



## OPEN ACCESS

## EDITED BY

Kalel Rossi,  
University of Oldenburg, Germany

## REVIEWED BY

Bruno Boaretto,  
Federal University of São Paulo, Brazil  
Everton Santos Medeiros,  
Universidade Estadual Paulista, UNESP, Brazil

## \*CORRESPONDENCE

Andrew Flynn,  
✉ andrew\_flynn@umail.ucc.ie

RECEIVED 19 June 2024

ACCEPTED 16 September 2024

PUBLISHED 03 October 2024

## CITATION

Flynn A and Amann A (2024) Exploring the origins of switching dynamics in a multifunctional reservoir computer. *Front. Netw. Physiol.* 4:1451812. doi: 10.3389/fnetp.2024.1451812

## COPYRIGHT

© 2024 Flynn and Amann. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Exploring the origins of switching dynamics in a multifunctional reservoir computer

Andrew Flynn<sup>1,2\*</sup> and Andreas Amann<sup>1,3</sup>

<sup>1</sup>School of Mathematical Sciences, University College Cork, Cork, Ireland, <sup>2</sup>INFANT Research Centre, University College Cork, Cork, Ireland, <sup>3</sup>Potsdam Institute for Climate Impact Research, Potsdam, Germany

The concept of multifunctionality has enabled reservoir computers (RCs), a type of dynamical system that is typically realized as an artificial neural network, to reconstruct multiple attractors simultaneously using the same set of trained weights. However, there are many additional phenomena that arise when training a RC to reconstruct more than one attractor. Previous studies have found that in certain cases, if the RC fails to reconstruct a coexistence of attractors, then it exhibits a form of metastability, whereby, without any external input, the state of the RC switches between different modes of behavior that resemble the properties of the attractors it failed to reconstruct. In this paper, we explore the origins of these switching dynamics in a paradigmatic setting via the “seeing double” problem.

## KEYWORDS

reservoir computer, multifunctionality, multistability, metastability, chaos, network physiology, chaotic itinerancy, machine learning

## 1 Introduction

Multifunctionality is the term used to describe a neural network that has the ability to perform multiple tasks without changing any of its connections. Multifunctionality is an essential property of certain biological neural networks and has been an active area of research in neuroscience since the mid-1980s, with seminal work published in [Mpitsos and Cohan \(1986\)](#) and [Getting \(1989\)](#), followed by further review papers by [Dickinson \(1995\)](#) and [Marder and Calabrese \(1996\)](#), and more recently, reviewed in [Briggman and Kristan, \(2008\)](#). These studies have identified that a multifunctional neural network in principle resembles a multistable dynamical system. In this sense, for each task that the network performs, there is an attractor associated with it. This attractor is in coexistence with several other attractors in the network’s state space, and each attractor is related to one of the tasks that the network performs. Therefore, in order to perform a given task, the multifunctional network requires a cue in the form of an initial condition in the basin of attraction of the attractor associated with the task.

Taking all of the above into account, where this ability to harness multistability becomes immediately relevant is in the domain of machine learning (ML), as multifunctionality can be used to unlock additional computational capabilities of artificial neural networks (ANNs) that would otherwise have remained dormant. In [Flynn et al. \(2021b\)](#), multifunctionality was achieved in an artificial setting for the first time via the reservoir computing approach to ML. This involved training a “reservoir computer” (RC), which in this case was a dynamical system in the form of an ANN, to reconstruct a coexistence of chaotic attractors from different dynamical systems using the same set of trained weights. This RC was driven with

input from these chaotic attractors, and the RC's response dynamics to the different driving inputs were used to obtain a readout layer to replace the drive, after which the RC became a multistable system that reconstructed a coexistence of the chaotic attractors. In this example, to perform a particular task, i.e., to reconstruct a particular chaotic attractor, the multifunctional RC is like any other multistable dynamical system and only needs to be initialized with an initial condition in the basin of attraction of the corresponding attractor.

There are many additional phenomena that can arise and also factors to consider when training an RC to reconstruct more than one attractor simultaneously. For instance, it was shown in Flynn et al. (2021b) that multifunctionality becomes increasingly difficult to achieve with the increase in the difference of the time scales of the attractors that the RC is trained to reconstruct. Furthermore, in Flynn et al. (2023), where the RC was trained to solve the "seeing double" problem that involves training the RC to construct a coexistence of attractors that describe clockwise and anticlockwise trajectories on two circular orbits, it was shown that by manually shifting the location of the training data describing these orbits, the closer the orbits are to one another, the more difficult it is for the RC to achieve multifunctionality. Remarkably, for a small range of training parameters, it was found that the RC achieves multifunctionality even when the orbits are overlapping in state space (in the sense that the training data used to drive the RC contain identical data points from the different orbits). In Flynn et al. (2023) and Flynn (2023), it was shown that in certain cases, when the RC fails to achieve multifunctionality, it instead produces a variety of episodic switching patterns between different metastable states that resemble the dynamics it failed to reconstruct. Through further investigation of the seeing double problem, we have found a similar phenomenon to occur when the orbits are moved closer together. The purpose of this paper is to examine the origins of these switching dynamics in much greater detail.

We explore the origins of the transition from multifunctionality to metastable switching dynamics in much greater detail. We find that for a small change in the spectral radius of the RC's internal connectivity matrix, the RC first fails to reconstruct one of the orbits as the corresponding reconstructed attractor becomes unstable, and it is only after a relatively long transient that the RC approaches the other reconstructed orbit (which is the only stable attractor present in the system). After another small change in the spectral radius, the other reconstructed orbit also becomes unstable, and this results in RC switching between the dynamics of these two unstable states. On closer inspection, we find that when the second attractor becomes unstable, there is a new attractor created that facilitates these switching dynamics. This new attractor is created through this sequence of attractors becoming unstable because due to the RC's design, and it is prohibited from becoming globally unstable. We show that these switching dynamics appear when the orbits are brought closer together, touch, and overlap. From computing the probability density of different residence times in each of the metastable states, we find a sawtooth-like pattern consisting of multiple branches of exponentially distributed points, where each branch describes a particular path taken by the RC on each of the metastable states.

## 2 Methods

In this section, we introduce the particular RC that is studied, describe how this RC is trained to achieve multifunctionality, and outline the specifics of the seeing double problem, the task that the RC is trained to solve. We follow the same procedure as in Flynn et al. (2023).

### 2.1 Reservoir computing

#### 2.1.1 Central philosophy of reservoir computing

Today, the term "reservoir computer" (RC) is generally used to describe a dynamical system that, for instance, can be realized as an ANN and trained to solve certain machine learning (ML) problems without explicitly training the internal structure of the system. As outlined in Nakajima and Fischer (2021), the reservoir computing approach to ML received its name in Verstraeten et al. (2005), where the term reservoir computer was coined as a means to establish a new ML framework based on the common concepts of echo-state networks (ESNs) Jaeger (2001) and liquid-state machines (LSMs) Maass et al. (2002). These are two independently proposed designs of ANNs with recurrent connections (RNNs) that share the following philosophy: instead of training all the weights in a network, it is sufficient to only optimize the weights of a readout layer in order to solve a particular problem. This ideological shift in training RNNs 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. The role of this reservoir is to enable the state of the RC to become a representation of the history of training input signals related to a particular task, and only a readout layer needs to be found in order to project this information out of the RC to solve the given task. Multifunctionality extends this philosophy by demonstrating that an RC's response to several different sequences of training input signals, each of which is related to a particular task, can be harnessed to produce a single RC that performs all of these tasks using the same readout layer.

#### 2.1.2 RC formulation

The RC that is studied throughout this paper was introduced in Lu et al. (2018); before this RC is trained, it is defined as the following ANN in the form of a non-autonomous dynamical system, which we refer to as the "open-loop RC":

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

$$\mathbf{r}(0) = \mathbf{0}^T. \quad (2)$$

In Equation 1,  $\mathbf{r}(t) \in \mathbb{R}^N$  describes the state of the open-loop RC at a given time  $t$  and  $N$  is the number of artificial neurons in the network. Solutions of Equation 1 are computed using the 4<sup>th</sup> order Runge–Kutta method with time step  $\tau = 0.01$ .  $\gamma$  is a decay-rate parameter that arises during the derivation of Equation 1 from the discrete-time design proposed by Jaeger (2001). The tanh "activation function" is a pointwise operation and is defined as  $\tanh(\cdot): \mathbb{R}^N \rightarrow \mathbb{R}^N$ . The adjacency matrix,  $\mathbf{M} \in \mathbb{R}^{N \times N}$ , plays the role of the "reservoir." The input strength parameter,  $\sigma$ , and the input matrix,  $\mathbf{W}_{in} \in \mathbb{R}^{N \times D}$ , when multiplied together represent the weight given to the  $D$ -dimensional driving input,  $\mathbf{u}(t) \in \mathbb{R}^D$ , as it is

projected into the open-loop RC. We use the superscript  $T$  to denote the vector transpose operation. The initial condition, specified in Equation 2, is the same for all experiments that were carried out.

The elements of  $\mathbf{M}$  and  $\mathbf{W}_{in}$  are the same that are used in Flynn et al. (2023) in order to provide a direct comparison to the results of this present paper.  $\mathbf{M}$  was designed by first constructing a random sparse matrix, where each element is chosen independently to be nonzero with probability  $P = 0.04$  (i.e., sparsity =  $P$  or degree =  $N/P$ ), and these nonzero elements are chosen uniformly from  $(-1, 1)$ . The elements of this random sparse matrix are subsequently rescaled so that the resultant matrix, which we then call  $\mathbf{M}$ , has a specific spectral radius denoted by  $\rho$ , which is the magnitude of the largest eigenvalue of  $\mathbf{M}$ . The corresponding input matrix,  $\mathbf{W}_{in}$ , was designed such that each row has only one nonzero randomly assigned element that is chosen uniformly from  $(-1, 1)$ . As a result, each neuron is driven with only one component of  $\mathbf{u}(t)$ .

Building on the results of Flynn et al. (2023),  $\rho$  is again shown to play a significant role in producing the switching dynamics that are studied in this paper.  $\rho$  has also been a key parameter in previous results on training an RC to achieve multifunctionality; see: Flynn et al. (2021b,a, 2022); Morra et al. (2023). One of the main reasons why  $\rho$  is such an influential parameter of this RC is that it is used to tune how previous states of the RC impact the current state. This becomes particularly important in scenarios involving overlapping training data because there must be a sufficiently large weight placed on previous states in order for the RC to distinguish between identical data points from the different sets of training data. In this paper, we find that when  $\rho$  is not sufficiently large, the RC cannot easily distinguish between the different orbits, which, in certain scenarios, leads to the state of RC switching between the orbits.

