Symmetry kills the square in a multifunctional reservoir computer

ⓘ Andrew Flynn, ⓘ Joschka Herteux, ⓘ Vassilios A. Tsachouridis, et al.

COLLECTIONS

Paper published as part of the special topic on In Memory of Vadim S. Anishchenko: Statistical Physics and Nonlinear Dynamics of Complex Systems

open access

View Online   Export Citation   CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Multifunctionality in a reservoir computer
Chaos: An Interdisciplinary Journal of Nonlinear Science **31**, 013125 (2021); https://doi.org/10.1063/5.0019974

On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD
Chaos: An Interdisciplinary Journal of Nonlinear Science **31**, 013108 (2021); https://doi.org/10.1063/5.0024890

Introduction to Focus Issue: Recent advances in modeling complex systems: Theory and applications
Chaos: An Interdisciplinary Journal of Nonlinear Science **31**, 070401 (2021); https://doi.org/10.1063/5.0061767

Symmetry kills the square in a multifunctional reservoir computer

View Online   Export Citation   CrossMark

Andrew Flynn,[1,a] Joschka Herteux,[2] Vassilios A. Tsachouridis,[3] Christoph Räth,[2] and Andreas Amann[1]

AFFILIATIONS

[1]School of Mathematical Sciences, University College Cork, Cork T12 XF62, Ireland
[2]Institut für Materialphysik im Weltraum, Deutsches Zentrum für Luft-und Raumfahrt, Münchner Str. 20, 82234 Wessling, Germany
[3]Collins Aerospace—Applied Research and Technology, Cork T23 XN53, Ireland

Note: This paper is part of the Focus Issue, In Memory of Vadim S. Anishchenko: Statistical Physics and Nonlinear Dynamics of Complex Systems.
[a]**Author to whom correspondence should be addressed:** andrew_flynn@umail.ucc.ie

ABSTRACT

The learning capabilities of a reservoir computer (RC) can be stifled due to symmetry in its design. Including quadratic terms in the training of a RC produces a "square readout matrix" that breaks the symmetry to quell the influence of "mirror-attractors," which are inverted copies of the RC's solutions in state space. In this paper, we prove analytically that certain symmetries in the training data forbid the square readout matrix to exist. These analytical results are explored numerically from the perspective of "multifunctionality," by training the RC to specifically reconstruct a coexistence of the Lorenz attractor and its mirror-attractor. We demonstrate that the square readout matrix emerges when the position of one attractor is slightly altered, even if there are overlapping regions between the attractors or if there is a second pair of attractors. We also find that at large spectral radius values of the RC's internal connections, the square readout matrix reappears prior to the RC crossing the edge of chaos.

Symmetry is a notorious issue in the training of a reservoir computer (RC). For example, when training some RC designs to mimic trajectories on a given attractor, a "mirrored" version of the attractor is automatically created in the RC's state space due to symmetry. Fortunately, we can suppress the influence these "mirror-attractors" exert on the RC's learning capacity by breaking the symmetry. In this paper, we investigate the behavior of the "square readout matrix," which is produced to destroy the mirror-attractors by including quadratic terms in the training of the RC's readout layer. We consider these symmetry associated phenomena from the perspective of "multifunctionality" by specifically training the RC to reconstruct a coexistence of attractors related by symmetry. We prove analytically that under certain conditions, symmetries in the training data prohibit the square readout matrix from existing. The consequences of these analytical results are explored in various numerical experiments. We examine the behavior of the square readout matrix when the RC is trained to exhibit multifunctionality by reconstructing a
pair of mirrored-Lorenz attractors. Changing the location of one Lorenz attractor results in the square readout matrix appearing. This effect persists even in the more complex cases of overlapping regions between the pair of Lorenz attractors or the coexistence of two mirror-attractor pairs. We also show that at critical spectral radius values of the RC's internal layer, the square readout matrix appears even when driving the RC with a single source of symmetrical input training data.

I. INTRODUCTION

Despite the growing number of machine learning applications, many of these methods are considered to be "black-boxes," which oftentimes results in very little being understood about how the machine actually learns. Fortunately, reservoir computing[1–3] is one machine learning approach that can be rigorously assessed through the lens of dynamical systems.[4–8]

A reservoir computer (RC) can be realized as a type of artificial neural network consisting of three layers: input, internal, and read-out layer. If the internal layer, known as the "reservoir," is suitably designed, then only the weights of the readout layer need to be determined in order to train the RC. Nonlinear time-series prediction and attractor reconstruction are two of the most well known application areas of RCs.[9–13]

There are many ways to design the RC's readout layer with respect to a given problem. While originally it was mostly linear terms that were used to train the readout layer, in recent years, the inclusion of quadratic terms in the training to produce a "square readout matrix" has become popular after its introduction by Lu *et al.*,[14] where it was employed to handle symmetries of the Lorenz system.[15] This "squaring technique" was empirically found to improve a RC's performance in more general cases.[11,16] This can, in part, be attributed to the ability of the square readout matrix to break harmful symmetries present in the equations describing a RC.[17] In the context of attractor reconstruction, these harmful symmetries must be broken in order to prevent the creation of a "mirror-attractor," an inverted version of the attractor that the RC was specifically trained to reconstruct.

In this paper, we consider these phenomena associated with symmetry from the perspective of "multifunctionality." This neuroscientific term describes a neural network with the ability to perform more than one task without changing any synapses. There are numerous examples of multifunctionality found to occur in nature; for further reading, we suggest Ref. 18, and references therein. Recently Flynn *et al.*[19] translated multifunctionality to a machine learning setting. Here, it was demonstrated that a RC can be trained to create a coexistence of multiple attractors from different dynamical systems. The multifunctional RC is directly analogous to its biological counterparts as it can perform different tasks depending on a given initial condition. The same result has also been shown by Herteux and Räth[17] with a similar training technique.

By including certain symmetries in the training data, we are able to "open the black-box" and expose some of the fundamental properties of a multifunctional RC. In Sec. III, we prove analytically that a point or inversion symmetry of the training data necessitates a disappearance of the square readout matrix in order for the RC to exhibit multifunctionality or reconstruct a single anti-symmetric trajectory.

In Sec. IV, we use the analytical results from Sec. III as a platform to extend our study of the relationship between symmetry and multifunctionality to several numerical experiments. We focus on training the RC to specifically reconstruct a coexistence of the Lorenz attractor and its mirror-attractor. It is only when the symmetry in the training data is broken that we see a contribution from the square readout matrix.

We also show that symmetry still "kills the square" for two more complex examples. In the first case, we manipulate the training data from the Lorenz attractor such that the trajectories appear to intersect with its mirror-attractor in state space. We train the RC to reconstruct a coexistence of these overlapping attractors and show that it is only when the location of one attractor is slightly shifted that the square readout matrix appears. In the second case, we investigate the behavior of the square readout matrix when there are overlapping regions between two pairs of mirrored-Lorenz attractors. We show that it is sufficient to break the symmetry between only one pair of attractors for the square readout matrix to emerge.

Using a simplistic example of anti-symmetric training data, we find that at a certain spectral radius value of the RC's internal connections, the square readout matrix appears. By slightly increasing the spectral radius past this critical point, the elements of the matrix grow in magnitude as chaos begins to conquer the RC dynamics.

The rest of the paper is outlined as follows: in Sec. II, we introduce the type of RC we consider in this paper and the particular squaring technique we use to break the symmetry, and the role of symmetry in the RC is discussed in greater length. The specifics of training the RC to achieve multifunctionality are also outlined. In Sec. III, we present our analytical results that are explored numerically in Sec. IV. We provide some concluding remarks in Sec. V.

II. RESERVOIR COMPUTING

Echo-State Networks (ESNs)[2] and Liquid-State Machines (LSMs)[3] are two independently proposed designs of "recurrent neural networks" that are the foundation of reservoir computing.[1] Central to the philosophy behind this machine learning approach is that instead of training all the weights in a recurrent neural network, it is sufficient to optimize only the weights of a readout layer. This ideological shift stems from the design of a suitable internal layer, known as the "reservoir," which does not need to be trained according to a given task. Many physical mediums can play the role of a reservoir.[20,21] In this paper, we construct the reservoir as a network of artificial neurons with a sparse Erdös–Renyi topology. We make use of the continuous-time formulation devised by Lu *et al.*,[9] which is expressed in the following equation:

$$\dot{\boldsymbol{r}}(t) = \gamma \left[-\boldsymbol{r}(t) + \tanh\left(\mathbf{M}\,\boldsymbol{r}(t) + \sigma \mathbf{W}_{in}\,\boldsymbol{u}(t) \right) \right]. \quad (1)$$

Here, $\boldsymbol{r}(t) \in \mathbb{R}^N$ is the state of the RC at a given time t and N is the number of neurons in the network. γ is the decay-rate parameter. $\mathbf{M} \in \mathbb{R}^{N \times N}$ is the adjacency matrix describing the reservoir. σ is the input strength parameter and $\mathbf{W}_{in} \in \mathbb{R}^{N \times D}$ is the input matrix; when multiplied together, this represents the weight given to the D-dimensional input time-series, $\boldsymbol{u}(t) \in \mathbb{R}^D$, as it is projected into the reservoir. Solutions of Eq. (1) are computed using the fourth order Runge–Kutta method with time step $\tau = 0.01$. In the Appendix, we outline how $\mathbf{M}$ and $\mathbf{W}_{in}$ are designed.

