

Received 7 July 2023, accepted 22 July 2023, date of publication 27 July 2023, date of current version 7 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299296

 SURVEY

A Survey on Reservoir Computing and Its Interdisciplinary Applications Beyond Traditional Machine Learning

HENG ZHANG¹ AND DANILO VASCONCELLOS VARGAS^{1,2}

¹Department of Information Science and Technology, Kyushu University, Fukuoka 819-0395, Japan

²Department of Electrical Engineering and Information Systems, The University of Tokyo, Tokyo 113-8656, Japan

Corresponding author: Heng Zhang (rogerzhangheng@gmail.com)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Challenging Exploratory Research under Grant JP22534665, in part by the Japan Science and Technology Agency (JST) Strategic Basic Research Promotion Program (Advanced Integrated Intelligence Platform (AIP) Accelerated Research) under Grant 22584686, in part by the JSPS Research on Academic Transformation Areas (A) under Grant 22572551, and in part by JST Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2136.

ABSTRACT Reservoir computing (RC), first applied to temporal signal processing, is a recurrent neural network in which neurons are randomly connected. Once initialized, the connection strengths remain unchanged. Such a simple structure turns RC into a non-linear dynamical system that maps low-dimensional inputs into a high-dimensional space. The model's rich dynamics, linear separability, and memory capacity then enable a simple linear readout to generate adequate responses for various applications. RC spans areas far beyond machine learning, since it has been shown that the complex dynamics can be realized in various physical hardware implementations and biological devices. This yields greater flexibility and shorter computation time. Moreover, the neuronal responses triggered by the model's dynamics shed light on understanding brain mechanisms that also exploit similar dynamical processes. While the literature on RC is vast and fragmented, here we conduct a unified review of RC's recent developments from machine learning to physics, biology, and neuroscience. We first review the early RC models, and then survey the state-of-the-art models and their applications. We further introduce studies on modeling the brain's mechanisms by RC. Finally, we offer new perspectives on RC development, including reservoir design, coding frameworks unification, physical RC implementations, and interaction between RC, cognitive neuroscience and evolution.

INDEX TERMS Reservoir computing, neural networks, recurrent neural networks, nonlinear dynamical systems, cognitive neuroscience.

I. INTRODUCTION

Artificial neural networks (ANNs) attract attention in the fields of artificial intelligence, neuroscience, computer science, and machine learning. These ANNs can be mainly divided into two architectures: (1) feed-forward neural networks (FFNNs) and (2) recurrent neural networks (RNNs) [1]. In the field of neuroscience, it has been realized that the convergent feed-forward circuit observed in the cerebral cortex of mammals is a method used to encode relations, allowing cognitive objects to be represented through

multi-layered feed-forward architectures [2]. In machine learning, training FFNNs is a process that usually involves the optimization of a highly non-convex problem using gradient descent based methods to find the optimum. One of the biggest advantages of FFNNs is their ability to deal with static (non-temporal) data processing tasks such as image recognition [3], object detection [4] and semantic segmentation [5]. However, samples are normally independently processed in FFNNs, making it hard to handle temporally correlated events without memory.

On the other hand, RNNs are models where neurons are recurrently coupled with feedback connections. The recurrent connections provide rich non-linear dynamics and memory,

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda¹.

which are essential for temporal data and sequential processing. However, RNNs can be challenging to train. This is mainly because they must deal with vanishing and exploding gradient problems, along with other problems such as longer training time and the need for careful weight initialization [6], [7]. Back-propagation-through-time (BPTT) [8] and Long Short-Term Memory (LSTM) [9] networks are two solutions to some of the problems mentioned above. However, the learning difficulty still exists.

Reservoir computing (RC) is a bio-inspired, RNN based computational framework mainly originated from two independent models in the early 2000s: the Echo State Network (ESN) [10] and the Liquid State Machine (LSM) [11]. Similar and related concepts include backpropagation-decorrelation (BPDC) learning algorithm [12] and the cortico-striatal models realized in prefrontal cortex [13]. These proposed models were then unified as the RC computational framework by Verstraeten et al. [14], and were summarized in two reviews [15], [16].

A typical RC network includes a *reservoir* and a *readout* layer. Specifically, the reservoir is a network with randomly connected neurons. It can generate complex and high-dimensional transient responses to input. Such responses are considered as reservoir's states, forming unique trajectories for each input. These high-dimensional reservoir's states are then processed with a simple readout layer, generating the output. During RC training, the weight connections between neurons in the reservoir are usually fixed to their initial random values (i.e., remain untrained), and only the readout layer is trained by relatively simple learning algorithms (e.g., linear regression).

RC has several advantages, such as a fast training process, simplicity of implementation, reduction of computational cost, and no vanishing or exploding gradient problems, among others. Moreover, they do have memory and, for this reason, are capable of tackling temporal and sequential problems. These advantages have attracted increasing interest in RC research and many of its applications since its conception. However, due to the impressive wide-range results of gradient descent based neural networks—the deep learning revolution, RC research receded into a niche for a few years after the 2010s. Recently, there has been a resurgence of interest in RC as a compelling biological model of neural networks, driven by its potential for scalability through physical implementations. Additionally, problems in deep learning such as adversarial attacks and robustness/adaptiveness issues have also contributed to an increase in interest in alternative paradigms [17], [18], [19], [20], [21], [22].

A. AIMS

The complex dynamics in the reservoir indicate that RC is not only just a machine learning framework, but also a concept that highly correlates to physics, biology, and neuroscience. In 2019, a review in physical RC (PRC) comprehensively summarizes various physical materials and hardware com-

ponents that can be used for PRC implementations [23]. In biology and neuroscience, it was reported in 2013 that parts of the brain's mechanisms are similar to those of reservoirs, including mixed selectivity [24] and neuronal oscillations [2]. Although valuable surveys exist in specific research branches, there is still a notable gap in the literature regarding understanding the intricate interplay between RC and its connections to physics, biology, and neuroscience. The absence of such a comprehensive review hinders the advancement of our knowledge and the exploration of RC's potential across diverse domains. Addressing this need and providing a holistic overview would significantly contribute to bridging this gap, facilitating further research, and unlocking new insights into the various aspects of RC.

This survey provides a clear exposition of recent developments in RC. While the literature on RC is vast and fragmented, we aim to provide a uniform introduction to RC. We begin in **Section II** by introducing the fundamental concepts of RC. In **Section III**, we explore different network architectures and optimization techniques that can enhance reservoir performance. **Section IV** further highlights the recent trends of RC, facilitated by improved training schemes. Various physical hardware solutions are also reviewed, covering electronic, optical, and biophysical approaches, among others. Furthermore, we demonstrate the wide range of applications of RC in **Section V**, including practical engineering, natural science, and social/data science, etc. These applications often involve time-dependent data, requiring memory and memory-related processing. Next, by discussing the cortico-striatal models and coupled oscillator networks in **Section VI**, we show that high-dimensional responses triggered by the reservoir's dynamics offer insights into brain mechanisms that also exploit a high-dimensional dynamical process. Additionally, our contributions to RC reside in **Section VII**, where we present fresh perspectives on the development of RC and pinpoint open problems that require further research.

II. BASIC FRAMEWORK OF RC

In this section, we first review the historical developments in RC. By tracing its trajectory from initial theoretical foundations to the present state, our emphasis will then be on highlighting significant works that have played a crucial role in shaping the current landscape of RC.

A. HISTORY OF RC DEVELOPMENTS

1) THE FIRST PROTOTYPE OF RESERVOIR COMPUTING

Perhaps the birth of the RC framework was in the 1980s-90s. During that period, some researchers were focusing on the characterization of the fast eye movements (i.e., the oculomotor saccade) in the corticostriatal system—the interactions between cortex and basal ganglia [25]. In a pioneering work, Barone and Joseph [26] examined the function of the corticostriatal system by carrying saccade experiments on macaque monkeys. They found that some neurons have

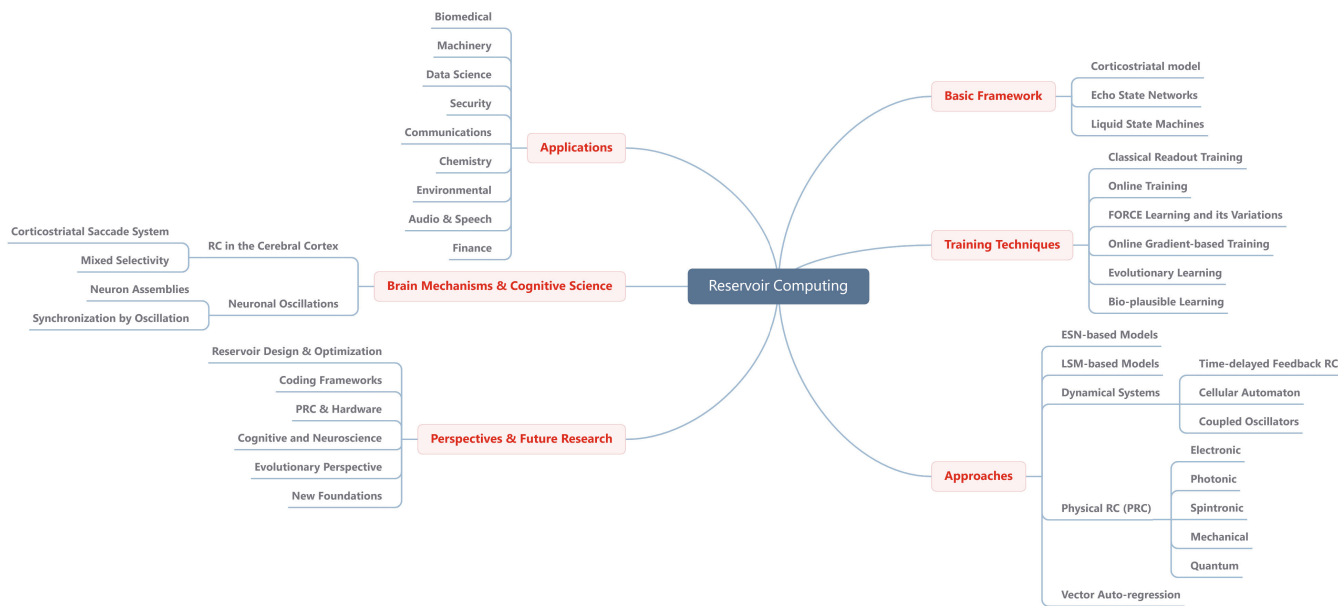


FIGURE 1. Paper structure showing an overall picture and future trends of research in reservoir computing.

a preferred spatial saccade amplitude and direction, resulting in selective responses to a particular sequential order. This finding was further characterized as *mixed selectivity* by [24] in 2013, which is one of the important principles in both RC and cognitive science (see Section VI).

Taking inspiration from the experiments of the cortico-striatal saccade system [26], [27], Dominey [13], [28] developed the very first prototype of the reservoir network in 1995 (see Section VI and Fig. 15 for detail). Specifically, the model was built based on a recurrent prefrontal cortex (PFC) system (the reservoir), and a reward-related learning method in PFC-to-caudate connections (the readout). Since they found that the modifications of the recurrent connections are considerably computationally costing, they decided to initialize the PFC layer with a mixture of fixed inhibitory and excitatory recurrent connections (i.e., a reservoir with fixed connections between neurons). The reservoir was then connected to the caudate or striatum to obtain the readout.

2) EARLY RESEARCH AND PROBLEMS IDENTIFIED IN GENERAL RNNs

The main branch of RC was originated in the fields of temporal and sequential pattern recognition using RNNs. In contrast to FFNNs that aim to approximate non-linear input-output *functions*, RNNs are capable of representing *dynamical systems* and processing sequential inputs with recurrent connections [15], [23].

Early studies of RNNs include a well-known model called Hopfield network [29] in the 1980s. The network topologies were specifically formulated with symmetrical weights connections and were trained in unsupervised ways. This special type of network normally experiences chaotic or stochastic dynamics with the mathematical background of statistical physics [15]. Another type of RNN features a deterministic

update dynamics and directed weighted connections. Systems from this type of RNN are usually made of high dimensional hidden states with non-linear dynamics, resulting in a transformation from an input sequence into an output sequence. Two standard examples are (1) back-propagation-through-time (BPTT) by [8] and [30]; and (2) real-time recurrent learning (RTRL) by [31]. Even though these learning methods showed great potential in complex sequential processing, they struggled to tackle real-world problems due to the high computational costs and difficulties of training, especially the vanishing and exploding gradient problems that make them hard to capture long-term dependencies [6], [7]. In 1997, a well-known architecture, Long Short Term Memory (LSTM) [9], was then proposed to address these problems. For more details on the gradient-based RNNs, please refer to an early review [32].

3) UNIFICATION OF RESERVOIR COMPUTING

In 2000, [32] proposed a new algorithm based on error gradient approximation, which efficiently reduces the computational complexity and shows faster convergence in recurrent network training. This work, referred to Atiya and Parlos Recurrent Learning (APRL) in later literature, identified that an RNN can be divided into two parts: the quickly changing output weights, and the slowly adapting hidden weights. Therefore, APRL is considered the algorithm to bridge between general RNNs and reservoir computing [15]. Besides, another predecessor of RC, the backpropagation-decorrelation (BPDC) learning algorithm, further simplified APRL and made it an online learning algorithm [12].

Later in the early 2000s, two types of fundamental reservoir computing algorithms were independently invented by Maass et al. [11] as Liquid State Machine (LSM), and by Jaeger [10] as Echo State Network (ESN). The two

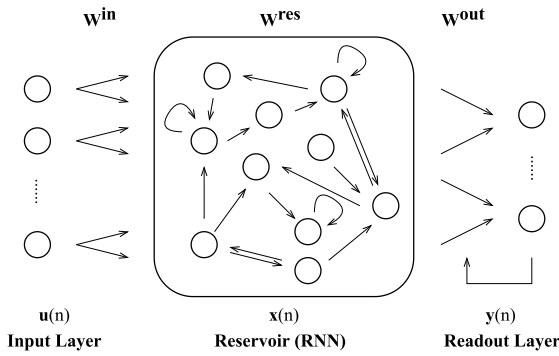


FIGURE 2. Simplified structure of Echo State Network (ESN).

algorithms, as well as other related works such as BPDC and works in neuroscience fields such as Dominey’s research [13], were unified as a computational framework called *reservoir computing* (RC) [14]. In this unified framework, the low-dimensional input data is transformed into spatio-temporal patterns in a high-dimensional space by the *reservoir*—an RNN with fixed topologies and unchanged weights. The high-dimensional responses generated by the reservoir are then processed by the *readout*—an output layer which can be trained with simple learning algorithms such as linear regression. In other words, during training, the values of the weight connections within the reservoir remain unchanged while only the readout weights are trained based on specific tasks. Before going deep into recent advances in RC, we will first introduce the basic concepts of ESN and LSM in the following subsections.

B. ECHO STATE NETWORKS

Echo State Network (ESN) was first proposed by [10]. This pioneering work is based on the fact that training only the readout layer of an RNN can achieve acceptable performance, if the network has sufficiently rich dynamics. ESN is normally implemented with leaky-integrated, non-spiking, discrete-time and continuous-value artificial neurons (see Fig. 2 the network structure). To illustrate the technical details, here we use the notations by [33]. Consider a temporal processing task, where the input signal is $\mathbf{u}(n) \in \mathbb{R}^{N_u}$ and the desired target signal is $\mathbf{y}^{target}(n) \in \mathbb{R}^{N_y}$, given $n = 1, \dots, T$ with T being the total number of discrete data points. The goal is to generate an output signal $\mathbf{y}(n) \in \mathbb{R}^{N_y}$ that matches $\mathbf{y}^{target}(n)$ as optimally as possible by minimizing the error between the two signals (e.g., Mean-Square Error, MSE). The simplified update equations of the reservoir part in ESNs are given by:

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}^{in}\mathbf{u}(n) + \mathbf{W}\mathbf{x}(n-1)), \quad (1)$$

$$\mathbf{x}(n) = (1 - \alpha)\mathbf{x}(n-1) + \alpha\tilde{\mathbf{x}}(n), \quad (2)$$

where $\tilde{\mathbf{x}}(n) \in \mathbb{R}^{N_x}$ is the update at time step n , $\mathbf{x}(n) \in \mathbb{R}^{N_x}$ is the state vector of the reservoir neurons (also known as the resulting states or the *echo* of its input history [15]), $\mathbf{W}^{in} \in \mathbb{R}^{N_x \times N_u}$ and $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$ are the weight matrices of the input-reservoir connections and the recurrent connections

Algorithm 1 Simplified Procedure of ESN Training

- 1: Initialize the network by generating random \mathbf{W}^{in} , \mathbf{W} . It is common to use uniform distributed randomization $U(-1, 1)$.
- 2: Run the model with input signal $\mathbf{u}(n)$, $n = 1, \dots, T$, it will generate the same length of the reservoir states $\mathbf{x}(n)$ by Eq. 1-2.
- 3: Collect all $\mathbf{x}(n)$, then calculate and get the output signal $\mathbf{y}(n)$ by Eq. 3.
- 4: Minimize the MSE between $\mathbf{y}(n)$ and $\mathbf{y}^{target}(n)$ using techniques such as linear regression. This should obtain a well-trained \mathbf{W}^{out} .
- 5: Take unseen data $\mathbf{u}_{test}(n)$ and obtain the predicted and/or generated output $\mathbf{y}_{test}(n)$.

inside the reservoir, respectively. $\tanh()$ is the non-linear activation function applied element-wise. α is the leaking rate that mainly controls the speed of the dynamics.

The readout layer is normally linearly defined as:

$$\mathbf{y}(n) = \mathbf{W}^{out}\mathbf{x}(n), \quad (3)$$

where $\mathbf{y}(n) \in \mathbb{R}^{N_y}$ is the output vector and $\mathbf{W}^{out} \in \mathbb{R}^{N_y \times N_x}$ is the weight matrix of the reservoir-readout connections. Alternatively, one can also introduce a bias value in both reservoir and readout, as well as integrate the input signal directly to the readout layer. In this case, $\mathbf{u}(n)$ in Eq. 1 becomes $[1; \mathbf{u}(n)]$ and $\mathbf{x}(n)$ in Eq. 3 becomes $[1; \mathbf{u}(n); \mathbf{x}(n)]$, where $[\cdot; \cdot]$ represents concatenation. A brief training procedure is shown in Alg. 1.

During conventional ESNs training, \mathbf{W}^{in} and \mathbf{W} remain unchanged, and only \mathbf{W}^{out} is trained in order to minimize the error between the network output and the target output (teacher signal), usually by using linear regression such as ridge regression [34]. Alternatively, many new proposed learning rules for ESN training exist, including but not limited to online FORCE learning [35], weights pre-training [36], gradient-based training [37], and evolutionary learning [38]. For details on training a RC model, please refer to section III.

1) ECHO STATE PROPERTY

An essential condition (requisite) that a standard ESN must meet is the echo state property (ESP), which ensures a condition of asymptotic state convergence of the reservoir. This property is under the influence of both the reservoir and the given input. On one hand, ESP is an algebraic property that is controlled by the reservoir’s weight matrix \mathbf{W} . It has been mathematically analyzed in [15], [39], and [33] that the spectral radius (SR , i.e., the maximum eigenvalue of \mathbf{W}) smaller than unity ensures ESP in most situations. As a result, many RNN-based RCs in literature consider $SR < 1$ as a necessary condition to make the models work (see the introduction section in [16]). However, it had been proved in [39] that $SR < 1$ is neither sufficient nor necessary for the ESP. The author claim that “it is not required to scale the spectral radius below 1, and there is no general benefit in scaling

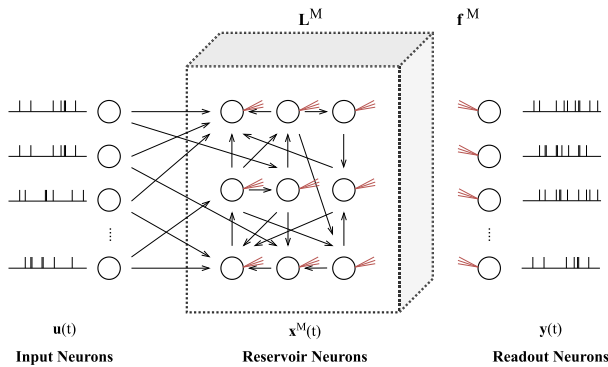


FIGURE 3. Simplified structure of Liquid State Machine (LSM). Note that LSM is implemented by spiking integrate-and-fire neurons, rather than non-spiking artificial neurons used in ESN.

the spectral radius toward the Edge of Chaos”. The same paper also proposed new sufficient conditions for the ESP. Please refer to [39] for mathematical details. On the other hand, ESP has been empirically studied in the presence of driving inputs of varied strength [40], without looking at the mathematics. Reference [40] shows that for an input-driven reservoir and a proper input scaling, the actual range of ESP validity (i.e., SR), is much wider than what is covered by the above literature conditions.

2) MEMORY CAPACITY AND EDGE OF CHAOS

As a special type of RNNs, ESN also has the characteristic of short-term memory. Analytical results that characterize the dynamical short-term memory capacity of reservoirs were discussed in [41] and [42]. Meanwhile, it can be found in a good deal of literature that reservoirs are claimed to work best when they are tuned to operate at the so-called “edge of chaos” [43], [44]. Here, the edge of chaos refers to a region of parameter settings which makes the dynamical system operates at the boundary between the chaotic and non-chaotic behavior. However, this is also a misnomer, as claimed by Jaeger in [45] and [39] that the “edge” in question here is the edge of the ESP, not the edge of chaos. For a detailed discussion, please refer to [46].

C. LIQUID STATE MACHINES

Liquid State Machine (LSM) was proposed by [11] from the perspective of computational neuroscience in order to study the brain mechanisms and model the neural microcircuits. In contrast to ESN that uses non-spiking artificial neurons, LSM is more biologically plausible as it is based on the Spiking Neural Networks (SNNs) with recurrent reservoir structures. Inside the reservoir, usually a 3D structured and locally connected network of spiking integrate-and-fire (IF) neurons is randomly created and stimulated by external input spike train signals (see Fig. 3). Intuitively, the reservoir in LSM is often called liquid, since they follow a metaphor of excited states as ripples on the surface of a pool of water [11]. Similar to ESNs in its form, here we use $\mathbf{u}(t)$, $\mathbf{x}(t)$ and $\mathbf{y}(t)$ to represent the input, reservoir state and output, respectively.

The reservoir dynamic of LSM is given by:

$$\mathbf{x}^M(t) = \mathbf{L}^M \mathbf{u}(t), \quad (4)$$

where t represents continuous time, \mathbf{x}^M is the reservoir state, \mathbf{u} represents the input spikes, and \mathbf{L}^M is the *liquid filter* for input-reservoir state transformation. The readout is given by:

$$\mathbf{y}(t) = \mathbf{f}^M(\mathbf{x}^M(t)), \quad (5)$$

where $\mathbf{y}(t)$ is the output vector and \mathbf{f}^M is a “memory-less” *readout map*. The readout here can also be trained using simple algorithms.

1) SEPARATION AND APPROXIMATION PROPERTIES

LSMs have two mathematical preconditions, namely separation property (SP) and approximation property (AP). These two properties ensure that the network has fading memory (i.e., echo state property in ESN). Specifically, SP addresses the degree of separation between different internal states \mathbf{x} caused by different input \mathbf{u} (condition is met if the *liquid filter* \mathbf{L}^M satisfies the point-wise separation property), whereas AP addresses the capability of the readout layer to produce the target outputs given different liquid states \mathbf{x} (condition is met if the *readout map* \mathbf{f}^M satisfies the approximation property). For the mathematical basis of SP and AP, please refer to [11]. Overall, these two properties in LSMs, together with the ESP and memory capacity in ESNs literature, ensure the RNN-based reservoirs function properly.

D. COMPARISON OF ESNs & LSMs

ESNs and LSMs are two similar RNNs with reservoir structures. The main difference between them is that LSM used spiking IF neurons, while ESN is based on non-spiking neurons. This makes LSMs more biologically plausible to be used in investigating biological mechanisms of information processing in the brain. In terms of model implementation, although both models show noticeable advantages in reducing computation cost and training time, LSM, with its biologically inspired characteristics and spiking implementation, becomes more suitable for new types of hardware such as neuromorphic chips [47]. Therefore, LSM is reported with several hardware designs and applications.

ESNs show better flexibility in model modifications, as many variants of ESNs were proposed to enhance the network performances (see Section III). These modifications are mainly to overcome the disadvantages of conventional ESNs. The first drawback is that the fixed connection weights could limit the performance of ESNs, since they are randomly initialized without the process of tuning or optimizing. Moreover, it is shown that the improvement of ESNs performance will reach a saturation as the size of the reservoir increases to a certain amount. This means that only increasing reservoir size may not result in a better performance. As a result, researchers have been focusing on constructing multi-reservoir ESNs, such as multi-layered ESNs [48], [49], [50] and parallel reservoir computing [51]. Meanwhile, several

optimization approaches have been proposed to fine-tune the networks weights and hyperparameters by using either evolutionary algorithms [50], [52] or gradient based optimization techniques [37]. Detailed discussions of recent ESN and LSM models are covered in section IV.

III. TRAINING A RC MODEL

In this section, we will delve into an array of methodologies for training and optimizing RC models (see Table 1). Beyond simply training the readout, we will explore diverse techniques that aspire to enhance the construction of reservoirs from various aspects, thereby enabling more effective and efficient models. These include but does not limit to (1) classical readout training such as ridge regression [34]; (2) online learning such as least mean square method and FORCE learning [35]; (3) pre-training such as particle swarm optimization [52]; (4) online gradient based learning with back-propagation [75], [76]; (5) evolutionary learning such as Evolino [66]; and (6) biologically plausible learning techniques such as Hebbian learning [77].

