius, “A practical guide to applying echo state networks,” in *Neural Networks: Tricks of the Trade: Second Edition*, edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer, Berlin, 2012), pp. 659–686.
- ⁴⁵M. Cucchi, S. Abreu, G. Ciccone, D. Brunner, and H. Kleemann, “Hands-on reservoir computing: A tutorial for practical implementation,” *Neuromorphic Comput. Eng.* **2**, 032002 (2022).
- ⁴⁶J. A. Platt, A. Wong, R. Clark, S. G. Penny, and H. D. I. Abarbanel, “Robust forecasting using predictive generalized synchronization in reservoir computing,” *Chaos* **31**, 123118 (2021).
- ⁴⁷J. A. Platt, S. G. Penny, T. A. Smith, T.-C. Chen, and H. D. Abarbanel, “A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics,” *Neural Networks* **153**, 530–552 (2022).
- ⁴⁸C. Grebogi, S. W. McDonald, E. Ott, and J. A. Yorke, “Final state sensitivity: An obstruction to predictability,” *Phys. Lett. A* **99**, 415–418 (1983).
- ⁴⁹D. Canaday, D. Kalra, A. Wikner, D. A. Norton, B. Hunt, and A. Pomerance, “Rescompy 1.0.0: Fundamental methods for reservoir computing in python,” GitHub (2024).
- ⁵⁰A. Wikner, J. Harvey, M. Girvan, B. R. Hunt, A. Pomerance, T. Antonsen, and E. Ott, “Stabilizing machine learning prediction of dynamics: Novel noise-inspired regularization tested with reservoir computing,” *Neural Networks* **170**, 94–110 (2024).
- ⁵¹A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola, “Regularization methods,” in *Numerical Methods for the Solution of Ill-Posed Problems* (Springer Netherlands, Dordrecht, 1995), pp. 7–63.
- ⁵²L. Grigoryeva, B. Hamzi, F. P. Kemeth, Y. Kevrekidis, G. Manjunath, J.-P. Ortega, and M. J. Steynberg, “Data-driven cold starting of good reservoirs,” *Phys. D* **469**, 134325 (2024).
- ⁵³G. Duffing, *Erzwungene Schwingungen Bei Veränderlicher Eigenfrequenz Und Ihre Technische Bedeutung* (F. Vieweg & Sohn, 1918), 41.
- ⁵⁴A. E. Motter, M. Gruiz, G. Károlyi, and T. Tél, “Doubly transient chaos: Generic form of chaos in autonomous dissipative systems,” *Phys. Rev. Lett.* **111**, 194101 (2013).
- ⁵⁵J. Lü, G. Chen, and D. Cheng, “A new chaotic system and beyond: The generalized Lorenz-like system,” *Int. J. Bifurcation Chaos* **14**, 1507–1537 (2004).
- ⁵⁶X. Chen, T. Nishikawa, and A. E. Motter, “Slim fractals: The geometry of doubly transient chaos,” *Phys. Rev. X* **7**, 021040 (2017).

⁵⁷Y. Zhang and W. Gilpin, “Zero-shot forecasting of chaotic systems,” in *The Thirteenth International Conference on Learning Representations (ICLR, 2025)*.

⁵⁸F. Hess, Z. Monfared, M. Brenner, and D. Durstewitz, “Generalized teacher forcing for learning chaotic dynamics,” in *Proceedings of the 40th International Conference on Machine Learning, ICML’23 (JMLR.org, 2023)*.

⁵⁹Y. Feng and Y. Tu, “The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima,” *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2015617118 (2021).

⁶⁰M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15849–15854 (2019).

⁶¹W. Gilpin, “Chaos as an interpretable benchmark for forecasting and data-driven modelling,” in *Advances in Neural Information Processing (NeurIPS, 2021)*.

⁶²W. Gilpin, “Model scale versus domain knowledge in statistical forecasting of chaotic systems,” *Phys. Rev. Res.* **5**, 043252 (2023).