A key parameter involved in this RC setup is the spectral radius, ρ, of the internal connection, $\mathbf{M}$. ρ is associated with the RC's memory as it is used to change the weight the RC places on its own internal dynamics. In Sec. IV, ρ plays a key role in several instances.

The RC in Eq. (1) is driven by the input $\boldsymbol{u}(t)$ from $t = 0$ to time $t = t_{listen}$ in order to remove any dependency $\boldsymbol{r}(t)$ has on its initial condition $\boldsymbol{r}(0) = (0, 0, \ldots, 0)^T = \mathbf{0}^T$. The training data are generated by continuing to drive the RC with $\boldsymbol{u}(t)$ from $t = t_{listen}$ to $t = t_{train}$.

A suitable readout layer needs to be calculated in order to train the RC. We replace the training input signal, $\boldsymbol{u}(t)$, in Eq. (1) with a post-processing function, $\hat{\psi}(\cdot)$. If the training is successful, then we say that

$$\hat{\psi}\left(\boldsymbol{r}(t)\right) = \hat{\boldsymbol{u}}(t) \approx \boldsymbol{u}(t) \quad \text{for } t > t_{train}, \quad (2)$$

where $\hat{\boldsymbol{u}}(t)$ is the predicted time-series. This layer closes the loop of the nonautonomous equation (1) and provides a map from the high-dimensional state space of the RC, $\mathbb{S}$, to the D-dimensional "prediction state space," $\mathbb{P}$.

In this paper we choose $\hat{\psi}\left(\boldsymbol{r}(t)\right) = \mathbf{W}_{out}\boldsymbol{q}(\boldsymbol{r}(t))$ where $\mathbf{W}_{out}$ is the readout matrix and we use $\boldsymbol{q}(\boldsymbol{r}(t))$ to break the symmetry using the "squaring technique" described by

$$\boldsymbol{q}(\boldsymbol{r}(t)) = \begin{pmatrix} \boldsymbol{r}(t) \\ \boldsymbol{r}^2(t) \end{pmatrix}. \tag{3}$$

We then write

$$\hat{\psi}\left(\boldsymbol{r}(t)\right) = \mathbf{W}_{out}\boldsymbol{q}(\boldsymbol{r}(t)) = \left(\mathbf{W}_{out}^{(1)} \, \mathbf{W}_{out}^{(2)}\right) \begin{pmatrix} \boldsymbol{r}(t) \\ \boldsymbol{r}^2(t) \end{pmatrix}$$

$$= \mathbf{W}_{out}^{(1)}\boldsymbol{r}(t) + \mathbf{W}_{out}^{(2)}\boldsymbol{r}^2(t), \tag{4}$$

where $\mathbf{W}_{out}^{(1)}$ is the linear readout matrix and $\mathbf{W}_{out}^{(2)}$ is the square readout matrix.

To calculate $\mathbf{W}_{out}$, we use a ridge regression given by

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T + \beta\,\mathbf{I}\right)^{-1}, \tag{5}$$

where

$$\mathbf{X} = \left[\begin{pmatrix} \boldsymbol{r}(t_{listen}) \\ \boldsymbol{r}^2(t_{listen}) \end{pmatrix} \begin{pmatrix} \boldsymbol{r}(t_{listen} + \tau) \\ \boldsymbol{r}^2(t_{listen} + \tau) \end{pmatrix} \cdots \begin{pmatrix} \boldsymbol{r}(t_{train}) \\ \boldsymbol{r}^2(t_{train}) \end{pmatrix} \right] \tag{6}$$

is the response of the RC to the input data

$$\mathbf{Y} = \begin{bmatrix} \boldsymbol{u}(t_{listen}) & \boldsymbol{u}(t_{listen} + \tau) & \cdots & \boldsymbol{u}(t_{train}) \end{bmatrix}. \tag{7}$$

β is the regularization parameter that is used to discourage overfitting and $\mathbf{I}$ is the identity matrix.

We then write the "predicting RC" as

$$\dot{\hat{\boldsymbol{r}}}(t) = \gamma\Big[-\hat{\boldsymbol{r}}(t) + \tanh\Big(\mathbf{M}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}^{(1)}\hat{\boldsymbol{r}}(t)$$

$$+ \sigma\mathbf{W}_{in}\mathbf{W}_{out}^{(2)}\hat{\boldsymbol{r}}^2(t)\Big)\Big], \tag{8}$$

where $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}(t_{train})$.

A. Symmetry

An important property to consider when designing a RC is the symmetry of the equations as it will usually harm the RC's learning capability. For example, if we simply choose $\boldsymbol{q}(\hat{\boldsymbol{r}}(t)) = \hat{\boldsymbol{r}}(t)$ in our setup, we get a common symmetry as the equation

$$\dot{\hat{\boldsymbol{r}}}(t) = \gamma\left[-\hat{\boldsymbol{r}}(t) + \tanh\left(\mathbf{M}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\hat{\boldsymbol{r}}(t)\right)\right] \tag{9}$$

is invariant under inversion of the sign $\hat{\boldsymbol{r}}(t) \to -\hat{\boldsymbol{r}}(t)$. This results in a symmetric partner for every solution of Eq. (9), whereby for any attractor, $\mathcal{A}$, in $\mathbb{S}$, there is a corresponding "mirror-attractor," $\mathcal{A}'$. Additionally, the linear readout results in pairs of attractors and mirror-attractors appearing in $\mathbb{P}$.

This symmetry induced phenomenon and, in particular, how to overcome the issues related to it was explored in great detail by Herteux and Räth.[17] Here, it was proven analytically that changing the sign of the input and target data in the listening stage also leads

to a sign change in the resulting reservoir states used as training data given the reservoir has the echo state property. Because of this, the calculation of the readout matrix $\mathbf{W}_{out}$ will give the same weights. It was also highlighted that symmetry becomes most problematic if the location of the training data is not chosen correctly as the attractors in the $\mathbb{S}$ may merge with one another.

There are numerous ways to break this symmetry to mitigate the restraints it imposes, for example, using a bias in the output, a shift in the input, or including square terms in the readout.[17] In this paper, we focus on the square terms in the readout described by Eq. (3). In this case, we see that the RC is not invariant under the change of sign like before as

$$\hat{\psi}\left(-\hat{\boldsymbol{r}}(t)\right) \approx \mathbf{W}_{out}^{(1)}\left(-\hat{\boldsymbol{r}}(t)\right) + \mathbf{W}_{out}^{(2)}\left(-\hat{\boldsymbol{r}}(t)\right)^2 \tag{10}$$

$$= -\mathbf{W}_{out}^{(1)}\hat{\boldsymbol{r}}(t) + \mathbf{W}_{out}^{(2)}\hat{\boldsymbol{r}}^2(t) \neq -\hat{\psi}\left(\hat{\boldsymbol{r}}(t)\right). \tag{11}$$

$\mathbf{W}_{out}^{(2)}\hat{\boldsymbol{r}}^2(t)$ breaks the symmetry and the mirror-attractors are destroyed. In the current paper, we consider this phenomenon from the perspective of multifunctionality and find that the opposite is also true. More specifically, in Sec. III, we prove analytically that $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ when there are certain symmetries present in the training data. In Sec. IV, we show that when the RC in Eq. (1) is trained using the squaring technique in Eq. (3) to reconstruct a coexistence of the Lorenz attractor and its mirror-attractor, then $\mathbf{W}_{out}^{(2)}$ does not come into existence.

We remark that the effect of symmetries in reservoir computing has been investigated in a number of further previous studies. Carroll and Pecora[22] show how symmetries in the structure of the network lead to a lower covariance rank and negatively affect the training error. A permutation symmetry in the reservoir states was found to severely impact the performance of a swarm based RC in Ref. 23. On the other hand, matching the symmetry of the input data with a suitable handcrafted RC design was shown to cause drastic improvements in two tasks by Barbosa et al.[24] In this work, we examine how a RC can adapt to and take on the symmetry of the trained system spontaneously.

Next, we outline how to train a RC to exhibit multifunctionality.

B. Multifunctionality

A multifunctional neural network can perform multiple tasks by changing its activity patterns according to a particular input without the need of changing any connections. Multifunctionality is prevalent in networks that are used to switch between mutually exclusive tasks, for example, swimming or crawling,[25,26] and regular breathing or sighing or gasping.[27]

Translating multifunctionality to a RC is the same as saying that the RC must be trained to reconstruct the behavior of more than one attractor using the same readout layer.[19] In other words, if for a given $\mathbf{M}$ and $\mathbf{W}_{in}$ a post-processing function, $\hat{\psi}\,(\cdot)$, can be found such that

$$\hat{\psi}\left(\hat{\boldsymbol{r}}_{\mathcal{A}}(t)\right) \approx \boldsymbol{u}_{\mathcal{A}}(t) \quad \text{and} \quad \hat{\psi}\left(\hat{\boldsymbol{r}}_{\mathcal{B}}(t)\right) \approx \boldsymbol{u}_{\mathcal{B}}(t) \tag{12}$$

for $t > t_{train}$, then we say that the RC is multifunctional as a single network can successfully reproduce the long term behavior of

two different time-series $\boldsymbol{u}_{\mathcal{A}}$ and $\boldsymbol{u}_{\mathcal{B}}$ based on a particular initial condition.

