

PROGRESS REVIEW • OPEN ACCESS

An organized view of reservoir computing: a perspective on theory and technology development

To cite this article: Gisy Abdal *et al* 2024 *Jpn. J. Appl. Phys.* **63** 050803

View the [article online](#) for updates and enhancements.

You may also like

- [Reservoir computing and decision making using laser dynamics for photonic accelerator](#)
Atsushi Uchida, Kazutaka Kanno, Satoshi Sunada et al.
- [Exploring reservoir computing: implementation via double stochastic nanowire networks](#)
Jian-Feng Tang, , Lei Xia et al.
- [Feasibility and advantage of reservoir computing on single-electron devices](#)
Takahide Oya



An organized view of reservoir computing: a perspective on theory and technology development

Gisya Abdi¹, Tomasz Mazur¹, and Konrad Szacilowski^{1,2*}

¹AGH University of Krakow, Academic Centre for Materials and Nanotechnology, al. Mickiewicza 30, 30-059 Kraków, Poland

²Unconventional Computing Lab, University of the West of England, Bristol BS16 1QY, United Kingdom

*E-mail: szacilow@agh.edu.pl; konrad.szacilowski@uwe.ac.uk

Received February 1, 2024; revised March 20, 2024; accepted March 31, 2024; published online May 9, 2024

Reservoir computing is an unconventional computing paradigm that uses system complexity and dynamics as a computational medium. Currently, it is the leading computational paradigm in the fields of unconventional *in materia* computing. This review briefly outlines the theory behind the term ‘reservoir computing,’ presents the basis for the evaluation of reservoirs, and presents a cultural reference of reservoir computing in a haiku. The summary highlights recent advances in physical reservoir computing and points out the importance of the drive, usually neglected in physical implementations of reservoir computing. However, drive signals may further simplify the training of reservoirs’ readout layer training, thus contributing to improved performance of reservoir computer performance. © 2024 The Author(s). Published on behalf of The Japan Society of Applied Physics by IOP Publishing Ltd

1. Introduction

The current development of artificial intelligence (AI) is possible due to enormous progress in digital computing. New processor architectures, improved fabrication technologies, and progress in thermal management enables massive parallel computation, which the general audience perceives as intelligent. Current excitement about Chat GPT and other similar systems is the best illustration of the chances- and also threats-related AI approach. In principle, most of the currently enjoyed AI tools are based on a machine-learning approach; the quality and performance of these systems is related to their size and training protocols.

Training of an artificial neural network is a tedious and energy/time-consuming process, in which synaptic weights of all connections within the network must be modified according to the desired output. This implies, in simple cases, solving sets of millions of linear equations, optimizing the network architecture and activation function of nodes, and repeating these steps until the output meets the expected one. Naturally, the larger the network, the higher the cost of training, both in terms of energy and time. Therefore, various approaches, in which training is simplified or restricted to only part of the system, have been considered for years.^{1–3)} Furthermore, development of in-memory computing approaches,^{4,5)} the free von Neumann bottleneck,^{6,7)} requires completely new computing paradigms, different from commonly used Turingian algorithms.^{8–11)}

There are two main issues limiting computing efficiency: the von Neumann bottleneck and the informational “black hole” problem. The problem with central processing unit-memory information transfer can be solved by the in-memory computing approach, using, for example, memristors or other similar devices.^{6,12)} The “black hole” problem is related to a big data issue—humankind and all widely used information technology devices produce and store so much data that it can never be accessed and processed in a reasonable way, so most of the data stored are hidden behind the informational black hole event horizon.¹³⁾ This situation was already

envisioned in 1928 by American writer H.P. Lovecraft in his novelette *The Call of Cthulhu*.¹⁴⁾

“The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.”

In recent years, significant advancements have been witnessed in the development of artificial neural networks (ANNs), playing a pivotal role across diverse applications, including object detection, security, natural language processing, autonomous driving, and so on. ANNs are broadly classified into feedforward neural networks (FNNs), exemplified by the convolutional neural network, adept at handling static spatial patterns, and recurrent neural networks (RNNs) as typical temporal neural networks, designed for processing temporal signals by encapsulating historical information within internal states to facilitate short-term memory. The training of RNNs poses challenges attributable to the complexities of exploding or vanishing gradients inherent in recurrent structures.