### 2.1.3 Training a RC to achieve multifunctionality

We now outline the steps involved in training Equation 1 to achieve multifunctionality. To illustrate this procedure, we consider the case of training the open-loop RC in Equation 1 to reconstruct a coexistence of two attractors,  $\mathcal{A}_1, \mathcal{A}_2 \subset \mathbb{R}^D$ , given access to a trajectory on each attractor described by  $\mathbf{u}_{(\mathcal{A}_1)}(t) \in \mathcal{A}_1$  and  $\mathbf{u}_{(\mathcal{A}_2)}(t) \in \mathcal{A}_2$ . In the case of multifunctionality, the aim of the training is to determine a “readout function/layer,” defined as  $\hat{\psi}(\cdot): \mathbb{R}^N \rightarrow \mathbb{R}^D$ , which enables us to replace  $\mathbf{u}(t)$  in Equation 1 with  $\hat{\psi}(\cdot)$  and form a new “closed-loop RC,” which is capable of reconstructing a coexistence of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . In this paper,  $\hat{\psi}(\cdot)$  is constructed as:

$$\hat{\psi}(\mathbf{r}(t)) = \mathbf{W}_{out} \mathbf{q}(\mathbf{r}(t)), \tag{3}$$

where  $\mathbf{W}_{out} \in \mathbb{R}^{D \times 2N}$  is the “readout matrix,” and  $\mathbf{q}(\mathbf{r}(t)) \in \mathbb{R}^{2N}$  is given by:

$$\mathbf{q}(\mathbf{r}(t)) = (\mathbf{r}(t) \ \mathbf{r}^2(t))^T, \tag{4}$$

where  $\mathbf{r}^2(t) = (r_1^2(t), r_2^2(t), \dots, r_N^2(t))^T$ . The purpose of  $\mathbf{q}(\cdot)$ , as specified in Equation 4 is to prevent the occurrence of “mirror-attractors,” which can impede the ability of the RC to reconstruct attractors, as reported in Herteux and R ath (2020) and Flynn et al. (2021a). To compute  $\mathbf{W}_{out}$  in Equation 3, we use a ridge regression technique, which consists of solving the following equation:

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

where  $\beta$  is the “ridge parameter” and is tuned to reduce the magnitudes of elements in  $\mathbf{W}_{out}$  in order to discourage overfitting,  $\mathbf{I}$  is the identity matrix, and  $\mathbf{X}_C$  and  $\mathbf{Y}_C$  are the training data matrices, which are both constructed as concatenations of two smaller matrices, where  $\mathbf{X}_C = [\mathbf{X}_{\mathcal{A}_1}, \mathbf{X}_{\mathcal{A}_2}]$  and  $\mathbf{Y}_C = [\mathbf{Y}_{\mathcal{A}_1}, \mathbf{Y}_{\mathcal{A}_2}]$ . The elements of these  $\mathbf{X}_{\mathcal{A}_1}$  and  $\mathbf{X}_{\mathcal{A}_2}$  matrices are computed as follows: we first drive the open-loop RC in Equation 1 with input  $\mathbf{u}_{(\mathcal{A}_1)}(t) \in \mathcal{A}_1$  for  $0 < t \leq t_{train}$  and then repeat this process for  $\mathbf{u}_{(\mathcal{A}_2)}(t) \in \mathcal{A}_2$ . The corresponding responses of the open-loop RC to these driving inputs are denoted by  $\mathbf{r}_{(\mathcal{A}_1)}(t)$  and  $\mathbf{r}_{(\mathcal{A}_2)}(t)$ . It is these responses that are used to generate the elements of  $\mathbf{X}_{\mathcal{A}_1}$  and  $\mathbf{X}_{\mathcal{A}_2}$ , where

$$\mathbf{X}_{\mathcal{A}_1} = [\mathbf{q}(\mathbf{r}_{(\mathcal{A}_1)}(t_{listen})) \ \mathbf{q}(\mathbf{r}_{(\mathcal{A}_1)}(t_{listen} + \tau)) \ \dots \ \mathbf{q}(\mathbf{r}_{(\mathcal{A}_1)}(t_{train}))], \tag{6}$$

and similarly for  $\mathbf{X}_{\mathcal{A}_2}$ . The elements of the corresponding  $\mathbf{Y}_{\mathcal{A}_1}$  and  $\mathbf{Y}_{\mathcal{A}_2}$  matrices are defined as:

$$\mathbf{Y}_{\mathcal{A}_1} = [\mathbf{u}_{(\mathcal{A}_1)}(t_{listen}) \ \mathbf{u}_{(\mathcal{A}_1)}(t_{listen} + \tau) \ \dots \ \mathbf{u}_{(\mathcal{A}_1)}(t_{train})], \tag{7}$$

and similarly for  $\mathbf{Y}_{\mathcal{A}_2}$ . The time  $t_{listen}$  is chosen such that at this time, both  $\mathbf{r}_{(\mathcal{A}_1)}(t)$  and  $\mathbf{r}_{(\mathcal{A}_2)}(t)$  are determined by a history of driving inputs and are no longer dependent on the open-loop RC’s initial condition; the duration of time from  $t = 0$  to  $t = t_{listen}$  is known as “the listening stage.” The time  $t_{train}$  is chosen such that  $\mathbf{X}_{\mathcal{A}_1}$  and  $\mathbf{X}_{\mathcal{A}_2}$  contain a sufficiently long representation of a trajectory on  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , and the duration of time from  $t = t_{listen}$  to  $t = t_{train}$  is known as “the training stage.” It is important to highlight that  $\mathbf{M}$ ,  $\mathbf{W}_{in}$ , and all training parameters remain identical when generating  $\mathbf{X}_{\mathcal{A}_1}$  and  $\mathbf{X}_{\mathcal{A}_2}$ .

### 2.1.4 The “closed-loop” RC

After following the steps outlined in the previous section and obtaining  $\mathbf{W}_{out}$  from Equation 5,  $\mathbf{u}(t)$  in Equation 1 can then be replaced by  $\hat{\psi}(\mathbf{r}(t))$ . In Equation 8, we now define the resulting closed-loop RC as the following autonomous dynamical system:

$$\dot{\hat{\mathbf{r}}}(t) = \gamma [-\hat{\mathbf{r}}(t) + \tanh(\mathbf{M} \hat{\mathbf{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out} \mathbf{q}(\hat{\mathbf{r}}(t)))], \tag{8}$$

$$\hat{\mathbf{r}}(0) = \mathbf{r}(t_{train}), \tag{9}$$

where  $\hat{\mathbf{r}}(t)$  denotes the state of the closed-loop RC at a given time  $t$ . While  $\hat{\mathbf{r}}(t)$  and  $\mathbf{r}(t)$  are both  $N$ -dimensional vectors, the purpose of this notation is to distinguish between the dynamics of the closed-loop and open-loop RCs. Furthermore, we consider  $\hat{\mathbf{r}}(t) \in \mathbb{S}$ , where  $\mathbb{S}$  is referred to as the “RC’s state space” and is used henceforth when discussing the dynamics of the closed-loop RC in  $\mathbb{R}^N$ . By computing the solutions of Equation 8, predictions of  $\mathbf{u}(t)$  for  $t > t_{train}$ , denoted as  $\hat{\mathbf{u}}(t)$ , are given by:

$$\hat{\mathbf{u}}(t) = \hat{\psi}(\hat{\mathbf{r}}(t)). \tag{10}$$

Again, while both  $\mathbf{u}(t)$  and  $\hat{\mathbf{u}}(t)$  are  $D$ -dimensional vectors, we use the same convention to indicate that  $\hat{\mathbf{u}}(t)$  is a prediction of  $\mathbf{u}(t)$  at time  $t$ . We also define  $\hat{\mathbf{u}}(t) \in \mathbb{P}$ , where  $\mathbb{P}$  is referred to as the “projected state space” and is used henceforth when discussing these projected dynamics of the closed-loop RC.

To test whether the closed-loop RC has achieved multifunctionality, as indicated by Equation 9, we initialize Equation 8 with  $\hat{\mathbf{r}}(0) = \mathbf{r}_{(\mathcal{A}_1)}(t_{train})$  and  $\mathbf{r}_{(\mathcal{A}_2)}(t_{train})$ , and from

these initial conditions, we examine the long-term projected dynamics of the closed-loop RC in  $\mathbb{P}$ . We say the closed-loop RC has achieved multifunctionality if the long-term dynamical characteristics of  $\hat{\mathbf{u}}_{(\mathcal{A}_1)}(t)$  and  $\hat{\mathbf{u}}_{(\mathcal{A}_2)}(t)$ , defined according to Equation 10, are indistinguishable from  $\mathbf{u}_{(\mathcal{A}_1)}(t)$  and  $\mathbf{u}_{(\mathcal{A}_2)}(t)$ . If this is the case, then we can say that there exists a coexistence of attractors  $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{S}$ , and when the state of the closed-loop RC approaches either  $\mathcal{S}_1$  or  $\mathcal{S}_2$ , the corresponding projected dynamics in  $\mathbb{P}$  are referred to as the “reconstructed attractors,”  $\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2 \subset \mathbb{P}$ , which resemble the long-term dynamics of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . By resembling the long-term dynamics, it is meant that, for instance,  $\mathcal{A}_1$  and  $\hat{\mathcal{A}}_1$  will have nearly identical Poincaré sections when computed for the same region of  $\mathbb{R}^D$  and  $\mathbb{P}$  as  $t \rightarrow \infty$ . If multifunctionality is achieved, then we refer to the resulting multistable closed-loop RC as the “multifunctional RC.”

We comment that  $\hat{\mathbf{r}}(0) = \mathbf{r}_{(\mathcal{A}_1)}(t_{\text{train}})$  and  $\mathbf{r}_{(\mathcal{A}_2)}(t_{\text{train}})$  are not the only initial conditions that will allow the closed-loop RC to reconstruct  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , so long as the closed-loop RC is initialized with a point in the basin of attraction of either  $\mathcal{S}_1$  or  $\mathcal{S}_2$ , the corresponding attractor will be reconstructed in  $\mathbb{P}$ .

## 2.2 Seeing double

The specifics of the “seeing double” problem are outlined in this section. This numerical experiment was introduced in Flynn et al. (2023) as a means to systematically study the issues related to multifunctionality and overlapping training data.

### 2.2.1 Numerical experiment setup

The seeing double problem consists of training an RC to construct a coexistence of attractors such that their dynamics in  $\mathbb{P}$  follow trajectories along two circular orbits of equal radius and rotate in opposite directions. The difficulty of this task is varied by moving the centers of these orbits closer together or further apart. When these orbits are overlapping, the RC is therefore required to distinguish between points in  $\mathbb{R}^D$  that are common to both cycles in order to exhibit multifunctionality.

The driving input to the RC is generated via

$$\mathbf{u}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} b_x \cos(t) + x_{cen} \\ b_y \sin(t) \end{pmatrix}, \quad (11)$$

for  $t = 0, \tau, 2\tau, \dots$ , using the time-step  $\tau = 0.01$ . The resultant time-series of  $\mathbf{u}(t)$ , given by Equation 11, corresponds to a trajectory around a circle of radius  $b = |b_x| = |b_y|$  and centered at  $(x_{cen}, 0)$ .

As in Flynn et al. (2023), for a given  $x_{cen}$ , we set  $b_x = b_y = 5$  in Equation (11) to generate a trajectory about the counter-clockwise circular orbit that we denote as  $\mathcal{C}_A$ , and points along this orbit are written as  $\mathbf{u}_{(\mathcal{C}_A)}(t)$ . For the corresponding  $-x_{cen}$ , we generate a trajectory about the clockwise circular orbit that we denote as  $\mathcal{C}_B$ , and by setting  $b_x = -5$  and  $b_y = 5$ , points along this orbit are written as  $\mathbf{u}_{(\mathcal{C}_B)}(t)$ . By changing  $x_{cen}$ , the centers of these cycles are moved equidistantly along the line  $y = 0$ . An overlapping region between  $\mathcal{C}_A$  and  $\mathcal{C}_B$  exists whenever  $|x_{cen}| < b = 5$ , i.e.,  $\mathcal{C}_A \cap \mathcal{C}_B \neq \emptyset \forall |x_{cen}| < 5$ . Furthermore,  $\mathcal{C}_A$  and  $\mathcal{C}_B$  are said to be “entirely/completely overlapping” when  $x_{cen} = 0$ . In this extreme case, the only difference between  $\mathcal{C}_A$  and  $\mathcal{C}_B$  is the direction of rotation on both cycles.