In order to train the RC to exhibit multifunctionality, we use the "blending technique" presented in Ref. 19. The first step of this technique is to drive the reservoir in Eq. (1) with $\boldsymbol{u}_{\mathcal{A}}(t)$ from $t = 0$ to $t = t_{train}$ and collect the corresponding training data in the matrices $\mathbf{X}_{\mathcal{A}}$ and $\mathbf{Y}_{\mathcal{A}}$ like in Eqs. (6) and (7). We then repeat this process for the other input source $\boldsymbol{u}_{\mathcal{B}}$ to obtain $\mathbf{X}_{\mathcal{B}}$ and $\mathbf{Y}_{\mathcal{B}}$. Both reservoir training data matrices are then concatenated in the following way:

$$\mathbf{X}_C = (\alpha \mathbf{X}_{\mathcal{A}}, \ (1 - \alpha)\mathbf{X}_{\mathcal{B}}). \tag{13}$$

The same process is applied to construct the equivalent input concatenation matrix, $\mathbf{Y}_C$. The parameter $\alpha \in [0, 1]$ "blends" two different sources of data together by applying more weight to one set of data over the other. This proved particularly useful when training a RC to reconstruct a coexistence of chaotic attractors with different dynamical characteristics.[19] We take the additional step to randomly reorder each column of the matrices, $\mathbf{X}_C$ and $\mathbf{Y}_C$, corresponding to the input and reservoir state at a given time. The matrices, $\mathbf{X}_C$ and $\mathbf{Y}_C$, are then used in the ridge regression formula in Eq. (5).

Once the training is successful, if the predicting RC in Eq. (8) is initialized with either $\hat{\boldsymbol{r}}_{\mathcal{A}}(0)$ or $\hat{\boldsymbol{r}}_{\mathcal{B}}(0)$, then the RC will predict the future evolution of $\boldsymbol{u}_{\mathcal{A}}(t)$ or $\boldsymbol{u}_{\mathcal{B}}(t)$ from $t = t_{train}$.

We remark that similar results exist in the literature regarding RCs and the learning of multiple attractors. For example, Inoue *et al.*[28] and Lu and Bassett[29] have shown that a RC can switch between learning to reconstruct different periodic and chaotic attractors using an online training scheme. Furthermore, Ceni *et al.*[30] demonstrated that an external input can be used to force a RC to switch between different fixed points.

The key difference between these results and multifunctionality is that a single readout matrix is calculated to create multistability of attractors in the RC's high-dimensional state space, $\mathbb{S}$, that resemble the target behavior when projected to the lower-dimensional prediction state space, $\mathbb{P}$.

III. ANALYTICAL RESULTS

A. Anti-symmetric training data leads to $\mathbf{W}_{out}^{(2)} = 0$

Let us now study the properties of the ridge regression formula in Eq. (5) if it is presented with anti-symmetric data. Let us assume that $\mathbf{X}$ and $\mathbf{Y}$ both have $2n$ columns and are obtained from data $\boldsymbol{r}_i$ and $\boldsymbol{u}_i$ with the property

$$\boldsymbol{u}_{i+n} = -\boldsymbol{u}_i, \tag{14}$$

$$\boldsymbol{r}_{i+n} = -\boldsymbol{r}_i. \tag{15}$$

Then, we have

$$\mathbf{X} = \left[\begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_1^2 \end{pmatrix} \ \cdots \ \begin{pmatrix} \boldsymbol{r}_n \\ \boldsymbol{r}_n^2 \end{pmatrix} \ \begin{pmatrix} -\boldsymbol{r}_1 \\ \boldsymbol{r}_1^2 \end{pmatrix} \ \cdots \ \begin{pmatrix} -\boldsymbol{r}_n \\ \boldsymbol{r}_n^2 \end{pmatrix} \right], \tag{16}$$

$$\mathbf{Y} = \begin{bmatrix} \boldsymbol{u}_1, & \cdots, & \boldsymbol{u}_n, -\boldsymbol{u}_1, & \cdots, & -\boldsymbol{u}_n \end{bmatrix}. \tag{17}$$

This means that

$$\mathbf{Y}\mathbf{X}^T = \left[2 \sum_{i=1}^{n} \boldsymbol{u}_i \boldsymbol{r}_i^T, \quad \mathbf{0} \right], \tag{18}$$

$$\mathbf{X}\mathbf{X}^T = \begin{bmatrix} 2 \sum_{i=1}^{n} \boldsymbol{r}_i \boldsymbol{r}_i^T & \mathbf{0} \\ \mathbf{0} & 2 \sum_{i=1}^{n} \boldsymbol{r}_i^2 \left(\boldsymbol{r}_i^2 \right)^T \end{bmatrix}. \tag{19}$$

Plugging this into Eq. (5), we obtain

$$\mathbf{W}_{out} = \left[2 \left(\sum_{i=1}^{n} \boldsymbol{u}_i \boldsymbol{r}_i^T \right) \left(2 \sum_{i=1}^{n} \boldsymbol{r}_i \boldsymbol{r}_i^T + \beta \mathbf{I} \right)^{-1}, \quad \mathbf{0} \right]. \tag{20}$$

Therefore, whenever the training data fulfills the conditions in Eqs. (14) and (15), then it follows that

$$\mathbf{W}_{out}^{(2)} = \mathbf{0}. \tag{21}$$

B. Sources of anti-symmetric training data

1. Multiple trajectories related by symmetry

Now that we know that anti-symmetric training data lead to a vanishing $\mathbf{W}_{out}^{(2)} = \mathbf{0}$, and we would like to investigate under what conditions such anti-symmetric training data naturally arise. In the context of multifunctionality, this can happen if we train the system with two separate trajectories from two attractors $\boldsymbol{u}_1(t)$ and $\boldsymbol{u}_2(t)$, which are related by

$$\boldsymbol{u}_1(t) = -\boldsymbol{u}_2(t). \tag{22}$$

Time discretization and concatenation of the two trajectories according to the blending technique of Eq. (13) with $\alpha = 1/2$ then immediately shows that Eq. (14) holds. To show that Eq. (15) is also fulfilled, we remember that both $\boldsymbol{r}_1(t)$ and $\boldsymbol{r}_2(t)$ are obtained from Eq. (1). Let us define

$$f(\boldsymbol{r}, \boldsymbol{u}) = \gamma \left[-\boldsymbol{r} + \tanh \left(\mathbf{M}\boldsymbol{r} + \sigma \mathbf{W}_{in}\boldsymbol{u} \right) \right] = -f(-\boldsymbol{r}, -\boldsymbol{u}), \tag{23}$$

where in the second equation, we observe that f is anti-symmetric. If now $\boldsymbol{r}_1(t)$ is a solution of Eq. (1) for some initial condition $\boldsymbol{r}_1(0)$ then $\boldsymbol{r}_2(t) = -\boldsymbol{r}_1(t)$ is a solution for initial condition

$$\boldsymbol{r}_2(0) = -\boldsymbol{r}_1(0). \tag{24}$$

This is demonstrated by

$$\dot{\boldsymbol{r}}_2(t) = -\dot{\boldsymbol{r}}_1(t) = -f(\boldsymbol{r}_1(t), \boldsymbol{u}_1(t)) \tag{25}$$

$$= -f(-\boldsymbol{r}_2(t), -\boldsymbol{u}_2(t)) = f(\boldsymbol{r}_2(t), \boldsymbol{u}_2(t)). \tag{26}$$

Therefore, Eq. (15) holds. This means that if the system is trained with two distinct anti-symmetric trajectories and the initial condition of the reservoir state is anti-symmetric, the resulting $\mathbf{W}_{out}^{(2)}$ matrix automatically vanishes. This result can be easily generalized to any even number of distinct trajectories $\boldsymbol{u}_k(t)$, which are pairwise anti-symmetric,

$$\boldsymbol{u}_{2k}(t) = -\boldsymbol{u}_{2k+1}(t). \tag{27}$$

2. Single anti-symmetric trajectory

Another source of anti-symmetric data is the case of periodic trajectories of period T, which are of the form

$$\boldsymbol{u}\left(t + \frac{T}{2}\right) = -\boldsymbol{u}(t). \tag{28}$$

The reservoir trajectory $\boldsymbol{r}(t)$ is the response of the system to the periodic drive $\boldsymbol{u}(t)$ according to Eq. (1). By the symmetry, there always

TABLE I. Hyperparameters and time parameters used to generate the results displayed in each of the specified figures. T is the period of the circular orbit described in Eq. (31). These parameters were chosen using a grid search method.