A. CLASSICAL READOUT TRAINING

The original works of ESN and LSM state that the readout of a reservoir with rich dynamics can be trained by using any statistical classification or regression methods [16]. Using ESNs for example, it is recommended to apply simple linear regression technique to single-layer readout. Again, we use the notations by [33] for illustration. First, notice that Eq. 3 can be rewritten and extended in a matrix form as:

$$\mathbf{Y} = \mathbf{W}^{out} \mathbf{X} \approx \mathbf{Y}^{target}, \quad (6)$$

where $\mathbf{Y} \in \mathbb{R}^{N_y \times T}$ stands for the collection of all $\mathbf{y}(n)$. Similar notation goes to \mathbf{X} and \mathbf{Y}^{target} . It is clear that \mathbf{W}^{out} needs to be optimized to minimize the difference between \mathbf{Y} and \mathbf{Y}^{target} . The most common technique is the ridge regression [34], where \mathbf{W}^{out} is obtained by:

$$\mathbf{W}^{out} = \mathbf{Y}^{target} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1}, \quad (7)$$

where \mathbf{I} is the identity matrix with β being the regularization factor. Detailed implementation can be found in a practical guide of ESN training in [33]. Once \mathbf{X} is obtained in an off-line way, one can tune β to reach the best performance without any model retraining. Ridge regression often shows sufficiency in many concrete tasks, when the reservoir provides rich non-linear dynamics. Moreover, it is easy and fast to train, which attracts many researchers coming from non-machine learning backgrounds.

B. ONLINE LEARNING TECHNIQUES

Reference [33] provides many empirical solutions on how to produce a reservoir with good initialization. However, problems still exist, as one cannot guarantee that the reservoir is always well-initialized. Another effective way to solve this problem is online adaptation. In this manner, a feedback loop between the reservoir and readout is usually introduced.

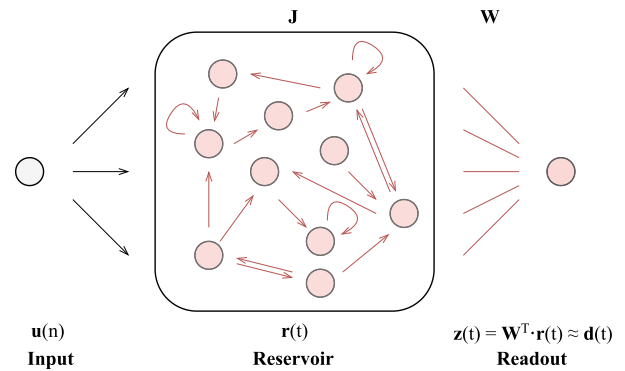


FIGURE 4. FORCE learning overview. The red color represents that the weight connections can be modified during training.

1) LEAST MEAN SQUARES (LMS) AND RECURSIVE LEAST SQUARES (RLS) METHODS

Originated in the area of adaptive signal processing, LMS and RLS are the two standard online learning methods for reservoir computing models [78]. The mathematical description of LMS and RLS are presented in [53] and [55]. To illustrate, LMS is a gradient-based error minimization method in which an error is exponentially discounted propagating back through time, yet this method might be unstable because of the large eigenvalue spreads of the cross-correlation matrix (i.e., $\mathbf{X} \mathbf{X}^T$). Moreover, it is reported that LMS struggles to capture the history-dependent temporal data [79]. Compared to LMS, RLS is more popular due to its robustness/insensitivity to the effect of eigenvalue spreads mentioned above, as well as its faster second-order convergence speed. Therefore, RLS method for RC has been widely studied in [59], [60], and [55]. Although RLS has advantages over LMS, it is more computationally costly with $O(N^2)$ time complexity, while LMS only requires $O(N)$ in most situations [41], where N is the number of variables. Having said that, RLS is employed by FORCE learning, which creates a new branch of reservoir computing with regard to cognitive science and brain mechanisms.

2) FORCE LEARNING

It has been shown that RNNs often experience spontaneous chaotic activity, and algorithms such as BPTT [30] are usually not able to converge if the network exhibits chaotic activity. ESN models address the chaotic activity by ensuring echo state property (ESP), so that the models do not operate in a chaotic manner. Instead of avoiding spontaneous activity like ESNs, *Fisrt-Order Reduced and Controlled Error* (FORCE) learning, which is perhaps the most popular online learning method of RC, reveals that the results are better when the reservoirs exhibit chaotic behavior before training [35]. By modifying the synaptic strengths of the reservoir (either internal or external), models trained with FORCE learning show the effectiveness of suppressing autonomous chaotic activity while turning it into a wide variety of desired output patterns. Since the original mathematics are quite complex, here we aim to provide a simplified description of the online

TABLE 1. Various training techniques of RC.

Category of techniques	Training approaches
Classic readout training	Linear regression
	Ridge regression [34]
Online learning techniques	Least mean squares (LMS) [53], [54]
	Recursive least squares (RLS) [55]
	FORCE learning [35]
	FORCE variations [55]–[62]
Online gradient based training	LSM method [53], [54]
	GD-based optimizer [37]
	BackPropagation-DeCorrelation (BPDC) [15]
	Back-propagation Through Time (BPTT) ACTRNN [63]
Evolutionary learning techniques	Genetic algorithm (GA) [64]
	Particle swarm optimization (PSO) [52], [65]
	Competitive swarm optimizer (CSO) [50]
	Evolino [66]
Biologically plausible learning techniques	Hebbian learning [67]
	Spike Timing Dependent Plasticity (STDP) [68]–[70] Intrinsic Plasticity (IP) [71]–[74]

readout training for completeness [80]. Please refer to [35] for details.

Consider the reservoir state as $\mathbf{r}(t)$ (i.e., $\mathbf{x}(n)$ in ESNs), the network output $\mathbf{z}(t)$ (i.e., $\mathbf{y}(n)$ in ESNs) is defined as:

$$\mathbf{z}(t) = \mathbf{w}^T \mathbf{r}(t), \tag{8}$$

where \mathbf{w} is the weights connecting reservoir and readout. Note that here the output dimension is restricted to one, while it can be easily generalized to multidimensional. Training of \mathbf{w} happens at every time interval Δt . Before updating at time t , the error is denoted by:

$$e_-(t) = \mathbf{w}^T(t - \Delta t)\mathbf{r}(t) - f(t), \tag{9}$$

where $f(t)$ is the predefined target function (i.e., $\mathbf{y}^{target}(n)$ in ESNs). The FORCE algorithm uses a modified RLS method to update the weights by:

$$\mathbf{w}(t) = \mathbf{w}(t - \Delta t) - e_-(t)\mathbf{P}(t)\mathbf{r}(t), \tag{10}$$

where $\mathbf{P}(t)$ is a square matrix that is updated at the same time as the weights according to

$$\mathbf{P}(t) = \mathbf{P}(t - \Delta t) - \frac{\mathbf{P}(t - \Delta t)\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{P}(t - \Delta t)}{1 + \mathbf{r}^T(t)\mathbf{P}(t - \Delta t)\mathbf{r}(t)}, \tag{11}$$

and is initialized as:

$$\mathbf{P}(0) = \frac{\mathbf{I}}{\alpha}, \tag{12}$$

where \mathbf{I} is the identity matrix with α as constant. After training, the error becomes

$$e_+(t) = e_-(t)(1 - \mathbf{r}^T(t)\mathbf{P}(t)\mathbf{r}(t)). \tag{13}$$

Finally, the training will end when it reaches

$$\frac{e_+(t)}{e_-(t)} \approx 1. \tag{14}$$

The above modified RLS method is applied to suppress the output errors and frequently adapt the weight matrices

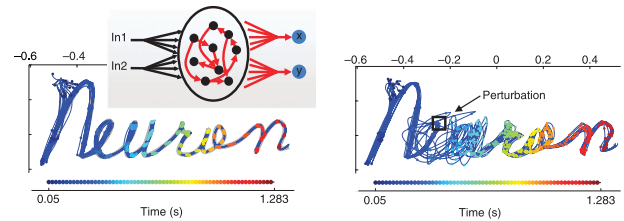


FIGURE 5. Generation and stability of complex spatio-temporal motor patterns by FORCE learning (modified from [58]). Left panel shows the ten trials to re-generate the target signal (in gray line) after training. Right panel shows the same trials under the addition of a perturbation.

in the reservoir or readout (see Fig. 4). This makes FORCE learning disparate from other traditional iterative training schemes—the errors in FORCE learning are always small during training, even at the beginning, suggesting that the aim is not to reduce errors but rather to keep the errors small. When training is done, the network will autonomously generate the desired output. As the author claimed, FORCE learning helps to construct machine learning based RNNs that “generate complex and controllable patterns of activity either in the absence of or in response to input”. It provides an interesting link between computational and biological neuroscience. In short, FORCE learning can be seen as a useful tool for optimizing RC, and simultaneously, it presents a potential model that could help understand biological neural circuits [35].

3) FORCE LEARNING VARIATIONS AND IMPLEMENTATIONS

One of the disadvantages of FORCE learning is that the trained network is too complex to analyze the neuron activities. Also, for complex real-world problems such as speech recognition, networks trained by FORCE require many more units to match the performance of gradient based networks, as reported in [56].

Therefore, some studies have been published aiming at improving FORCE learning, ranging from neuroscience (e.g., spiking networks) to physical and hardware implementations. An “extended” FORCE [57], for example, was proposed for more general internal learning by using the desired output to generate targets for every internal neuron in the network. This so-called “target-generating” network is then improved by [58], in which the reservoir network can preserve time information and generate complex and high-dimensional trajectories even under high levels of noise (see Fig. 5). To this end, the FORCE variations are still infeasible to be implemented as a spiking network. Later, [59] proposed a full-FORCE algorithm. Compared with FORCE learning, it requires fewer neurons, achieves significantly better performance in noisy environments, and can also be applied to SNNs (see implementation in [60]). For hands-on implementation, the full-FORCE has been realized using a Python spiking network framework called Nengo [81]. Other recent improvements of FORCE learning include (1) Two-Step FORCE that converges faster than the original work [61]; (2) Transfer-FORCE learning which takes the advantages of both LMS and RLS methods for better learning performance [55]; (3) R-FORCE that aims to model multidimensional sequences [62]. Very recently, [82] built an object-oriented, open-source Python package that implements a TensorFlow / Keras API for FORCE.

C. ONLINE GRADIENT BASED TRAINING

Reservoir based on non-spiking artificial neurons (e.g., ESNs) can be trained by using gradient descent (GD) approaches. The LMS method mentioned earlier is one of the candidates which is first proposed in [54], yet using such approach shows poor stability even when trying to stabilize the network by adding noise. Reference [37] proposed a more stable version of GD-based reservoir to optimize four hyperparameters: the input scaling, spectral radius, leaking rate, and regularization parameter. Besides, BackPropagation-DeCorrelation (BPDC) algorithm is another powerful method for online training of single-layer readouts with feedback connected back to reservoirs. This algorithm is robust to the random initialization of the reservoir weights, and it is also capable of tracking quickly changing signals. Detailed discussions of BPDC are presented in a survey [15]. Meanwhile, the classical Back-propagation Through Time (BPTT) approach for RNN training can also be applied to RC model. It is worth noting that a network architecture called Adaptive Continuous Time Recurrent Neural Network (ACTRNN) [63] shows some similarity to GD-based RC. Please refer to Section VI for details.

D. EVOLUTIONARY LEARNING TECHNIQUES

1) EVOLUTIONARY ALGORITHM

One of the disadvantages of the traditional RC models is that the performance is highly reliant on the random initialization of the weights and hyperparameters. While the

optimal hyperparameters can be found by grid-search techniques, using such techniques to find the optimal weights’ initialization is nearly infeasible [33], given a concrete task. Therefore, instead of applying online learning rules, another possible direction, taking inspiration from above, is to train (or pre-train) the reservoirs using evolutionary algorithms (EA). When EAs are applied to evolve any type of neural networks (including reservoirs) they usually receive the name of neuroevolution.

Various types of EA can be used to evolve a reservoir, including (1) genetic algorithms (GA) [64]; (2) particle swarm optimization (PSO) [52], [65] and its variants [50]; and (3) artificial bee colony [83]. For example, a GA was applied to a double-reservoir ESN for parameter optimization, yet without optimizing weights of input and the reservoirs [36]. Inspired by LSTM [9], another EA for RC was proposed called Evolino [66]. Evolino constructs units that are capable of preserving memory for long periods of time, in which the weights of the reservoir are trained using evolutionary methods. A performance comparison of several EAs for RC are presented in [84].

Particle swarm optimization (PSO), which is an efficient and widely used technique for finding optimal regions on complex spaces, has also applied to reservoir weight optimization. The first two attempts of using PSO technique include using a binary PSO to find the optimal reservoir-readout connections [38], as well as a supervised PSO algorithm by [65] for better initializing the input weights of RC. However, only a subset of the weights was tuned in the latter model, due to the high computational cost. Reference [52] further developed the PSO algorithm for RC, where a portion of fixed weights in an ESN is pre-trained via PSO. The results show improvements on model generation as well as a faster convergence time, yet the network architecture is rather simple, and some hyperparameters should be selected empirically and carefully. The latest version of PSO based RC is the competitive swarm optimizer (CSO) for fault diagnosis problems [50], which is a hybrid evolutionary algorithm combining both a variant of PSO and local search (LS).

E. BIOLOGICALLY PLAUSIBLE LEARNING TECHNIQUES

1) HEBBIAN LEARNING AND SPIKE TIMING DEPENDENT PLASTICITY

Reservoir computing takes inspiration partially from biological systems. LSM-based reservoirs, for example, are implemented using spiking neurons. This indicates that some biologically plausible adaptation methods can be applied for reservoir training. Inspired by synaptic plasticity in human brains, the first attempts would be to use Hebbian and anti-Hebbian learning to try to decrease the eigenvalue spread in ESNs but failed [67]. Later, it is reported in [41] that the reservoir trained by Hebbian learning “makes neurons prefer inputs that are easy to predict and weaken connections from those that carry more information”. In terms of LSM,

the spike-time-dependent plasticity (STDP), which is based on Hebbian learning and is often integrated with SNNs, is reported to improve the separation property (SP) in some real-world speech data [68]. STDP was further developed to reduce memory storage load to make RC more hardware-friendly [69], [70].

2) INTRINSIC PLASTICITY

Another biologically plausible way of adaptation is based on Intrinsic Plasticity (IP), which is an unsupervised learning rule used for adapting the intrinsic excitability of the reservoir neurons. Here, intrinsic excitability refers to a phenomenon called long-term potentiation, in which brief and high-frequency stimulation tends to produce an increased ability to generate spikes [85]. Early research of the integration of RC and IP mainly focuses on reservoir pre-training and global optimizations [71], [72], [73]. In 2019, an IP with a local search scheme was proposed to improve the flexibility of the IP rule by allowing hyperparameters such as learning rate to be different [86]. In 2022, [87] applied IP learning to successfully tune the parameter in MEMS-based RC. As a side note, experiments on intrinsic plasticity have shown that the output distributions of real biological neurons may have different forms in different brain regions among various species [74].

IV. RECENT APPROACHES IN RC

A. OVERVIEW

RC has witnessed a significant development in recent years. On one hand, traditional RC models such as ESN and LSM have been improved by many new proposed models. Some of these models are built on top of the original ones to achieve better performances, while several new architectures of RC have been proposed to solve problems with increasing difficulties. On the other hand, recent studies have demonstrated that the idea of reservoir (i.e., a dynamical system that can generate high-dimensional and non-linear responses) can be implemented by using various materials, such as electronic devices, physical systems, and biological realizations.

In this section, we aim to cover recent approaches in RC from several perspectives up until 2023, including ESN-based RC, LSM-based RC, dynamical systems, and physical RC. These approaches are highly interdisciplinary and are usually tested in several benchmark tasks. To introduce some, benchmark tasks for pattern classification includes spoken digit recognition [88], waveform [89] and handwritten digit image recognition [90]. Besides, the non-linear Autoregressive Moving Average (NARMA) time series [53] was widely used in time series forecasting, while a channel equalization benchmark [54] was introduced to evaluate the RC performance on adaptive filtering and control. In addition, temporal XOR task [43] and memory capacity task [42] were also commonly used in studies that focus on system approximation and short-term memory.

B. RECENT TRENDS OF ESN-BASED RC

ESNs represent one of the foundational RC models. The simplicity of implementation makes them an approachable entry point of RC. Therefore, improving and extending ESNs is not only a key pursuit within the RC community, but its impacts extend far beyond, proving particularly influential for researchers outside the field, and thus allowing more interdisciplinary research. In the following, we review recent trends of ESN-based RC models.

1) MULTIPLE RESERVOIRS

a: DEEP ESN

Apart from the performance saturation problems, there is another limitation in the conventional ESN, i.e., the single large-scale reservoir is poor in simultaneously dealing with different timescales [91]. In this concern, some studies started to investigate multiple timescale dynamics of reservoir structure, as it has been found that stacking recurrent networks with different topologies can generate multiple timescales at different layers [49].

In 2016, [49] proposed a deep reservoir computing model to achieve hierarchical timescale representation. This model, called deep Echo State Network (DeepESN), stacks multiple reservoirs one on top of the other, as shown in Fig. 6. Mathematically, consider N the number of reservoir layers. The update equations, extended from Eq. 1-2, is given by:

$$\tilde{\mathbf{x}}^{(i)}(n) = \tanh(\mathbf{W}_{fwd}^{(i)} \mathbf{x}^{(i-1)}(n) + \mathbf{W}_{rec}^{(i)} \mathbf{x}^{(i)}(n-1)), \quad (15)$$

$$\mathbf{x}^{(i)}(n) = (1 - \alpha^{(i)}) \mathbf{x}^{(i)}(n-1) + \alpha^{(i)} \tilde{\mathbf{x}}^{(i)}(n), \quad (16)$$

where $\tilde{\mathbf{x}}^{(i)}(n)$, $\mathbf{x}^{(i)}(n)$ and $\alpha^{(i)}$ are the update, the reservoir state vector, and the leaking rate at layer i , respectively. $\mathbf{W}_{fwd}^{(i)}$ is the weight matrix connecting layers $i-1$ and i , and $\mathbf{W}_{rec}^{(i)}$ is the weight matrix of the reservoir at layer i , $i = 1 \dots N$. The number of input layer is denoted by $i = 0$ and $\mathbf{x}^{(0)}(n) = \mathbf{u}(n)$. At each time step n , the composition of the states in all the reservoir layers $\mathbf{x}(n)$ is given by:

$$\mathbf{x}(n) = \langle \mathbf{x}^{(1)}(n), \dots, \mathbf{x}^{(N)}(n) \rangle. \quad (17)$$

Experiments show that the deep RC structure can achieve (1) multiple timescale representation, where the timescales are ordered along the network's hierarchy; (2) multiple frequency representation, where progressively higher layers focus on progressively lower frequencies. Additionally, when there is a perturbation at the input, the effects of this perturbation last longer for higher layers in the stack, and this differentiation is drastically attenuated when input is provided to every layer. Therefore, having deeper layers while increasing distance from the input is a key architectural factor for obtaining a time-scales separation.

DeepESN shows potential for designing more efficient RC learning algorithms used for sequential and temporal data processing. From 2022, A branch of deep ESN is recently studied by the research team of Tanaka et al. modified the deep network architectures were proposed, combining with other techniques such as sequence resampling [92]

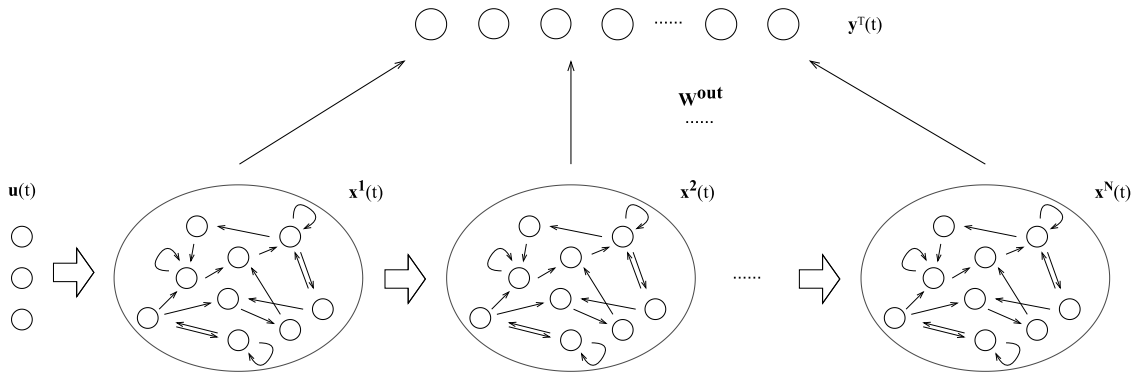


FIGURE 6. Architecture of DeepESN. For the readout organization for DeepESN, at each time step the reservoir states of all layers are used as input to the output layer. Figure modified from [49].

in 2022 and Hodrick–Prescott filter [93] in 2023. These model are claimed to have high prediction performance in time-series prediction tasks with relatively low training cost.

b: DEEP FUZZY ESN

In 2019, [48] proposed a novel deep ESN model with fuzzy tuning called Deep Fuzzy ESN (DFESN). Here, two reservoirs are stacked, where the first reservoir is applied for feature extraction and dimensional reduction, and the second one is used for feature reinforcement based on fuzzy clustering. In other words, the output of the previous reservoir was extracted as features for the next reservoir input, followed by a feature reinforcing process performed by fuzzy clustering for classification enhancement. In DFESN, back propagation is no longer necessary, since the feature reinforcement process can be considered as a layer-wise fuzzy tuning that replaces the back propagation algorithm with lower computational costs. As claimed by the author, input samples are clustered more easily, thus improving the final classification performance.

2) ESN WITH EVOLUTIONARY ALGORITHMS

a: MULTI-LAYERED ECHO STATE NETWORK AUTOENCODER

Autoencoder (AE) is a type of common feed-forward network for dimensionality reduction and feature detection, in which non-linear transformations are performed in each hidden layer to regenerate a new effective data representation from the originals. This technique was introduced to RC area by [94] as the first recurrent and non-gradient descent-based AE in the literature. In [94] an autoencoder was implemented by using multilayered ESN, with a bi-level evolutionary algorithms for optimizing the network architecture and weights. Particularly, PSO was applied for the bi-level optimization, where the first level is the architecture determination and the second one is the weights optimization. Classification results on various benchmarks showed that the performance of the evolved model is improved compared with the conventional ESN as well as other CNN or SVM based models.

b: COMPETITIVE SWARM OPTIMIZER

Pre-training an ESN using PSO introduces some extra hyper-parameters, which are usually determined empirically. Furthermore, when dealing with high-dimensional optimization tasks, PSO is likely to experience stagnation or premature convergence [50]. To address this, [50] designed a deep ESN model with a competitive swarm optimizer (CSO) and used it for fault diagnosis—a precise classification task. CSO avoids the problem of optimizing too many parameters at once in PSO with its powerful particle update rule: the particles are updated by evaluating a pre-defined fitness function, and the winner particle will go straight into the next iteration. For the implementation, CSO is combined with a local search technique to further optimize the deep ESN structure. The work shows that deep reservoir networks based on evolutionary algorithms are suitable not only for time series prediction but can also be used to deal with classification problems with adequate results.

3) OTHER TYPES OF ESNs

a: NON-LINEAR FUNCTIONS READOUT

As mentioned earlier, single reservoir ESN may not be able to create rich enough non-linear dynamics. Reference [91] proposed a new method called Non-linear ESN based on non-linear functions and successfully decreased the internal states of the network while increasing dynamic complexity, thus reducing the computational load. Specifically, recall that $x(n)$ is the internal reservoir state vector (x for short), while in this method, it is replaced by a non-linear function:

$$x_{NESN} = f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n. \quad (18)$$

where a_n being the constant. With this modification, non-linear complexity and the learning capability are increased, which results in a higher accuracy in time series forecasting. Moreover, as this method remains simple structure design, it does not require extensive training, parameter tuning or complex optimization process.

b: SMALL-WORLD TOPOLOGY

Small-world (SW) network was first proposed by [95]. For a regular topological network, each node is usually connected

to its neighboring nodes. For the connection to other randomly chosen nodes (not adjacent), we denote the connection probability as p , where $p = 0$ remain regular topology, $p = 1$ remain random topology and $p \approx 0.1$ as the SW topology (see Fig. 18).

To further investigate the echo state property and learning performance of ESNs, [96] presented an SW based ESN (SW-ESN). In this study, the input and output nodes are segregated, and the reservoir remains as an SW topology; that is, neurons connected to the input are different from neurons connected to the output. Experiments showed that the SW topology enables the input to flow to the output nodes, and the cluster organizations of the topology guarantee a larger range of echo state property, thus improving the robustness and learning performance of the ESN.

C. RECENT TRENDS OF LSM-BASED RC

In this section, recent trends of LSM-based RC models will be reviewed.

1) SPIKE TIMING DEPENDENT PLASTICITY (STDP)

STDP is a local unsupervised self-organizing learning rule based on Hebbian learning [69]. The main idea of STDP is that if the firing neuron A tends to induce/inhibit spikes from another neuron B , the synaptic connection w from A to B is likely to be potentiated/depressed [47]. In other words, the synaptic connection w from A to B is potentiated if a causal order (i.e., the presynaptic neuron fires before the postsynaptic neuron) is observed, or depressed otherwise. As a spiking neural network, LSM was shown able to be trained by this adopted learning rule in an online learning manner, and therefore significantly reducing memory storage load and computational cost, as well as making LSM becomes more hardware-friendly for physical implementations [69], [70]. Specially, a recent hands-on implementation of LSM based on [97] is realized using NEST simulator [98].

2) EVOLUTIONARY ALGORITHMS

Similarly to the small world topology, the percentage connectivity indicates the connection probability between neurons within the liquid. Finding a proper percentage of connectivity is then an important factor for improving the accuracy of LSM. Too high/low connectivity will harm the performance, which also suggests that there is an optimal connectivity for a given task. Particularly, [99] proposed an evolutionary algorithm to optimize the number of neurons and percentage connectivity on a single liquid. Meanwhile, [100] used a covariance matrix adaptation evolution strategy to optimize three parameters, i.e., percentage connectivity, weight distribution and membrane time constant in one liquid. However, [101] pointed out that the above algorithms “only perform parameter optimization in a single liquid and do not optimize the architectures of LSM,” and proposed a Neural Architecture Search (NAS) based framework to optimize both architecture and parameters of LSM model. Furthermore,

the presented framework introduced a three-steps search for LSM, where the first step is architecture optimization, the second step is the search for the number of neurons and the final step is parameters optimization. Experimental results show that the proposed model achieves comparable accuracy on classification tasks of the three datasets (i.e., MNIST, Noisy MNIST and FSDD).