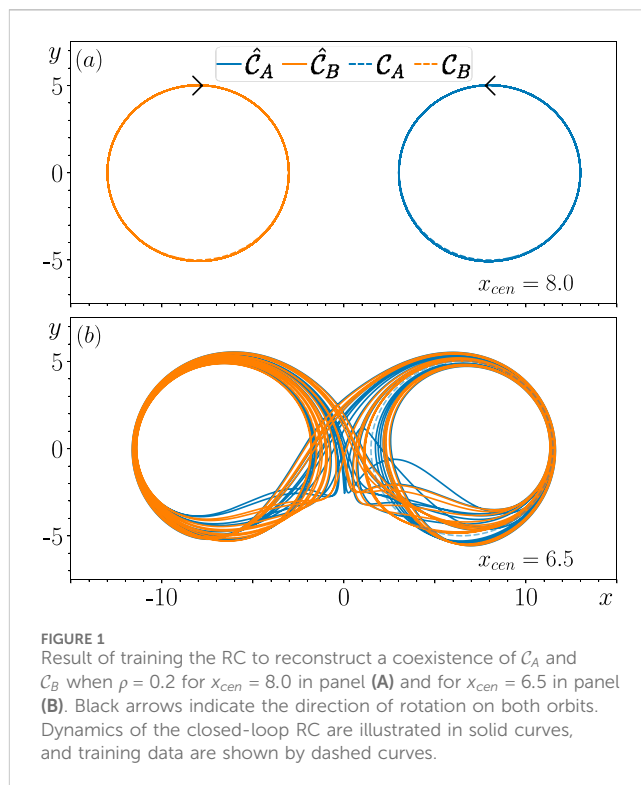


FIGURE 1 Result of training the RC to reconstruct a coexistence of  $\mathcal{C}_A$  and  $\mathcal{C}_B$  when  $\rho = 0.2$  for  $x_{cen} = 8.0$  in panel (A) and for  $x_{cen} = 6.5$  in panel (B). Black arrows indicate the direction of rotation on both orbits. Dynamics of the closed-loop RC are illustrated in solid curves, and training data are shown by dashed curves.

The values of  $\mathbf{u}_{(\mathcal{C}_A)}(t)$  and  $\mathbf{u}_{(\mathcal{C}_B)}(t)$  are used as the input to the open-loop RC in Equation 1 for  $0 \leq t \leq t_{\text{train}}$ . The open-loop RC’s responses to these driving input signals are denoted as  $\mathbf{r}_{(\mathcal{C}_A)}(t)$  and  $\mathbf{r}_{(\mathcal{C}_B)}(t)$ . Following the steps outlined in Section 2.1.3, the values of  $\mathbf{r}_{(\mathcal{C}_A)}(t)$ ,  $\mathbf{r}_{(\mathcal{C}_B)}(t)$ ,  $\mathbf{u}_{(\mathcal{C}_A)}(t)$ , and  $\mathbf{u}_{(\mathcal{C}_B)}(t)$  for  $t \in [t_{\text{listen}}, t_{\text{train}}]$  are used to produce the corresponding training data matrices  $\mathbf{X}_{(\mathcal{C}_A)}$ ,  $\mathbf{X}_{(\mathcal{C}_B)}$ ,  $\mathbf{Y}_{(\mathcal{C}_A)}$ , and  $\mathbf{Y}_{(\mathcal{C}_B)}$  as per Equations 6, 7 in order to compute  $\mathbf{W}_{\text{out}}$  in Equation 5. This  $\mathbf{W}_{\text{out}}$  is then used to create the closed-loop RC in Equation 8.

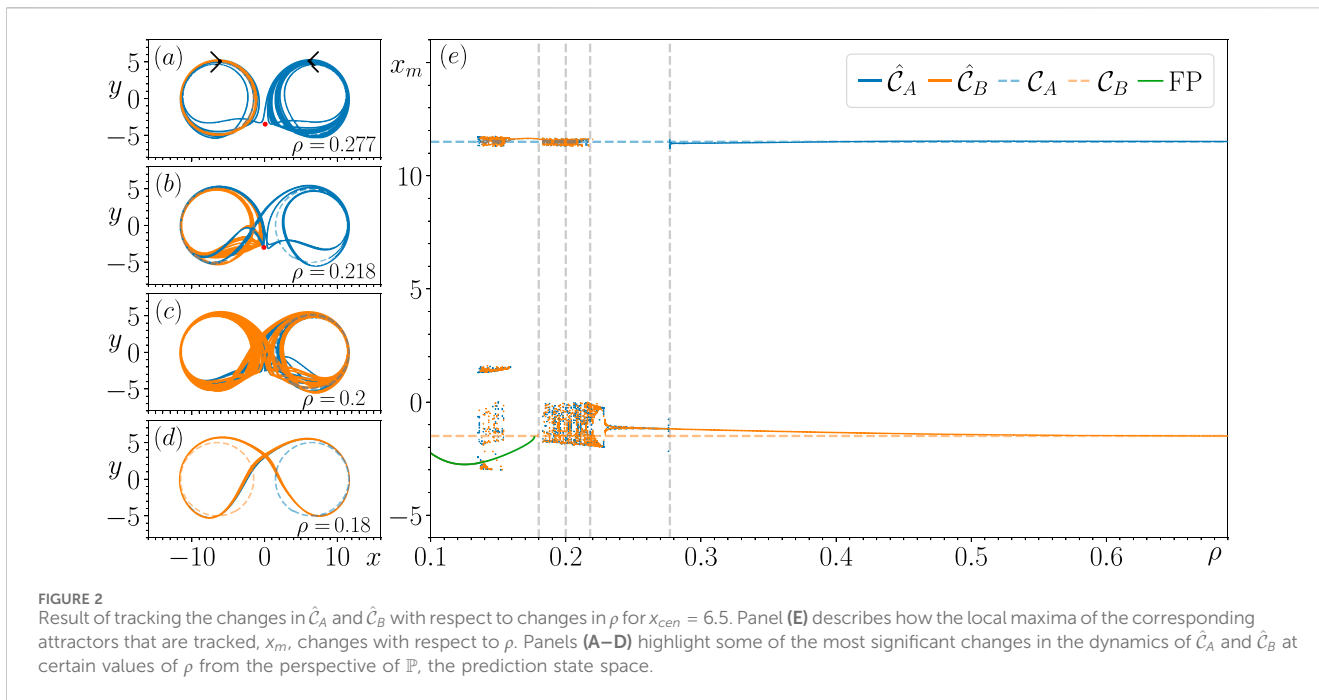
We say that this closed-loop RC achieves multifunctionality and solves the seeing double problem once it reconstructs a coexistence of  $\mathcal{C}_A$  and  $\mathcal{C}_B$ . To do this, the RC must construct a coexistence of two attractors,  $\mathcal{S}_A$  and  $\mathcal{S}_B$ , that exist in  $\mathbb{S}$  and resemble  $\mathcal{C}_A$  and  $\mathcal{C}_B$  when projected to  $\mathbb{P}$  using  $\hat{\psi}(\cdot)$  in Equation 3, with  $\mathbf{W}_{\text{out}}$  computed as mentioned above. As per the same convention used earlier, the projected dynamics of  $\mathcal{S}_A$  and  $\mathcal{S}_B$  are referred to as the reconstructed attractors and are denoted by  $\hat{\mathcal{C}}_A$  and  $\hat{\mathcal{C}}_B$ . To reconstruct the dynamics of  $\mathcal{C}_A$  or  $\mathcal{C}_B$  using this multifunctional RC, we initialize Equation 8 with  $\hat{\mathbf{r}}(0) = \mathbf{r}_{(\mathcal{C}_A)}(t_{\text{train}})$  or  $\hat{\mathbf{r}}(0) = \mathbf{r}_{(\mathcal{C}_B)}(t_{\text{train}})$  or some known point in the basin of attraction of  $\mathcal{S}_A$  or  $\mathcal{S}_B$ . The subsequent states of Equation 8 when approaching  $\mathcal{S}_A, \mathcal{S}_B \subset \mathbb{S}$  (i.e.,  $\hat{\mathcal{C}}_A, \hat{\mathcal{C}}_B \subset \mathbb{P}$ ) are written as  $\hat{\mathbf{r}}_{(\mathcal{C}_A)}(t)$  and  $\hat{\mathbf{r}}_{(\mathcal{C}_B)}(t)$ .

## 3 Results

### 3.1 Outline of experiments

The main aim of this paper is to improve our current understanding of how metastable switching dynamics emerge in an RC that fails to achieve multifunctionality. Figure 1 illustrates the





particular phenomenon we are interested in studying. In panel (a), we show that when  $C_A$  and  $C_B$  are sufficiently far apart (when  $x_{cen} = 8.0$ ), then for  $\rho = 0.2$ , the closed-loop RC achieves multifunctionality as  $\hat{C}_A$  and  $\hat{C}_B$  are more or less identical to  $C_A$  and  $C_B$ . However, panel (b) shows that when the same RC is trained with  $x_{cen} = 6.5$ , when  $C_A$  and  $C_B$  are slightly closer together but do not overlap, then the closed-loop RC fails to achieve multifunctionality, and instead, its state switches between regions of  $\mathbb{P}$  associated with  $C_A$  and  $C_B$ . To investigate the origins of these switching dynamics, we conduct the following experiments.