Figure	ρ	σ	γ	β	α	t_{listen}	t_{train}	$t_{predict}$
1 and 2	0.2	0.2	10	10^{-2}	0.5	100	100	100
5	1.7	0.3	5	10^{-1}	0.5	100	100	100
6	1.7	0.22	5	10^{-1}	n/a	100	100	100
7 and 8	[1.4, 2.4]	0.3	10	10^{-2}	0.5	30T	30T	30T

exists a solution of the form

$$r\left(t + \frac{T}{2}\right) = -r(t). \qquad (29)$$

However, this solution is not necessarily stable. In our numerical experiments in Sec. III B, we find that the solution is stable up to a certain value of ρ and in this case, Eq. (15) is fulfilled, which again gives rise to vanishing $\mathbf{W}_{out}^{(2)}$.

We remark that there is scope to extend the results in this section to the other symmetry breaking terms identified in Ref. 17 and we leave this for future work.

We now investigate how these analytical results can be realized in different numerical experiments.

IV. NUMERICAL RESULTS

In this section, we discuss the results of our numerical experiments regarding symmetry and multifunctionality. The RC hyperparameters used to produce each of the figures are given in Table I. Also, note that in each case the predicted trajectory for a given input/attractor, $\mathcal{A}$, is represented by $\hat{\mathcal{A}}$, which is in keeping with the notation provided in Sec. II.

A. Multifunctionality with mirrored-Lorenz attractors

First, we consider the case described in Eq. (22) by training the RC in Eq. (1) with the squaring technique in Eq. (3) to reconstruct the chaotic Lorenz attractor, $\mathcal{L}$, and its mirror-attractor, $\mathcal{L}'$. The input data are generated using the fourth order Runge–Kutta method with time step $\tau = 0.01$. The mirror-attractor is created by taking the negative of the Lorenz input time-series, $\mathbf{u}_{\mathcal{L}}(t) = (x(t), y(t), z(t))^T$, where $x(t)$, $y(t)$, and $z(t)$ are the dynamical variables of the Lorenz system.[15]

We use the methods outlined in Sec. II B to calculate $\mathbf{W}_{out} = \left(\mathbf{W}_{out}^{(1)} \; \mathbf{W}_{out}^{(2)}\right)$ in order to train the RC to exhibit multifunctionality by reconstructing either $\mathcal{L}$ or $\mathcal{L}'$ based on a given initial condition. The result of this is shown in Figs. 1(a) and 1(b).

In Fig. 1(a), we plot the reconstructed attractors, $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}'}$, in the prediction state space $\mathbb{P}$ along with the target attractors $\mathcal{L}$ and $\mathcal{L}'$. In this case, we see that multifunctionality was achieved by the RC as both attractors were successfully reconstructed using the same $\mathbf{W}_{out}$ matrix. However, in Fig. 1(b), we see that despite breaking the symmetry in the training by using the squaring technique defined in Eq. (3), the square readout matrix, $\mathbf{W}_{out}^{(2)}$, does not come into existence. In keeping with the analytical results in Sec. III,

the symmetry in the training data forces the RC to also become symmetric.

Next, we consider breaking the symmetry in the training data by shifting $\mathcal{L}'$ by a factor χ in the x-direction, i.e., $\mathbf{u}_{\mathcal{L}'_\chi}(t) = (-x(t) - \chi, -y(t), -z(t))^T$. In Figs. 1(c) and 1(d), we display the results for $\chi = 2$.

In Fig. 1(c), we see that the RC exhibits multifunctionality as it can reconstruct both $\mathcal{L}$ and $\mathcal{L}'_\chi$ using the same readout matrix. However, we see in Fig. 1(d) that once the symmetry in the training data is broken by slightly shifting the location of the mirror-attractor in state space, then $\mathbf{W}_{out}^{(2)}$ comes into existence.

We now explore what happens to $\mathbf{W}_{out}^{(2)}$ when shifting the location of the mirror-attractor for $\chi \in [-5, 5]$. In Fig. 2, we plot histograms of $\mathbf{W}_{out}^{(2)}$ as a function of χ. Here, each color represents the number of $\mathbf{W}_{out}^{(2)}$ elements at a specific value, and the more elements there are at a certain value, the darker the color is. For $\chi = 0$, we see the known result from Fig. 1(b) where all elements of $\mathbf{W}_{out}^{(2)}$ equal to 0. We see that the elements of $\mathbf{W}_{out}^{(2)}$ grow larger as the magnitude of χ is increased. So, we can say that the more we break the symmetry in the training data, the greater the contribution from $\mathbf{W}_{out}^{(2)}$ in the dynamics of the multifunctional RC.

The conditions defined in Eqs. (14) and (15) rely on the assumption that both training data sets have precisely the same number of points. We now explore the behavior of $\mathbf{W}_{out}^{(2)}$ as the RC is trained with input from trajectories on $\mathcal{L}$ and $\mathcal{L}'$ using the respective training times $t_{train}^{\mathcal{L}}$ and $t_{train}^{\mathcal{L}'}$. The resultant imbalance in the data sets is characterized by the parameter $\kappa \in [0, 1]$ defined as

$$\kappa = \frac{t_{train}^{\mathcal{L}'}}{t_{train}^{\mathcal{L}} + t_{train}^{\mathcal{L}'}}. \qquad (30)$$

Therefore, when $\kappa = 0.5$, the conditions in Eqs. (14) and (15) are satisfied as $t_{train}^{\mathcal{L}} = t_{train}^{\mathcal{L}'}$, which results in $\mathbf{W}_{out}^{(2)} = \mathbf{0}$. In Fig. 3, we illustrate the behavior of $\mathbf{W}_{out}^{(2)}$ for $\kappa \in [0, 1]$ and $t_{train}^{\mathcal{L}} + t_{train}^{\mathcal{L}'} = 200$. We observe that for a small change in κ away from 0.5, the histogram of $\mathbf{W}_{out}^{(2)}$ elements is still strongly peaked at zero. To provide further details on the changes of $\mathbf{W}_{out}^{(2)}$ away from $\kappa = 0.5$, we plot the cumulative density function of the absolute values of the elements of $\mathbf{W}_{out}^{(2)}$ for a number of κ values in Fig. 4. This shows that for $\kappa = 0$, which corresponds to the case of training the RC with data only from $\mathcal{L}$, roughly 50% of the magnitude of $\mathbf{W}_{out}^{(2)}$ element values are greater than 0.08. However, when 20% of the total amount of training data is taken from $\mathcal{L}'$, i.e., $\kappa = 0.2$, then 50% of the magnitude of $\mathbf{W}_{out}^{(2)}$ element values are less than 0.03. This means that including only a modest amount of data from $\mathcal{L}'$ during the training reduces the magnitude of $\mathbf{W}_{out}^{(2)}$ element values by nearly a factor of 3. This trend continues as κ is further increased, and for $\kappa = 0.48$, which corresponds to a slight imbalance in the amount of training data, more than 50% of $\mathbf{W}_{out}^{(2)}$ elements have a magnitude of less than 0.002. While $\mathbf{W}_{out}^{(2)}$ completely vanishes only when $\kappa = 0.5$, it is still strongly suppressed even when there is an imbalance in the amount of training data.