D. DYNAMICAL SYSTEMS

The key essence of RC lies in its approach to use large, fixed random networks, i.e., the *reservoirs*, exhibiting a rich set of dynamical behaviors. These reservoirs provide high-dimensionality and memory in which input data can be transformed and stored, making it easier to model complex temporal patterns and perform machine learning tasks. On the other hand, dynamical systems, characterized by their temporal evolution and behavior, offer crucial insights into the working of these reservoirs, shaping how we understand and optimize them. In the following subsections, we show how RC provides a practical framework for studying dynamical systems, while the theories underlying dynamical systems give a solid mathematical foundation to the operation of RC.

1) SINGLE-NODE TIME-DELAYED FEEDBACK RESERVOIR

a: DEFINITION

Classical RC models process a low-dimensional temporal input through a high-dimensional reservoir state space. This high-dimensional state space is achieved by creating many randomly connected artificial (ESN) or spiking (LSM) neurons as a reservoir, so as to receive input data coupled into the reservoir with weight (synaptic) connections. It turns out that RC can be implemented by using only a single hardware node. In 2011, the concept of using a single dynamical node as a reservoir to generate a high-dimensional space, was introduced in [102]. In contrast to network-based reservoirs consisting of many neurons such as ESN and LSM, single node RC shows great simplicity especially for physical implementation. Theoretically, the proposed delay system refers to non-linear delayed feedback system, which is a type of dynamical system described by delay differential equations given by [105]:

$$\frac{dx(t)}{dt} = F(t, x(t), x(t - \tau)), \quad (19)$$

where t and x are continuous time and state variables, respectively. F is the function of this system with $\tau > 0$ the delay factor. This system is usually employed by using electronic circuits with a feedback loop (see Fig. 7A). The workflow is as follows:

- A low-dimensional input signal (e.g., one-dimensional temporal signal) is first processed using a time-multiplexing masking function, and modulates the state of the node.
- The single node samples the pre-processed input states and holds them for a delay period of τ .

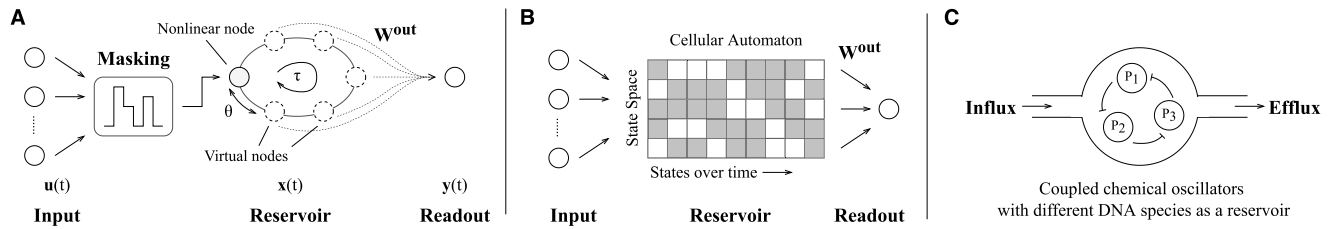


FIGURE 7. RC with dynamical systems. (A) Single-node time-delayed feedback reservoir [102]. (B) Cellular automaton [103]. (C) DNA oscillator [104].

- Meanwhile, N virtual nodes are set that equally divide τ with the time interval of $\theta = \tau/N$, forming a delay line.
- When the signal reaches the end of the delay line, it is fed back into the node, influencing the node's future states.
- The current state of the node and the states stored in the delay line are then fed to the readout layer with trainable weighted connections.

b: EXTENSIONS

The single-node delayed feedback RC was experimentally investigated on spoken digit recognition task and NARMA time series prediction task [42], [88]. Some variations of delayed feedback structure were proposed, such as (1) Different ensembles of delay-based RC with several delayed neurons by [106]. and (2) two circular connected time-delayed based reservoirs with a longer delay line by [107].

2) CELLULAR AUTOMATON

Another type of dynamical system that can be used as an RC is the cellular automaton (CA). CA is a collection of a cell-grid of specified shape that evolves (interacts with its neighbors and changes its state) through a number of discrete time steps [108]. CA's new state is determined by a pre-defined set of update rules and neighboring cells, resulting in a rich dynamic (see Fig. 7B).

The concept of CA system was further extended to RC. A series of work on building CA-based RC was made by [109] and [103], where an evolution rule was introduced to create a space-time volume in the automaton state space (i.e., the reservoir). The proposed CA system was reported to be suitable for combining other types of discrete dynamical systems such as Boolean logic and symbolic processing [110]. Recent studies on CA-based RC also include modifications and extensions of network architecture [111], [112]; as well as improvements of evolution rules in CA [113]. More recently, [114] explored the advantages of critical spacetime patterns generated by elementary cellular automata (ECAs) in reservoir computing, specifically focusing on the distractor's length in time series data and proposing asynchronously tuned ECAs (AT-ECAs) to generate universally critical spacetime patterns.

3) COUPLED OSCILLATORS

a: DEFINITION

RC models have been successfully applied in research areas such as machinery, chemistry, biology and physical systems (see section V). Among these implementations, many RCs

are built using coupled oscillators. A general representation of the dynamics of coupled oscillators is given by an ordinary differential equation:

$$\frac{dx_i(t)}{dt} = F(x_i(t)) + G(x_1(t), \dots, x_N(t)), \quad (20)$$

where $i = 1, \dots, N$ is the index of a total of N coupled oscillators, $x_i(t)$ is the state of i^{th} the oscillator at time t , F and G are an isolated function and a coupling function. The rich dynamics are provided by each oscillator and the interactions between them [23]. In the following, we aim to review studies that utilize coupled oscillators as a building block of RC. Additionally, we will discuss oscillation mechanism of brain in Section VI.

b: MECHANICAL OSCILLATORS

The first category of RCs using coupled oscillators is based on mechanical oscillators. Reference [115] built a network with *anharmonic* (i.e., non-linear) oscillators, where the components include masses coupled linear or non-linear springs, making the system power-efficient to solve a bit-stream computation task and a spoken words classification task.

c: DNA OSCILLATORS

In the field of molecular computing, a deoxyribonucleic oscillator (DNA) reservoir was first proposed in [104]. This RC consists of coupled deoxyribozyme based oscillators. Specifically, a microfluidic reaction chamber was used to construct a reservoir, since different DNA species can interact (see Fig. 7C). Here, the microfluidic reaction chamber is a specific kind of chamber used to carry out chemical or biological reactions under well-controlled conditions. Rich transient dynamics were then generated in the reservoir, where the reservoir state consists of the time-varying concentration of various species inside the chamber. A signal-tracking task was then performed by using a reservoir with three DNA species that exhibits oscillatory behavior. Recent developments of DNA oscillators include a random chemical RC model [116], where the random chemical circuits (i.e., DNA strand displacement) provide complex non-linear dynamics, making them suitable for RC implementation. A novel RC using DNA oscillators was reported in [117] which solves the problem of the lack of readout layer in the previous work [104], and was then demonstrated for the handwritten digit recognition and a second-order non-linear prediction.

d: CHEMICAL REACTION NETWORKS (CRNs)

Related to the DNA oscillators, the chemical reaction networks also show the capabilities to RC implementations. One of the initial studies on CRN-based RC model was presented in a presentation [118], from which the reservoir dynamic is given by a set of ordinary differential equations (ODEs), while the readout layer is to learn the Hamming distance between input bit-streams. In 2022, the author further proposed a chemical RC for single stranded DNA (ssDNA) analysis [119]. Additionally, [120] used a modular framework to implement a RC model. The main advantage of this work, compared with the previous DNA oscillators [104], is that molecular computing allows changing the size of CRNs on-the-fly. Another new chemical RC architecture was proposed by [121], where the reservoir was implemented through electrochemical reactions. Also, it is reported that the Polyoxometalate molecule (POM) in this chemical RC increases the diversity of the response current and thus improves their abilities to predict periodic signals. POM-based RC was further integrated with the so-called *single-walled carbon nanotubes* as a random dense network [122]. Adequate results were obtained in tasks including waveform reconstruction, non-linear autoregressive modelling and memory capacity testing.

e: OTHER RCs WITH OSCILLATORS

It is reported that oscillatory behavior can be restricted to the phase domain [123]. This makes it possible to apply phase oscillators that exhibit rich dynamics to RC [124]. A RC using two coupled relaxation oscillators built on VO_2 switches was reported [125], where the oscillators show high order synchronization that allows simulating the XOR operation. Besides, RC can also be implemented by using spin-torque nano-oscillators in neuromorphic computing [126]. See Section IV-E for detail of spintronic RC.

E. PHYSICAL RC

Recall that the key feature of RC models is to transform sequential/temporal inputs into a high-dimensional non-linear dynamical space (i.e., the reservoir). If the reservoir provides rich enough dynamics, the desired output can be read out by using simple learning methods such as linear regression (see section III for details). Therefore, in principle, any kind of “non-linear, high-dimensional dynamical systems which satisfies some conditions”, has the potential to be a reservoir.

In particular, RC has become popular in a wide range of research fields focusing on hardware design; that is, recent trend of RC implementation have shifted to many domains of physical reservoir computing (PRC) such as optical systems, neuromorphic devices, chemical reactions, quantum computing, to name a few.

Several reviews have tried to organize this highly interdisciplinary topic of PRC. A comprehensive overview of recent PRC implementations was reported in [23], while [161] focuses on the recent advances in photonic RC. A book

series with detailed and special issues on designing PRC was published [45]. In the following sections, we briefly outline the recently proposed PRC models for completeness. Readers may refer to the articles above for a more systematic and theoretical understanding of PRC.

1) ELECTRONIC RC

a: ANALOG CIRCUITS

Various electronic circuits are potential components for hardware RC. Previously, we discuss the single-node time-delayed feedback RC that was first proposed by [102] and [127]. In fact, this type of RC can be formed by electronic circuits with other digital hardware elements [162], [163]. Additionally, a single-node RC implemented by using spiking circuits was also reported [130], [131]. The advantages of analog circuit based RC include (1) less hardware requirements and (2) power efficiency in spiking implementations. Therefore, RCs based on analog circuits were successfully applied to tasks including (1) spoken digit recognition and memory capacity estimation [127]; (2) time series prediction and ECG signal processing [128]; and (3) non-temporal non-linear tasks [129].

b: FIELD-PROGRAMMABLE GATE ARRAY (FPGA)

FPGA board, as a hardware friendly element, has been proven to be suitable for RC implementations. An early attempt at the combination of RC and FPGA board was reported by [164]. In 2014, an FPGA board with stochastic logic was used to implement RC for non-linear time series prediction task [165]. The offline learning in the work above was further modified to an online learning scheme [132], where the units in the reservoir exhibit sigmoid activation and learn with a gradient descent algorithm. In terms of spiking implementation, the conventional LSM models were successfully built on FPGAs in an early research [133] (see Fig. 8A), as well as some models featured with parallel processing [134] and STDP learning rule [166]. In fact, FPGA-based RC have shown (1) better re-configurability; (2) much faster processing speed; (3) less energy costs compared to general CPUs; and (4) more biologically plausible (SNN-based LSM). Several benchmark tests were taken, indicating FPGAs based RC can be applied to (1) channel equalization problems [132]; (2) image and isolated digit recognition [133], [134]; and (3) short input and waveform patterns classification [135], [136].

c: MEMRISTIVE RC

A memristive device, or memristor, is a type of passive circuit element that maintains a relationship between the time integrals of current and voltage across a device. Some studies of RC focus on using memristive elements. Here, the property of memristive elements is different from other circuit ones, since they vary the resistance depending on the current flow at different times. There are two main types of memristive components suitable for RC. The first type is based on neuromemristive circuits. Specifically, memristors

TABLE 2. Types of physical implementations of RC: components and applications.

RC Type	Name	Components / Methods	Benchmark tasks / Applications
Electronic RC	Analog Circuits	1. Various electronic elements with digital hardware. 2. Spiking circuit implementations.	Spoken digit recognition [127]. Memory capacity estimation [127]. Time-series prediction [128]. ECG signal processing [128]. Non-temporal non-linear task [129]. Efficient spiking implementation [130], [131].
	FPGAs	1. FPGAs board with stochastic logic. 2. Recurrent SNN on FPGA. 3. Parallel neuromorphic hardware.	Channel equalization problems [132]. Image and isolated digit recognition [133], [134]. Short input and waveform-patterns classification [135], [136].
	Memristor	Neuromemristive components: 1. Double crossbar arrays. Memristive devices without neurons: 1. Random memristor networks. 2. Memristor with volatility. 3. Memcapacitors. 4. Atomic switch networks.	Time-series prediction [137]. Waveform pattern generation/classification [138]. Associative memory task [138]. Image recognition [139].
Photonic RC	Spatially distributed Optical nodes	1. Semiconductor optical amplifiers (SOAs) with digital masking. 2. Photonic crystal platform. 3. Nodes with free-space optics principles.	Optical packet header identification [140]. Spoken digit classification [141]. Logical function prediction [142]. Waveform prediction [143]. Memory capacity estimation [144].
	Time-delayed feedback loop	1. Opto-electronic feedback loop. 2. All-optical reservoir. 3. Coherently driven passive cavity.	Non-linear channel equalization [145]. Spoken digit recognition [145].
Spintronic RC	Spin-torque oscillators	Magnetic tunnel junction.	Short-term memory task [146]. Macro-magnetic simulation [147].
	Spin wave	Thin Yttrium iron garnet film between a magneto-electric coupling layer and a conductive substrate.	Temporal XOR problems [148].
	Magnetic skyrmions	1. Nanoscale magnetic vortex. 2. Skyrmion fabrics. 3. Magnetic skyrmion memristor.	Image classification task [149]. Handwritten digit recognition [149].
	Dipole-Coupled Nanomagnets	1. Magnetic nonodots array. 2. Magnetic random access memory (MRAM) technology.	NARMA10 task [150]. large-scale RC implementation [150].
Mechanical RC	Mass-spring-damper systems	1. Mass-spring-damper simulating softbodied robots. 2. Soft robotic arm (octopus). 3. Pneumatically driven robotic arm.	Active shape discrimination [151]. Learning to emulate timers-delays and parity [152], [153]. Robot crawling [154]. Foraging learning task [155].
	Sensors	State Weaving Environment Echo Tracker (SWEET) sensing.	Ion concentration analysis [156].
	Tensegrity robots and Central pattern generator	1. Tensegrity-based structure with tensile elements and compressive elements. 2. Spiking-based reservoir acting as CPG.	Locomotion and sensing tasks [157].
Quantum RC	Quantum circuits / computers	1. Analog quantum dynamics under a time-dependent Hamiltonian. 2. Nuclear magnetic resonance (NMR). 3. Quantum circuits with quantum gates.	Quantum-chaotic system-implementation [158]–[160].

are used to model the synaptic plasticity between neurons. Reference [137] successfully built an ESN based on memristor (see Fig. 8B), yet the performance was worse than the conventional ESN in terms of time-series prediction task. Other studies include using double crossbar arrays as reservoir in ESNs [167] and LSMs [168].

Another branch of studies has shown that memristive devices without neurons can also generate rich non-linear dynamics for RC implementations. The first attempt was made by [138] for a wave pattern classification task, where a memristive topology was applied as a reservoir (Fig. 8C).

In 2017, another RC model using dynamic memristors for hard digits recognition was reported [139]. Other memristive networks were also explored to have potential of constructing RC, such as (1) random memristor networks [169]; (2) memristor with volatility [170]; (3) memcapacitors [171] and its hierarchy extension [172]; and (4) atomic switch networks [173].

2) PHOTONIC RC

Optical computing is another paradigm suitable for RC implementations, where the complex non-linear and

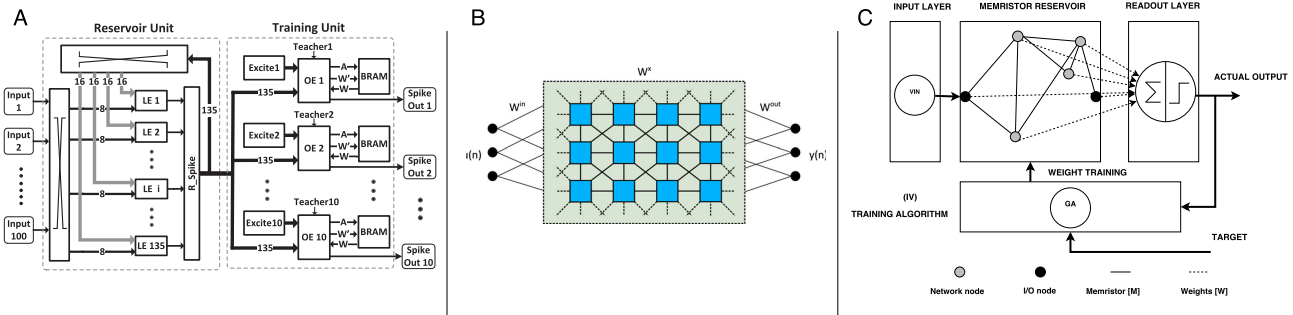


FIGURE 8. Electronic RC. (A) Hardware LSM architecture using FPGAs. BRAM refers to block RAM, where W and W' are the old and updated synaptic weights, respectively [134]. (B) RC implemented by memristors with local connections [137]. (C) Memristor-based reservoir computing using GAs for training [138].

high-dimensional dynamics can be achieved in the intensity and phase of the optical field. A wide range of studies aim to uncover this specific area of RC. In principle, there are two main directions of photonic RC implementations: (1) spatially distributed optical nodes, and (2) time-delayed based photonic RC. For more detailed investigation and discussion, please refer to two comprehensive reviews in [161] and [23].

a: SPATIALLY DISTRIBUTED OPTICAL NODES

It has been realized that the fixed and randomly connected topologies in the conventional RC models (e.g., ESN and LSM) can be implemented by spatially extended photonic networks using spatially distributed optical nodes. Perhaps the first optical RC was proposed and subsequently developed in [174] and [140]. Here, an on-chip network of semiconductor optical amplifiers (SOAs) was constructed to efficiently compute the \tanh function in the reservoir. Later, a digital masking approach was proposed to overcome the short time delay and high operation rate in the previous photonic RC [175]. From 2014 on, several techniques based on optical nodes were reported, including (1) photonic crystal platform [143], [144]; and (2) nodes with free-space optics principles [176]. In fact, optical nodes RC are not only operating with lower power consumption, but more importantly, they are extremely fast in computation. In terms of benchmark tasks and applications, RC with optical nodes were simulated and applied to (1) optical packet header identification and spoken digit classification [140], [141]; (2) logical function prediction [142]; (3) waveform prediction task [143]; and (4) memory capacity task [144].

b: TIME-DELAYED FEEDBACK RC

In section IV-D1, we review RC with single-node time-delayed feedback loop. After the first electronic time-delayed reservoir proposed by [102], optical devices were quickly used to implement such systems [89], [177]. In specific, these pioneering systems applied opto-electronic feedback loops, where the optical parts with long fiber provide non-linearity and time-delay while the electronic parts play the role of input processing and output extraction [161]. Later, the electronic parts of the opto-electronic based RC were replaced by active

optical devices (i.e., SOAs or fiber coupler) [178], forming an all-optical delayed based RC. Moreover, it is reported that a significant improvement of high-speed, low-consumption can be achieved by using passive devices (see Fig. 9D). An example is a RC with a coherently driven passive cavity proposed in [145], in which a simple linear fiber cavity was used as a reservoir to solve tasks such as non-linear channel equalization and spoken digit recognition with remarkable performance.

3) SPINTRONIC RC

Spintronics, or spin electronics, is a branch of physics (particularly condensed matter physics) and nanotechnology that uses the intrinsic spin of the electron and its associated magnetic moment, in addition to its fundamental electronic charge, in solid-state devices. It is reported independently that several spintronic elements can be candidates for RC implementations, including but not limited to spin-torque oscillators, spin waves, magnetic skyrmion; and Dipole-coupled nano-magnets. Spintronic RC were reviewed in [23], as well as in [146] specially for spin-torque oscillators.

a: SPIN-TORQUE OSCILLATORS

The first experiments of building a spintronic RC can be found in [179] and [180], where the spin-torque oscillators were used to provide non-linearity. Here, the so-called magnetic tunnel junction is the key component and is used as a reservoir. The advantage of spin-torque oscillators is that the whole neural network can be emulated by the fast magnetization dynamics generated by simple components. The performances of spin oscillators based RC were evaluated with short-term memory estimation experiments, and were quantified by macro-magnetic simulation [146], [147].

b: SPIN WAVES RC

Another branch of spintronic RC is using spin waves as a reservoir. The first RC based on spin waves was proposed by [181]. A bit sequence input was used and a stripe magnetic domain structure is introduced in a continuous magnetic garnet film where spin waves propagate, thus generating spatially distributed rich dynamics. In 2021, spin waves RC was

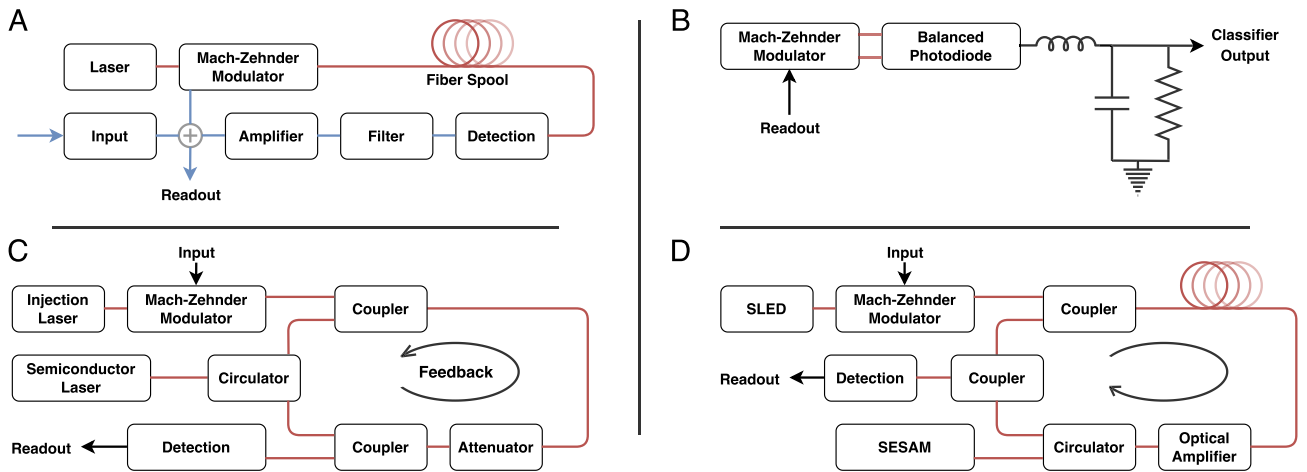


FIGURE 9. RC implemented with electronic and optical elements [161]. (A) Opto-electronic RC system. The optical (electronic) path is depicted in red (blue) color. (B) Readout layer used in photonic RC by [178]. (C) All-optical RC based on a semiconductor laser subject to delayed optical feedback. (D) All-optical RC based on passive devices.

numerically evaluated in [148], and was further shown to be low power consuming [182]. For the applications, spin waves RC in the above studies shows state-of-the-art performances in numerical experiments of temporal XOR problems and memory capacity tasks, once the model is properly tuned.

c: MAGNETIC SKYRMIONS

Magnetic skyrmions are small swirling topological defects in the magnetization texture, in which the stabilization and dynamics depend strongly on the topological properties of skyrmions [183]. In 2018, the first prototype of RC based on skyrmion fabrics was proposed in [184] and [185]. Here, the skyrmion fabrics refer to the phases that interpolate between single skyrmions, skyrmion crystals and magnetic domain walls. Owing to their random phase structures, they are claimed to be suitable for RC implementation. Another application on RC based on magnetic skyrmion is to implement physical RC based on a single magnetic skyrmion memristor (MSM) for image classification task (i.e., handwritten digit recognition) [149].

d: DIPOLE-COUPLED NANOMAGNETS

A novel magnetic nanodots array was reported for a new way of RC implementations [150]. The proposed system is a static magnetic system, in which the dynamics are enriched by increasing the number of nanomagnets. Specifically, the reservoir is formed by dipole-coupled nanomagnets (nodes). Similarly to the conventional RNN-based RC, some nodes were connected to the input, while all nodes were connected to the output and were then read out by the magnetic random access memory (MRAM) technology. A good aspect of nanomagnets based RC is that the magnetic interconnections solve the wiring problem of hardware RNN implementations, showing a great potential to build a large-scale RC system. The system performance was evaluated in the NARMA10 task with adequate results.

4) MECHANICAL RC

a: MASS-SPRING-DAMPER SYSTEMS

Several mechanical RC models have been proposed, including but not limited to soft robots and sensors networks. The idea of employing robot’s body and its dynamics as a computational resource for RC, is originated from the early works [186], [187]. The so-called mass-spring-damper systems are used to replace the conventional neurons in the reservoir (e.g., artificial nodes in ESNs). Specifically, mass-spring-damper systems serve as good models to simulate the biological bodies and soft-bodied robots, where both systems show rich non-linear dynamics that can be used in physical RC implementations. Inspired by the pioneering works above, [151] built a mass-spring-damper system for active shape discrimination. From a more biologically plausible perspective, [188] created a soft robotic arm based on an octopus. The work implies that control can partially be outsourced to the physical body and the interaction with the environment without being processed by the brain or a controller. The authors further made a series of work on the octopus-inspired robotic RC, showing that the implementations can learn to emulate timers, delays and parity [152], [153]. For more applications of mechanical RC, please see Section V.

b: SENSORS

RC had been shown potential for processing sensor data (see section V). Yet some researchers like [156] argued that those reservoirs focusing on sensing are often exploited in a somewhat passive manner, being a separated post-processing component that receives data from sensors. Therefore, [156] proposed the State Weaving Environment Echo Tracker (SWEET) sensing approach, where the RC is considered the sensing element itself for novel sensing applications such as ion concentration analysis.

c: TENSEGRITY ROBOTS AND CENTRAL PATTERN GENERATOR (CPG)

For completeness, tensegrity based robots and CPGs are briefly reviewed here. To illustrate, tensegrity refers to a stable structure that consists of tensile elements connected by additional compressive elements [189], which is considered to be as adaptive and resilient as the biological systems. Another concept, the central pattern generator (CPG), is a neural network that can produce rhythmic patterned outputs without relying on rhythmic sensory or central inputs [190], [191], [192], [193]. On top of the mass-spring systems, [157] first developed a RC based on a tensegrity based structure and applied it to locomotion and sensing tasks. Tensegrity structure can be further used as computational resources for modelling biological structures like human bodies and cells due to its stability. For example, CPG signals were generated by a tensegrity based RC for locomotion [194], [195]. Another CPG-related RC was reported in [196], where the FORCE learning algorithm was used to train a spiking-based reservoir that acts as a CPG. Reference [197] further showed that the biologically plausible tensegrity robots are capable of adaptation to environmental changes.