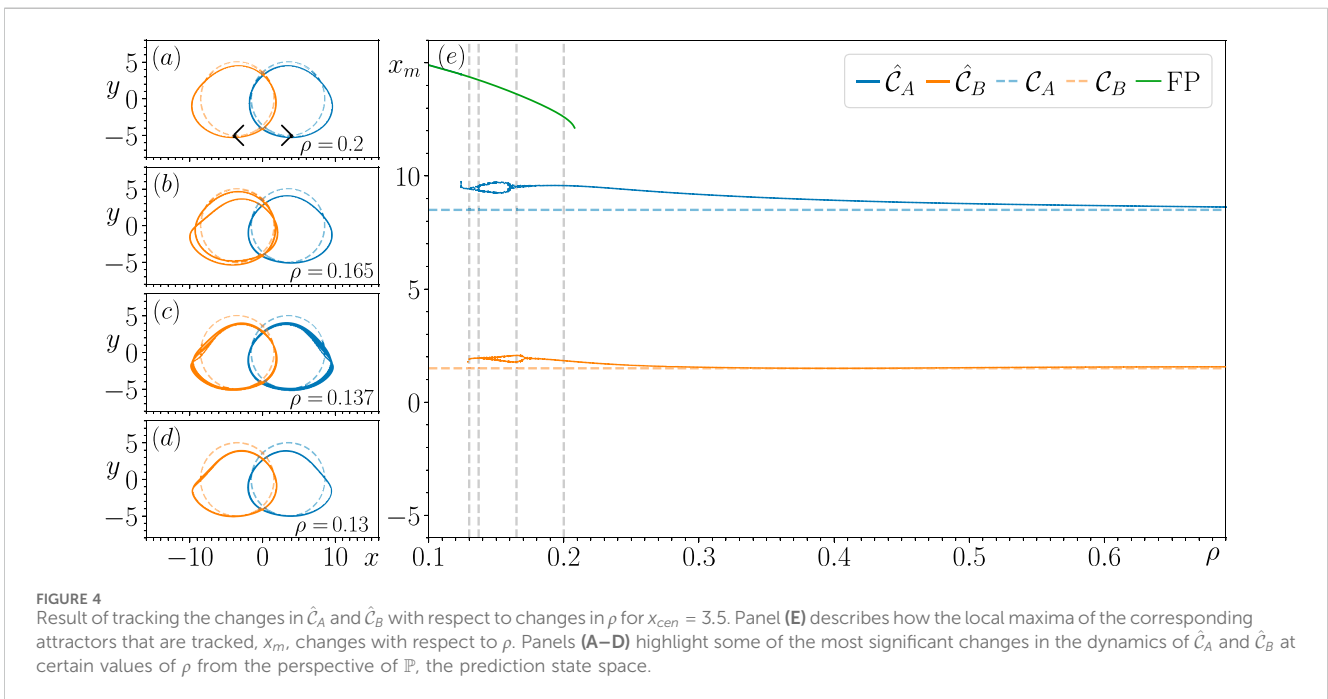
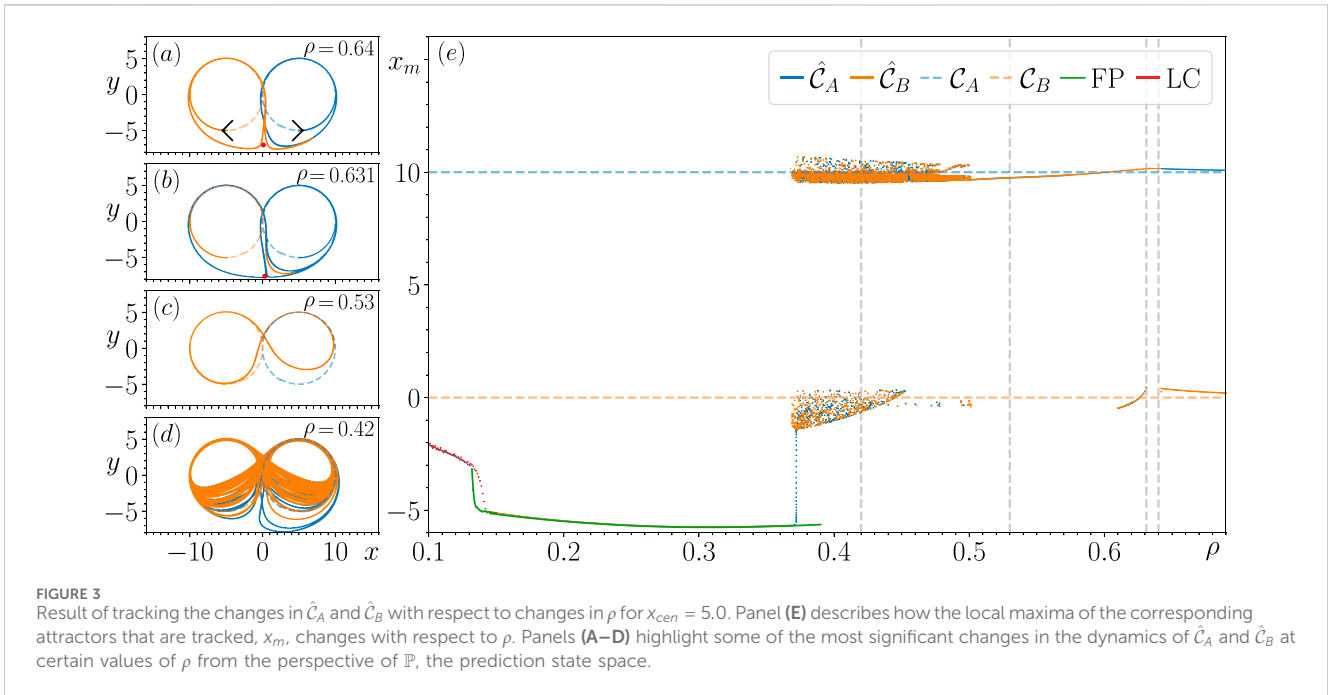
The results from the experiments reported in this section consist of training the RC in Equation 1 to solve the seeing double problem for  $x_{cen} = 6.5, 5.0, 3.5,$  and  $2.0$ . To illustrate the differences in the closed-loop RC's (Equation 8) dynamics when trained at these values of  $x_{cen}$ , we chose a common  $\rho$  value ( $\rho = 0.7$ ), where multifunctionality is achieved for each  $x_{cen}$ . We then decrease  $\rho$  in small steps of 0.001 and track the changes in the dynamics of the reconstructed attractors,  $\hat{C}_A$  and  $\hat{C}_B$ , by initializing the closed-loop RC with an initial condition corresponding to each attractor at the previous step and integrating the closed-loop RC forward in time up to  $t = 200$ . If  $\hat{C}_A$  or  $\hat{C}_B$  can no longer be tracked, i.e., have become unstable or cease to exist, we continue decreasing  $\rho$  and track the changes in the attractor that the state of the closed-loop RC subsequently approaches until  $\rho = 0.1$ . This method of attractor continuation enables us to investigate the origin of the switching dynamics we see in  $\mathbb{P}$  at certain values of  $\rho$  and  $x_{cen}$ .

The results of this continuation procedure at each of the specified  $x_{cen}$  values are shown in panel (e) of Figures 2–5, where we plot the local maxima of the reconstructed  $x$  variable, denoted by  $x_m$ . In panels (a)–(d) of Figures 2–5, we illustrate some of the most significant changes in the closed-loop RC's dynamics at particular  $\rho$  values, highlighting how the switching dynamics emerge. In Figures 2–5, the dashed blue and orange curves illustrate the location of  $C_A$

and  $C_B$ , the corresponding solid curves describe the closed-loop RC's reconstruction of  $C_A$  and  $C_B$  (denoted by  $\hat{C}_A$  and  $\hat{C}_B$ ), and the blue and orange points are the corresponding  $x_m$  values obtained from tracking the changes in the dynamics of  $\hat{C}_A$  and  $\hat{C}_B$  and the subsequent attractor that the closed-loop RC's state approaches when it fails to reconstruct  $\hat{C}_A$  or  $\hat{C}_B$ , respectively. In circumstances where the closed-loop RC fails to reconstruct  $C_A, C_B$ , or produce switching dynamics between regions of  $\mathbb{P}$  associated  $C_A$  and  $C_B$ , the “untrained attractor” (an attractor that the closed-loop RC produces that was not present during the training) that the state of the closed-loop RC subsequently approaches is depicted using the color specified in the associated plot legends. While there may be other untrained attractors present in  $\mathbb{P}$  for  $\rho < 0.1$ , in order to maintain the focus of this paper (which is to explore the origins of the switching dynamics), we only track the changes in the attractors that the state of the closed-loop RC approaches when it fails to reconstruct  $C_A, C_B$  or produce the switching dynamics. In addition, we also initialize the state of the closed-loop RC from many random initial conditions at several different  $\rho$  values when tracking the changes in  $\hat{C}_A$  and  $\hat{C}_B$ , but we do not find any untrained attractors.

In panel (e) of Figures 2–5, the vertical dashed gray lines indicate the  $\rho$  values that the corresponding dynamics in  $\mathbb{P}$  are illustrated in the accompanying panels (a)–(d). The black arrows in panel (a) of Figures 2–5 indicate  $C_A$  and  $C_B$ 's direction of rotation. These illustrations are generated by training the RC in Equation 1 at the specified  $\rho$  values and initializing the closed-loop RC with  $r_{(C_A)}(t_{train})$  and  $r_{(C_B)}(t_{train})$  (the last point in the training data corresponding to  $C_A$  and  $C_B$ ), i.e., following the description in Section 2.1.3. We do this in order to observe whether there exists any transient behavior associated with  $C_A$  or  $C_B$  when the RC fails to reconstruct these attractors.

Furthermore, we also conducted this analysis for the case of  $x_{cen} = 8.0$ , but we choose not to show these results as we found no

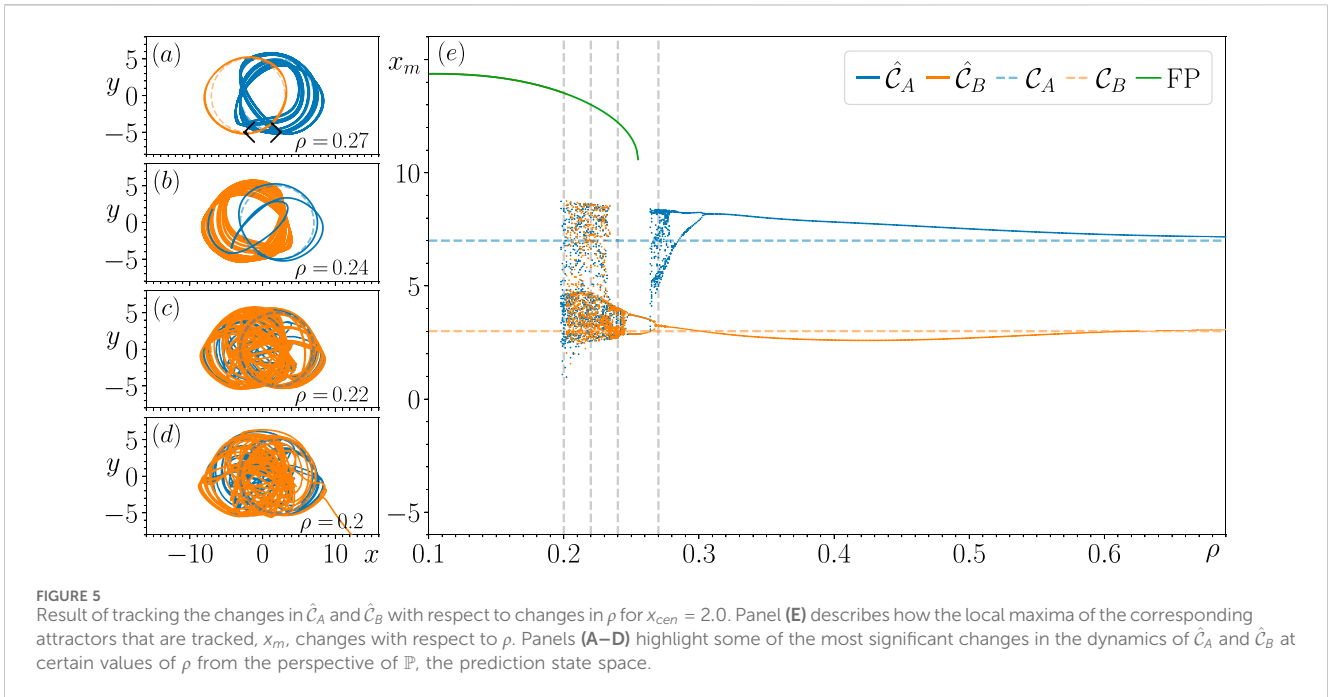


switching dynamics nor significant changes in the closed-loop RC's dynamics for changes in  $\rho$ . For  $x_{cen} = 8.0$ , the closed-loop RC achieves multifunctionality with a nearly perfect reconstruction of  $C_A$  and  $C_B$  for the range of  $\rho$  values that were investigated.

The results of the continuation analysis for each of the selected  $x_{cen}$  values are outlined in Secs. 3.2–3.5. In Section 3.6, we examine the residence and escape times associated with the switching and transient dynamics observed in Secs. 3.2–3.5.

### 3.2 Continuation analysis for $x_{cen} = 6.5$

Figure 2E shows that for  $x_{cen} = 6.5$ , there are no significant changes in the closed-loop RC's ability to reconstruct a coexistence of  $C_A$  and  $C_B$  until  $\rho = 0.277$ , where  $\hat{C}_A$  becomes unstable. In Figure 2A, we take a closer look at this behavior, and we see that the state of the closed-loop RC follows a significantly long chaotic transient when initialized with the associated  $r_{(C_A)}(t_{train})$  before



eventually approaching  $\hat{C}_B$ , which is the only stable attractor present in  $\mathbb{P}$  (confirmed by initializing the closed-loop RC from many random initial conditions). Based on the structure of this transient and the evidence of a saddle present at  $(x, y) \approx (0.1, -3.5)$  (indicated by the red point in Figure 2A), there is evidence that  $\hat{C}_A$  becomes unstable by colliding with this saddle.