In Fig. 3, we also highlight the range of κ values ($0.1 \leq \kappa \leq 0.9$) in which the RC is able to achieve multifunctionality. This indicates
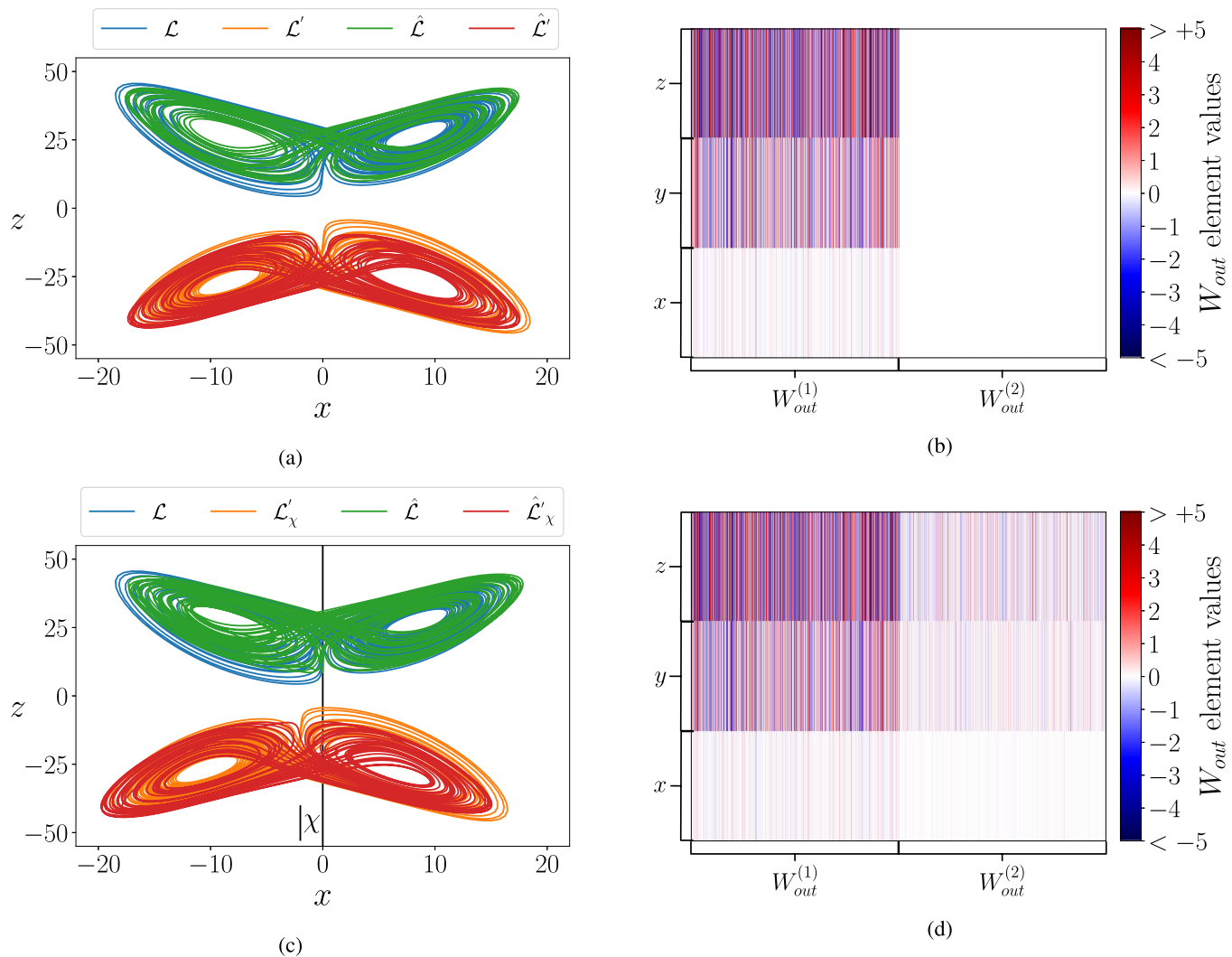
FIG. 1. Illustrating the behavior of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to exhibit multifunctionality in cases of mirror-attractors. (a) Reconstruction of the Lorenz attractor, $\mathcal{L}$ and its mirror-attractor $\mathcal{L}'$ and in (b) color plot of the corresponding trained $\mathbf{W}_{out}$ matrix individual elements. Each of the i rows in $\mathbf{W}_{out}$ is denoted by the variable it reconstructs. Each column represents the weight, $w_{i,j}$, given to the jth component of $\boldsymbol{q}(\boldsymbol{r}(t)) = (\boldsymbol{r}(t), \boldsymbol{r}^2(t))^T$. The distinction is made between which columns belong to $\mathbf{W}_{out}^{(1)}$ and $\mathbf{W}_{out}^{(2)}$. (c) Reconstruction of $\mathcal{L}$ and the slightly shifted $\mathcal{L}'_\chi$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix.

that even when there is a strong imbalance in the training data used from each attractor, the RC is still able to exhibit multifunctionality.

B. Multifunctionality with overlapping attractors

Symmetry in the RC equations can oftentimes lead to catastrophic consequences once the training data intersect with its mirrored version. This symmetry induced phenomenon was discussed in Ref. 17. In this paper, we consider this problem from the perspective of multifunctionality.

By introducing overlapping regions between the different sources of input data, it becomes much more evident what needs

to happen in order for a RC to achieve multifunctionality. In Sec. II B, we say that a multifunctional RC creates multistability of attractors in $\mathbb{S}$ such that when these attractors are projected to $\mathbb{P}$ using $\mathbf{W}_{out}$, they resemble the target behavior. We now explore this numerically.

In this section, we investigate the behavior of $\mathbf{W}_{out}^{(2)}$ in two different cases of overlapping attractors. First, we focus on the scenario where the RC is trained to reconstruct a coexistence of the Lorenz attractor and its mirror-attractor, which share mutual regions of state space. We then consider training the RC to reconstruct two pairs of overlapping mirrored-Lorenz attractors.
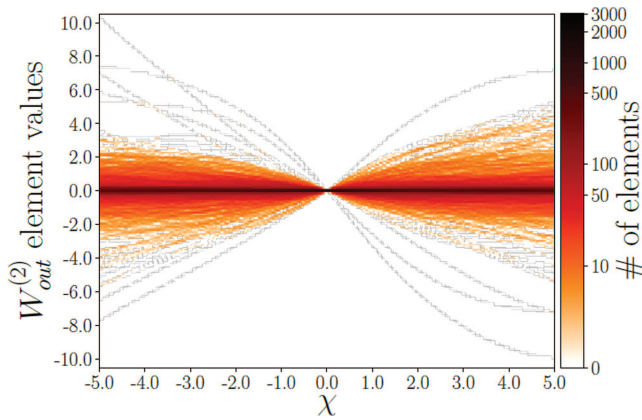
FIG. 2. Histograms of $\mathbf{W}_{out}^{(2)}$ as a function of χ.

1. One pair of mirrored-Lorenz attractors

In order to produce the overlapping regions in the data, we simultaneously shift the location of both $\mathcal{L}$ and $\mathcal{L}'$ along the z-direction by a factor ζ in order to preserve the symmetry. In other words, we use $\boldsymbol{u}_{\mathcal{L}_\zeta}(t) = \left(x(t), y(t), z(t) + \zeta\right)^T$ and $\boldsymbol{u}_{\mathcal{L}'_\zeta}(t) = -\boldsymbol{u}_{\mathcal{L}_\zeta}(t)$ as the input data to train the RC to exhibit multifunctionality. The result of this for $\zeta = -31$ is shown in Figs. 5(a) and 5(b).

In Fig. 5(a), we see that the RC is able to reconstruct the overlapping Lorenz attractors. The particular $\mathbf{W}_{out}$ matrix that is used to achieve multifunctionality is shown in Fig. 5(b). Here, we see that even in the case when the attractors are overlapping, the symmetry in the training data eliminates any contribution from $\mathbf{W}_{out}^{(2)}$ to the RC dynamics.

We now check if $\mathbf{W}_{out}^{(2)}$ will come into existence once the symmetry in the training data is broken by shifting the mirror-attractor along the x-axis by a factor χ like in Sec. IV B. In this case, we now train the RC with $\boldsymbol{u}_{\mathcal{L}_\zeta}(t)$ as before and $\boldsymbol{u}_{\mathcal{L}'_{\chi,\zeta}}(t)$
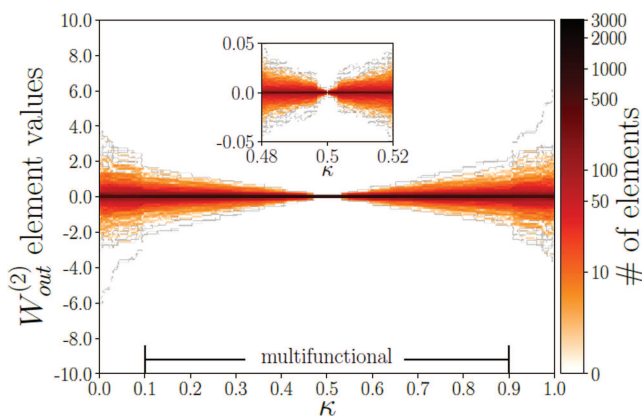

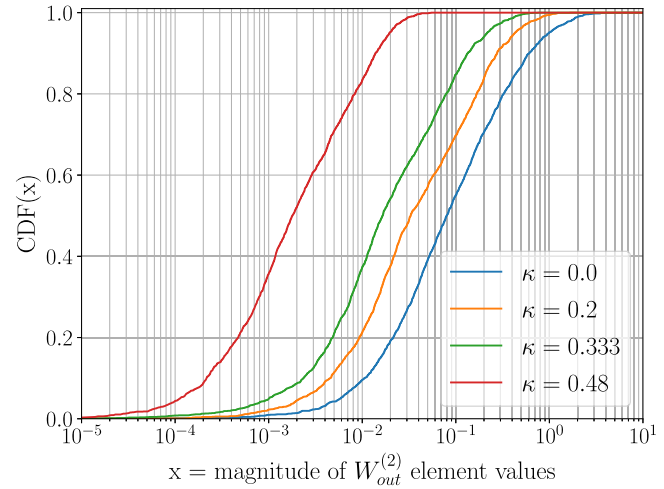**FIG. 3.** Histograms of $\mathbf{W}_{out}^{(2)}$ as a function of κ.

FIG. 4. Cumulative density function of the magnitude of $\mathbf{W}_{out}^{(2)}$ element values.

$= \left(-x(t) - \chi, -y(t), -z(t) - \zeta\right)^T$. In Figs. 5(c) and 5(d), the result of the training for $\chi = 8$ and $\zeta = -31$ is shown.