5) QUANTUM RC

Quantum reservoir computing (QRC) is an intersection of quantum computing and RC. Reviews on this topic can be found in [198], [199], and [200]. The platform of QRC was first proposed by [201]. The idea is to use analog quantum dynamics under a time-dependent Hamiltonian, where the parameters are randomly chosen without tuning. Here, the Hamiltonian is a mathematical operator used to describe the total energy of a quantum system. It plays a central role in the Schrödinger equation, which describes how a quantum state evolves over time. Several improvements of QRC have been explored, including (1) boosting computing power [202]; (2) enhancing memory capacity [203]; and (3) using Nuclear Magnetic Resonance (NMR) [204]. Taking the advantage that any quantum-chaotic system can be used for implementations, QRC are investigated in many other studies, including (1) using quantum circuits with quantum gates [158]; (2) single non-linear oscillator [159]; and (3) dynamical phase transitions [160].

F. OTHER RC MODELS

RC With Non-Linear Vector Autoregression (NVAR): It turns out that RC can be realized as a general, universal approximator of dynamical systems, in which the RNN part contains non-linear activation neurons while the readout layer is a weighted linear sum of the reservoir states. A novel concept was proposed by [205] and [206], that RC with linear activation of neurons followed by a non-linear readout is equivalent to a universal approximator. In this case, such a RC would become mathematically identical to a non-linear vector autoregression (NVAR) machine [207]. By identifying the limitations of random reservoir and taking inspiration

from [207] and [208] proposed a so-called next generation reservoir computing (NG-RC) model based on NVAR. The proposed model was built without the requirements of random matrices and many meta-parameters, and the feature vector of the NVAR was introduced equivalent to the readout of RC. For mathematical details, please refer to [208]. By applying NG-RC to three RC benchmark tasks including Lorenz attractor prediction, the model showed faster computational time while at the same time requiring only a small number of sample and few meta-parameters for training. A possible application would be using NG-RC to create a digital twin for dynamical systems.

V. RECENT APPLICATIONS OF RESERVOIR COMPUTING

As a special type of recurrent neural network, RC avoids the main problem of a difficult, unstable and resource-consuming training process. In the past decade, however, various deep learning algorithms, took advantage of the intricacies of gradient based RNN training with greater computational power and finally became main-stream. This led reservoir computing research into a niche for a few years. As scientists in various research fields have found many new ways of RC implementations and applications (see Table 3), RC have prompted renewed interest among researchers from disparate domains. This section presents a comprehensive review of these recent trends, showcasing the widespread applicability of RC from the realms of engineering and computer science to the diverse fields of physical and social science.

A. BIOMEDICAL

1) ELECTROCARDIOGRAM (ECG)

A modified ESN is applied to *cardiac monitoring* [209]. Specifically, the experimental data includes two classes of ECG signals from MIT-BIH databases with highly imbalanced number of instances. The reservoirs are used as patient-adaptable classifiers. These classifiers can not only produce accurate results, but also show the potential to implement ECG classifiers by using neuromorphic hardware with spiking neural networks. Similarly, [210] used an ESN for *abnormal cardiac activity detection* (Fig. 10A). The main objective is to apply an ECG monitoring model in Medical Internet of Things (MIoT) devices with fast speed and low power consumption. The proposed RC model shows better performance and generalization in AHA and MIT-BIH-SVDM datasets than patient-adaptable methods, and also suggests that RC can be implemented in wearable wireless devices. Besides, in the problem of ECG signal *denoising*, a single-node RC is applied to solve this problem by minimizing the EMG signal that impairs the ECG signal [211].

In 2021, [252] reported a new type of reservoir for *arrhythmic heartbeats* classification. Inspired by the pioneering work on the combination of organic electrochemical transistors (OECTs) and reservoir computing, they implemented dendritic networks using OECTs for real-time classification. In detail, the reservoir is created by using coupled dendritic

TABLE 3. Recent applications of RC in various research fields.

Research Fields	Applications
Biomedical	ECG, EMG and MCG signal processing [209]–[214]. Medical images segmentation and classification [215].
Machinery	Active shape discrimination [151]. Temporal information processing [149]. Robotic crawling [154]. UAVs and sensors control [156], [216]–[221]. Fault diagnosis [48], [50], [222]–[224].
Data Science	Series chunking and clustering [225]–[227]. Similarity learning [228].
Security	Attack detection [229]–[232]. Specific emitter identification [233], [234].
Communications	Optical communications [235]. Network traffic flows analysis [236], [237]. Symbol detection in MIMO systems [238], [239].
Chemistry	Remaining useful lifetime prediction of PEMFC [240]–[243]. Real-time gas concentration prediction- using chemosensors [244]. DNA oscillators and Magnetic skyrmions [104], [116], [117], [120]. Chemical reaction networks [119]–[121].
Environmental	Wind forecasting [91], [245]. Wind power generation [230].
Audio and Speech	Speech recognition [47], [88], [246]–[248]. Music classification [249].
Finance	Stock market prediction [250]. Financial system modelling [251].

fibers. Once these fibers are excited through the electrolyte, they create a strong enough non-linear dynamic of the input signals. The proposed networks show the potential use for *biofluid monitoring* and biosignal analysis with high accuracy, indicating that bio-compatible computational platforms can interact with body and biological analysis.

2) ELECTROMYOGRAPHY (EMG)

Some studies on EMG also apply RC models. Reference [253] proposed a scalable and reconfigurable neuromemristive reservoirs architecture for EEG and EMG signal analysis. A further EMG application using LSM model was proposed by [212], where the EMG signal collected by the surface EMG (sEMG) sensors are classified by an LSM-based neuromorphic hardware. In 2021, another spiking RC model that applies CRITICAL plasticity rule [254] for synaptic connection optimization was proposed for hand gesture recognition [255] (Fig. 10B). The author also proposed a novel approach to evaluate and convert the raw EMG signals to spikes encoding.

3) MAGNETOCARDIOGRAPHY (MCG)

Checking the ECG is not possible for everyone. Alternatively, magnetocardiography (MCG) signals can be detected by measuring the magnetic field produced by the electrical currents in the heart and can be converted into ECG signals. Reference [213] built the first physical RC model for MCG monitoring. Specifically, the noisy sensed MCG signals take as input to the reservoir (i.e., the spintronic sensors, see Section IV-E) and the output is the predicted ECG signals (Fig. 10C). This lightweight RC model is claimed to be much

power-saving and lower memory-required while achieving comparable performance with a deep learning based filtering approach [214]. A similar model was later proposed by [256] to extract ECG signals from MCG signals.

4) OTHER BIOMEDICAL APPLICATIONS of RC

In early research of molecular reservoir computing, the coupled deoxyribozyme oscillators is shown to be a type of reservoir [104]. This refers to DNA reservoir computing (see Section IV). In addition to DNA based RC, a medical image classification with distributed representations on cellular automata RC was reported by [257] (see cellular automata RC in Section IV). Besides, a recent study on spatio-temporal feature learning used RC model for T-cell consistent segmentation [215]. Instead of only applying a single reservoir, the model used multiple reservoirs for image segmentation and classification, where each reservoir focuses on a specific area of the image to obtain local interactions.

B. MACHINERY

1) ROBOTICS

In section IV, we gave a brief review of mechanical RC. The so-called mass-spring-damper systems were built as the first type of mechanical RC by [186] and [187] (see Fig. 11A). This new type of RC was then used for *active shape discrimination* [151]. Meanwhile, [188] made a series of works on the octopus-inspired soft robotic RC (Fig. 11B), showing that the implementations can learn to emulate timers, delays, and parity [152], [153], [188]. Another application of robotic RC was reported to learn and reproduce various end point trajectories by using a new RC-based soft robot

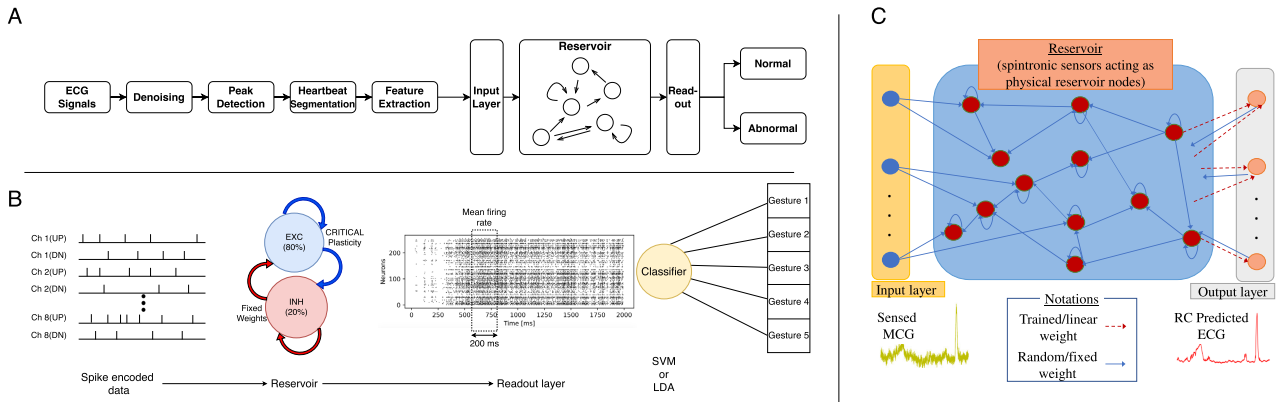


FIGURE 10. (A) Ventricle heartbeat classifier using RC [210]. (B) Gesture recognition using a plastic reservoir (EMG signal processing) by [255]. (C) MCG Monitoring using physical RC with spintronic sensors acting as physical reservoir nodes [213].

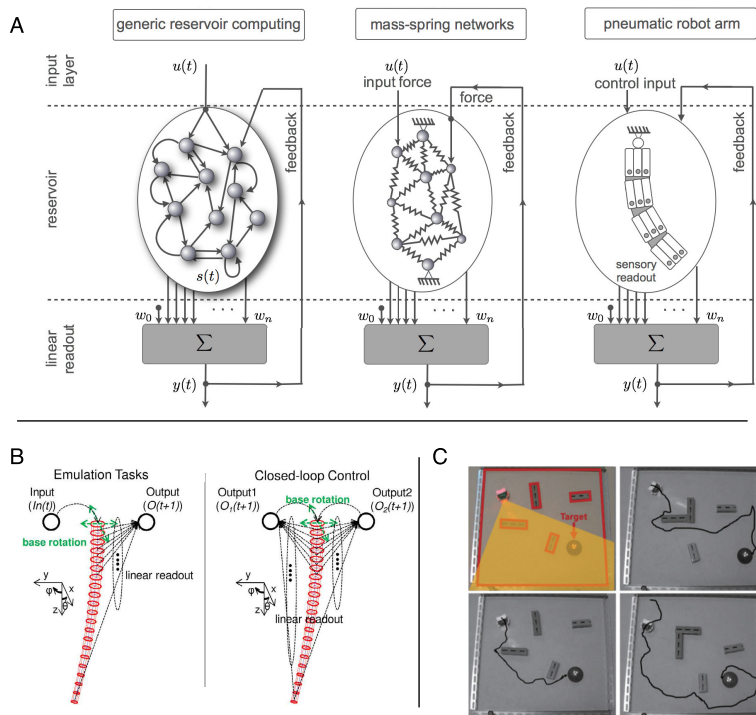


FIGURE 11. (A) Non-linear mass-spring-damper systems as a RC proposed by [186] (middle panel). Pneumatically driven, modular robot arm as a RC proposed by [258] (right panel). (B) A bio-inspired RC using octopus as soft robotic arm by [188]. (C) A foraging learning task [155]. A vehicle robot was placed in an environment with obstacles and was directed toward the goal. The system used FORCE learning to generate a coherent signal output from a living neuronal culture.

system with a highly complex *pneumatically driven* robotic arm [258]. RC has also been applied in robotic crawling by using the *origami* – a traditional play of folding paper into sophisticated and 3D shapes. Reference [154] shows that an origami structure based PRC can be designed to build a soft robotic controller for earthworm-like peristaltic crawling. Reference [155] embedded FORCE learning into a robot to learn in a living neuronal culture (i.e., foraging learning task), in which the robot was placed on square fields with various obstacles and was directed toward the target objects (see Fig. 11C).

2) UNMANNED AERIAL VEHICLES (UAVs)

RC can be applied to UAV systems. Particularly in telecommunications, RC models were used in the so-called cache-enabled UAVs for optimizing *resource allocation* over the LTE licensed and unlicensed bands. The first attempt was to use ESN in such a system to predict each user's content request distribution and its mobility pattern when limited information on the states of users and the network is available [216]. An LSM-based model was further proposed, which can predict more context information of the users and thus improves the prediction accuracy [217].

Another branch of UAV applications includes one that uses a deep ESN based reinforcement learning algorithm for UAV path planning by [218]. In this system, each UAV uses an ESN to optimize paths and learns transmission power at different locations. Besides, ESN can also be used to control rotorcraft UAVs, which outperforms linear models in robustness to disturbances [259]. In 2021, [260] proposed a method for controlling flapping-wing UAVs in different wind directions, where strain sensors are applied to measure the wind movements, and a physical RC is used as a classifier to recognize the wind stream from the sensor data (see Fig. 12A).

3) SENSORS

RC models can also be integrated into wireless sensor networks (WSNs). Generally, the sensor devices in WSNs are distributed and computationally constrained, and the collected data usually consist of temporal information, which makes RC inherently suitable for embedding on the WSN devices [219]. One of the real-world WSN applications using RC is activity recognition in Ambient Assisted Living (AAL) tasks [220]. Specifically, RC-based multi-sensors were used for feature collection and extraction. The sensed data were then further processed by an ESN which provides a good activity recognition accuracy with low computational costs. In 2021, a bio-inspired in-sensor RC was demonstrated to be effective for classifying short sentences of language [221].

It is worth noting that although RC had been shown potential for processing sensor data, some researchers like [156] argued that those reservoirs focusing on sensing are often exploited in a somewhat passive manner, being a separated post-processing component that receives data from sensors. Therefore, [156] further proposed the State Weaving Environment Echo Tracker (SWEET) sensing approach. Here, RC was considered as the sensing element itself for novel sensing applications such as *ion* concentration analysis.

4) FAULT DIAGNOSIS

Fault diagnosis generally refers to the process of detecting errors in physical systems while attempting to identify the source of the problems. Built on the deep ESN architecture suggested by [49] and [50] proposed evolving deep ESN models for 3-D printer fault diagnosis, with a developed version of particle swarm optimization (i.e., competitive swarm optimizer, CSO). This RC model uses evolutionary optimization and is shown to be state-of-the-art and computationally economic, which is a complement to deep learning algorithms, rather than a competitor. Meanwhile, the same research group proposed another solution for 3-D printer fault diagnosis [48], [224]. Specifically, deep ESNs were used to improve feature extraction performance, where the features were reinforced throughout the hidden layers by using fuzzy clustering as a tuning step. This low computational costing model also provides the optimal solution in all experiments, with a total of 26 different condition patterns in fault diagnosis data.

TABLE 4. Comparison of the Accuracy (%) of recent RC models in image recognition benchmarks.

Model Type	Model	MNIST	Fashion MNIST	CIFAR-10
Standard RC	[267]	99.07%	-	-
	[261]	99.19%	-	-
Hybrid (CNN+RC)	[262]	99.25%	-	-
	[266]	98.71%	86.27%	-
	[263]	98.38%	91.04%	64.49%
	[268]	97.70%	-	-
Physical RC	[172]	73.86%	-	12.96%
	[269]	98.20%	89.90%	-
	[270]	98.08%	-	-
	[271]	98.90%	-	-
Hybrid (CNN+PRC)	[272]	99.03%	-	60.57%

In addition, RC can also be applied to chemical fault diagnosis in the *proton exchange membrane fuel cell* (PEMFC) system. The first RC application in PEMFC system diagnosis was made by [222], where a delayed feedback RC was used to detect four fault types yet in the static operating conditions only. Instead of processing voltage signal in the original data space, a newer variant based on previous models was proposed, which performs abnormal detection in the reservoir computing based model space (current-voltage model) without requiring additional feature extraction [223].

C. DATA SCIENCE

1) IMAGE RECOGNITION

Image recognition stands as a prominent field within computer vision and machine learning. While RC shows impressive performance in finding and generating temporal features, it has also been adapted for image recognition tasks, either alone or integrated with other techniques such as deep networks with convolutional layers. Table 4 shows the performance comparison between different RC models. To illustrate, when using RC alone to deal with 2D or 3D inputs (i.e., images or videos), data are usually flattened into 1D signals prior to feeding them into the reservoir, as discussed in [33]. When combining with convolutional layers, these layers preprocess images and videos, transforming them into intermediate representations which the reservoir can process temporally [261], [262], [263], [264], [265], [266]. Although RC models demonstrate competitive performances with other machine learning methods like CNNs on simpler datasets such as MNIST, their performance significantly declines when faced with datasets exhibiting higher spatial complexity, such as CIFAR-10. This performance gap highlights the inherent limitations of RC in dealing with complex spatial correlations, and underscores the need for further exploration and investigations in this field. On the other hand, studies have proved that convolutional neural networks are likely to misclassify even if small perturbations are added to original samples [17], [18], [19], [20], [21], [22]. Thus, this also highlights a strong incentive for RC-based methods to tackle high-dimensional inputs with strong 2D/3D correlations, as it was shown that higher degrees of nonlinearity in the model are related to more robust neural networks, and nonlinearity is where the RC really shines.

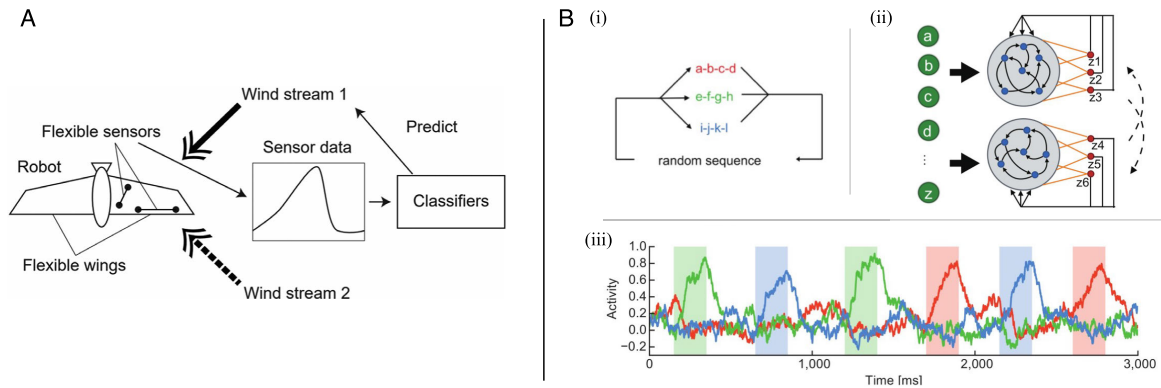


FIGURE 12. (A) A flapping-wing UAV for recognition of the wind direction in which PRC approach was used to classify the wind stream from sensors [260]. (B) A dual-reservoir structure for chunking temporal information streams [227]. (i) Chunking problems example. Three fixed chunks and noise (random sequence between chunks) are repeated in the input sequence with equal probabilities. (ii) The dual-reservoir structure. (iii) Selective readout responses to the individual chunks, which are colored according to their selectivity to the chunks.

2) CLUSTERING

One of the applications of RC in data science is *clustering*. As a special case of clustering, *time-series clustering* introduces several additional issues when compared with static data clustering. For example, the lengths of time-series usually vary, and some of them may be infinite (e.g., video and audio sequences collected from CCTV cameras). Moreover, temporal dependencies in different parts of a particular time-series cannot be captured by making a fixed detecting window (i.e., the dynamical behaviors in sequences contain to both short- and long-term correlation). As a result, similarity measurement techniques, such as calculating Euclidean distance among temporal data, are not inherently suitable for time-series clustering.

Reference [225] proposed the first dynamic clustering algorithm using conventional ESNs. The idea is to apply a clustering method inside every step of the reservoir's state update, where the author claimed that any unsupervised clustering methods can be used in principle (e.g., k-means or any other iterative, partitioning clustering methods). The proposed method overcomes the above-mentioned issues and produces more compact clusters when applied to a hard classification problem of detecting patients with eye disease in eye movements datasets (saccades). In 2020, deep reservoir structure was introduced in time-series clustering [226]. The proposed algorithm was applied to more common benchmark datasets and showed better clustering quality than the previous algorithm and static clustering methods.

3) CHUNKING

Related to clustering problems, some studies have been proposed for sequence *chunking* [273], [274]. Here, the main difference between time-series clustering and sequence chunking is that chunking finds the temporal correlation between state variables, instead of clustering homogenous time-series together based on a certain similarity measure (see Fig. 12B in the upper-left panel). Reference [227] used dual-reservoir networks that supervise each other to mimic

the partner's responses to the given input. Here, a challenge of chunking sequences with uniform transition probabilities, which can be easily processed by humans in basal ganglia, was successfully solved by the proposed model while conventional statistical approaches fail to chunk (see Fig. 12B). This suggests that reservoirs can predict dynamical response patterns to sequence input other than to directly learn transition patterns.

4) SIMILARITY LEARNING

Reference [228] applied RC to learn the similarity between image pairs with limited data. The reservoir here acts as a non-linear filter that projects the images into a high-dimensional state space, in which the state trajectories represent different dynamical patterns that reflect the corresponding relationship of given image pairs. The proposed model was tested on MNIST dataset and images taken from a moving camera. Compared to deep Siamese Neural Networks, this RC model showed significantly better performance in generalization tasks. The generalized combinations of relationships provide robust and effective image pair classification.

D. SECURITY

1) ATTACK DETECTION IN SMART GRID

RC has been proven to efficiently solve *false data injection* (FDI) and to improve the reliability in smart grid systems [229]. The first attempt was to use a modified delayed feedback network (i.e., single-node time-delayed RC) as a reservoir combined with a multilayer perceptron (MLP) as a readout for *single-period attack detection* [230]. This RC model with MLP architecture produces a high attack detection rate (99%) and shows strong robustness in various attack types. Later, the author extended the pioneering work to the more challenging dynamic attack detection in smart grid [231], where a bio-inspired learning rule called *precise-spike-detection* (PSD) [276] is used for spiking reservoir training.

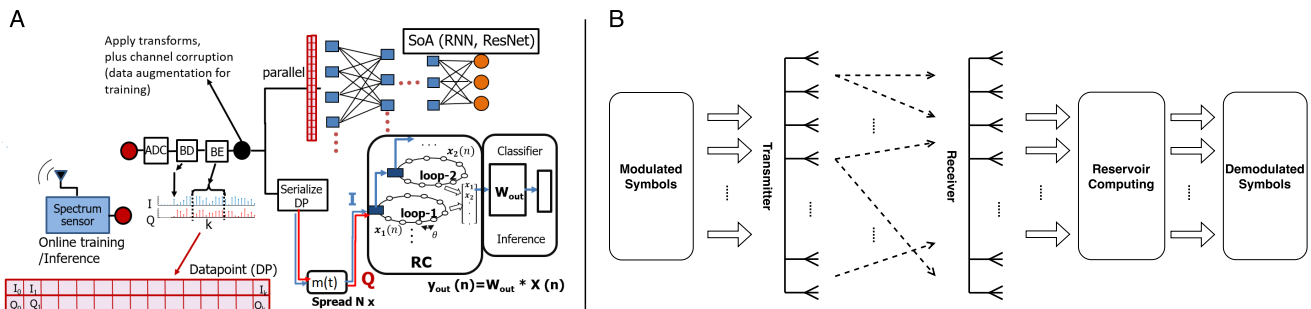


FIGURE 13. (A) Diagram of Specific Emitter Identification (SEI) using RC [234]. A multi-layer time-delayed feedback reservoir structure was introduced to enable the linear classifier for emitter identification. (B) RC used for symbol detection (modified from [275]).

Regarding the attack detection, recent applications of RC include detecting malware and micro-architectural attack, which is reported in [232] using a CMOS-based RC neural network embedded in a 65nm CMOS chip.

2) SPECIFIC EMITTER IDENTIFICATION (SEI)

SEI is capable of extracting rich non-linear characteristics of internal components within a transmitter to distinguish one transmitter from another. Since the fingerprint of SEI cannot be emulated, it is widely used in IoT devices to prevent MAC address spoofing attacks. A reservoir with delay loops for SEI was first proposed by [233] and further adapted to edge computing [234] where the RC architectures include a digital loop (FPGA) and a photonic one (Fig. 13A).

E. COMMUNICATIONS

1) OPTICAL COMMUNICATIONS

In the high speed optical fiber communication systems, RC was applied for *digital equalization*. Reference [235] quantified the equalization performance of the optoelectronics RC. Experiment results show that the optoelectronics RC outperforms traditional equalizers under the same transmission conditions, taking the advantage of its ring topology for better correlation between adjacent data as well as its lower complexity and computational cost.

2) NETWORK TRAFFIC

Reference [236] proposed a method for application identification for network traffic by physical RC, which processes traffic flows as dynamical time series data and enables fast and real-time identification. Another RC application is reported by [237] for road traffic analysis.