There is stronger evidence to support the above claim in Figure 2B as we see for  $\rho = 0.218$ , the structure of the transient is highly influenced by this saddle (now located at  $(x, y) \approx (0, -3)$  in  $\mathbb{P}$ , as indicated by the red point in Figure 2B), whose unstable directions appear to point predominately along the x-axis and stable directions point along the y-axis in  $\mathbb{P}$ . While we see here that the state of the closed-loop RC takes significantly less time to escape this transient, what is particularly interesting about this transient is that the state of the closed-loop RC follows a path that encircles  $\hat{C}_B$  and switches back to the portion of  $\mathbb{P}$  associated with  $C_A$  before switching back to and remaining on  $\hat{C}_B$  for all future time.  $\hat{C}_B$  is the only stable attractor present in  $\mathbb{P}$  for  $\rho \in [0.218, 0.277]$ , as further indicated by the dashed blue horizontal line associated with the reconstruction of  $C_A$  that is visible within this range of  $\rho$  values. It is also during this range of  $\rho$  values that  $\hat{C}_B$  becomes chaotic through what appears to be a period-doubling bifurcation, which is quickly followed by a torus bifurcation. Furthermore, it appears that the closed-loop RC's trajectory on  $\hat{C}_B$  for  $\rho = 0.218$  comes arbitrarily close to the saddle, indicating that a similar fate to  $\hat{C}_A$  awaits  $\hat{C}_B$  at smaller  $\rho$  values.

Figure 2E illustrates that when tracking the changes in  $\hat{C}_B$  for decreasing  $\rho$  further, there are multiple values of  $x_m$  obtained that surround both blue and orange dashed horizontal lines. This indicates the emergence of the switching dynamics between regions of  $\mathbb{P}$  where the previously stable  $\hat{C}_A$  and  $\hat{C}_B$  existed. These switching dynamics are found to occur for  $\rho \in [0.135, 0.217]$ , and a variety of different switching patterns are exhibited. For instance, in Figure 2C, we show that these

switching dynamics resemble a Lorenz-like chaotic attractor for  $\rho = 0.2$ , whereas in Figure 2D, a periodic switching pattern appears in the form of a limit cycle, which emerges from the chaotic attractor. Section 3.6 describes the long-term dynamics of the chaotic attractor shown in Figure 2C. Figure 2E shows that this periodic switching pattern returns to an aperiodic switching pattern at  $\rho = 0.155$ , indicated by the three clusters of  $x_m$  values.

The switching dynamics come to an end at  $\rho = 0.135$ , and the state of the closed-loop RC subsequently approaches a stable fixed point (FP), indicated by the sequence of green points, and we continue to track the changes in this FP until  $\rho = 0.1$ . We find a small range of  $\rho$  values where the FP coexists with the switching patterns by tracking the changes in this FP for increasing  $\rho$  values until it becomes unstable at  $\rho = 0.178$ . At this point, the state of the closed-loop RC returns to the limit cycle associated with the periodic switching pattern mentioned in the paragraph above.

### 3.3 Continuation analysis for $x_{cen} = 5.0$

For  $x_{cen} = 5.0$ , we find that by moving  $C_A$  and  $C_B$  closer together so that they touch at  $(x, y) = (0, 0)$ , the switching dynamics begin at much larger  $\rho$  values, persist for a greater range of  $\rho$  values, and the switching patterns are first found to occur periodically before becoming chaotic. Figure 3E shows that as  $\rho$  is decreased,  $\hat{C}_B$  becomes unstable at  $\rho = 0.64$ . Figure 3A illustrates the transient dynamics of the closed-loop RC when initialized with the associated  $r_{(C_B)}(t_{train})$ . Here, we see that the state of the closed-loop RC follows one loop around the dashed orange circle ( $C_B$ ) before diverging away to approach and then remain on the slightly oval-shaped  $\hat{C}_A$ . It is from these unstable directions of flow along the x-axis and stable flow along the y-axis that the nature of this transient also provides us with some evidence that there is a saddle located at  $C_B$  ( $(x, y) \approx (0.1, -7)$ ), as indicated by the red point in Figure 3A.

Figure 3B provides us with further information about this saddle (now located at  $(x, y) \approx (0.3, -7.6)$ ), as indicated by the red point in Figure 3B) as it appears that  $\hat{C}_A$  has become unstable at  $\rho = 0.631$  by colliding with the saddle. Moreover, it is through this second collision that a new limit cycle is created that produces the switching dynamics nearby the point at which  $C_A$  and  $C_B$  touch in  $\mathbb{P}$ . This limit cycle consists of two weakly attracting connected regions of flow around  $C_A$  and  $C_B$ . Taking a closer look at the transient dynamics exhibited by the closed-loop RC when initialized with  $\mathbf{r}_{(C_A)}(t_{\text{train}})$ , we see that its state comes arbitrarily close to the saddle before completing one loop around the dashed blue circle associated with  $C_A$ ; however, on the second loop, the trajectory diverges away from  $C_A$  nearby the saddle and then approaches and subsequently remains on the new larger limit cycle that consists of loops around regions of  $\mathbb{P}$  associated with  $C_A$  and  $C_B$ . Initially, there are two values obtained for  $x_m$ , the local maxima associated with  $C_A$  and a point nearby the saddle and the small branch of points nearby the dashed orange horizontal line seen in Figure 3E. Correspondingly, the sharp turning point on the new limit cycle nearby the saddle point shown in Figure 3B does not persist for many subsequent  $\rho$  values as the limit cycle starts to resemble a figure of 8 in  $\mathbb{P}$ , like the example shown in Figure 3C for  $\rho = 0.53$ .

As shown in Figure 3E, additional values of  $x_m$  are found for  $\rho = 0.5$  as the limit cycle transitions to a chaotic attractor. In Figure 3D, we provide an example of the aperiodic switching patterns exhibited by this chaotic attractor for  $\rho = 0.42$ . Section 3.6 describes the long-term dynamics of this chaotic attractor. We are unable to track the changes in this chaotic attractor for  $\rho < 0.37$ , and the state of the closed-loop RC subsequently approaches an FP, whose behavior with respect to changes in  $\rho$  is described in the green branch of points in Figure 3E. There is a relatively small range of  $\rho$  values where this FP coexists with the chaotic attractor associated with the aperiodic switching patterns for  $\rho \in [0.37, 0.39]$ . When tracking the changes in this FP for decreasing  $\rho$ , we find a smaller range of  $\rho$  values where this FP coexists with a different period-1 limit cycle, whose corresponding  $x_m$  is described by the branch of red points in Figure 3E.

### 3.4 Continuation analysis for $x_{cen} = 3.5$

When decreasing  $x_{cen}$  to 3.5, we find that by moving  $C_A$  and  $C_B$  closer together so that they overlap and share two common points, this improves the performance of the RC as it achieves multifunctionality for a much larger range of  $\rho$  values and does not produce any switching dynamics. Figure 4E shows that by tracking the changes in  $\hat{C}_A$  and  $\hat{C}_B$  for decreasing  $\rho$ , there is a growing difference between the obtained values for  $x_m$  and the corresponding true values with respect to  $C_A$  and  $C_B$ . Figure 4A provides further insights into how  $\hat{C}_A$  and  $\hat{C}_B$  deform as  $\rho$  decreases. Figure 4B shows how  $\hat{C}_A$  and  $\hat{C}_B$  increasingly lose their resemblance to  $C_A$  and  $C_B$ , with  $\hat{C}_B$  having undergone a period-doubling bifurcation as  $\rho$  is decreased to  $\rho = 0.165$ . Figure 4C illustrates that for  $\rho = 0.137$ , both  $\hat{C}_A$  and  $\hat{C}_B$  display aperiodic dynamics (indicated by the increased thickness of the corresponding blue and orange curves). Figure 4E shows that as  $\rho$  is decreased further,  $\hat{C}_B$  becomes unstable at  $\rho \approx 0.129$  and the state of the closed-loop RC subsequently approaches  $\hat{C}_A$ , and at  $\rho \approx 0.124$ , we find that  $\hat{C}_A$

becomes unstable and the state of the closed-loop RC subsequently approaches the FP described by the branch of green points. By tracking the changes in this FP for increasing  $\rho$ , we find that this FP coexists with  $\hat{C}_A$  and  $\hat{C}_B$  until it becomes unstable at  $\rho \approx 0.21$  and the state of the closed-loop RC returns to  $\hat{C}_A$ . Figure 4D illustrates that prior to  $\hat{C}_A$  and  $\hat{C}_B$  becoming unstable, these attractors are no longer chaotic and have returned to period-1 limit cycles.

### 3.5 Continuation analysis for $x_{cen} = 2.0$

When decreasing  $x_{cen}$  to 2.0, we find that by increasing the amount of overlap between  $C_A$  and  $C_B$ , the closed-loop RC produces switching dynamics within a relatively small range of  $\rho$  values in a similar fashion to those found for  $x_{cen} = 6.5$  and 5.0. Figure 5E shows that as  $\rho$  decreases, there is an increasingly large offset between the values of  $x_m$  for  $\hat{C}_A$  and  $C_A$  and a small but noticeable difference between the values of  $x_m$  for  $\hat{C}_B$  and  $C_B$ .

Figure 5E shows that as  $\rho$  is decreased from  $\rho \approx 0.3$ ,  $\hat{C}_A$  undergoes a sequence of period-doubling bifurcations, which results in  $\hat{C}_A$  transitions to a chaotic attractor. In Figure 5A, we illustrate the dynamics of the closed-loop RC for  $\rho = 0.27$ , which shows the coexistence of the chaotic  $\hat{C}_A$  and periodic  $\hat{C}_B$  which closely resemble the dynamics of  $C_B$ . Figure 5E shows that by decreasing  $\rho$  further,  $\hat{C}_B$  undergoes a period-doubling bifurcation starting from  $\rho \approx 0.269$ .  $\hat{C}_A$  becomes unstable at  $\rho \approx 0.261$ , and after a bout of transient dynamics, the state of the closed-loop RC subsequently approaches the period-2  $\hat{C}_B$ . We then continue to track the changes in  $\hat{C}_B$ , which also becomes chaotic at  $\rho \approx 0.245$ . In Figure 5B, we illustrate the chaotic dynamics of  $\hat{C}_B$  and the relatively short duration of transient dynamics exhibited by the closed-loop RC when initialized from  $\mathbf{r}_{(C_A)}(t_{\text{train}})$ . This transient completes one loop around the region of  $\mathbb{P}$  associated with  $C_A$ ; however, on its second loop, the state of the closed-loop RC approaches the point  $(x, y) \approx (-4.5, -4.5)$ , where it subsequently reverses along its trajectory to this point and then approaches the chaotic  $\hat{C}_B$ , remaining on  $\hat{C}_B$  thereafter.

The densely populated range of  $x_m$  values, which spans across both dashed horizontal lines associated with  $C_A$  and  $C_B$  in Figure 5E, shows that the switching dynamics emerge at  $\rho \approx 0.23$ . For  $\rho = 0.22$ , in Figure 5C, we illustrate the dynamics of the large chaotic attractor that is born at  $\rho \approx 0.23$ , and trajectories on this attractor resemble aperiodic switching dynamics between regions of  $\mathbb{P}$  associated with  $C_A$  and  $C_B$ . For  $\rho = 0.2$ , Figure 5D illustrates that when the closed-loop RC is initialized with  $\mathbf{r}_{(C_B)}(t_{\text{train}})$ , its state follows a chaotic transient before approaching an FP located at  $(x, y) \approx (13.5, -9.5)$ , which is just outside the portion of  $\mathbb{P}$  shown here.