Figures 5(c) and 5(d) precisely illustrate what the analytical work in Sec. III predicts. Once the symmetry in the training data is broken, then $\mathbf{W}_{out}^{(2)}$ exists even if there is an overlap between the different trajectories described by the training data.

We now investigate how this result scales to the case of four overlapping Lorenz attractors.

2. Two pairs of mirrored-Lorenz attractors

There are many different ways in which we can realize the extent of our analytical results in Sec. III in a numerical setting. As long as the training data satisfy the conditions in Eqs. (14) and (15), then we see from Eq. (27) that in general "symmetry kills the square" when training the RC to reconstruct any number of pairs of mirrored-attractors.

In order to illustrate this numerically, we adapt the blending technique from Sec. II B to train a RC to achieve multifunctionality with more than two attractors. Instead of using the blending parameter α in the matrix concatenation step we follow the training procedure presented in Ref. 17. In this approach, the resultant $\mathbf{X}_C$ and $\mathbf{Y}_C$ matrices that are used in the Eq. (5) are a concatenation of all the individual reservoir and input data training matrices, respectively.

In this section, we explore the behavior of $\mathbf{W}_{out}^{(2)}$ when training the RC to reconstruct two pairs of mirrored-Lorenz attractors with overlapping regions between all four attractors. We use the original Lorenz data, $\boldsymbol{u}_{\mathcal{L}}(t)$, to generate the training input to the RC for these four attractors. Using the shifting parameters, χ and ζ, we write the input from the four Lorenz attractors as $\boldsymbol{u}_{\mathcal{L}_1} = \left(x(t) + \chi, y(t), z(t) + \zeta\right)^T$, $\boldsymbol{u}_{\mathcal{L}'_1} = -\boldsymbol{u}_{\mathcal{L}_1}$, $\boldsymbol{u}_{\mathcal{L}_2} = \left(-x(t) - \chi, -y(t), z(t) + \zeta\right)^T$, and $\boldsymbol{u}_{\mathcal{L}'_2} = -\boldsymbol{u}_{\mathcal{L}_2}$. The result of this for $\chi = -15$ and $\zeta = -40$ is illustrated in Figs. 6(a) and 6(b).
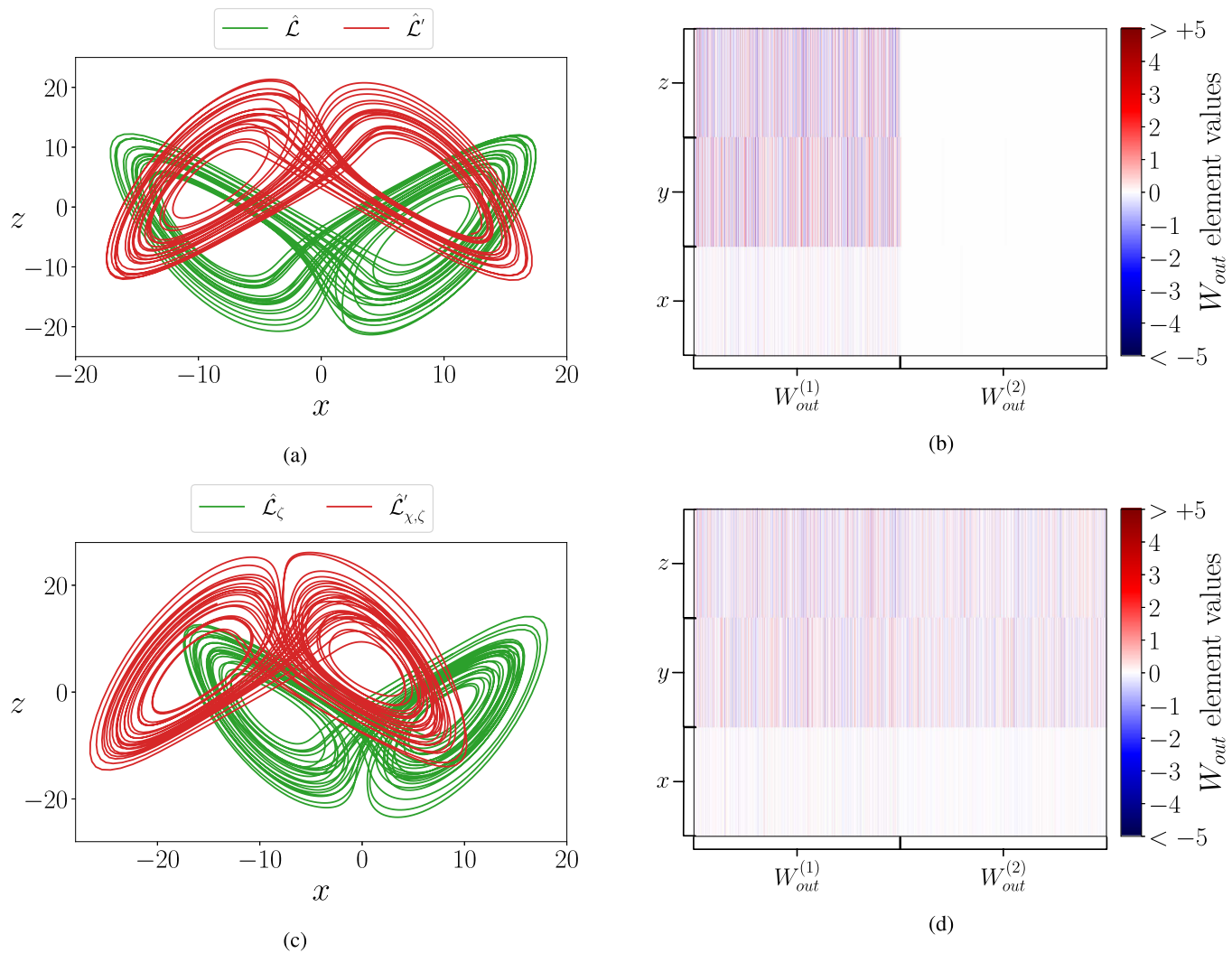
FIG. 5. Illustrating the behavior of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to exhibit multifunctionality in cases of overlapping mirror-attractors. (a) Reconstruction of the overlapping $\mathcal{L}_{\zeta}$ and its mirror-attractor $\mathcal{L}'_{\zeta}$ and in (b) plot of the corresponding trained $\mathbf{W}_{out}$ matrix. (c) Reconstruction of $\mathcal{L}_{\zeta}$ and the slightly shifted $\mathcal{L}'_{\chi,\zeta}$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix. Description of $\mathbf{W}_{out}$ same as in Fig. 1.

In Fig. 6(a), we see that the RC exhibits multifunctionality by successfully reconstructing each of the four Lorenz attractors. The analytical results from Sec. III continue to hold as we see from Fig. 6(b) that $\mathbf{W}_{out}^{(2)}$ is prohibited from contributing to the prediction due to the symmetry between each individual pair of the mirrored-Lorenz attractors in the training data.

To break the symmetry, we shift the location of only $\mathcal{L}'_1$ in the positive x and z directions. In Figs. 6(c) and 6(d), we see that despite the symmetry being present for $\mathcal{L}_2$ and $\mathcal{L}'_2$, it is sufficient to only break the symmetry between the other Lorenz attractors, $\mathcal{L}_1$ and $\mathcal{L}'_1$, for $\mathbf{W}_{out}^{(2)}$ to come into existence.