3) SYMBOL DETECTION IN MIMO-OFDM SYSTEMS

In wireless communication domains, multiple-input multiple-output with orthogonal frequency division multiplexing (MIMO-OFDM) is a key enabling technology in the 5G cellular network. Symbol detection is an important technique due to the severe non-linear distortion during transmission (Fig. 13B). Thus, an accurate estimation of MIMO-OFDM channel is usually required. The first integration of RC and MIMO-OFDM systems was proposed by [238]. Specifically,

an ESN was used for system modelling and predicting non-linear dynamics, where the MIMO-OFDM channel estimation is no longer necessary. Further, inspired by deep RC architectures [49], [239] extended the existing shallow RC to form a deep neural network [239], [275], which significantly mitigates the frequency distortion.

F. CHEMISTRY

1) FUEL CELLS (FC)

ESNs have shown their effectiveness on *remaining useful lifetime* (RUL) prediction for *proton exchange membrane fuel cell* (PEMFC). Reference [240] developed the first RC model for FC prognostics using conventional ESN. Later, several variants of RC model have been proposed for improving the prediction performance. These include an ESN combined with ANOVA method [241], as well as a multi-reservoir ESN [242]. However, recent research states that the above proposals assume that FCs are operated in constant nominal operating conditions [243]; that is, only the degradation is considered the factor of the deviation of stack voltage. Another open problem is that the prognostic results in long-term experiments show that prediction will become inaccurate when disturbances occur.

2) CHEMOSENSOR

Metal oxide (MOX) based sensors are a common choice for tasks of chemical detection, yet the time response of these chemical sensors is usually excessively slow. It is stated that algorithms based on batch or sequential measurements are not suitable for continuous sensing scenarios. Reference [244] used RC algorithms to overcome the slow temporal dynamics of the chemosensors and applied RC for real-time gas concentration prediction by observing the sensors' time series in response to the changes in the composition of a gas sample. Still, problems were reported, such as the drift of sensor response over time.

3) MAGNETIC SKYRMIONS

RC using magnetic *skyrmions* are reviewed previously in Section IV, where the random phase structures of the skyrmion fabrics are suitable for RC implementation [185].

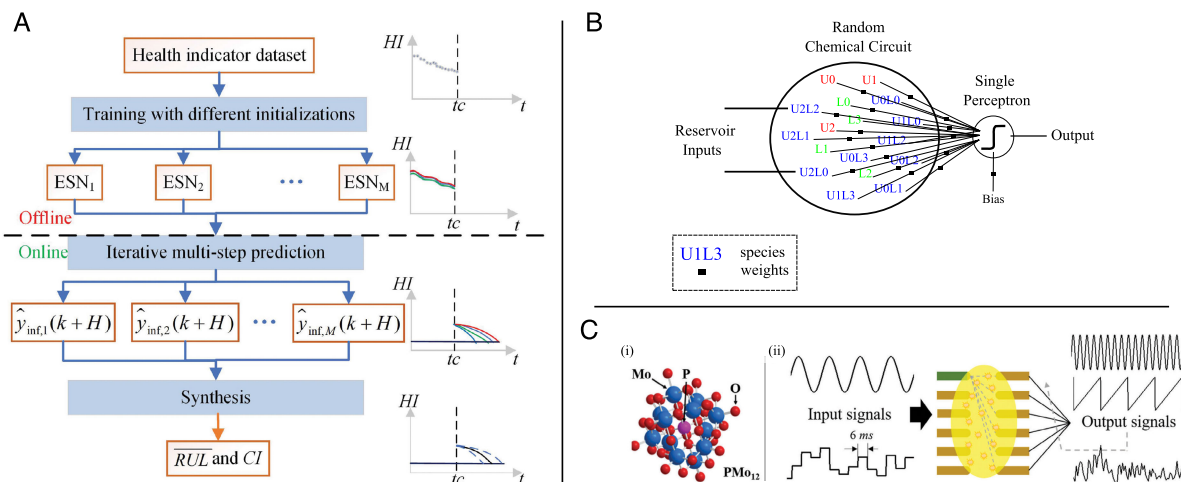


FIGURE 14. (A) An ESN was used to predict the remaining useful lifetime (RUL) of Fuel Cells [243]. (B) A chemical RC implemented by a random chemical circuit with different DNA species [116]. (C) An electrochemical-reaction-based reservoir proposed by [121]. (i) Structure of the Polyoxometalate (POM) molecule. (ii) Testing procedure.

Recent applications in this field is to implement RC based on a single magnetic skyrmion memristor (MSM) for image classification task (i.e., handwritten digit recognition) [149].

4) COUPLED DEOXYRIBOZYME OSCILLATORS AND DNA OSCILLATORS

Taking the inspiration of DNA reservoir computing approach [104], [116] proposed a random chemical RC model, where the random chemical circuits (i.e., DNA strand displacement) provide complex non-linear dynamics, making them suitable for RC implementation (Fig. 14B). The proposed model outperforms the previous deoxyribozyme oscillator RC [120] in short and long-term memory tasks. Another recent RC using DNA oscillators was reported in [117] which solves the problem of the lack of readout layer [104], and then applied to a handwritten digit recognition and a second-order non-linear prediction task.

5) CHEMICAL REACTION NETWORKS (CRNs)

As mentioned earlier, reservoir’s dynamic can be generated by a set of ordinary differential equations (ODEs). An extension was proposed for single stranded DNA (ssDNA) analysis [119]. Besides, [120] used a modular framework for molecular computing to implement a RC model. The main advantage of this work, compared with previous DNA oscillators [104], is that molecular computing allows tuning the size of CRNs. Another new chemical RC architecture was proposed by [121], where the reservoir is implemented through electrochemical reactions since the chemical dynamic is shown to be computing resources (Fig. 14C). As claimed by the author, the Polyoxometalate molecule in the solution “increases the diversity of the response current and thus improves their abilities to predict periodic signals”.

G. ENVIRONMENTAL

1) WIND FORECASTING AND WIND POWER GENERATION

Reference [91] proposed a RC based wind speed and wind direction forecasting model. In fact, the proposed model is a new type of non-linear echo state network, which is discussed in Section IV. Instead of deterministic forecasting, a recent study focuses on probabilistic wind power forecasting [245] by using a time warping invariant echo state network [277]. In addition, the wind turbines were used as a major source of power generation for smart grids, where the delayed feedback RC was applied for attack detection [230].

H. AUDIO AND SPEECH

1) AUDIO PROCESSING

Audio signals cover a wide range of temporal sequences such as speech, sounds and music. In 2009, RC was first applied as a general framework for non-linear audio processing by [278]. Three main potential applications were proposed with simulations, including tube amplifier plugin identification, non-linear audio prediction and music information retrieval (MIR). RC was claimed suitable for non-linear audio processing because of the inherently temporal processing capability. In terms of real-time audio processing, a cascaded discrete-time RC was proposed for black-box system identification [279]. Albeit much effort was made to reduce computation consumption, the cascaded structure is stated considerably more complex to tune than the conventional RC. In 2019, [249] proposed randomly weighted CNNs for music classification. The proposed model shares similarity to RC, where weight connections remain untrained during training.

2) SPEECH RECOGNITION

The earliest RC application of speech recognition was presented in [88]. Here, an LSM-based RC with spiking integrate-and-fire neurons was implemented recognizing isolated digits, where the readout is trained using ridge

regression. The problem of the model is that it requires intermediate data storage for offline learning. Similarly to the work above, [246] trained output neurons using back-propagation based MLPs. Inspired by Hebbian learning, [47] further proposed a variant of Hebbian online learning rule to train an LSM without requiring data storage for speech recognition. In detail, the analog input speech signal is pre-processed by the *Lyon passive ear* model and further converted into spikes by BSA algorithm [280] before feeding into LSM. Other types of RC can also be applied to speech recognition. For example, an ESN combined with extreme kernel machines was used for Arabic speech recognition [247]. In 2020, a RC based on nano-oscillators was also applied to TI-46 database [248].

I. FINANCE

1) STOCK MARKET PREDICTION

A successful prediction of a stock's future price could yield significant profit. An early attempt of short-term stock price prediction was reported in [281], who used an ESN as a basic network with the *Hurst exponent* to select a persistent subseries with the greatest predictability for training from the original training set. Instead of using basic ESN, three RC network structures were investigated in stock price prediction [250], including the small-world topology discussed in an earlier section.

2) FINANCIAL SYSTEM MODELLING

Reference [251] used RC for financial system modelling. In this study, an ESN was first applied to predict a pre-defined financial system behavior. The model was further proved to effectively re-generate only the required data based on limited known information.

VI. RC WITH BRAIN MECHANISMS AND COGNITIVE SCIENCE

A. RESERVOIR IN THE CEREBRAL CORTEX

RNNs have been shown to have rich, complex, non-linear and high-dimensional dynamics. In the cerebral cortex, especially the prefrontal cortex (PFC), massive recurrent connections of neurons were found, and it is progressively recognized that some parts of the brain operate as reservoirs [282]. Moreover, the cortex is shown able to extract the desired outputs (readout) from the high-dimensional neural representations (reservoir). In this section, we review studies on using RC to model the cerebral cortex.

1) DOMINEY'S DECADE-LONG RESEARCH: THE BIRTH OF RC Dominey et al. developed the first RC prototype in a series of neurocognitive studies on corticostriatal systems [13], [28]. During the period of 80s-90s, many researchers were focusing on the characterization of the fast eye movements (i.e., the oculomotor saccade) in the corticostriatal system, which refers to the interactions between cortex and basal ganglia [25]. Particularly, [26] examined the function of

the corticostriatal system by carrying saccade experiments on macaque monkeys. The experiments showed that some neurons (1) have a preferred spatial saccade amplitude and direction; (2) selective to response to a particular sequential order. In 2013, [24] characterized this finding as *mixed selectivity*, which became one of the important principles in RC and cognitive science.

Suggested by the two experiments of the *corticostriatal* saccade system [26], [27], the first corticostriatal RC model was built based on (1) a recurrent prefrontal cortex (PFC) system (i.e., the reservoir), and (2) the reward-related learning in PFC-to-caudate connections (i.e., the readout). Fig. 15 shows the architecture of the model. Since they found that the modification of the recurrent connections are considerably computational costing, they decided to initialize PFC layer (the reservoir) with a mixture of fixed inhibitory and excitatory recurrent connections. The reservoir layer was then connected to the *caudate* or *striatum* to obtain the readout. This pioneering RC model, as the author claimed, can be seen as a dedicated temporal recurrent network (TRN), which shows the inherent capabilities and sensitivity to temporal and sequential structure by providing a rich spatio-temporal dynamic [283].

Recent Extension of TRN: Dominey et al. further proposed a series of works on the previous corticostriatal RC model. These include a combination of RC and neuro-physiological models of language processing [284], [285], as well as a performance improvement of the RC learning algorithm [286]. Reader may refer to a more detailed review of the corticostriatal RC model and its historical developments in a review paper by [287]. In 2013, one of the important cortical activities was obtained in randomly connected recurrent networks (e.g., reservoir) and was then characterized as *mixed selectivity* [24] (see the following section). In 2016, the representational power and dynamical properties of mixed selectivity were investigated by training a RC model to perform a complex cognitive explore-exploit task initially developed for monkeys [288]. By comparing the neural activity of the reservoir and the primate dACC neurons, it is found that not only mixed selectivity was observed in the two types of neurons, but more strikingly, the distributions of neurons were quite similar in terms of the epoch (explore/exploit), the task phase, and the target choice, which strongly supports the argument that the cortex behaves computationally as a reservoir (see Figures 3 and 4 in [288]).

2) RECENT STUDIES OF THE CORTEX

Dimensionality Implies Selectivity: One of the complex neural activity phenomena in PFC and in the model by [13] is that the firing rates of some neuron populations were modulated by the combinations of conditions such as spatial location and sequential order [289]. This cortical activity was then characterized as mixed selectivity by [24] in an object sequence memory task. Specifically, pure selectivity refers to neurons whose responses are selective only to

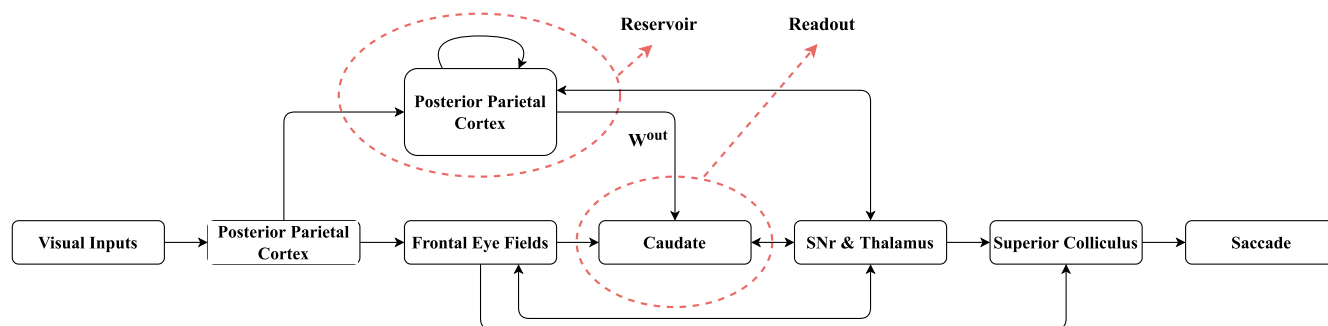


FIGURE 15. The first RC model built based on a recurrent PFC system as reservoir (modified from [28]).

an individual task-relevant aspect, whereas mixed selectivity refers to neurons whose responses are explained by a non-linear superposition of responses to the individual task-relevant aspects. In the object sequence memory task, monkeys were required to watch a sequence of two subsequently displayed images. After that, they had to (1) recognize the two images under distraction (recognition task), or (2) recall the order of the two images (recall task). The *dimensionality* of the neural spaces was then estimated (i.e., the minimal number of coordinate axes needed to specify the position of all points in neurons' firing rate space). It is observed that the dimensionality was higher if neurons having mixed selectivity were included. More importantly, the neural population was estimated to have a higher dimensionality when the monkeys performed correctly on a trial. That is, in the error trials, a collapse in dimensionality was observed, which impairs the ability of downstream readout neurons to produce the correct response. Moreover, [24] and [290] showed that RC models can be compared with a randomly connected recurrent structure in the monkey prefrontal cortex, which can generate high-dimensional mixed selective dynamics to assure the separability in the downstream readout units. The higher the dimensionality of the population coding, the better the performance on the task [289]. Regarding the readout, it was reported that the brain implements mixed selectivity even when it does not enable behaviorally useful linear decoding (i.e., simple linear readout), suggesting that mixed selectivity may be the key of population encoding for reliable and efficient neural representations [291].

Recent studies have shown that mixed selectivity not only plays an important role in PFC, but in other parts of the brain. Reference [292] found strong mixed selectivity in the subiculum (i.e., the area between the entorhinal cortex and the CA1 subfield), where individual neurons respond conjunctively to task-related aspects including position, head direction, and speed. In 2022, mixed selectivity was observed in the thalamus of a weakly electric fish [293]. Here, the mixed selectivity strategy was implemented to encode interactions in the recurrent networks in pallium, which is related to courtship and rivalry in terms of dominance in male-male competition and female-mate selection.

B. NEURONAL OSCILLATIONS

Neuronal oscillations refer to the temporally structured activity generated in mammalian brains, where neurons undergo periodic changes in excitability. These oscillations had been found in neuron assemblies, a concept to describe the behaviors by a population of neurons. In this section, we first present the cognitive science research on neuronal oscillations, and then we discuss the relationship between oscillations and RC models, followed by several examples of the existing research. Contents are partially from [2].

1) NEURON ASSEMBLIES

a: FEED-FORWARD CIRCUITS

It is widely believed that there are two frameworks of processing in natural systems in the cortex [2]: (1) convergent feed-forward circuits and (2) neuronal assemblies. In the framework of feed-forward circuits, specific neurons fire to particular features, and the information is propagated from the former layer to the next higher layer. In this way, higher-level features (e.g., cognitive objects) are extracted through a multi-layered structure. The encoding scheme here refers to spatial encoding, which is well-suited for simultaneously presenting features such as images. However, due to the lack of short-term memory functions, feed-forward circuits are less apt to tackle the relations among temporally segregated events.

b: NEURON ASSEMBLIES

On the other hand, a complementary framework in cognitive brains is the neuronal assemblies. Unlike the feed-forward networks which include explicit layered structures, the assemblies of neurons usually form coupled recurrent networks with non-linear, high-dimensional and self-organizing dynamic [2]. Relations among cognitive objects are translated into the weighted connections between neurons; in other words, high-level features are represented by the amplified reverberations (echoing) of neuronal assemblies. With the reverberating responses, the rich dynamics provided by the coupled recurrent connections have short-term memory (fading memory), and become efficient to handle temporally related sequential events.

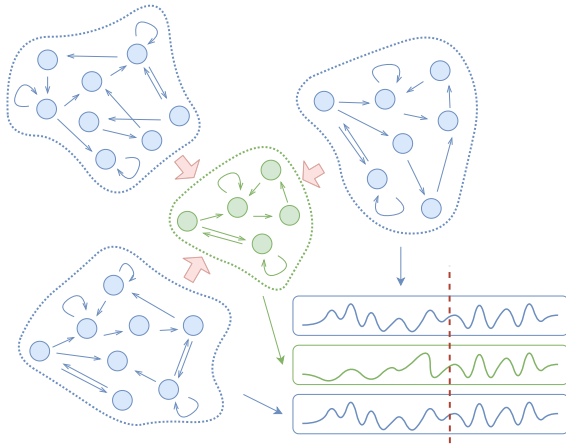


FIGURE 16. The spread of synchrony. Here, the blue neuron populations are already synchronized, and they might entrain other neurons (those in green) to become part of the same overall assembly after some point in time (see the dashed red line), thus resulting from the spread of synchronized activity through lateral connections. Figure modified from [299].

2) BLINDING PROBLEM

a: PROBLEM OF NEURONAL ASSEMBLIES

One of the challenging problems of the neuron assemblies is the blinding problem, which refers to the segregation of simultaneously active assemblies. According to [77], if assemblies were solely distinguished by enhanced activity (i.e., discharge rate) of the constituting neurons, it becomes difficult to distinguish which of the more active neurons actually belong to which assembly. Moreover, if the given objects share some common features and overlap in space (e.g., blind source separation and cocktail party problem), the corresponding feature-selective nodes would have to be shared by several assemblies [2].

b: SOLUTIONS TO BLINDING PROBLEM

A possible solution is multiplexing, in which various active assemblies are segregated in time. Because of the discharge rate of cortical neurons is relatively low (i.e., the integration needs time), multiplexing becomes problematic if only discharge rate is considered for distinguishing assemblies [294]. Therefore, it is only capable in a slow timescale [295].

In the 1990s, Gray and Singer proposed that “neurons temporarily bound into assemblies are distinguished not only by an increase of their discharge rate, but also by the precise synchronization of their action potentials” [296], [297], [298]. They also predicted that neurons that respond to the same sensory object might fire in temporal synchrony, with a precision in the millisecond range. Synchronization by oscillation is briefly introduced in the following section.

3) SYNCHRONIZATION BY OSCILLATION

a: DEFINITION OF SYNCHRONIZATION

The periodic changes of excitability of the neurons are considered neuronal oscillations among different brain areas [2]. It has been identified that these oscillations vary in terms

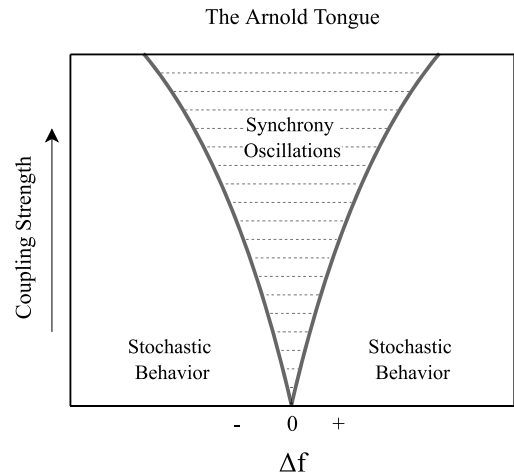


FIGURE 17. The Arnold tongue regime in coupled oscillators. This is a graphical representation modified from [300] to illustrate the synchronization behavior relating the difference in preferred frequency with increasing coupling strength, which results in a ‘tongue’-shape of possible synchronization regimes.

of the frequency, ranging from approximately 0.05 Hz to 500 Hz [301]. According to [302], synchronization by neuronal oscillations has been widely detected across various natural systems, in which different oscillations can “coexist and often synchronized to each other or nested into each other”. This observation is called the spread of synchrony. As an example shown in Fig. 16, the synchronized larger-scale populations can entrain other smaller local assemblies with different oscillations to be overall synchronized. This spread of synchronized activity was then believed to be a reinterpretation of the represented objects [299].

b: ARNOLD TONGUE REGIME

One of the interesting observations of synchronization behavior in coupled oscillators is the Arnold tongue regime. An early experiment by Van Huygens revealed that the beats of pendulum clocks can be synchronized when having the same timber; that is, if the preferred frequencies of the oscillators are similar, weak mutual interactions are enough for oscillatory synchronization [300]. This observation was then summarized as Arnold tongue regime by [303]. As shown in Fig. 17, the coupling strength should increase in order to assure a stable synchronization when the preferred frequencies between coupled oscillators become increasingly different, thus resulting in a tongue-shaped pattern. Synchronization would become unstable if the difference between preferred frequencies exceeds a critical point [300].

4) HOW RC RELATES TO OSCILLATIONS?

Recall that in most of the existing literature, RC models should normally meet several requirements to be efficient and functional for various tasks. These are also considered the properties of a RC model [23], which include (1) High-dimensionality: low-dimensional inputs are mapped into a high-dimensional space, which allows originally inseparable

or temporal inputs to be linearly separable as shown by the Cover Theorem [304]. (2) Non-linearity: non-linear mapping transforms the input into linearly separable reservoir states which can be read out by readout layer. (3) Separation property: a RC model should be capable of separating different inputs into different classes, under small fluctuations or in noisy environments [11]. (4) Fading memory: also known as short-term memory or echo state property. This algebraic property eliminates the effect of initial network condition. In other words, it ensures that the reservoir state is dependent on recent-past inputs (reverberating responses), but not distant-past inputs (responses faded).

In a RC review, [2] made a proposal to link the concept of RC in machine learning and that of neuronal behaviors in the cognitive brain. Readers may refer to the article for details. Based on the proposal, here we aim to discuss how the dynamics of coupled oscillators in mammalian brains could be exploited to accomplish the abovementioned four characteristics in RC.

a: HIGH-DIMENSIONALITY AND NON-LINEARITY

The first fact is that the cortex is reported to have consistent and random high-dimensional oscillations, which refers to the “resting activity” [305]. Meanwhile, it is believed that brains are likely to have an internal model of the external world (i.e., prior knowledge, which can be updated by learning). When the input comes in, the input stimuli “activate” some feature-sensitive neurons, thus making the dynamics of the network collapse into a stimuli-specific substate (e.g., oscillatory synchronization). All of these may suggest that once a reservoir enters a substate (i.e., synchronization), it is likely that the dynamics can be tuned selectively to specific stimuli or generating specific output signals.

b: SEPARATION PROPERTY

As mentioned above, the stimuli-specific substate with rhythmic oscillations, according to [2], “would have a lower dimensionality and to exhibit less variance than the resting activity, to possess a specific correlation structure and be metastable due to reverberation (rhythmic oscillations) among nodes supporting the respective substate”. Note that all these activities, including the stimuli-specific substates, are happening in a high-dimensional state space; thus, different inputs can be well-separated and classified even linearly, similar to the readout of RC models.

c: FADING MEMORY

Moreover, fading memory refers to the short-term memory in RC literature, in which the reservoir state should depend on recent inputs but not distant-past inputs. From an oscillation point of view, there exist several experiments in the cat’s visual cortex, which can support the mechanism of fading memory [306]. As stated in [2], these experiments show that “(1) the information about a particular stimulus persists in the activity of the network for up to a second after the end

of the stimulus (fading memory). (2) Two subsequent stimuli and the order of their presentation can be correctly classified with a linear classifier sometime after the end of the second stimulus, suggesting that the network is capable of performing non-linear XOR operations and (3) Stimulus identity is distributed across many neurons (>30) and encoded both in the rate vector and the temporal correlation structure of the responses”. The above evidence may explain the fading memory from the perspective of neuronal oscillations, suggesting that short-term memory is not only a property of the networks, but also a consequence of the oscillations and reverberations.

In addition, long-term memory is also important, and it is a more complex one. Early experiments showed that the “default” state of the unperturbed, sleeping brain is a complex system of numerous self-governed oscillations, particularly in the thalamocortical system [301], [305]. The content of these oscillations reflects spike sequence patterns created by prior waking experience. Moreover, these oscillations are spontaneously replayed (e.g., during sleeping), leading to an “off-line” synaptic modification. Such replays might be the way to the formation of long-term memory. Overall, a reason why short-term memory rather than long-term memory is one of the necessary requirements for building a RC model might be that we usually keep the random connections in the reservoir fixed without modifications. As a result, RC models generally struggle to form long-term memory, since replays in terms of synaptic modifications are required (i.e., investigating long-term memory is a more challenging task). In fact, some learning algorithms, such as STDP [307] and FORCE learning [35], are trying to modify the synaptic connections, and therefore they are likely to be able to possess long-term memory.