### 3.6 Closer inspection of switching dynamics at $x_{cen} = 6.5$ and 5.0

In this section, we aim to shed further light on the nature of the switching dynamics discussed so far. We consider the two examples of  $x_{cen} = 6.5$  and  $\rho = 0.2$ , which we refer to as case 1, and  $x_{cen} = 5.0$  and  $\rho = 0.42$ , which we refer to as case 2. We generate a much longer trajectory on these chaotic attractors in order to determine the distribution of residence times that the state of the closed-loop RC



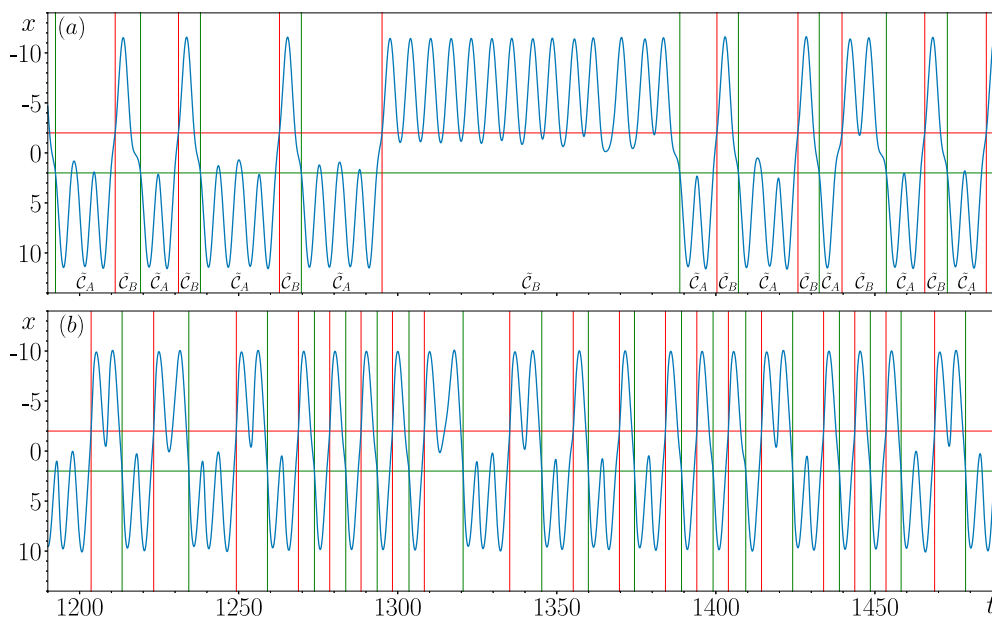


FIGURE 6 Obtaining the transition times between  $\tilde{C}_A$  and  $\tilde{C}_B$  for case 1 (A) and case 2 (B).

spends in the respective  $C_A$  and  $C_B$  regions of  $\mathbb{P}$ . When the state of the closed-loop RC is in the region of  $\mathbb{P}$  associated with  $C_A$ , we consider the system to be in a metastable state denoted as  $\tilde{C}_A$ , and similarly, for  $C_B$ , we consider to system to be in a different metastable state denoted as  $\tilde{C}_B$ .

### 3.6.1 Algorithm to detect transitions

In order to identify when the state of the closed-loop RC is in  $\tilde{C}_A$  or  $\tilde{C}_B$ , we construct a relatively simple algorithm based on the concept of a “non-ideal relay” (Krasnosel’skii and Pokrovskii, 2012). We use this algorithm to detect transition times from  $\tilde{C}_A$  to  $\tilde{C}_B$  and *vice versa*. The non-ideal relay aspect of the algorithm involves choosing two threshold values,  $\alpha$  and  $\beta$ , where we say that the closed-loop RC is in  $\tilde{C}_A$  once its state crosses  $\beta$  and remains below  $\alpha$ , and it is in  $\tilde{C}_B$  once its state crosses  $\alpha$  and remains above  $\beta$ . The benefit of using these two thresholds as opposed to one threshold is that it allows us to improve our estimate of when the system is in a particular state by reducing the effect of false alarm scenarios where, for instance, the state of the closed-loop RC is in  $\tilde{C}_A$  but briefly dips below the single threshold and does not spend any significant amount of time in the portion of  $\mathbb{P}$  associated with  $C_B$ .

The result of using this algorithm to detect transitions from  $\tilde{C}_A$  to  $\tilde{C}_B$  and *vice versa* in case 1 is illustrated in Figure 6A and that for case 2 is illustrated in Figure 6B, where we set  $\alpha = -2$  and  $\beta = 2$ . The green and red horizontal lines in Figure 6 are used to illustrate these threshold values. The green and red vertical lines shown here correspond to the detected transition times where the state of the closed-loop RC first enters  $\tilde{C}_A$  and  $\tilde{C}_B$ , respectively. The residence times in  $\tilde{C}_A$  and  $\tilde{C}_B$  are then calculated based on these transition times.

The benefit of using this double threshold algorithm is made clear in Figure 6A; if a single threshold of 0 was used

instead, then when the state of the RC crosses 0 without switching from one metastable state to the other, like at  $t \approx 1200, 1230, 1245, 1315, 1415, 1465, 1485$ , then all of these crossings would be considered transitions, which is evidently false.

Furthermore, what is also evident from Figure 6 is that there are at least three distinct types of switching patterns present where the state of the closed-loop RC can rapidly switch between  $\tilde{C}_A$  and  $\tilde{C}_B$  or spend a particular amount of time in  $\tilde{C}_A$  and  $\tilde{C}_B$  before switching.

### 3.6.2 Residence times

In order to construct a reasonably well-distributed sample of residence times in  $\tilde{C}_A$  and  $\tilde{C}_B$ , we generate 10,000 examples of switchings between  $\tilde{C}_A$  and  $\tilde{C}_B$ . To do this, we integrate the closed-loop RC forward in time up to  $t \approx 300,000$  for case 1 and up to  $t \approx 157,000$  for case 2. This tells us there are nearly twice as many switchings in a given duration of time for case 2 in comparison to case 1. From this sample of 10,000 switchings, we found that for case 1, the maximum and minimum residence times (in the arbitrary units of  $t$ ) in  $\tilde{C}_A$  were  $\approx 65$  and  $6.5$ , respectively, and for  $\tilde{C}_B$ , they were  $\approx 270$  and  $6.7$ , respectively. For case 2, the maximum and minimum residence times in  $\tilde{C}_A$  were  $\approx 23.5$  and  $4.8$ , respectively, and for  $\tilde{C}_B$ , they were  $\approx 20.3$  and  $4.5$ , respectively. We then compute the probability density of these residence times by generating a histogram of residence times with 100 bins chosen from numbers spaced evenly on a log scale with limits set to the max and min values specified above. The resulting probability density of these residence times for case 1 is shown in Figure 7A and for case 2 is shown in Figure 7B.

What is most striking about the results shown in Figure 7 is that there is no single branch of exponentially distributed points; instead, for both cases 1 and 2, the probability density of the residence times in  $\tilde{C}_A$  and  $\tilde{C}_B$  are organized into a number of branches of exponentially distributed points.

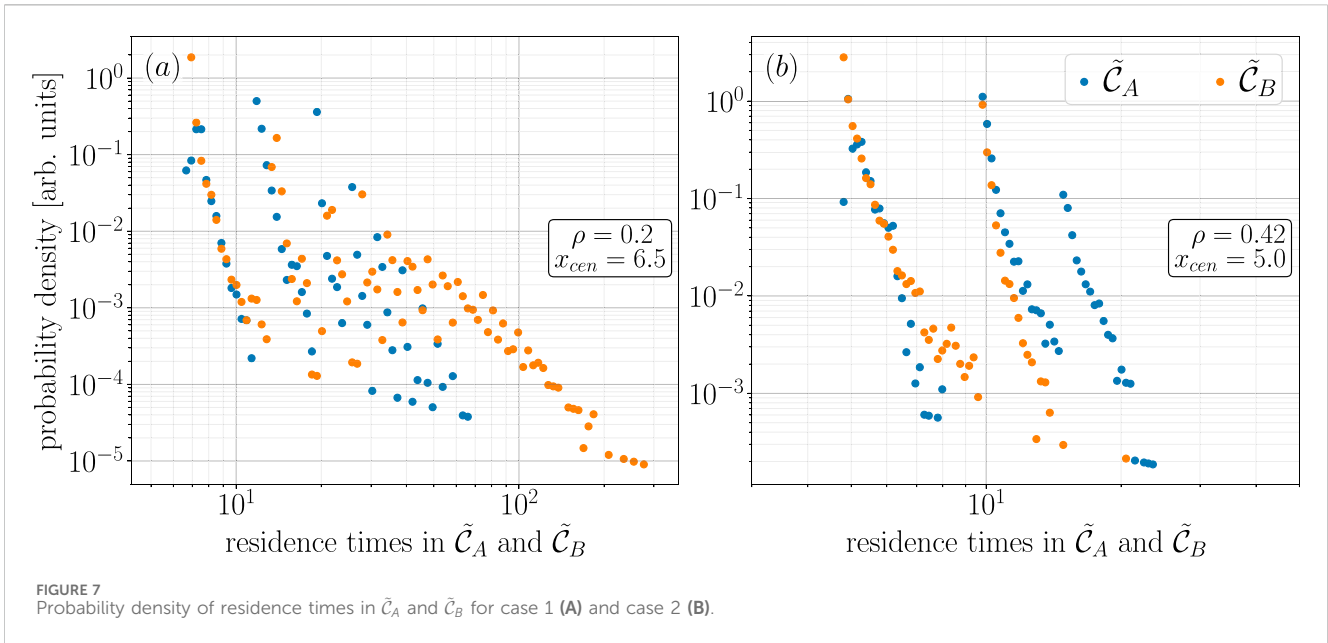


FIGURE 7 Probability density of residence times in  $\tilde{C}_A$  and  $\tilde{C}_B$  for case 1 (A) and case 2 (B).

We first outline the results shown in Figure 7B for case 2 as it is more straightforward to discuss. The probability density of residence times in  $\tilde{C}_A$  is organized into three branches of exponentially distributed points and two branches of exponentially distributed points for  $\tilde{C}_B$ . From further investigation, we find that the points on these different branches correspond to scenarios where the state of the closed-loop RC follows either one or two loops (or partial loops) around  $C_B$  before switching to  $C_A$  and can follow up to three loops (or partial loops) about  $C_A$  before switching to  $C_B$ . By partial loops, we mean that the state of the closed-loop RC may switch from  $\tilde{C}_A$  to  $\tilde{C}_B$  without completing a full loop around  $C_A$ . Furthermore, from the dynamics of the chaotic attractor that produces these switching dynamics illustrated in Figure 3D, it is reasonable to have anticipated the exponential distribution of points on these branches shown in Figure 7B. It is also reasonable to have anticipated that the state of the closed-loop RC spends slightly longer amounts of time in  $\tilde{C}_A$  than  $\tilde{C}_B$  since  $\hat{C}_B$  becomes unstable before  $\hat{C}_A$  and is, therefore, relatively less attracting when the switching dynamics begin.