We remark that our results significantly depend on an appropriate choice of the spectral radius, ρ. The state of the RC must have sufficient knowledge of its previous states in order to avoid spontaneously switching over to any of the other attractors when approaching a region of overlap. By increasing the value of ρ from 0.2 in Sec. IV A to 1.7 in this current section, we found that the RC was then able to distinguish between the different input data sources in order to exhibit multifunctionality. When increasing the amount of overlap between the attractors, the RC required larger values of ρ in order to achieve multifunctionality. On the other hand, if ρ is too large, then there is a danger that the RC can be pushed beyond the edge of chaos.[5]
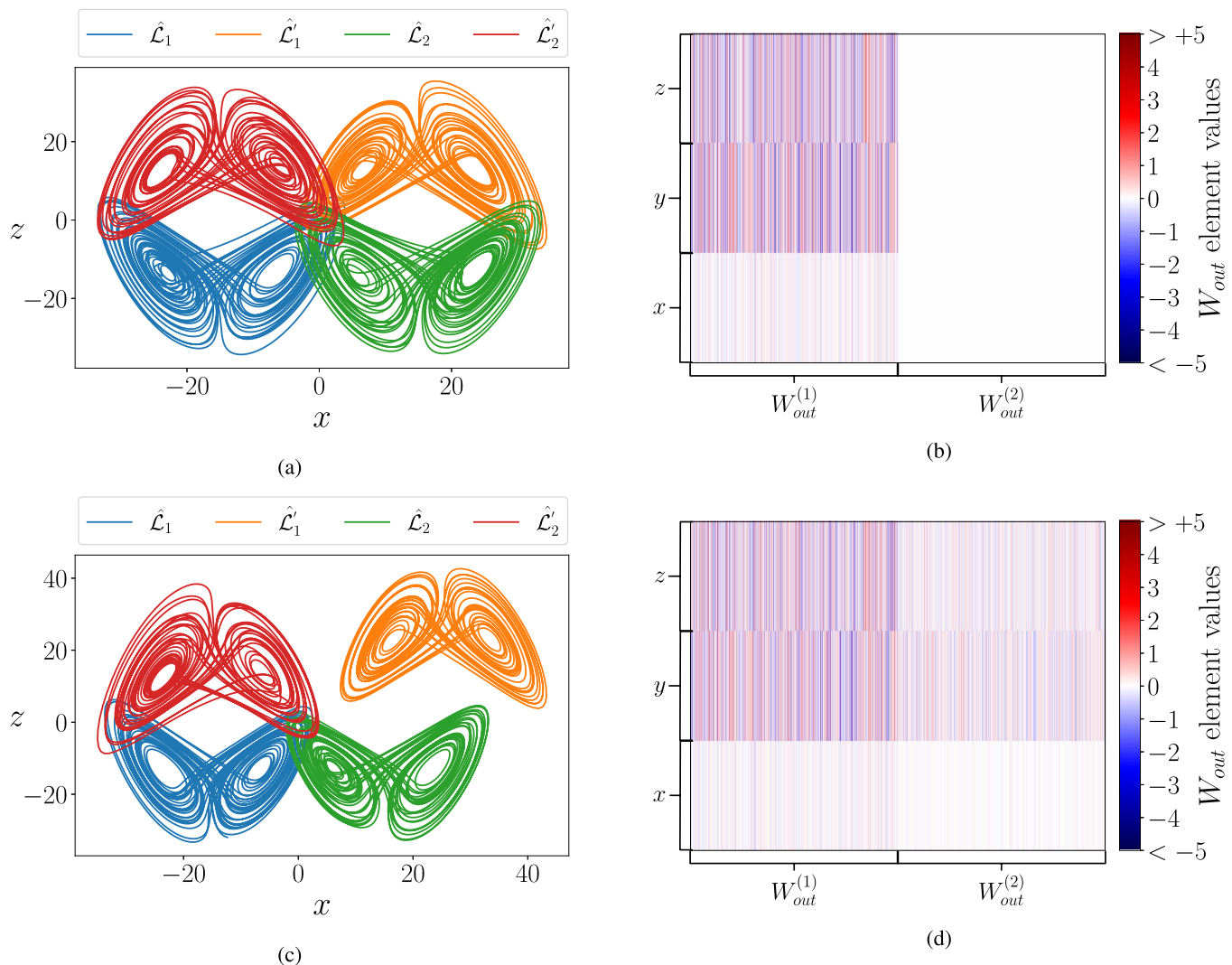
FIG. 6. Illustrating the behavior of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to exhibit multifunctionality in cases of two pairs of overlapping mirror-attractors. (a) Reconstruction of the overlapping $\mathcal{L}_1$, $\mathcal{L}_1'$, $\mathcal{L}_2$, and $\mathcal{L}_2'$ and in (b) plot of the corresponding trained $\mathbf{W}_{out}$ matrix. (c) Reconstruction of overlapping $\mathcal{L}_1$, $\mathcal{L}_1'$, $\mathcal{L}_2$, and the slightly shifted $\mathcal{L}_2'$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix. Description of $\mathbf{W}_{out}$ same as in Fig. 1.

However, studying the relationship between ρ, overlapping data, and number of attractors goes beyond the current scope of this paper and we choose to leave this for future work.

Next, we investigate the analytical results in Sec. III B 2 using a paradigmatic example that shows that at large values of ρ, $\mathbf{W}_{out}^{(2)}$ appears despite training the RC using an anti-symmetric input.

C. $\mathbf{W}_{out}^{(2)}$ reappears at large ρ

So far, in our numerical experiments, we have only been able to observe the appearance of $\mathbf{W}_{out}^{(2)}$ when both Eqs. (14) and (15) are violated. In this section, we show that $\mathbf{W}_{out}^{(2)}$ can come into existence in

the case where Eq. (14) holds but Eq. (15) does not at critical values of the spectral radius, ρ.

We focus on the case outlined in Sec. III B where a single source of input training data is already anti-symmetric as described by Eq. (28). We construct this scenario numerically using the following source of periodic training data:

$$\boldsymbol{u}_{\mathcal{C}}(t) = \begin{pmatrix} 5\cos t \\ 5\sin t \end{pmatrix}, \qquad (31)$$

which describes an orbit on a circle of radius 5 rotating anti-clockwise at constant angular velocity with period $T = 2\pi$.
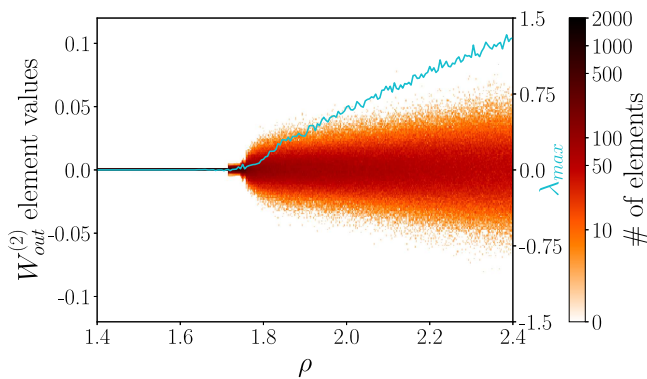
FIG. 7. Histograms of $\mathbf{W}_{out}^{(2)}$ as a function of ρ. Also plotted is the largest Lyapunov exponent of the predicting RC in Eq. (8), $\lambda_{\max}$, as a function of ρ.

This input has the symmetry described in Eq. (28). Therefore, when $\boldsymbol{u}_{\mathcal{C}}(t)$ is used to drive the RC in Eq. (1) and the state of the RC fulfills the condition in Eq. (29), then Eqs. (14) and (15) are satisfied. So, $\mathbf{W}_{out}^{(2)}$ is prevented from coming into existence.

Tuning the spectral radius, ρ, is crucial to a successful training outcome. We find in our numerical experiments that if ρ is too large, then Eq. (29) does not hold despite Eq. (28) being fulfilled. We now explore the behavior of $\mathbf{W}_{out}^{(2)}$ when the RC is trained with $\boldsymbol{u}_{\mathcal{C}}(t)$ as input to Eq. (1) for $\rho \in [1.4, 2.4]$.

In Fig. 7, we plot a histogram of the $\mathbf{W}_{out}^{(2)}$ matrix elements as a function of ρ. The darker the color each point is in Fig. 7, the more there are elements of $\mathbf{W}_{out}^{(2)}$ at a given value. Also included in Fig. 7 is the largest Lyapunov exponent of the predicting RC in Eq. (8), $\lambda_{\max}$, as a function of ρ.

In this picture, we see that for $\rho < \rho_s \approx 1.715$, all elements of $\mathbf{W}_{out}^{(2)}$ are 0 and $\lambda_{\max} = 0$, indicating that the RC obeys the symmetry conditions specified in Sec. III B. However, we see that for $\rho_s < \rho < \rho_c = 1.78$, a small number of $\mathbf{W}_{out}^{(2)}$ elements are now nonzero. At the same time, $\lambda_{\max} \approx 0$ within this range of ρ values. For $\rho > \rho_c$,

the RC goes beyond the edge of chaos and as $\lambda_{\max}$ increases from 0, we see the elements of $\mathbf{W}_{out}^{(2)}$ begin to grow much faster.

The RC's predicted trajectory on $\mathcal{C}$ at each of these stages, $\rho < \rho_s$, $\rho_s < \rho < \rho_c$, and $\rho > \rho_c$ is shown in Fig. 8.

Here, we see for increasing values of ρ, the reconstruction of $\mathcal{C}$ goes from good, to bad, to ugly. For $\rho = 1.6$, the RC can successfully reconstruct trajectories on $\mathcal{C}$. However, when $\rho = 1.75$, the reconstruction of $\mathcal{C}$ as viewed in $\mathbb{P}$ is neither chaotic nor symmetric. The RC fails to reconstruct trajectories on the cycle $\mathcal{C}$ for $\rho = 2.2$.

We remark that the figures provided in this section correspond to one sample of 100 different random realizations of $\mathbf{M}$ and $\mathbf{W}_{in}$. Qualitatively, the behavior was relatively similar for all instances. This step was taken to ensure that our results are not an artifact of a particular initialization of the RC but are direct characteristics of the RC itself.

V. CONCLUSION

By including certain symmetries in the training data, we are able to "open the black-box" and provide a greater understanding of how a RC learns to perform certain tasks in addition to exposing some of the underlying phenomena that arise when translating multifunctionality to a RC.

Breaking the symmetry intrinsic to certain formulations of a RC is necessary in order to mitigate the influence of "mirror-attractors" on the RC's learning capacity.[17] We focus on the "squaring technique," introduced in Ref. 14 and described in Eq. (3). This approach results in the creation of a "square readout matrix," $\mathbf{W}_{out}^{(2)}$, that breaks the symmetry in the RC's readout layer and destroys the mirror-attractors.