5) EXAMPLES OF SYNCHRONIZATION IN RC

a: MULTIPLE RESERVOIRS

Deep reservoir computing was proposed in [49], in which multiple reservoirs are concatenated together to form a hierarchical network structure. Detailed model descriptions are discussed in Section IV. It has been proved that the deep RC structure can achieve (1) multiple timescale representation, ordered along the network’s hierarchy; (2) multiple frequency representation, where progressively higher layers focus on progressively lower frequencies. According to the Arnold tongue regime shown in Fig. 17, if we keep the coupling strengths at a low level (weak synaptic links, or even zero weight connections), neurons with similar preferred frequencies can be synchronized, at different frequency bands. Therefore, this may explain the reason why multiple reservoirs can achieve these while the conventional ESN cannot, from the perspective of neuronal oscillations.

b: GACTRNN

Gating Adaptive Continuous Time Recurrent Neural Network (GACTRNN), is another research taking the inspirations from

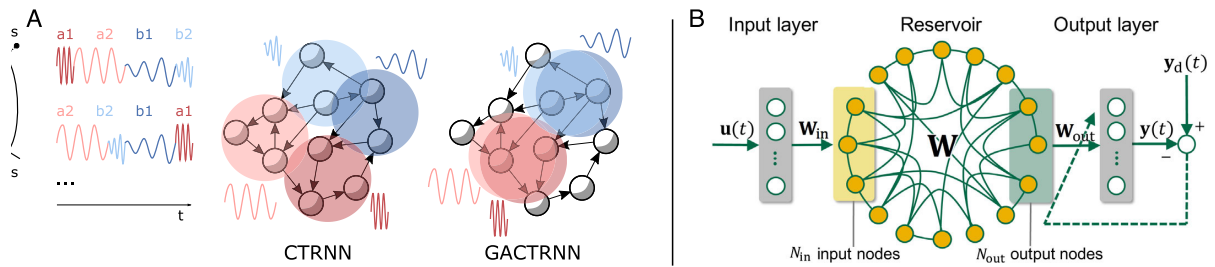


FIGURE 18. Neuroscientists suggest that neural oscillations reveal a highly complex interplay of neural populations and local integrations by coupled oscillations and synchronization, in which multiple timescales in hierarchical processing streams can be achieved. (A) and (B) show examples of how these mechanisms work on a computing and processing level. (A) GACTRNNs [63], a type of neural network where neurons in a population can learn to simultaneously represent temporally different primitives, by changing their timescales. (B) An ESN with a small-world topology. The network rewiring probability is around 10% [96].

neuronal oscillation proposed in machine learning by [63] in 2020. It extended the classic RNN to adaptive timescales RNN, which shares some similarities to reservoir computing models. GACTRNN is claimed to be able to learn to gate its timescale characteristic during activation and thus dynamically change the timescales in processing sequences; in other words, by changing their timescales during processing, neurons can learn to simultaneously represent temporally different primitives (Fig. 18A).

c: SMALL-WORLD TOPOLOGY

An ESN based on the topology of small-world (SW) wiring was proposed by [96] (Fig. 18B). The model incorporated SW structure with RC and further investigated echo state property. It was found that the SW topology plays the roles of both efficient signal propagation and enhancement of the ESP in neural computation. In fact, this idea partially originated from the cortical anatomical connectivity of the human brain. According to [301], “complex brains have developed specialized mechanisms for the grouping of principal cells into temporal coalitions”. In order to reduce the complexity of the connections without excessive wiring, the number of long-range connections between neurons decreases in growing brains; in other words, the synaptic path lengths between distant cell assemblies are reduced, keeping the path lengths short and maintaining fundamental functions.

VII. PERSPECTIVES AND FUTURE RESEARCH

Reservoir computing is becoming increasingly popular due to its simple network structure, hardware-friendly features, low computational cost, and fast training process. These benefits enable RC to extend far beyond machine learning into a wide range of research fields. In this paper, we provide a thorough overview of RC’s history, strengths and weaknesses from the perspectives of machine learning, dynamical systems, physics, biology, and neuroscience. We also summarize recent advanced approaches and architectures for RC optimizations and implementations. Besides, applications of RC are reviewed, from which we have seen how this interdisciplinary idea can be applied in various research areas.

While RC still remains an unconventional computational framework compared to other machine learning techniques like deep learning, its impact can be enhanced by addressing various challenges. Recent developments have unveiled new directions and perspectives for RC, indicating its untapped potential and promising prospects that may even surpass those of mainstream methods. In this section, we present perspectives and discuss the open problems that motivate further research in this field.

A. RESERVOIR DESIGN AND OPTIMIZATION

A consensus view of conventional RC is that initializing a random RNN as a reservoir is not the optimal solution, and that connecting a linear readout with the reservoir limits the generation of the downstream responses. It is also known that neurons in cortical networks in the brain are not randomly connected, while their structures and synapses exploit an evolutionary and developmental process [308]. Recent research, especially on ESNs and LSM, mainly focuses on network structure designs (e.g., deep reservoir), parameter optimizations (e.g., particle swarm optimization) and training rule determinations (e.g., STDP and Hebbian learning). Even if the optimal synaptic weights were discovered, the performance of various concrete tasks would still vary. As pointed out by Jaeger [45], “currently available insights are mostly distilled from experimental studies of timescale profiles or frequency spectra in input data and provide no comprehensive guides for optimizing reservoir designs”. In other words, one should find a way to analyze and abstract both the characteristics of input/output and task specifications, which can be used to design the reservoir dynamics. One possible solution is called reservoir Learning-to-learn (L2L) [308], in which a set of (hyper)parameters of the reservoir are optimized by BPTT for a whole family of learning tasks; note that this shares some similarity to meta learning in machine learning and neuroscience. This L2L method was investigated on LSM models and it can also be implemented by other RC architectures. Moreover, to have better and faster learning, it is possible to train the reservoir by the L2L method even without changing synaptic weights to readout neurons. Nevertheless, whether

the performance takes advantage of other reservoirs is still an open question.

B. EASY-ACCESS TOOLS, CODING FRAMEWORKS AND RECIPES

One pressing open problem in RC is the relative lack of user-friendly coding environments, libraries, and computation frameworks. Unlike the well-developed infrastructure supporting deep learning and other scientific computational paradigms, the coding ecosystem for RC remains relatively underdeveloped and fragmented. As shown in this paper that although numerous models and architectures have been proposed for RC, there is a notable dearth of unified frameworks that researchers can leverage with ease. This poses a significant challenge as it lowers the speed and efficiency of research, requiring additional time and effort to navigate through a variety of individual tools and frameworks, and often necessitates the development of custom code for each research project.

However, this does not imply that no progress is being made towards building a more unified and accessible coding infrastructure for RC. Indeed, we have seen some promising developments over the years. Back in the early 2010s, [33] provided a comprehensive guide on how to implement an ESN, which widely impacts future studies. The author also released a demonstration of coding ESN from scratch in Julia, Matlab, Octave, Python, and R language. In 2012, a toolbox was developed for RC called *Oger* (OrGanic Environment for Reservoir computing) to train and evaluate recurrent neural networks, particularly ESNs and LSMs [309], [310]. From 2017 on, several tools were released in terms of different RC models, such as ESNs, LSMs, and FORCE-based algorithms. The *easyesn* library was released [311], providing a more easy-to-use API for automatic gradient based hyperparameter tuning (of ridge regression penalty, spectral radius, leaking rate and feedback scaling), as well as transient time estimation. Meanwhile, a hands-on LSM implementation using *NEST* simulator in Python was proposed [97], [98], which is considered to be the starting point for new researchers who are interested in spiking-based RC models. Later, another open-source spiking model framework, *Nengo*, was developed for FORCE learning and its variation implementations [81]. In 2018, *EchoTorch* was proposed, and perhaps it is the first Python package to simplify the evaluation and implementation of ESNs and RC [312]. In 2019, a Matlab toolbox for DeepESNs [49] was released that extends the RC paradigm towards deep networks. One of the most common deep learning frameworks, TensorFlow, also supports the ESN layer in 2020 (see TF Addons). More recently, in 2022, [82] present *tension*, an object-oriented, open-source Python package that implements a TensorFlow / Keras API for FORCE learning. Another Julia package for RC is *ReservoirComputing.jl* [313]. It aims to provide a simple and flexible framework to work with ESNs and other models. Additionally, [314] present a Python library that

facilitates the creation of RC architectures, from ESNs and FORCE learning, to complex networks such as DeepESNs and other advanced architectures with complex connectivity between multiple reservoirs with feedback loops.

Despite progress, the full potential of RC is yet to be realized, and the goal of a unified and accessible environment for RC still eludes us. There is a need for more work in this area to ensure that the full potential of RC can be explored and utilized effectively.

C. PHYSICAL RC AND EXTREMELY EFFICIENT HARDWARE

As mentioned earlier, RC has reawakened and gained attention because of the fast development of PRC designs. Unlike conventional RCs that suffer from the information processing speed limit, PRCs can overcome this limit and process massive amounts of data in real-time. We review PRC models that use different physical materials from different areas such as electronics, optics, chemistry, and quantum. As more and more PRC approaches are being proposed, there are several open problems to be solved. For example, some PRCs get rid of the massive recurrent connections in RNNs, yet they are hard to design and tune (e.g., implementing a delayed feedback loop of the single node reservoir is quite a challenging task). Similarly, setting hyperparameters for PRC is not straightforward. In 2021, [316] proposed a framework to evaluate what makes a good computing substrate, providing a new perspective on how to build and compare physical reservoir computers.

Another challenge lies in harnessing the full potential of RC for machine learning applications and achieving highly efficient hardware implementations. Very recently, this issue has been explored and discussed in [317], where the future directions of inquiry are segmented into three categories. Simply put, the first category focuses on the theoretical aspects of RC, aiming to drive efficient model design and ensure reliable RC applications. This includes works like reservoir memory machines [318], consistency capacity [319] and curve fitting abilities [320] analysis. The second category delves into the exploration of novel model designs and applications of RC, with an aim to enhance computational performance and efficiency in tasks related to pattern recognition. Possible solutions include (1) industrial applications such as adaptive practical nonlinear model predictive control [321] and digital twins [322]; (2) integrating RC with deep learning methods such as convolutional and graph neural networks [262], [263], [323], [324], [325]. Lastly, the third category is to keep investigating new architectures and mechanisms in physical hardware that are suitable for RC implementations. Latest research includes a new FPGA-based RC for low-power pattern recognition [326], networks based on the Schrödinger equation [327], and a new cellular automata implementing rule (CA90) [328]. Overall, these studies use a wide range of new hardware, showcasing efficient RC-based methods and stimulating further growth in this research domain.

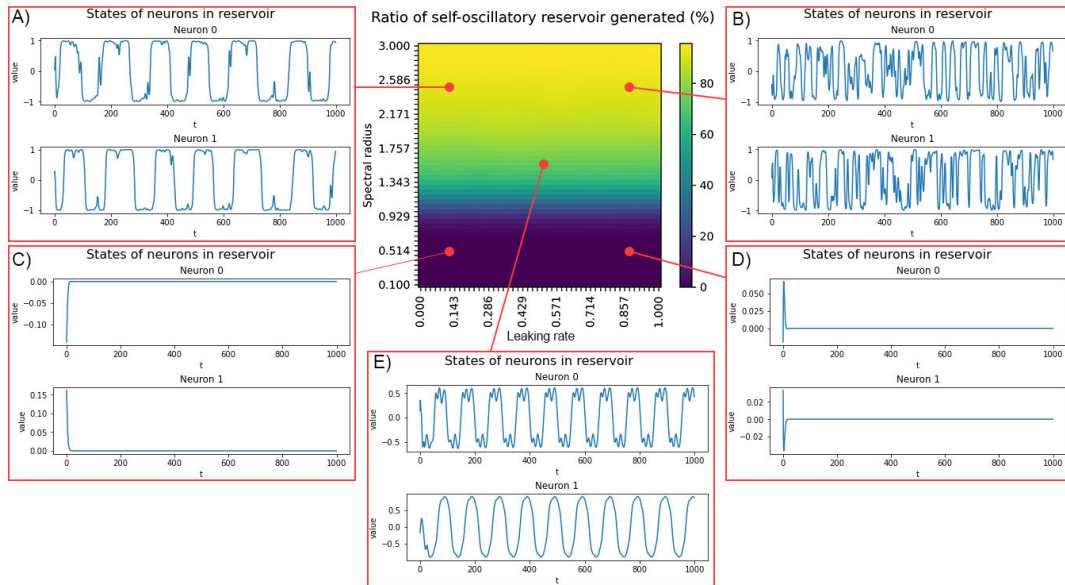


FIGURE 19. Self-oscillatory echo state network (SO-ESN) generating oscillations to reproduce desired waveforms without any inputs (figure from [315], with permission). The figure shows the *ratio* (%) of reservoir that triggers oscillations with regard to leaking rate (x-axis) and spectral radius (y-axis). A stable synchronization by oscillation can be seen in (E), in which neurons 0 and 1 were randomly selected from the population.

D. RC WITH COGNITIVE SCIENCE AND NEUROSCIENCE

In terms of modelling RC in cognitive science and neuroscience, we have reviewed several mechanisms of cognitive brains and tried to bridge the concept of neuron population to RC, such as *mixed selectivity* [24] which reveals that the prefrontal cortex can generate high-dimensional mixed selective dynamics to assure the separability in the downstream readout units. Future research directions include finding new analogies for biological characteristics in RC, allowing for a deeper understanding of brain's and bodies' mechanisms.

The domain of oscillation with synchronization, as previously discussed, is one area warranting further investigation. In fact, progress has been made in using ESNs to produce oscillatory outputs without any inputs, by mimicking the central pattern generators (CPGs) shown to be involved in rhythmic human movement [190], [191], [192], [193]. CPGs are important circuits present in the neural system of live beings. In fact, vertebrates have a spinal cord composed of many of CPG circuits, and it was shown that the spinal cord and mostly CPGs are sufficient for complex locomotion (e.g., walking and running in cats) [329], [330]. In 2023, Tham and Vargas [315] show that even the most basic ESN can be trained to reproduce the trajectory of dynamical systems from simple sinusoidal and square waves to complex Lorenz chaotic time series with high precision, without any external excitation. Fig. 19 depicts the different probabilities of having a reservoir that triggers oscillations in terms of both leaking rate and spectral radius (i.e., the echo state property, ESP), where pure yellow represents the highest probability. Here, it is important to note that the ESP, by design, typically restricts spontaneous oscillation, as it ensures that the reservoir's internal state should eventually lose memory of its initial conditions, hence creating a fading "echo" of past inputs. This can be seen in the cases of Fig. 19C-D, that the

reservoirs' states converge to 0 and remain stable afterward (refer to damped oscillation in [315]). In contrast, as shown in Fig. 19E, oscillations occur when the ESP is most likely not to be guaranteed. Therefore, this study clearly indicates the necessity for additional research, particularly regarding the outcomes and implications when the ESP is not strictly adhered to.

On the other hand, high-dimensionality is another feature shared by the encoding of neuron populations and RC models. Recent studies have also proposed other approaches to understanding the role of high-dimensionality in biological and artificial neural networks. Neural population geometry, for example, is an approach that provides a useful population-level mechanistic descriptor underlying task implementation. Here, the geometry of representation can be represented by high-dimensional neural activity, and it is further observed that the neural activity lies on lower-dimensional subspaces, i.e., the so-called intrinsic dimensionality or neural manifolds [331]. Similar to the reservoir responses, these lower-dimensional subspaces can then be well-separated by using simple classifiers.

Although all of these remain in an early stage of development, the investigations look promising for getting deeper insights into both artificial and natural intelligence.

E. RC FROM AN EVOLUTIONARY PERSPECTIVE

In addition to artificial intelligence and neuroscience, some researchers are exploring RC through an evolutionary lens, which brings a fresh angle to the discussion. Natural systems exhibiting reservoir-like behaviors are prevalent in nature (e.g., nonlinearities in liquids [11], soft robotic in muscles [188], electric and chemical dynamics in neural networks [116], [332], and brain mechanisms [2], [13], [24], [28]). This suggests that such reservoirs might have *evolved*

as advantageous structures for processing complex information, similarly to other computational approaches such as feed-forward networks [1], attractor networks [29], and self-organized maps [333] that have been influenced by and have influenced biology. As stated by [334], however, the current evidence is not as compelling as the well-established similarities between, for example, the structure of the human visual system and deep convolutional neural networks. From an evolutionary perspective, [334] argue that although reservoir-like systems may initially emerge due to their simplicity, their long-term persistence could be hindered by evolutionary trends towards specialization and scaling. In other words, as the reservoir evolves to specialize, integrate diverse sensory information, or scale up, the generalizing properties of the reservoir may become less advantageous compared to highly specialized circuits. This leads to another question: under what conditions might reservoirs maintain their original architecture with redundant dynamics, and when might they evolve towards more specialized configurations? Overall, understanding the evolutionary constraints is essential in evaluating its potential and limitations both in biological systems and engineering applications.

F. HYBRIDS AND NEW FOUNDATIONS

Last but not least, some recent works from 2019 have merged RC with other systems or paradigms such as deep learning [49], [50], [92], [117], [218], [239], [262], [263], [323], [324], [325]. Moreover, some authors have started investigating modifying the foundations (e.g., NG-RC [208]). In fact, research on random networks [96], [252], [335], [336] shows that there is a wide range of methods taking inspiration from some core mechanics of RC to build novel approaches that might challenge reservoir computing's current foundations in the near future.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [2] W. Singer, "The cerebral cortex: A delay-coupled recurrent oscillator network?" in *Reservoir Computing*. Singapore: Springer, 2021, pp. 3–28.
- [3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.
- [4] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
- [5] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," 2017, *arXiv:1704.06857*.
- [6] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, "Beyond exploding and vanishing gradients: Analysing RNN training using attractors and smoothness," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2370–2380.
- [7] B. Hanin, "Which neural net architectures give rise to exploding and vanishing gradients?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.
- [8] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [10] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks—with an erratum note," *German Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Tech. Rep. 148*, 2001, pp. 13–34.
- [11] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, Nov. 2002.
- [12] J. J. Steil, "Backpropagation-decorrelation: Online recurrent learning with O(N) complexity," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 843–848.
- [13] P. F. Dominey, "Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning," *Biol. Cybern.*, vol. 73, no. 3, pp. 265–274, Aug. 1995.
- [14] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Netw.*, vol. 20, no. 3, pp. 391–403, Apr. 2007.
- [15] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, Aug. 2009.
- [16] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. 15th Eur. Symp. Artif. Neural Netw.*, 2007, pp. 471–482.
- [17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 86–94.
- [19] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," 2018, *arXiv:1805.12152*.
- [20] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
- [21] S. Kotlyan and D. V. Vargas, "Evolving robust neural architectures to defend from adversarial attacks," in *Proc. CEUR Workshop*, 2020, pp. 1–8.
- [22] S. Kotlyan and D. V. Vargas, "Adversarial robustness assessment: Why in evaluation both L_0 and L_∞ attacks are necessary," *PLoS ONE*, vol. 17, no. 4, Apr. 2022, Art. no. e0265723.
- [23] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Netw.*, vol. 115, pp. 100–123, Jul. 2019.
- [24] M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, vol. 497, no. 7451, pp. 585–590, May 2013.
- [25] C. J. Bruce and M. E. Goldberg, "Primate frontal eye fields. I. Single neurons discharging before saccades," *J. Neurophysiol.*, vol. 53, no. 3, pp. 603–635, Mar. 1985.
- [26] P. Barone and J.-P. Joseph, "Prefrontal cortex and spatial sequencing in macaque monkey," *Exp. Brain Res.*, vol. 78, no. 3, pp. 447–464, Dec. 1989.
- [27] P. F. Dominey and M. A. Arbib, "A cortico-subcortical model for generation of spatially accurate sequential saccades," *Cerebral Cortex*, vol. 2, no. 2, pp. 153–175, 1992.
- [28] P. Dominey, M. Arbib, and J.-P. Joseph, "A model of corticostriatal plasticity for learning oculomotor associations and sequences," *J. Cogn. Neurosci.*, vol. 7, no. 3, pp. 311–336, Jul. 1995.
- [29] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.
- [30] D. E. Rumelhart and J. L. McClelland, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362. [Online]. Available: <https://ieeexplore.ieee.org/document/6302929>
- [31] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.
- [32] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: Unifying the algorithms and accelerating convergence," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 697–709, May 2000.
- [33] M. Lukoševičius, "A practical guide to applying Echo State Networks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 659–686.
- [34] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970.

- [35] D. Sussillo and L. F. Abbott, "Generating coherent patterns of activity from chaotic neural networks," *Neuron*, vol. 63, no. 4, pp. 544–557, Aug. 2009.
- [36] S. Zhong, X. Xie, L. Lin, and F. Wang, "Genetic algorithm optimized double-reservoir Echo State Network for multi-regime time series prediction," *Neurocomputing*, vol. 238, pp. 191–204, May 2017.
- [37] L. A. Thiede and U. Parlitz, "Gradient based hyperparameter optimization in Echo State Networks," *Neural Netw.*, vol. 115, pp. 23–29, Jul. 2019.
- [38] H. Wang and X. Yan, "Optimizing the Echo State Network with a binary particle swarm optimization algorithm," *Knowl.-Based Syst.*, vol. 86, pp. 182–193, Sep. 2015.
- [39] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural Netw.*, vol. 35, pp. 1–9, Nov. 2012.
- [40] C. Gallicchio, "Chasing the echo state property," 2018, *arXiv:1811.10892*.
- [41] M. Lukoševičius, "Reservoir computing and self-organized neural hierarchies," Ph.D. dissertation, School Eng. Sci., Jacobs Univ., Bremen, Germany, 2012.
- [42] H. Jaeger, "Short term memory in Echo State Networks," GMD-German Nat. Res. Inst. Comput. Sci., Bonn, Germany, GMD Tech. Rep. 152, 2002. [Online]. Available: <http://www.faculty.jacobs-university.de/hjaeger/pubs/STMEchoStatesTechRep.pdf>
- [43] N. Bertschinger and T. Natschläger, "Real-time computation at the edge of chaos in recurrent neural networks," *Neural Comput.*, vol. 16, no. 7, pp. 1413–1436, Jul. 2004.
- [44] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural Netw.*, vol. 20, no. 3, pp. 323–334, Apr. 2007.
- [45] K. Nakajima and I. Fischer, *Reservoir Computing*. Singapore: Springer, 2021.
- [46] B. Cramer, D. Stöckel, M. Kreft, M. Wibral, J. Schemmel, K. Meier, and V. Priesemann, "Control of criticality and computation in spiking neuromorphic networks with plasticity," *Nature Commun.*, vol. 11, no. 1, p. 2853, Jun. 2020.
- [47] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital Liquid State Machine with biologically inspired learning and its application to speech recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2635–2649, Nov. 2015.
- [48] S. Zhang, Z. Sun, M. Wang, J. Long, Y. Bai, and C. Li, "Deep fuzzy Echo State Networks for machinery fault diagnosis," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1205–1218, Jul. 2020.
- [49] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, Dec. 2017.
- [50] J. Long, S. Zhang, and C. Li, "Evolving deep Echo State Networks for intelligent fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4928–4937, Jul. 2020.
- [51] M. L. Alomar, E. S. Skibinsky-Gitlin, C. F. Frasser, V. Canals, E. Isern, M. Roca, and J. L. Rosselló, "Efficient parallel implementation of reservoir computing systems," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2299–2313, Apr. 2020.
- [52] N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of Echo State Network parameters for time series forecasting," *Appl. Soft Comput.*, vol. 55, pp. 211–225, Jun. 2017.
- [53] H. Jaeger, "Adaptive nonlinear system identification with Echo State Networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, 2002, pp. 1–8.
- [54] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.
- [55] H. Tamura and G. Tanaka, "Transfer-RLS method and transfer-FORCE learning for simple and fast training of reservoir computing models," *Neural Netw.*, vol. 143, pp. 550–563, Nov. 2021.
- [56] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.
- [57] D. Sussillo and L. F. Abbott, "Transferring learning from external to internal weights in echo-state networks with sparse connectivity," *PLoS ONE*, vol. 7, no. 5, May 2012, Art. no. e37372.
- [58] R. Laje and D. V. Buonomano, "Robust timing and motor patterns by taming chaos in recurrent neural networks," *Nature Neurosci.*, vol. 16, no. 7, pp. 925–933, Jul. 2013.
- [59] B. DePasquale, C. J. Cueva, K. Rajan, G. S. Escola, and L. F. Abbott, "Full-FORCE: A target-based method for training recurrent networks," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0191527.
- [60] W. Nicola and C. Clopath, "Supervised learning in spiking neural networks with FORCE training," *Nature Commun.*, vol. 8, no. 1, p. 2208, Dec. 2017.
- [61] H. Tamura and G. Tanaka, "Two-step FORCE learning algorithm for fast convergence in reservoir computing," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2020, pp. 459–469.
- [62] Y. Zheng and E. Shlizerman, "R-FORCE: Robust learning for random recurrent neural networks," 2020, *arXiv:2003.11660*.
- [63] S. Heinrich, T. Alpay, and Y. Nagai, "Learning timescales in gated and adaptive continuous time recurrent neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 2662–2667.
- [64] A. Sharma and R. Rani, "An optimized framework for cancer classification using deep learning and genetic algorithm," *J. Med. Imag. Health Inform.*, vol. 7, no. 8, pp. 1851–1856, Dec. 2017.
- [65] S. Basterrech, E. Alba, and V. Šnášel, "An experimental analysis of the Echo State Network initialization using the Particle Swarm Optimization," in *Proc. 6th World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Jul. 2014, pp. 214–219.
- [66] J. Schmidhuber, D. Wierstra, M. Gagliolo, and F. Gomez, "Training recurrent networks by Evolino," *Neural Comput.*, vol. 19, no. 3, pp. 757–779, Mar. 2007.
- [67] H. Jaeger, "Reservoir riddles: Suggestions for echo state network research," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul./Aug. 2005, pp. 1460–1462.
- [68] D. Norton and D. Ventura, "Preparing more effective Liquid State Machines using Hebbian learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2006, pp. 4243–4248.
- [69] Y. Jin, Y. Liu, and P. Li, "SSO-LSM: A Sparse and Self-Organizing architecture for Liquid State Machine based neural processors," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2016, pp. 55–60.
- [70] S. Luo, H. Guan, X. Li, F. Xue, and H. Zhou, "Improving Liquid State Machine in temporal pattern classification," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 88–91.
- [71] J. J. Steil, "Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning," *Neural Netw.*, vol. 20, no. 3, pp. 353–364, Apr. 2007.
- [72] B. Schrauwen, M. Wardermann, D. Verstraeten, J. J. Steil, and D. Stroobandt, "Improving reservoirs using intrinsic plasticity," *Neurocomputing*, vol. 71, nos. 7–9, pp. 1159–1171, Mar. 2008.
- [73] F. Xue, Q. Li, H. Zhou, and X. Li, "Reservoir computing with both neuronal intrinsic plasticity and multi-clustered structure," *Cogn. Comput.*, vol. 9, no. 3, pp. 400–410, Jun. 2017.
- [74] C. Li, "A model of neuronal intrinsic plasticity," *IEEE Trans. Auto. Mental Develop.*, vol. 3, no. 4, pp. 277–284, Dec. 2011.
- [75] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets," 2019, *arXiv:1901.09049*.
- [76] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Commun.*, vol. 11, no. 1, p. 3625, Jul. 2020.
- [77] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. London, U.K.: Psychology Press, 2005.
- [78] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*. Hoboken, NJ, USA: Wiley, 2013.
- [79] C. Beer and O. Barak, "One step back, two steps forward: Interference and learning in recurrent neural networks," *Neural Comput.*, vol. 31, no. 10, pp. 1985–2003, Oct. 2019.
- [80] G. S. Duane, "'FORCE' learning in recurrent neural networks as data assimilation," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 27, no. 12, Dec. 2017, Art. no. 126804.
- [81] T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. C. Stewart, D. Rasmussen, X. Choo, A. R. Voelker, and C. Eliasmith, "Nengo: A Python tool for building large-scale functional brain models," *Frontiers Neuroinform.*, vol. 7, p. 48, Jan. 2014.
- [82] L. B. Liu, A. Losonczy, and Z. Liao, "Tension: A Python package for FORCE learning," *PLoS Comput. Biol.*, vol. 18, no. 12, Dec. 2022, Art. no. e1010722.
- [83] H. Badem, A. Basturk, A. Caliskan, and M. E. Yuksel, "A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited-memory BFGS optimization algorithms," *Neurocomputing*, vol. 266, pp. 506–526, Nov. 2017.
- [84] A. A. Ferreira and T. B. Ludermitz, "Comparing evolutionary methods for reservoir computing pre-training," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 283–290.