Figure 7A illustrates that for case 1, there are a number of less strongly defined branches of exponentially distributed points. The two most well-defined branches on the left hand side of this figure correspond to scenarios where the state of the closed-loop RC completes one or two loops (or partial loops) around  $C_A$ ,  $C_B$  or rapidly switches between  $\tilde{C}_A$  and  $\tilde{C}_B$ . The well-defined branch of orange points on the right-hand side of this figure corresponds to the significantly longer amounts of time that the state of the closed-loop RC spends in  $\tilde{C}_B$ , like in the example shown in Figure 6A, where the state of the closed-loop RC is in  $\tilde{C}_B$  from  $t \approx 1,295$  to  $1,390$ . From further analysis, we find that by increasing the number of bins, the cloud of points in the middle Figure 6A corresponds to scenarios where the state of the closed-loop RC completes several loops (or partial loops) around  $C_A$  and  $C_B$ . However, by increasing the number of bins, we also find that this results in an increasingly large accumulation of points at the bottom of these branches, which, in our opinion, diminishes the clarity of the message behind this

figure, and for that reason, we do not present a version of Figure 6A with a larger number of bins.

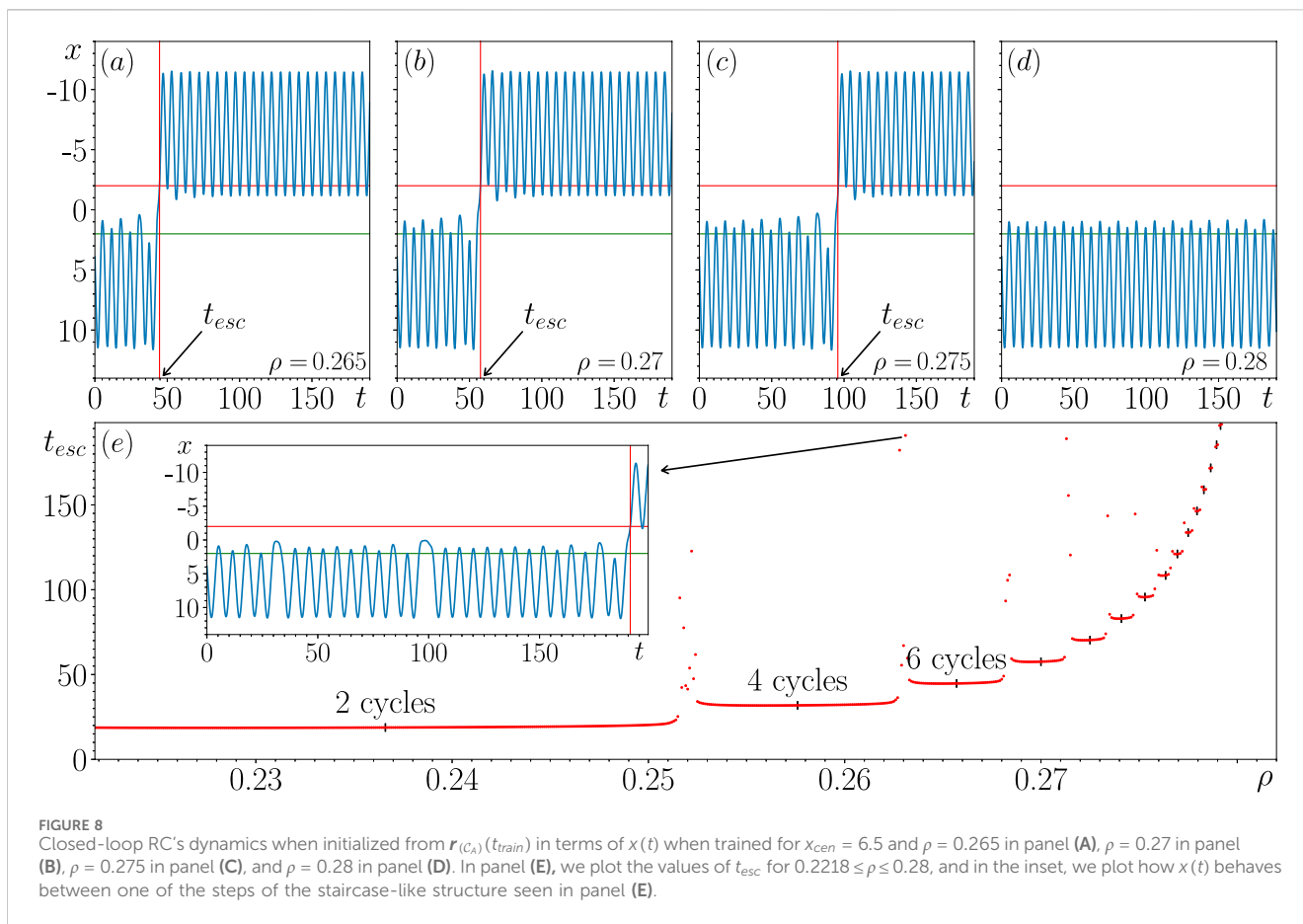
### 3.6.3 Escape times

The purpose of this section is to provide further insights into the interesting transient dynamics associated with  $\hat{C}_A$  becoming unstable when  $x_{cen} = 6.5$ , as discussed in Section 3.2. Using the transition detection algorithm, we calculate the time it takes for the closed-loop RC to escape from transient behavior when initialized from  $r_{(C_A)}(t_{train})$ , and we denote this duration of time as  $t_{esc}$ . We investigate the relationship between  $\rho$  and  $t_{esc}$  for values of  $\rho$  when no switching dynamics occur for  $0.2218 \leq \rho \leq 0.28$ . In panels (a)–(d) of Figure 8, we plot the time series of the reconstructed  $x(t)$  variable at particular  $\rho$  values. We use the same  $\alpha$  and  $\beta$  thresholds as in the previous section, indicated by the red and green horizontal lines, respectively. The vertical red line depicts the detected value of  $t_{esc}$ .

Figures 8A–D indicate that as  $\rho$  increases and approaches the point at which  $\hat{C}_A$  becomes stable,  $t_{esc}$  naturally increases. However, panel (e) shows that while  $t_{esc}$  increases as  $\rho$  increases,  $t_{esc}$  increases in a non-trivial staircase-like manner where the length of each successive step decreases as  $\rho$  increases.

For instance,  $t_{esc}$  is shown here to be relatively small and increasing at a relatively slow rate for  $\rho < 0.251$ ; however, by increasing  $\rho$  to 0.253, this results in a nearly two-fold increase in  $t_{esc}$ , but for  $0.251 < \rho < 0.253$ , we find relatively large values and large variations in the values of  $t_{esc}$  where  $\hat{C}_A$  appears to almost regain stability. The inset plot shows one of these relatively long transients between the steps at  $\rho = 0.2631$ . Here, we see from the change in time where local minima occur that the state of the closed-loop RC almost escapes from this transient activity at  $t \approx 32$ , as it does so for smaller  $\rho$  values, and again at  $t \approx 100$ . This change in time is indicative of the state of the closed-loop RC approaching the saddle point on the unstable  $\hat{C}_A$  but fails to cross its separatrix. It is only for  $t \approx 191$  that the RC escapes from the transient.

As indicated in panel (e), at each successive step along this staircase, the state of the closed-loop RC completes two additional



cycles about the unstable  $\hat{C}_A$  before escaping. While the calculated values of  $t_{esc}$  depend on the choice of the initial condition relative to the point of escape on the unstable  $\hat{C}_A$ , this behavior of completing two additional cycles at each successive step may be more strongly dependent on the nature of  $\hat{C}_A$  prior to becoming unstable as it exists as a period-2 limit-cycle (indicated by the two global maxima Figure 8D, also seen in Figure 2E albeit barely visible). Our results suggest that by increasing  $\rho$ , the saddle point on the unstable  $\hat{C}_A$  moves in a way that the state of the closed-loop RC needs to complete an additional round trip about the unstable period-2 nature of  $\hat{C}_A$  until it reaches the point of escape.

### 4 Discussion

In this paper, we explore how switching dynamics emerge in a dynamical system in the form of an RC when trained to achieve multifunctionality by solving the seeing double problem. This problem involves training the open-loop RC in Equation (1) to reconstruct a coexistence of two circular orbits  $C_A$  and  $C_B$ . We find that as  $C_A$  and  $C_B$  are moved closer together, the state of the closed-loop RC (Equation 8) begins to switch between what appears to be metastable states that resemble trajectories around regions of  $\mathbb{P}$  associated with  $C_A$  and  $C_B$ . To be more specific, we find that these switching dynamics occur just before  $C_A$  and  $C_B$  touch (as shown in Figure 2 for  $x_{cen} = 6.5$ ), as they touch (as shown in Figure 3 for

$x_{cen} = 5.0$ ), and after they touch (as shown in Figure 5 for  $x_{cen} = 2.0$ ), whereby there is an overlap between  $C_A$  and  $C_B$ . However, as shown in Figure 4, there is an intermediary regime whereby after  $C_A$  and  $C_B$  touch and begin to overlap (for  $x_{cen} = 3.5$ ), the RC recovers its ability to achieve multifunctionality and does not succumb to these switching dynamics. It is only after there is a sufficiently large amount of overlap between  $C_A$  and  $C_B$  (for  $x_{cen} = 2.0$ ) that the switching dynamics reappear.

Our results also shed further light on the key role played by  $\rho$  in this RC design and its connection to the concept of memory in terms of how the larger the value of  $\rho$ , the greater the influence of previous states on the current state of the RC. What our results indicate is that if the orbits are close to touching each other, like for  $x_{cen} = 6.5$ , or touch each other at only one point when  $x_{cen} = 5.0$ , this requires the RC to place a greater weight on previous states (i.e., large  $\rho$ ) in order to achieve multifunctionality as the dynamics nearby these touching regions are quite similar. On the other hand, if the orbits overlap and touch each other in two locations that are sufficiently far but not too far apart, like for  $x_{cen} = 3.5$ , then the RC does not need to place such a large weight on previous states in order to achieve multifunctionality. However, once there is a larger amount of overlap between the orbits, like for  $x_{cen} = 2.0$ , then the RC needs to place greater weight on previous states in order to achieve multifunctionality once again.

It is also worth noting that in panel (e) of Figures 2–5, prior to  $\hat{C}_A$  or  $\hat{C}_B$  becoming unstable as  $\rho$  decreases, there is a noticeable

difference in the obtained values for  $x_m$  and the corresponding true values. This is most evident in panels (a)–(d) of Figure 4, where we see  $\hat{C}_A$  and  $\hat{C}_B$  stretched toward larger positive and negative values of  $x$ , respectively. As  $x_{cen}$  is decreased further, this effect appears to become increasingly noticeable. A similar sequence of events was shown to occur in Figures 14, 15, and 21 in Flynn et al. (2023), where, for  $x_{cen} = 0$ , as  $\rho$  decreases,  $\hat{C}_A$  and  $\hat{C}_B$  are deformed in a similar way. This particular deformation may occur due to the design of  $\mathbf{W}_{in}$ , as each neuron receives input from only one component of the driving input signal because each row contains only one nonzero element; therefore, as  $\rho$  decreases, this increases the influence of the input, and this may increase the likelihood that the resulting dynamics of the closed-loop RC are stretched along the  $y = x$  and  $y = -x$  diagonals. However, in order to provide a more rigorous answer, this requires conducting an extensive analysis across several random realizations of  $\mathbf{M}$  and  $\mathbf{W}_{in}$  and testing whether such a deformation effect persists when using different design principles to construct  $\mathbf{M}$  and  $\mathbf{W}_{in}$ . We believe that such an investigation is highly worthwhile to conduct and is better suited to appear in a paper where this is the main focus.