In this paper, we explore the behavior of $\mathbf{W}_{out}^{(2)}$ in the case where there are already certain symmetries present in the training data. In Sec. III, we prove that when the RC is trained to reconstruct anti-symmetric training data when using the ridge regression technique described in Eq. (5), it then follows that $\mathbf{W}_{out}^{(2)}$ must vanish. We provide examples in Eqs. (22), (27), and (28) of anti-symmetric training data that can under certain conditions "kill the square." In Sec. IV, we explore each of these examples in a numerical setting. We
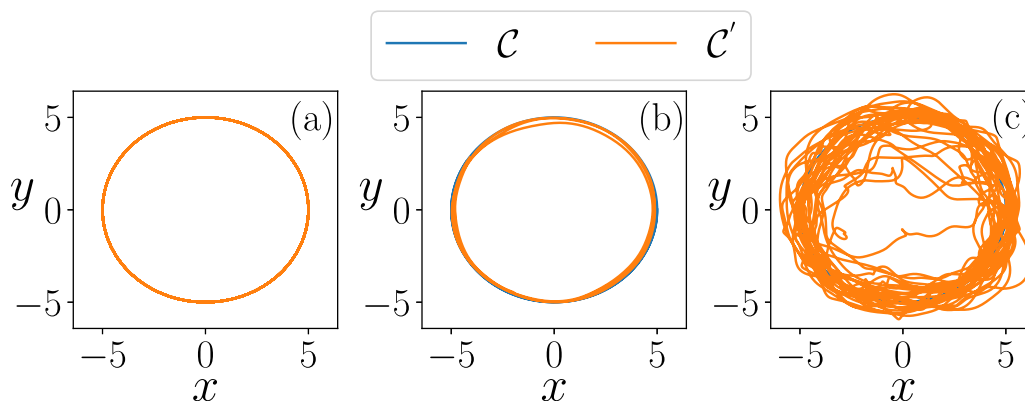

FIG. 8. Prediction of the circular trajectory $\mathcal{C}$. (a) $\rho = 1.6$, (b) $\rho = 1.75$, and (c) $\rho = 2.2$.

consider the cases described in Eqs. (22) and (27) from the perspective of multifunctionality. More specifically, we investigate the behavior of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to reconstruct a coexistence of the Lorenz attractor and its mirror-attractor. We show in Fig. 1 that when the location of one attractor is slightly shifted, which thereby conflicts with the conditions in Eqs. (14) and (15), only then does $\mathbf{W}_{out}^{(2)}$ appear. We see the same effect occur in Fig. 5 with the added complication of overlapping regions between the pair of mirrored-Lorenz attractors. This result demonstrates that multifunctionality can be used to overcome the harmful properties of symmetry as described in Ref. 17.

The results in Figs. 2–4 demonstrate that the conditions defined in Eqs. (14) and (15) are robust to small changes in both χ from 0 and κ from 0.5 as the majority of $\mathbf{W}_{out}^{(2)}$ element values remain relatively close to 0. However, the more we change χ and κ the larger the elements of $\mathbf{W}_{out}^{(2)}$ become.

To explore the square killing scenario described by Eq. (27) in a numerical setting, we consider training the RC to reconstruct two pairs of mirrored-Lorenz attractors in Sec. IV B 2. In Fig. 6, we demonstrate that it is sufficient to break the symmetry between one pair of mirrored-Lorenz attractors in order for $\mathbf{W}_{out}^{(2)}$ to come into existence.

In Sec. IV C, we construct the case of a single anti-symmetric trajectory considered in Eq. (28) by using a circular orbit described by Eq. (31). Here, we demonstrate that when the RC is successfully able to reconstruct a trajectory on this circular orbit, there is no contribution from $\mathbf{W}_{out}^{(2)}$. However, if the spectral radius of the RC's internal connections, ρ, is too large, then the prediction breaks down as the RC begins to exhibit chaos and $\mathbf{W}_{out}^{(2)}$ emerges.

This paper demonstrates the benefits of studying the effects of symmetries in a RC setup and serves as a route to better understand the fundamentals of this machine learning paradigm which we intend to further pursue. Overall, the two-armed approach of our analytical results together with our numerical experiments provide a much greater insight into the relationship between symmetry and multifunctionality in a RC. The inclusion of exotic cases like overlapping regions between attractors in the prediction state space and achieving multifunctionality with four attractors further increase the robustness of our results and showcase some of the interesting dynamics a RC can be trained to simultaneously reconstruct. We identify that there is a connection between these newly demonstrated abilities of a multifunctional RC and the spectral radius, ρ. However, exploring the important role of ρ in these scenarios goes beyond the scope of the current paper and we choose to leave this for future work.

ACKNOWLEDGMENTS

APPENDIX: RC DESIGN AND TRAINING PARAMETERS

In all of our numerical simulations, we set the number of neurons in the reservoir as $N = 1000$. The neurons are connected with a sparse Erdös–Renyi topology. In this case, the adjacency matrix, $\mathbf{M}$, is designed such that each element is chosen independently to be nonzero with probability $P = 0.04$ and these nonzero elements are chosen uniformly from $(-1, 1)$. This random sparse matrix is then rescaled such that it has a specific spectral radius, ρ. The elements of the input matrix, $\mathbf{W}_{in}$, are also chosen randomly such that each row has only one nonzero randomly assigned element, chosen uniformly from $(-1, 1)$.

The numerical results illustrated in Figs. 1–6 are generated using the same $\mathbf{M}$ and $\mathbf{W}_{in}$ in order to provide consistency in our discussion when relating one result to another. We remark that there are relatively small quantitative changes to our results when using different initializations of $\mathbf{M}$ and $\mathbf{W}_{in}$; however, the main characteristics of our results remain similar. The same is said for Figs. 7 and 8, but the dimension of $\mathbf{W}_{in}$ is decreased in order to take into consideration the lower-dimensional training data.

DATA AVAILABILITY

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

REFERENCES

[1] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, Neural Netw. **20**, 391 (2007).
[2] H. Jaeger, German National Research Center for Information Technology GMD Technical Report, Bonn, Germany, 2001, p. 148.
[3] W. Maass, T. Natschläger, and H. Markram, Neural Comput. **14**, 2531 (2002).
[4] L. Grigoryeva and J.-P. Ortega, Neural Netw. **108**, 495 (2018).
[5] T. L. Carroll, Chaos **30**, 121109 (2020).
[6] A. G. Hart, J. L. Hook, and J. H. Dawes, Physica D **421**, 132882 (2021).
[7] A. Hart, J. Hook, and J. Dawes, Neural Netw. **128**, 234 (2020).
[8] P. Verzelli, C. Alippi, and L. Livi, arXiv:2010.02860 (2020).
[9] Z. Lu, B. R. Hunt, and E. Ott, Chaos **28**, 061104 (2018).
[10] H. Jaeger and H. Haas, Science **304**, 78 (2004).
[11] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Phys. Rev. Lett. **120**, 024102 (2018).
[12] A. Griffith, A. Pomerance, and D. J. Gauthier, Chaos **29**, 123108 (2019).
[13] A. Haluszczynski and C. Räth, Chaos **29**, 103143 (2019).
[14] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, Chaos **27**, 041102 (2017).
[15] E. N. Lorenz, J. Atmos. Sci. **20**, 130 (1963).
[16] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, Nonlinear Process. Geophys. **27**, 373 (2020).
[17] J. Herteux and C. Räth, Chaos **30**, 123142 (2020).
[18] K. L. Briggman and W. Kristan, Jr., Annu. Rev. Neurosci. **31**, 271 (2008).
[19] A. Flynn, V. A. Tsachouridis, and A. Amann, Chaos **31**, 013125 (2021).
[20] K. Nakajima, Jpn. J. Appl. Phys. **59**, 060501 (2020).
[21] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Neural Netw. **115**, 100 (2019).
[22] T. L. Carroll and L. M. Pecora, Chaos **29**, 083130 (2019).
[23] T. Lymburn, S. D. Algar, M. Small, and T. Jüngling, Chaos **31**, 033121 (2021).
[24] W. A. Barbosa, A. Griffith, G. E. Rowlands, L. C. Govia, G. J. Ribeill, M.-H. Nguyen, T. A. Ohki, and D. J. Gauthier, arXiv:2102.00310 (2021).
[25] K. L. Briggman and W. B. Kristan, J. Neurosci. **26**, 10925 (2006).
[26] I. R. Popescu and W. N. Frost, J. Neurosci. **22**, 1985 (2002).
[27] S. Lieske, M. Thoby-Brisson, P. Telgkamp, and J. Ramirez, Nat. Neurosci. **3**, 600 (2000).
[28] K. Inoue, K. Nakajima, and Y. Kuniyoshi, Sci. Adv. **6**, eabb3989 (2020).
[29] Z. Lu and D. S. Bassett, Chaos **30**, 063133 (2020).
[30] A. Ceni, P. Ashwin, and L. Livi, Cognit. Comput. **12**, 330 (2020).