- [85] W. Zhang and D. J. Linden, "The other side of the engram: Experience-driven changes in neuronal intrinsic excitability," *Nature Rev. Neurosci.*, vol. 4, no. 11, pp. 885–900, Nov. 2003.
- [86] X. Wang, Y. Jin, and K. Hao, "Echo state networks regulated by local intrinsic plasticity rules for regression," *Neurocomputing*, vol. 351, pp. 111–122, Jul. 2019.
- [87] K. Nakada, S. Suzuki, E. Suzuki, Y. Terasaki, T. Asai, and T. Sasaki, "An information theoretic parameter tuning for MEMS-based reservoir computing," *Nonlinear Theory Appl. IEICE*, vol. 13, no. 2, pp. 459–464, 2022.
- [88] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the Liquid State Machine: A case study," *Inf. Process. Lett.*, vol. 95, no. 6, pp. 521–528, Sep. 2005.
- [89] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Sci. Rep.*, vol. 2, no. 1, pp. 1–6, Feb. 2012.
- [90] A. Jalalvand, G. Van Wallendael, and R. Van De Walle, "Real-time reservoir computing network-based systems for detection tasks on visual contents," in *Proc. 7th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jun. 2015, pp. 146–151.
- [91] M. A. Chitsazan, M. Sami Fadali, and A. M. Trzynadlowski, "Wind speed and wind direction forecasting using echo state network with nonlinear functions," *Renew. Energy*, vol. 131, pp. 879–889, Feb. 2019.
- [92] Z. Li and G. Tanaka, "Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction," *Neurocomputing*, vol. 467, pp. 115–129, Jan. 2022.
- [93] Z. Li, Y. Liu, and G. Tanaka, "Multi-reservoir Echo State Networks with Hodrick–Prescott filter for nonlinear time-series prediction," *Appl. Soft Comput.*, vol. 135, Mar. 2023, Art. no. 110021.
- [94] N. Chouikhi, B. Ammar, A. Hussain, and A. M. Alimi, "Bi-level multi-objective evolution of a multi-layered echo-state network autoencoder for data representations," *Neurocomputing*, vol. 341, pp. 195–211, May 2019.
- [95] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
- [96] Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," *Neural Netw.*, vol. 112, pp. 15–23, Apr. 2019.
- [97] J. Kaiser, R. Stal, A. Subramoney, A. Roennau, and R. Dillmann, "Scaling up Liquid State Machines to predict over address events from dynamic vision sensors," *Bioinspiration Biomimetics*, vol. 12, no. 5, Sep. 2017, Art. no. 055001.
- [98] M.-O. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.
- [99] J. J. M. Reynolds, J. S. Plank, and C. D. Schuman, "Intelligent reservoir generation for Liquid State Machines using evolutionary optimization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [100] Y. Zhou, Y. Jin, and J. Ding, "Evolutionary optimization of Liquid State Machines for robust learning," in *Proc. Int. Symp. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 389–398.
- [101] S. Tian, L. Qu, L. Wang, K. Hu, N. Li, and W. Xu, "A neural architecture search based framework for Liquid State Machine design," *Neurocomputing*, vol. 443, pp. 174–182, Jul. 2021.
- [102] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, no. 1, p. 468, Sep. 2011.
- [103] O. Yilmaz, "Machine learning using cellular automata based feature expansion and reservoir computing," *J. Cellular Automata*, vol. 10, no. 5, pp. 435–472, Jan. 2015.
- [104] A. Goudarzi, M. R. Lakin, and D. Stefanovic, "DNA reservoir computing: A novel molecular computing approach," in *Proc. Int. Workshop DNA-Based Comput.* Cham, Switzerland: Springer, 2013, pp. 76–89.
- [105] S. Lepri, G. Giacomelli, A. Politi, and F. T. Arecchi, "High-dimensional chaos in delayed dynamical systems," *Phys. D, Nonlinear Phenomena*, vol. 70, no. 3, pp. 235–249, Jan. 1994.
- [106] S. Ortín and L. Pesquera, "Reservoir computing with an ensemble of time-delay reservoirs," *Cogn. Comput.*, vol. 9, no. 3, pp. 327–336, Jun. 2017.
- [107] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, "Tutorial: Photonic neural networks in delay systems," *J. Appl. Phys.*, vol. 124, no. 15, Oct. 2018, Art. no. 152004.
- [108] S. Wolfram, *Cellular Automata and Complexity: Collected Papers*. Boca Raton, FL, USA: CRC Press, 2018.
- [109] O. Yilmaz, "Reservoir computing using cellular automata," 2014, *arXiv:1410.0162*.
- [110] D. Snyder, A. Goudarzi, and C. Teuscher, "Computational capabilities of random automata networks for reservoir computing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 87, no. 4, Apr. 2013, Art. no. 042808.
- [111] S. Nichele and M. S. Gundersen, "Reservoir computing using non-uniform binary cellular automata," 2017, *arXiv:1702.03812*.
- [112] S. Nichele and A. Molund, "Deep reservoir computing using cellular automata," 2017, *arXiv:1703.02806*.
- [113] N. McDonald, "Reservoir computing & extreme learning machines using pairs of cellular automata rules," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2429–2436.
- [114] D. Uragami and Y.-P. Gunji, "Universal criticality in reservoir computing using asynchronous cellular automata," *Complex Syst.*, vol. 31, no. 1, pp. 103–121, 2022.
- [115] J. C. Coulombe, M. C. A. York, and J. Sylvestre, "Computing with networks of nonlinear mechanical oscillators," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0178663.
- [116] H. Nguyen, P. Banda, D. Stefanovic, and C. Teuscher, "Reservoir computing with random chemical systems," in *Proc. Conf. Artif. Life (ALIFE)*. Cambridge, MA, USA: MIT Press, 2020, pp. 491–499.
- [117] X. Liu and K. K. Parhi, "Reservoir computing using DNA oscillators," *ACS Synth. Biol.*, vol. 11, no. 2, pp. 780–787, Feb. 2022.
- [118] H. Nguyen and C. Teuscher, "Biochemical reservoir computing," in *Proc. Student Res. Symp. 8*. Portland, OR, USA: Portland State Univ., 2018.
- [119] P. H. L. Nguyen, S. Rubin, P. Sarangi, P. Pal, and Y. Fainman, "SERS-based ssDNA composition analysis with inhomogeneous peak broadening and reservoir computing," *Appl. Phys. Lett.*, vol. 120, no. 2, Jan. 2022, Art. no. 023701.
- [120] W. Yahiro, N. Aubert-Kato, and M. Hagiya, "A reservoir computing approach for molecular computing," in *Proc. Conf. Artif. Life*. Cambridge, MA, USA: MIT Press, 2018, pp. 31–38.
- [121] S. Kan, K. Nakajima, T. Asai, and M. Akai-Kasaya, "Physical implementation of reservoir computing through electrochemical reaction," *Adv. Sci.*, vol. 9, no. 6, Feb. 2022, Art. no. 2104076.
- [122] M. Akai-Kasaya, Y. Takeshima, S. Kan, K. Nakajima, T. Oya, and T. Asai, "Performance of reservoir computing in a random network of single-walled carbon nanotubes complexed with polyoxometalate," *Neuromorphic Comput. Eng.*, vol. 2, no. 1, Mar. 2022, Art. no. 014003.
- [123] H. Nakao, "Phase reduction approach to synchronisation of nonlinear oscillators," *Contemp. Phys.*, vol. 57, no. 2, pp. 188–214, Apr. 2016.
- [124] T. Yamane, Y. Katayama, R. Nakane, G. Tanaka, and D. Nakano, "Wave-based reservoir computing by synchronization of coupled oscillators," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2015, pp. 198–205.
- [125] A. A. Velichko, D. V. Ryabokon, S. D. Khanin, A. V. Sidorenko, and A. G. Rikkiev, "Reservoir computing using high order synchronization of coupled oscillators," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 862, May 2020, Art. no. 052062.
- [126] M. Riou, F. A. Araujo, J. Torrejon, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, "Neuromorphic computing through time-multiplexing with a spin-torque nano-oscillator," in *IEDM Tech. Dig.*, Dec. 2017, pp. 3–36.
- [127] L. Appeltant, G. Van der Sande, J. Danckaert, and I. Fischer, "Constructing optimized binary masks for reservoir computing with delay systems," *Sci. Rep.*, vol. 4, no. 1, pp. 1–5, Jan. 2014.
- [128] J. Li, K. Bai, L. Liu, and Y. Yi, "A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system," in *Proc. 19th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2018, pp. 308–313.
- [129] J. H. Jensen and G. Tufte, "Reservoir computing with a chaotic circuit," in *Proc. 14th Eur. Conf. Artif. Life ECAL*. Cambridge, MA, USA: MIT Press, Sep. 2017, pp. 222–229.
- [130] C. Zhao, J. Li, L. Liu, L. S. Koutha, J. Liu, and Y. Yi, "Novel spike based reservoir node design with high performance spike delay loop," in *Proc. 3rd ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2016, pp. 1–5.
- [131] J. Li, C. Zhao, K. Hamedani, and Y. Yi, "Analog hardware implementation of spike-based delayed feedback reservoir computing system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3439–3446.

- [132] P. Antonik, A. Smerieri, F. Dupont, M. Haelterman, and S. Massar, "FPGA implementation of reservoir computing with online learning," in *Proc. 24th Belg.-Dutch Conf. Mach. Learn.*, 2015, pp. 1–8.
- [133] B. Schrauwen, M. D'Haene, D. Verstraeten, and J. V. Campenhout, "Compact hardware Liquid State Machines on FPGA for real-time speech recognition," *Neural Netw.*, vol. 21, nos. 2–3, pp. 511–523, Mar. 2008.
- [134] Q. Wang, Y. Li, and P. Li, "Liquid State Machine based pattern recognition on FPGA with firing-activity dependent power gating and approximate computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 361–364.
- [135] N. D. Haynes, M. C. Soriano, D. P. Rosin, I. Fischer, and D. J. Gauthier, "Reservoir computing with a single time-delay autonomous Boolean node," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, no. 2, Feb. 2015, Art. no. 020801.
- [136] M. L. Alomar, M. C. Soriano, M. Escalona-Morán, V. Canals, I. Fischer, C. R. Mirasso, and J. L. Rosselló, "Digital implementation of a single dynamical node reservoir computer," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 10, pp. 977–981, Oct. 2015.
- [137] X. Yang, W. Chen, and F. Z. Wang, "Investigations of the staircase memristor model and applications of memristor-based local connections," *Anal. Integr. Circuits Signal Process.*, vol. 87, no. 2, pp. 263–273, May 2016.
- [138] M. S. Kulkarni and C. Teuscher, "Memristor-based reservoir computing," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2012, pp. 226–232.
- [139] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, "Reservoir computing using dynamic memristors for temporal information processing," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, Dec. 2017.
- [140] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Commun.*, vol. 5, no. 1, pp. 1–6, Mar. 2014.
- [141] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Sci. Rep.*, vol. 8, no. 1, pp. 1–10, Feb. 2018.
- [142] J. Dong, S. Gigan, F. Krzakala, and G. Wainrib, "Scaling up echo-state networks with multiple light scattering," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jun. 2018, pp. 448–452.
- [143] M. A. A. Fiers, T. Van Vaerenbergh, F. Wyffels, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 344–355, Feb. 2014.
- [144] F. Laporte, A. Katumba, J. Dambre, and P. Bienstman, "Numerical demonstration of neuromorphic computing with photonic crystal cavities," *Opt. Exp.*, vol. 26, no. 7, pp. 7955–7964, 2018.
- [145] Q. Vinckier, F. Dupont, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, "High-performance photonic reservoir computer based on a coherently driven passive cavity," *Optica*, vol. 2, no. 5, pp. 438–446, 2015.
- [146] T. Taniguchi, S. Tsunegi, S. Miwa, K. Fujii, H. Kubota, and K. Nakajima, "Reservoir computing based on spintronics technology," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 331–360.
- [147] T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki, and S. Miwa, "Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions," *Phys. Rev. Appl.*, vol. 10, no. 3, Sep. 2018, Art. no. 034063.
- [148] R. Nakane, A. Hirose, and G. Tanaka, "Spin waves propagating through a stripe magnetic domain structure and their applications to reservoir computing," *Phys. Rev. Res.*, vol. 3, no. 3, Sep. 2021, Art. no. 033243.
- [149] W. Jiang, L. Chen, K. Zhou, L. Li, Q. Fu, Y. Du, and R. Liu, "Physical reservoir computing built by spintronic devices for temporal information processing," 2019, *arXiv:1901.07879*.
- [150] H. Nomura, H. Kubota, and Y. Suzuki, "Reservoir computing with dipole-coupled nanomagnets," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 361–374.
- [151] C. Johnson, A. Philippides, and P. Husbands, "Active shape discrimination with physical reservoir computers," in *Proc. 14th Int. Conf. Synth. Simul. Living Syst.* Cambridge, MA, USA: MIT Press, Jul. 2014, pp. 176–183.
- [152] K. Nakajima, T. Li, H. Hauser, and R. Pfeifer, "Exploiting short-term memory in soft body dynamics as a computational resource," *J. Roy. Soc. Interface*, vol. 11, no. 100, Nov. 2014, Art. no. 20140437.
- [153] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Exploiting the dynamics of soft materials for machine learning," *Soft Robot.*, vol. 5, no. 3, pp. 339–347, Jun. 2018.
- [154] P. Bhowad and S. Li, "Physical reservoir computing with origami and its application to robotic crawling," *Sci. Rep.*, vol. 11, no. 1, pp. 1–18, Jun. 2021.
- [155] Y. Yada, S. Yasuda, and H. Takahashi, "Physical reservoir computing with FORCE learning in a living neuronal culture," *Appl. Phys. Lett.*, vol. 119, no. 17, Oct. 2021, Art. no. 173701.
- [156] Z. Konkoli, "On developing theory of reservoir computing for sensing applications: The state weaving environment echo tracker (SWEET) algorithm," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 33, no. 2, pp. 121–143, Mar. 2018.
- [157] K. Caluwaerts and B. Schrauwen, "The body as a reservoir: Locomotion and sensing with linear feedback," in *Proc. 2nd Int. Conf. Morphological Comput. (ICMC)*, 2011, pp. 1–3.
- [158] S. Ghosh, T. Krisnanda, T. Paterek, and T. C. H. Liew, "Realising and compressing quantum circuits with quantum reservoir computing," *Commun. Phys.*, vol. 4, no. 1, pp. 1–7, May 2021.
- [159] L. C. G. Govia, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, and T. A. Ohki, "Quantum reservoir computing with a single nonlinear oscillator," *Phys. Rev. Res.*, vol. 3, no. 1, Jan. 2021, Art. no. 013077.
- [160] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, "Dynamical phase transitions in quantum reservoir computing," *Phys. Rev. Lett.*, vol. 127, no. 10, Aug. 2021, Art. no. 100502.
- [161] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, May 2017.
- [162] M. C. Soriano, S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. van der Sande, "Delay-based reservoir computing: Noise effects in a combined analog and digital implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 388–393, Feb. 2015.
- [163] M. C. Soriano, D. Brunner, M. Escalona-Morán, C. R. Mirasso, and I. Fischer, "Minimal approach to neuro-inspired information processing," *Frontiers Comput. Neurosci.*, vol. 9, p. 68, Jun. 2015.
- [164] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir computing with stochastic bitstream neurons," in *Proc. 16th Annu. Prorisc Workshop*, 2005, pp. 454–459.
- [165] M. L. Alomar, V. Canals, V. Martínez-Moll, and J. L. Rosselló, "Low-cost hardware implementation of reservoir computers," in *Proc. 24th Int. Workshop Power Timing Model., Optim. Simul. (PATMOS)*, Sep. 2014, pp. 1–5.
- [166] Y. Liu, Y. Jin, and P. Li, "Online adaptation and energy minimization for hardware recurrent spiking neural networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 1, pp. 1–21, Jan. 2018.
- [167] A. M. Hassan, H. H. Li, and Y. Chen, "Hardware implementation of echo state networks using memristor double crossbar arrays," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2171–2177.
- [168] N. Soares, L. Hays, and D. Kudithipudi, "Robustness of a memristor based Liquid State Machine," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2414–2420.
- [169] J. Burger, A. Goudarzi, D. Stefanovic, and C. Teuscher, "Computational capacity and energy consumption of complex resistive switch networks," 2015, *arXiv:1507.03716*.
- [170] J. P. Carbajal, J. Dambre, M. Hermans, and B. Schrauwen, "Memristor models for machine learning," *Neural Comput.*, vol. 27, no. 3, pp. 725–747, Mar. 2015.
- [171] S. J. D. Tran and C. Teuscher, "Memcapacitive reservoir computing," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2017, pp. 115–116.
- [172] S. J. D. Tran and C. Teuscher, "Hierarchical memcapacitive reservoir computing architecture," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Nov. 2019, pp. 1–6.
- [173] A. Z. Stieg, A. V. Avizienis, H. O. Sillan, C. Martin-Olmos, M. Aono, and J. K. Gimzewski, "Emergent criticality in complex Turing B-type atomic switch networks," *Adv. Mater.*, vol. 24, no. 2, pp. 286–293, Jan. 2012.
- [174] K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, R. Baets, P. Bienstman, and J. Van Campenhout, "Toward optical signal processing using photonic reservoir computing," *Opt. Exp.*, vol. 16, no. 15, pp. 11182–11192, 2008.

- [175] B. Schneider, J. Dambre, and P. Bienstman, "Using digital masks to enhance the bandwidth tolerance and improve the performance of on-chip reservoir computing systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2748–2753, Dec. 2016.
- [176] C. Mesaritikakis, A. Bogris, A. Kapsalis, and D. Syvridis, "High-speed all-optical pattern recognition of dispersive Fourier images through a photonic reservoir computing subsystem," *Opt. Lett.*, vol. 40, no. 14, pp. 3416–3419, 2015.
- [177] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, P. Bortolotti, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond Turing: An optoelectronic implementation of reservoir computing," *Opt. Exp.*, vol. 20, no. 3, pp. 3241–3249, 2012.
- [178] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Opt. Exp.*, vol. 20, no. 20, pp. 22783–22795, 2012.
- [179] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, vol. 547, no. 7664, pp. 428–431, Jul. 2017.
- [180] M. Riou, J. Torrejon, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, N. Leroux, D. Marković, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, "Reservoir computing leveraging the transient non-linear dynamics of spin-torque nano-oscillators," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 307–329.
- [181] R. Nakane, G. Tanaka, and A. Hirose, "Reservoir computing with spin waves excited in a garnet film," *IEEE Access*, vol. 6, pp. 4462–4469, 2018.
- [182] T. Taniguchi, A. Ogihara, Y. Utsumi, and S. Tsunegi, "Spintronic reservoir computing without driving current or magnetic field," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, Jun. 2022.
- [183] A. Fert, N. Reyren, and V. Cros, "Magnetic skyrmions: Advances in physics and potential applications," *Nature Rev. Mater.*, vol. 2, no. 7, pp. 1–15, Jun. 2017.
- [184] G. Bourianoff, D. Pinna, M. Sitte, and K. Everschor-Sitte, "Potential implementation of reservoir computing models based on magnetic skyrmions," *AIP Adv.*, vol. 8, no. 5, May 2018, Art. no. 055602.
- [185] D. Prychynenko, M. Sitte, K. Litzius, B. Krüger, G. Bourianoff, M. Kläui, J. Sinova, and K. Everschor-Sitte, "Magnetic skyrmion as a nonlinear resistive element: A potential building block for reservoir computing," *Phys. Rev. Appl.*, vol. 9, no. 1, Jan. 2018, Art. no. 014034.
- [186] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *Biol. Cybern.*, vol. 105, nos. 5–6, pp. 355–370, Dec. 2011.
- [187] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "The role of feedback in morphological computation with compliant bodies," *Biol. Cybern.*, vol. 106, no. 10, pp. 595–613, Nov. 2012.
- [188] K. Nakajima, H. Hauser, R. Kang, E. Guglielmino, D. G. Caldwell, and R. Pfeifer, "A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm," *Frontiers Comput. Neurosci.*, vol. 7, p. 91, Jul. 2013.
- [189] K. Fujita, S. Yonekura, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Environmental and structural effects on physical reservoir computing with tensegrity," in *Proc. 5th IIAE Int. Conf. Intell. Syst. Image Process.*, 2017, pp. 197–204.
- [190] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biol.*, vol. 11, no. 23, pp. R986–R996, Nov. 2001.
- [191] R. M. Harris-Warrick, "Neuromodulation and flexibility in central pattern generator networks," *Current Opinion Neurobiol.*, vol. 21, no. 5, pp. 685–692, Oct. 2011.
- [192] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Netw.*, vol. 21, no. 4, pp. 642–653, May 2008.
- [193] D. W. Sims, N. E. Humphries, N. Hu, V. Medan, and J. Berni, "Optimal searching behaviour generated intrinsically by the central pattern generator for locomotion," *eLife*, vol. 8, Nov. 2019, Art. no. e50316.
- [194] K. Fujita, S. Yonekura, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Physical reservoir computing in tensegrity with structural softness and ground collision dynamics," *J. Inst. Ind. Appl. Eng.*, vol. 6, no. 2, pp. 92–99, 2018.
- [195] K. Caluwaerts, J. Despraz, A. İççen, A. P. Sabelhaus, J. Bruce, B. Schrauwen, and V. Sunspiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *J. Roy. Soc. Interface*, vol. 11, no. 98, Sep. 2014, Art. no. 20140520.
- [196] A. Vandesompele, G. Urbain, F. Wyffels, and J. Dambre, "Populations of spiking neurons for reservoir computing: Closed loop control of a compliant quadruped," *Cogn. Syst. Res.*, vol. 58, pp. 317–323, Dec. 2019.
- [197] R. Terajima, K. Inoue, S. Yonekura, K. Nakajima, and Y. Kuniyoshi, "Behavioral diversity generated from body–environment interactions in a simulated tensegrity robot," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1597–1604, Apr. 2022.
- [198] P. Mujal, R. Martínez-Peña, J. Nekkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, "Opportunities in quantum reservoir computing and extreme learning machines," *Adv. Quantum Technol.*, vol. 4, no. 8, Aug. 2021, Art. no. 2100027.
- [199] K. Fujii and K. Nakajima, "Quantum reservoir computing: A reservoir approach toward quantum machine learning on near-term quantum devices," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 423–450.
- [200] S. Ghosh, K. Nakajima, T. Krisnanda, K. Fujii, and T. C. H. Liew, "Quantum neuromorphic computing with reservoir computing networks," *Adv. Quantum Technol.*, vol. 4, no. 9, Sep. 2021, Art. no. 2100053.
- [201] K. Fujii and K. Nakajima, "Harnessing disordered-ensemble quantum dynamics for machine learning," *Phys. Rev. Appl.*, vol. 8, no. 2, Aug. 2017, Art. no. 024030.
- [202] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, "Boosting computational power through spatial multiplexing in quantum reservoir computing," *Phys. Rev. Appl.*, vol. 11, no. 3, Mar. 2019, Art. no. 034021.
- [203] A. Kutvonen, K. Fujii, and T. Sagawa, "Optimizing a quantum reservoir computer for time series prediction," *Sci. Rep.*, vol. 10, no. 1, pp. 1–7, Sep. 2020.
- [204] M. Negoro, K. Mitarai, K. Nakajima, and K. Fujii, "Toward NMR quantum reservoir computing," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 451–458.
- [205] L. Gonon and J.-P. Ortega, "Reservoir computing universality with stochastic inputs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 100–112, Jan. 2020.
- [206] A. G. Hart, J. L. Hook, and J. H. P. Dawes, "Echo state networks trained by Tikhonov least squares are $L^2(\mu)$ approximators of ergodic dynamical systems," *Phys. D, Nonlinear Phenomena*, vol. 421, Jul. 2021, Art. no. 132882.
- [207] E. Bollt, "On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 31, no. 1, Jan. 2021, Art. no. 013108.
- [208] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, "Next generation reservoir computing," *Nature Commun.*, vol. 12, no. 1, pp. 1–8, Sep. 2021.
- [209] F. Hadaeghi, "Reservoir computing models for patient-adaptable ECG monitoring in wearable devices," 2019, *arXiv:1907.09504*.
- [210] Q.-U.-A. Mastoi, T. Wah, and R. Gopal Raj, "Reservoir computing based Echo State Networks for ventricular heart beat classification," *Appl. Sci.*, vol. 9, no. 4, p. 702, Feb. 2019.
- [211] A. N. Elbedwehy, M. E. Abo-Elhoud, and A. Elnakib, "ECG denoising using a single-node dynamic reservoir computing architecture," *Mansoura Eng. J.*, vol. 46, no. 4, pp. 47–52, Dec. 2021.
- [212] E. Donati, M. Payvand, N. Risi, R. Krause, K. Burelo, G. Indiveri, T. Dalgaty, and E. Vianello, "Processing EMG signals using reservoir computing on an event-based neuromorphic system," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2018, pp. 1–4.
- [213] S. Sakib, M. M. Fouda, M. Al-Mahdawi, A. Mohsen, M. Oogane, Y. Ando, and Z. M. Fadlullah, "Noise-removal from spectrally-similar signals using reservoir computing for MCG monitoring," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.
- [214] A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and Z. M. Fadlullah, "AI aided noise processing of spintronic based IoT sensor for magnetocardiography application," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
- [215] F. Hadaeghi, B.-P. Diercks, D. Schetelig, F. Damicelli, I. Wolf, and R. Werner, "Spatio-temporal feature learning with reservoir computing for T-cell segmentation in live-cell Ca^{2+} fluorescence microscopy," *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, 2021.