From closer inspection of the transitions between these metastable states, which we refer to as  $\tilde{C}_A$  and  $\tilde{C}_B$ , we find that there is a common sequence of events that occurs in each case in order to produce the switchings between  $\tilde{C}_A$  and  $\tilde{C}_B$ . Starting from a set of training parameters where the closed-loop RC achieves multifunctionality, we track how the dynamics of  $\tilde{C}_A$  and  $\tilde{C}_B$  change with respect to changes in  $\rho$ , the spectral radius of the RC's internal connectivity matrix. We find that by decreasing  $\rho$  from the point where  $\tilde{C}_A$  and  $\tilde{C}_B$  coexist and resemble  $C_A$  and  $C_B$ , there is a value of  $\rho$ , where, for instance,  $\tilde{C}_A$  collides with a nearby saddle and becomes unstable, but there still exists some transient dynamics that the state of the closed-loop follows when initialized from a point on the previously stable  $\tilde{C}_A$ . Then, by further decreasing  $\rho$ , we find that there is a value of  $\rho$  where  $\tilde{C}_B$  also becomes unstable by colliding with a nearby saddle. However, when  $\tilde{C}_B$  becomes unstable, there is a new attractor born that facilitates the switching dynamics between the metastable states,  $\tilde{C}_A$  and  $\tilde{C}_B$ , mentioned earlier. To be more specific, a trajectory on this new attractor consists of two regions of convergent flow where the trajectory inside these regions resembles a trajectory around  $C_A$  and  $C_B$  and a divergent flow whereby the state of the closed-loop RC switches from one region of convergent flow to the other.

We also investigate the long-term behavior of some of these new attractors that are born during the sequence of events discussed above. We integrated the closed-loop RC forward in time until we obtained 10,000 transitions between  $\tilde{C}_A$  and  $\tilde{C}_B$  for the chaotic attractors illustrated in Figure 2C, denoted as case 1, and in Figure 3D, denoted as case 2. We construct an algorithm based on the concept of a non-ideal relay to determine the time of transition between  $\tilde{C}_A$  and  $\tilde{C}_B$ . In Figure 6, we provide an example of the transition times detected by this algorithm. Interestingly, by computing the probability density of residence times in  $\tilde{C}_A$  and  $\tilde{C}_B$ , we obtain several branches of exponentially distributed points, as shown in Figure 7. From closer inspection, we find that each of these branches correspond to scenarios where the state of the closed-loop RC completes a given number of loops or partial loops around  $C_A$  and  $C_B$ .

We remark that while these switching dynamics are found for a particular random realization of  $\mathbf{M}$  and  $\mathbf{W}_{in}$  (the internal and input connectivity matrices), the results presented in this paper are not solely dependent on these particular weights as we see similar behavior emerging from further experiments not shown here. Furthermore, there is a noticeable imbalance in the behavior of  $\tilde{C}_A$  and  $\tilde{C}_B$  despite the symmetry present in the training data. We believe that this is due to the particular random realization of the  $\mathbf{M}$  and  $\mathbf{W}_{in}$  matrices happening to favor the reconstruction of one orbit over the other at particular parameter settings. From further analysis (also not shown in this paper), we find some small differences in the values of  $\rho$  and the order of when  $\tilde{C}_A$  and  $\tilde{C}_B$  become unstable for different realizations of  $\mathbf{M}$  and  $\mathbf{W}_{in}$ . As a further point, while the switching dynamics are induced by moving  $C_A$  and  $C_B$  closer together, it is still possible for switching dynamics to emerge between a reconstructed attractor and untrained attractors (attractors that the closed-loop RC produces that was not present during the training), or between the attractor, an RC with symmetry trained to reconstruct, and its mirrored counterpart as shown in Figure 2 in Herteux and R ath (2020). We suspect that when there is a competition between attractors, be it attractors that are manually moved closer together or attractors that compete with their mirrored counterpart or other untrained attractors, this sequence of attractors becoming unstable combined with the constraint that the RC is prohibited from exhibiting globally unstable dynamics (due to the choice of activation function) in turn creates a new attractor that is composed of different metastable states, which in turn produces these switching dynamics.

Out of the many examples of routes to metastable dynamics discussed in Rossi et al. (2024), there are a number of similarities between the results presented in this paper and phenomena such as chaotic itinerancy and heteroclinic cycles. In the case of chaotic itinerancy, which describes a switching process whereby the state of an autonomous dynamical system switches between several ‘‘attractor ruins’’ or ‘‘quasi-attractors’’ (these were previously coexisting attractors that retain much of their original features except trajectories on these quasi-attractors leak into each other), in our case, these quasi-attractors are described as the metastable states  $\tilde{C}_A$  and  $\tilde{C}_B$ . In terms of heteroclinic cycles, this typically occurs when the unstable manifold of one saddle intersects with a stable manifold of the other saddle, which, in our case, these saddles would be the chaotic transients associated with  $C_A$  and  $C_B$ . However, further work is required in order to determine which of these phenomena our results are most closely aligned with. Furthermore, a similar route to chaotic behavior has been observed in the past by Grebogi et al. (1985), whereby when two unstable orbits move toward each other by changing a parameter in the system, they coalesce at a bifurcation point and subsequently disappear; however, after the bifurcation, a chaotic transient is produced, which persists for parameter values far beyond the bifurcation point. In our case, we have one stable attractor and an unstable orbit/relatively long transient in the closed-loop RC that as  $\rho$  is varied, and there is a bifurcation where the stable attractor becomes unstable and a new attractor is born, which, depending on the circumstances, is either a chaotic attractor or limit cycle. Moreover, there is a valid reason why there is no transient produced after the second attractor becomes unstable. Due to the design of this closed-loop RC, it is

prevented from ever becoming globally unstable, and since there is no other stable attractor present in the closed-loop RC when the second attractor becomes unstable, there is no option but for there to be a stable attractor born through these sequence of attractors becoming unstable.

While the routes to metastable behaviour mentioned above are well-studied phenomena they only arise in certain circumstances and rather than relying on there being a parameter in a dynamical system that so happens to produce these switching dynamics, the major advantage of the multifunctional reservoir computing setup studied in this paper is that we are able to systematically induce these switching dynamics by adjusting the location of  $C_A$  and  $C_B$ . As a further remark, while the results presented in this paper are based on  $C_A$  and  $C_B$  rotating in opposite directions, this is not a necessary condition in order for switching dynamics to emerge in the RC. From additional experiments that are not reported on in the present paper, we find that when  $C_A$  and  $C_B$  rotate in the same direction then switching dynamics also emerge at particular values of  $\rho$  as  $C_A$  and  $C_B$  are moved closer together. In future work we intend to conduct a wider study that includes additional factors which may influence the emergence and behaviour of switching dynamics in a RC that are related to the training data, such as, in the context of the seeing double problem, differences in the frequency or relative size of  $C_A$  and  $C_B$ , and the relationship between the training data and RC training parameters. The benefit of conducting such a step-by-step sequence of increasingly sophisticated experiments is that it provides a reasonable point of reference when attempting to make sense of how switching dynamics in a RC can emerge in more exotic scenarios involving, for instance, multiple chaotic attractors, or working with experimental data where transitions occur between states and multistability is suspected to play a role. Given the rich variety of interesting dynamics that we see arise when training the RC to reconstruct a coexistence of two circular orbits we expect that in these more complicated scenarios there are even more interesting dynamics waiting to be explored.

As a final comment, the work presented throughout this paper highlights the importance of studying the behavior of saddles and the bifurcations which take place as an RC, or any dynamical system-based machine learning approach, is trained to solve a given task. As strongly emphasized in Sussillo and Barak (2013), in order to open the black-box of machine learning approaches, it is necessary that we improve our understanding of the interaction between stable and unstable dynamics and pay closer attention to the influence of saddles that are present in the system.

## References

- Briggman, K. L., and Kristan, W. (2008). Multifunctional pattern-generating circuits. *Annu. Rev. Neurosci.* 31, 271–294. doi:10.1146/annurev.neuro.31.060407.125552
- Dickinson, P. S. (1995). Interactions among neural networks for behavior. *Curr. Opin. Neurobiol.* 5, 792–798. doi:10.1016/0959-4388(95)80108-1
- Flynn, A. (2023). Theory and applications of multifunctional reservoir computers.
- Flynn, A., Heilmann, O., Köglmayr, D., Tsachouridis, V. A., R ath, C., and Amann, A. (2022). "Exploring the limits of multifunctionality across different reservoir computers," in *2022 international joint conference on neural networks (IJCNN)*, 1–8.
- Flynn, A., Herteux, J., Tsachouridis, V. A., R ath, C., and Amann, A. (2021a). Symmetry kills the square in a multifunctional reservoir computer. *Chaos* 31, 073122. doi:10.1063/5.0055699
- Flynn, A., Tsachouridis, V. A., and Amann, A. (2021b). Multifunctionality in a reservoir computer. *Chaos* 31, 013125. doi:10.1063/5.0019974
- Flynn, A., Tsachouridis, V. A., and Amann, A. (2023). Seeing double with a multifunctional reservoir computer. *Chaos* 33, 113115. doi:10.1063/5.0157648
- Getting, P. A. (1989). Emerging principles governing the operation of neural networks. *Annu. Rev. Neurosci.* 12, 185–204. doi:10.1146/annurev.ne.12.030189.001153

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

AF: conceptualization, investigation, methodology, writing–original draft, and writing–review and editing. AA: writing–original draft and writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was partly supported by the Deutsche Forschungsgemeinschaft Project No. 411803875 and PIK Werkvertrag 2023-0336.

## Acknowledgments

The authors would like to thank Aravind Kumar, and their Applied Mathematics colleagues at the School of Mathematical Sciences, in particular, Andrew Keane, Pierce Ryan, Serhiy Yanchuk, and Sebastian Wicczorek, for their influential conversations and input when discussing the contents of this paper.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.



- Grebogi, C., Ott, E., and Yorke, J. A. (1985). Super persistent chaotic transients. *Ergod. Theory Dyn. Syst.* 5, 341–372. doi:10.1017/s014338570000300x
- Herteux, J., and R ath, C. (2020). Breaking symmetries of the reservoir equations in echo state networks. *Chaos* 30, 123142. doi:10.1063/5.0028993
- Jaeger, H. (2001). *The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note*. Bonn, Germany: German National Research Center for Information Technology GMD Technical.
- Krasnosel'skii, M. A., and Pokrovskii, A. V. (2012). *Systems with hysteresis*. Springer Science and Business Media.
- Lu, Z., Hunt, B. R., and Ott, E. (2018). Attractor reconstruction by machine learning. *Chaos* 28, 061104. doi:10.1063/1.5039508
- Maass, W., Natschl ager, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi:10.1162/089976602760407955
- Marder, E., and Calabrese, R. L. (1996). Principles of rhythmic motor pattern generation. *Physiol. Rev.* 76, 687–717. doi:10.1152/physrev.1996.76.3.687
- Morra, J., Flynn, A., Amann, A., and Daley, M. (2023). "Multifunctionality in a connectome-based reservoir computer," in *2023 IEEE international conference on systems, man, and cybernetics (SMC)* (IEEE), 4961–4966.
- Mpitsos, G. J., and Cohan, C. S. (1986). Convergence in a distributed nervous system: parallel processing and self-organization. *J. Neurobiol.* 17, 517–545. doi:10.1002/neu.480170512
- Nakajima, K., and Fischer, I. (2021). *Reservoir computing*. Springer.
- Rossi, K. L., Budzinski, R. C., Medeiros, E. S., Boaretto, B. R. R., Muller, L., and Feudel, U. (2024). Dynamical properties and mechanisms of metastability: a perspective in neuroscience. *arXiv Prepr.* doi:10.48550/arXiv.2305.05328
- Sussillo, D., and Barak, O. (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* 25, 626–649. doi:10.1162/NECO\_a\_00409
- Verstraeten, D., Schrauwen, B., and Stroobandt, D. (2005). "Reservoir computing with stochastic bitstream neurons," in *Proceedings of the 16th annual ProRISC workshop*, 454–459.