- [216] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.
- [217] M. Chen, W. Saad, and C. Yin, "Liquid State Machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.
- [218] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.
- [219] D. Bacciu, P. Barsocchi, S. Chessa, C. Gallicchio, and A. Micheli, "An experimental characterization of reservoir computing in ambient assisted living applications," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1451–1464, May 2014.
- [220] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, "Human activity recognition using multisensor data fusion based on reservoir computing," *J. Ambient Intell. Smart Environ.*, vol. 8, no. 2, pp. 87–107, Mar. 2016.
- [221] L. Sun, Z. Wang, J. Jiang, Y. Kim, B. Joo, S. Zheng, S. Lee, W. J. Yu, B.-S. Kong, and H. Yang, "In-sensor reservoir computing for language learning via two-dimensional memristors," *Sci. Adv.*, vol. 7, no. 20, 2021, Art. no. eabg1455.
- [222] Z. Zheng, S. Morando, M.-C. Pera, D. Hissel, L. Larger, R. Martinenghi, and A. B. Fuentes, "Brain-inspired computational paradigm dedicated to fault diagnosis of PEM fuel cell stack," *Int. J. Hydrogen Energy*, vol. 42, no. 8, pp. 5410–5425, Feb. 2017.
- [223] Z. Zheng, M.-C. Pera, D. Hissel, L. Larger, N. Y. Steiner, and S. Jemei, "Fault diagnosis of PEMFC systems in the model space using reservoir computing," in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Aug. 2018, pp. 1–5.
- [224] S. Zhang, X. Duan, C. Li, and M. Liang, "Pre-classified reservoir computing for the fault diagnosis of 3D printers," *Mech. Syst. Signal Process.*, vol. 146, Jan. 2021, Art. no. 106961.
- [225] M. Atencia, C. Stoean, R. Stoean, R. Rodríguez-Labrada, and G. Joya, "Dynamic clustering of time series with Echo State Networks," in *Advances in Computational Intelligence*, I. Rojas, G. Joya, and A. Catala, Eds. Cham, Switzerland: Springer, 2019, pp. 73–83.
- [226] M. Atencia, C. Gallicchio, G. Joya, and A. Micheli, "Time series clustering with deep reservoir computing," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2020, pp. 482–493.
- [227] T. Asabuki, N. Hiratani, and T. Fukai, "Interactive reservoir computing for chunking information streams," *PLoS Comput. Biol.*, vol. 14, no. 10, Oct. 2018, Art. no. e1006400.
- [228] S. Krishnagopal, Y. Aloimonos, and M. Girvan, "Similarity learning and generalization with limited data: A reservoir computing approach," *Complexity*, vol. 2018, Nov. 2018, Art. no. 6953836.
- [229] M. Ozay, I. Esnaola, F. T. Yarmar Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.
- [230] K. Hamedani, L. Liu, R. Atat, J. Wu, and Y. Yi, "Reservoir computing meets smart grids: Attack detection using delayed feedback networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 734–743, Feb. 2018.
- [231] K. Hamedani, L. Liu, S. Hu, J. Ashdown, J. Wu, and Y. Yi, "Detecting dynamic attacks in smart grids using reservoir computing: A spiking delayed feedback reservoir based approach," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 253–264, Jun. 2020.
- [232] S. T. Chandrasekaran, A. P. Kuruvila, K. Basu, and A. Sanyal, "Real-time hardware-based Malware and micro-architectural attack detection utilizing CMOS reservoir computing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 349–353, Feb. 2022.
- [233] S. Kokalj-Filipovic, P. Toliver, W. Johnson, R. R. Hoare II, and J. J. Jezak, "Deep delay loop reservoir computing for specific emitter identification," 2020, *arXiv:2010.06649*.
- [234] S. Kokalj-Filipovic, P. Toliver, W. Johnson, and R. Miller, "Reservoir-based distributed machine learning for edge operation of emitter identification," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 96–101.
- [235] S. Wang, N. Fang, and L. Wang, "Signal recovery based on optoelectronic reservoir computing for high speed optical fiber communication system," *Opt. Commun.*, vol. 495, Sep. 2021, Art. no. 127082.
- [236] T. Yamane, J. B. Héroux, H. Numata, G. Tanaka, R. Nakane, and A. Hirose, "Application identification of network traffic by reservoir computing," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2019, pp. 389–396.
- [237] H. Ando and H. Chang, "Road traffic reservoir computing," 2019, *arXiv:1912.00554*.
- [238] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4694–4708, Oct. 2018.
- [239] Z. Zhou, L. Liu, V. Chandrasekhar, J. Zhang, and Y. Yi, "Deep reservoir computing meets 5G MIMO-OFDM systems in symbol detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1266–1273.
- [240] S. Morando, S. Jemei, R. Gouriveau, N. Zerhouni, and D. Hissel, "Fuel cells prognostics using Echo State Network," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2013, pp. 1632–1637.
- [241] S. Morando, S. Jemei, D. Hissel, R. Gouriveau, and N. Zerhouni, "ANOVA method applied to proton exchange membrane fuel cell ageing forecasting using an Echo State Network," *Math. Comput. Simul.*, vol. 131, pp. 283–294, Jan. 2017.
- [242] R. Mezzi, S. Morando, N. Y. Steiner, M. C. Péra, D. Hissel, and L. Larger, "Multi-reservoir Echo State Network for proton exchange membrane fuel cell remaining useful life prediction," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 1872–1877.
- [243] Z. Li, Z. Zheng, and R. Outhib, "Adaptive prognostic of fuel cells by implementing ensemble echo state networks in time-varying model space," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 379–389, Jan. 2020.
- [244] J. Fonollosa, S. Sheik, R. Huerta, and S. Marco, "Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring," *Sens. Actuators B, Chem.*, vol. 215, pp. 618–629, Aug. 2015.
- [245] J. Wang, T. Niu, H. Lu, W. Yang, and P. Du, "A novel framework of reservoir computing for deterministic and probabilistic wind power forecasting," *IEEE Trans. Sustain. Energy*, vol. 11, no. 1, pp. 337–349, Jan. 2020.
- [246] A. Ghani, T. M. McGinnity, L. Maguire, L. McDaid, A. Belatreche, and N. Shabtai, "Neuro-inspired speech recognition based on reservoir computing," in *Advances in Speech Recognition*, vol. 164. London, U.K.: IntechOpen, 2010.
- [247] A. Alalshkhumbarak and L. S. Smith, "On improving the classification capability of reservoir computing for Arabic speech recognition," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2014, pp. 225–232.
- [248] F. Abreu Araujo, M. Riou, J. Torrejon, S. Tsunegi, D. Querlioz, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, "Role of non-linear data processing on speech recognition task in the framework of reservoir computing," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Jan. 2020.
- [249] J. Pons and X. Serra, "Randomly weighted CNNs for (music) audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 336–340.
- [250] W.-J. Wang, Y. Tang, J. Xiong, and Y.-C. Zhang, "Stock market index prediction based on reservoir computing models," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115022.
- [251] R. Budhiraja, M. Kumar, M. K. Das, A. S. Bafila, and S. Singh, "A reservoir computing approach for forecasting and regenerating both dynamical and time-delay controlled financial system behavior," *PLoS ONE*, vol. 16, no. 2, Feb. 2021, Art. no. e0246737.
- [252] M. Cucchi, C. Gruener, L. Petrauskas, P. Steiner, H. Tseng, A. Fischer, B. Penkovsky, C. Matthus, P. Birkholz, H. Kleemann, and K. Leo, "Reservoir computing with biocompatible organic electrochemical networks for brain-inspired biosignal classification," *Sci. Adv.*, vol. 7, no. 34, Aug. 2021, Art. no. eabh0693.
- [253] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," *Frontiers Neurosci.*, vol. 9, p. 502, Feb. 2016.
- [254] S. Brodeur and J. Rouat, "Regulation toward self-organized criticality in a recurrent spiking neural reservoir," in *Proc. Int. Conf. Artif. Neural Netw.* Berlin, Germany: Springer, 2012, pp. 547–554.
- [255] N. Garg, I. Balafrej, Y. Beillaud, D. Drouin, F. Alibart, and J. Rouat, "Signals to spikes for neuromorphic regulated reservoir computing and EMG hand gesture recognition," in *Proc. Int. Conf. Neuromorphic Syst.*, Jul. 2021, pp. 1–8.

- [256] B. Shakya, M. M. Fouda, S. C. Chiu, and Z. M. Fadlullah, "A circuit-embedded reservoir computer for smart noise reduction of MCG signals," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IoT&IS)*, Nov. 2021, pp. 56–61.
- [257] D. Kleyko, S. Khan, E. Osipov, and S.-P. Yong, "Modality classification of medical images with distributed representations based on cellular automata reservoir computing," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 1053–1056.
- [258] M. Eder, F. Hisch, and H. Hauser, "Morphological computation-based control of a modular, pneumatically driven, soft robotic arm," *Adv. Robot.*, vol. 32, no. 7, pp. 375–385, Apr. 2018.
- [259] A. Vargas, M. Ireland, and D. Anderson, "Swing-free manoeuvre controller for rotorcraft unmanned aerial vehicle slung-load system using echo state networks," *Int. J. Unmanned Syst. Eng.*, vol. 3, no. 1, pp. 26–37, Jan. 2015.
- [260] K. Tanaka, S.-H. Yang, Y. Tokudome, Y. Minami, Y. Lu, T. Arie, S. Akita, K. Takei, and K. Nakajima, "Flapping-wing dynamics as a natural detector of wind direction," *Adv. Intell. Syst.*, vol. 3, no. 2, Feb. 2021, Art. no. 2000174.
- [261] A. Jalalvand, K. Demuyne, W. De Neve, and J.-P. Martens, "On the application of reservoir computing networks for noisy image recognition," *Neurocomputing*, vol. 277, pp. 237–248, Feb. 2018.
- [262] Z. Tong and G. Tanaka, "Reservoir computing with untrained convolutional neural networks for image recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1289–1294.
- [263] Y. Tanaka and H. Tamukoh, "Reservoir-based convolution," *Nonlinear Theory Appl. IEICE*, vol. 13, no. 2, pp. 397–402, 2022.
- [264] H. Chang and K. Futagami, "Reinforcement learning with convolutional reservoir computing," *Int. J. Speech Technol.*, vol. 50, no. 8, pp. 2400–2410, Aug. 2020.
- [265] H. Chang and K. Futagami, "Convolutional reservoir computing for world models," 2019, *arXiv:1907.08040*.
- [266] Y. Yonemura and Y. Katori, "Image recognition model based on convolutional reservoir computing," in *Proc. Annu. Conf. JSAI*, vol. JSAI2020, 2020, p. 4Rin183. [Online]. Available: https://www.jstage.jst.go.jp/article/pjsai/JSAI2020/0/JSAI2020_4Rin183/_article-char/en, doi: 10.11517/pjsai.JSAI2020.0_4Rin183.
- [267] N. Schaetti, M. Salomon, and R. Couturier, "Echo state networks-based reservoir computing for MNIST handwritten digits recognition," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC), 15th Int. Symp. Distrib. Comput. Appl. Bus. Eng. (DCABES)*, Aug. 2016, pp. 484–491.
- [268] P. Jacobson, M. Shirao, K. Yu, G.-L. Su, and M. C. Wu, "Image classification using delay-based optoelectronic reservoir computing," *Proc. SPIE*, vol. 11703, Mar. 2021, Art. no. 1170310.
- [269] Y. Yang, P. Zhou, T. Chen, Y. Huang, and N. Li, "Optical neuromorphic computing based on a large-scale laterally coupled laser array," *Opt. Commun.*, vol. 521, Oct. 2022, Art. no. 128599.
- [270] A. Morán, C. F. Frasser, and J. L. Rosselló, "Reservoir computing hardware with cellular automata," 2018, *arXiv:1806.04932*.
- [271] P. Jacobson, M. Shirao, K. Yu, G.-L. Su, and M. C. Wu, "Hybrid convolutional optoelectronic reservoir computing for image recognition," *J. Lightw. Technol.*, vol. 40, no. 3, pp. 692–699, Feb. 1, 2022.
- [272] Q. An, K. Bai, L. Liu, F. Shen, and Y. Yi, "A unified information perception using deep reservoir computing," *Comput. Electr. Eng.*, vol. 85, Jul. 2020, Art. no. 106705.
- [273] D. V. Vargas and T. Asabuki, "Continual general chunking problem and SyncMap," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 10006–10014.
- [274] T. Asabuki and T. Fukai, "Somatodendritic consistency check for temporal feature segmentation," *Nature Commun.*, vol. 11, no. 1, pp. 1–13, Mar. 2020.
- [275] K. Hamedani, Z. Zhou, K. Bai, and L. Liu, "The novel applications of deep reservoir computing in cyber-security and wireless communication," in *Intelligent System and Computing*. London, U.K.: IntechOpen, 2019.
- [276] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS ONE*, vol. 8, no. 11, Nov. 2013, Art. no. e78318.
- [277] M. Lukoševicius, D. Popovici, H. Jaeger, U. Siewert, and R. Park, "Time warping invariant echo state networks," *Inf. Resour. Center, Jacobs Univ. Bremen, Bremen, Germany, Constructor Univ. Tech. Rep. (2)*, 2006. [Online]. Available: <https://opus.constructor.university/frontdoor/index/index/docId/660>
- [278] G. Holzmam, "Reservoir computing: A powerful black-box framework for nonlinear audio processing," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Jun. 2009, pp. 1–8.
- [279] L. Keuninckx, J. Danckaert, and G. Van der Sande, "Real-time audio processing with a cascade of discrete-time delay line-based reservoir computers," *Cogn. Comput.*, vol. 9, no. 3, pp. 315–326, Jun. 2017.
- [280] B. Schrauwen and I. Van Campenhout, "BSA, a fast and accurate spike train encoding scheme," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2003, pp. 2825–2830.
- [281] X. Lin, Z. Yang, and Y. Song, "Short-term stock price prediction based on echo state networks," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 7313–7317, Apr. 2009.
- [282] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *Nature*, vol. 503, no. 7474, pp. 78–84, Nov. 2013.
- [283] P. F. Dominey and F. Ramus, "Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant," *Lang. Cogn. Processes*, vol. 15, no. 1, pp. 87–127, Feb. 2000.
- [284] P. F. Dominey and T. Inui, "Cortico-striatal function in sentence comprehension: Insights from neurophysiology and modeling," *Cortex*, vol. 45, no. 8, pp. 1012–1018, Sep. 2009.
- [285] P. F. Dominey, T. Inui, and M. Hoen, "Neural network processing of natural language: II. Towards a unified model of corticostriatal function in learning sentence comprehension and non-linguistic sequencing," *Brain Lang.*, vol. 109, nos. 2–3, pp. 80–92, May 2009.
- [286] X. Hinaut and P. F. Dominey, "Real-time parallel processing of grammatical structure in the fronto-striatal system: A recurrent network simulation study using reservoir computing," *PLoS ONE*, vol. 8, no. 2, Feb. 2013, Art. no. e52946.
- [287] P. F. Dominey, "Recurrent temporal networks and language acquisition—From corticostriatal neurophysiology to reservoir computing," *Frontiers Psychol.*, vol. 4, p. 500, 2013.
- [288] P. Enel, E. Procyk, R. Quilodran, and P. F. Dominey, "Reservoir computing properties of neural dynamics in prefrontal cortex," *PLoS Comput. Biol.*, vol. 12, no. 6, Jun. 2016, Art. no. e1004967.
- [289] P. F. Dominey, "Cortico-striatal origins of reservoir computing, mixed selectivity, and higher cognitive function," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 29–58.
- [290] S. Fusi, E. K. Miller, and M. Rigotti, "Why neurons mix: High dimensionality for higher cognition," *Current Opinion Neurobiol.*, vol. 37, pp. 66–74, Apr. 2016.
- [291] W. J. Johnston, S. E. Palmer, and D. J. Freedman, "Nonlinear mixed selectivity supports reliable neural computation," *PLoS Comput. Biol.*, vol. 16, no. 2, Feb. 2020, Art. no. e1007544.
- [292] D. Ledergerber, C. Battistin, J. S. Blackstad, R. J. Gardner, M. P. Witter, M.-B. Moser, Y. Roudi, and E. I. Moser, "Task-dependent mixed selectivity in the subiculum," *Cell Rep.*, vol. 35, no. 8, May 2021, Art. no. 109175.
- [293] A. Wallach, A. Melanson, A. Longtin, and L. Maler, "Mixed selectivity coding of sensory and motor social signals in the thalamus of a weakly electric fish," *Current Biol.*, vol. 32, no. 1, pp. 51–63, Jan. 2022.
- [294] P. M. Milner, "The functional nature of neuronal oscillations," *Trends Neurosci.*, vol. 15, no. 10, p. 387, Jan. 1992.
- [295] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends Neurosci.*, vol. 28, no. 1, pp. 1–4, Jan. 2005.
- [296] C. M. Gray, P. König, A. K. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, no. 6213, pp. 334–337, Mar. 1989.
- [297] W. Singer and C. M. Gray, "Visual feature integration and the temporal correlation hypothesis," *Annu. Rev. Neurosci.*, vol. 18, no. 1, pp. 555–586, Mar. 1995.
- [298] W. Singer, "Neuronal synchrony: A versatile code for the definition of relations?" *Neuron*, vol. 24, no. 1, pp. 49–65, Sep. 1999.
- [299] A. K. Engel, P. Fries, and W. Singer, "Dynamic predictions: Oscillations and synchrony in top-down processing," *Nature Rev. Neurosci.*, vol. 2, no. 10, pp. 704–716, Oct. 2001.
- [300] W. Singer, "Neuronal oscillations: Unavoidable and useful?" *Eur. J. Neurosci.*, vol. 48, no. 7, pp. 2389–2398, Oct. 2018.
- [301] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, Jun. 2004.
- [302] P. Fries, "Rhythms for cognition: Communication through coherence," *Neuron*, vol. 88, no. 1, pp. 220–235, Oct. 2015.

- [303] L. Glass and J. Sun, "Periodic forcing of a limit-cycle oscillator: Fixed points, Arnold tongues, and the global organization of bifurcations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 50, no. 6, pp. 5077–5084, Dec. 1994.
- [304] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.
- [305] B. György, "The brain's default state: Self-organized oscillations in rest and sleep," in *Rhythms of the Brain*. Oxford, U.K.: Oxford Univ. Press, Oct. 2006.
- [306] D. Nikolić, S. Häusler, W. Singer, and W. Maass, "Distributed fading memory for stimulus properties in the primary visual cortex," *PLoS Biol.*, vol. 7, no. 12, Dec. 2009, Art. no. e1000260.
- [307] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, Jul. 2008.
- [308] A. Subramoney, F. Scherr, and W. Maass, "Reservoirs learn to learn," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 59–76.
- [309] D. Verstraeten, B. Schrauwen, S. Dieleman, P. Brakel, P. Buteneers, and D. Pecevski, "Oger: Modular learning architectures for large-scale sequential processing," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2995–2998, 2012.
- [310] Oger. (2018). *Oger: Modular Learning Architectures For Large-Scale Sequential Processing*. [Online]. Available: <https://github.com/neuronalX/Oger>
- [311] Easyesn. (2017). *EasyESN: A Library for Recurrent Neural Networks Using Echo State Networks*. [Online]. Available: <https://github.com/kalekiu/easyesn>
- [312] N. Schaetti. (2018). *Echotorch: Reservoir Computing with PyTorch*. [Online]. Available: <https://github.com/nschaetti/EchoTorch>
- [313] F. Martinuzzi, C. Rackauckas, A. Abdelrehim, M. D. Mahecha, and K. Mora, "ReservoirComputing.JL: An efficient and modular library for reservoir computing models," *J. Mach. Learn. Res.*, vol. 23, no. 288, pp. 1–8, 2022.
- [314] N. Trouvain, N. Rougier, and X. Hinaut, "Create efficient and complex reservoir computing architectures with reservoirpy," in *Proc. 16th Int. Conf. Simulation Adapt. Behav. (SAB)*. Cergy-Pontoise, France: Springer, Sep. 2022, pp. 91–102.
- [315] T. Y. Foong and D. V. Vargas, "Generating oscillation activity with echo state network to mimic the behavior of a simple central pattern generator," 2023, *arXiv:2306.10927*.
- [316] M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer, "Reservoir computing in material substrates," in *Reservoir Computing*. Singapore: Springer, 2021, pp. 141–166.
- [317] G. Tanaka, C. Gallicchio, A. Micheli, J.-P. Ortega, and A. Hirose, "Guest editorial special issue on new frontiers in extremely efficient reservoir computing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2571–2574, Jun. 2022.
- [318] B. Paaßen, A. Schulz, T. C. Stewart, and B. Hammer, "Reservoir memory machines as neural computers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2575–2585, Jun. 2022.
- [319] T. Jüngling, T. Lymburn, and M. Small, "Consistency hierarchy of reservoir computers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2586–2595, Jun. 2022.
- [320] G. Manjunath, "An echo state network imparts a curve fitting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2596–2604, Jun. 2022.
- [321] B. B. Schwedersky, R. C. C. Flesch, and S. B. Rovea, "Adaptive practical nonlinear model predictive control for echo state network models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2605–2614, Jun. 2022.
- [322] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, "Reservoir computing as digital twins for nonlinear dynamical systems," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 33, no. 3, Mar. 2023, Art. no. 033111.
- [323] A. Jalalvand, J. Abbate, R. Conlin, G. Verdoolaege, and E. Kolemen, "Real-time and adaptive reservoir computing with application to profile prediction in fusion plasma," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2630–2641, Jun. 2022.
- [324] L. Pasa, N. Navarin, and A. Sperduti, "Multiresolution reservoir graph neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2642–2653, Jun. 2022.
- [325] L. Pedrelli and X. Hinaut, "Hierarchical-task reservoir for online semantic analysis from continuous speech," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2654–2663, Jun. 2022.
- [326] S. Gupta, S. Chakraborty, and C. S. Thakur, "Neuromorphic time-multiplexed reservoir computing with on-the-fly weight generation for edge devices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2676–2685, Jun. 2022.
- [327] M. Nakajima, K. Tanaka, and T. Hashimoto, "Neural Schrödinger equation: Physical law as deep neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2686–2700, Jun. 2022.
- [328] D. Kleyko, E. P. Frady, and F. T. Sommer, "Cellular automata can reduce memory requirements of collective-state computing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2701–2713, Jun. 2022.
- [329] R. Yuste, J. N. MacLean, J. Smith, and A. Lansner, "The cortex as a central pattern generator," *Nature Rev. Neurosci.*, vol. 6, no. 6, pp. 477–483, Jun. 2005.
- [330] J. Duysens and H. W. A. A. Van de Crommert, "Neural control of locomotion; Part 1: The central pattern generator from cats to humans," *Gait Posture*, vol. 7, no. 2, pp. 131–141, Mar. 1998.
- [331] M. Jazayeri and S. Ostojic, "Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity," *Current Opinion Neurobiol.*, vol. 70, pp. 113–120, Oct. 2021.
- [332] W. Maass, R. Legenstein, and N. Bertschinger, "Methods for estimating the computational power and generalization capability of neural microcircuits," in *Advances in Neural Information Processing Systems*, vol. 17. Cambridge, MA, USA: MIT Press, 2004.
- [333] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.
- [334] L. F. Seoane, "Evolutionary aspects of reservoir computing," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 374, no. 1774, Jun. 2019, Art. no. 20180377.
- [335] T. L. Carroll and L. M. Pecora, "Network structure effects in reservoir computers," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 29, no. 8, Aug. 2019, Art. no. 083130.
- [336] R. Berner, *Patterns of Synchrony in Complex Networks of Adaptively Coupled Oscillators*. Cham, Switzerland: Springer, 2021.



HENG ZHANG received the B.Eng. degree from the Department of Electronic Information Engineering, Shenzhen University, China, in 2019, and the M.Sc. degree from the Department of Electrical and Electronic Engineering, The University of Manchester, U.K., in 2020. He is currently pursuing the Ph.D. degree with Kyushu University, Japan. He has been receiving the SPRING Scholarship to study with Kyushu University, since 2021, which is supported by one of the biggest Japanese funding agency. His current research interests include bio-inspired machine learning, artificial intelligence, and robust and adaptive learning algorithms.



DANILO VASCONCELLOS VARGAS is an Associate Professor with Kyushu University, a Visiting Researcher with The University of Tokyo, and a CEO/a Founder of MiraiX. He has published numerous works in prestigious journals. His research interests include AI, evolutionary computation, and complex adaptive systems. He received several awards, including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, in 2022. He has presented tutorials at various conferences, co-organized and advised workshops, and leads the Laboratory of Intelligent Systems, focused on building robust and adaptive AI, funded by JST and JSPS.