To address this issue, reservoir computing (RC) emerged as a machine-learning framework and evolved through the amalgamation of specific recurrent neural network models, encompassing liquid state machines (LSMs), echo state networks (ESNs), and delay-feedback reservoirs.¹⁵⁾ RC originated from ESNs by Jaeger¹⁶⁾ in 2001 and the LSM by Maass et al.¹⁷⁾ in 2002. An ESN employs sparsely connected sigmoidal-response neurons in its random network architecture, while an LSM utilizes spiking neurons. Despite architectural differences, both methods leverage reservoirs of neurons to effectively address time-series problems, leading to their classification under RC.¹⁸⁾ The concept of an LSM (or echo state machine) helps us to both understand



and to harness the role of stochasticity and dynamics for computation.

RC simplifies training by concentrating on training only the weights connected to the output layer (also called the readout layer). A related concept is backpropagation through time (BPTT), a training algorithm to train the readout layer of the reservoir, which unrolls the RNN through time and treats it as a FNN over a sequence adjusting weights based on the error in the output.^{19–21} BPTT's efficacy lies in its versatility across network architectures, employing automatic differentiation, and taking advantage of graphics processing unit acceleration. Moreover, BPTT is implemented to train long-short-term memory, gated recurrent units, and unitary RNNs, allowing adjustment of parameters like forward-pass time steps and the number of previous time steps for backpropagation. Vlachas et al.²² compared the efficiency of RNNs with RC and BPTT for forecasting spatio-temporal dynamics in complex systems. RC excels with full-state dynamics, outperforming BPTT in predictive accuracy and requiring less training time. However, for reduced-order data, BPTT-trained RNNs demonstrate superior stability and forecasting abilities.¹⁹

It is not possible to decrease the total cost of computation below a certain limit; however, smart interplay and trade-off between cost and performance is possible.²³ In classical neural networks, the cost of training is related to the size of the network (more precisely, the number of synaptic connections within the network). On the other hand, the concept of RC,²⁴ which implies an unknown (and untrainable) inner architecture, followed by a relatively simple trainable perceptron (linear FNN) is a tempting alternative.²⁵ It seems simple, but the application of randomly arranged unknown nodes of a network is not a trivial task. This does not violate the “no free lunch” theorem due to preserved complexity and utilizing reservoir dynamics for computation instead of training an ANN.²⁶

Furthermore, utilization of inner dynamics for computation may bring some randomness, which may be beneficial for tasks like prediction of future trends of the basis of past time series (applied, for example, in macroeconomy^{27,28} and autonomous robotics^{29,30}) and advanced cryptography,^{31–33} and may be a seed for systems intended for mimicking human creativity.^{34,35} From a more trivial side, the application of RC may increase sensitivity and selectivity of chemosensors.^{36–40}

Physical RC brings a completely new perspective to information processing. It utilizes the internal dynamics (spontaneous or stimulated)⁴¹ of physical systems as well as their nonlinear responsiveness^{42–44} for computation. There is a plethora of systems that can be used in this context, provided that they present significant dynamic properties. In the words of Zoran Konkoli, “even rocks can compute,” but the complexity of computation, which can be performed in reasonable time with a such system, is rather low.^{45,46} The field of *in materia* neuromorphic computing, despite tremendous activity of numerous research groups, is still in its infancy. Replication of a complex biological information-processing circuitry of even the simplest neural system is out of reach for current technology. There are, however, many successful constructs mimicking neurons and synapses, and processes like sensory integration and

nociception.^{47–55} Physical neuromorphic computing is slowly coming into reality.^{56,57}

Furthermore, RC seems to be a perfect tool to understand the relation between the connectome (the connectivity map between all neurons in the nervous system) and the cognitive abilities of the neural system.^{58,59} Whereas the vision of a full understanding of the human brain is still far-fetched, the development of new tools will be helpful for a better understanding of ANNs and also physical neuromimetic systems.^{60–62}

2. Reservoir computing in dynamic systems

In principle, any dynamic system can be considered a computational engine; however, its performance strongly depends on the complexity of its dynamics. The reservoir can be understood as a function \mathbf{F} acting on an input data space Ω and transforming it into the output space Ω' :

$$\mathbf{F}: \Omega \rightarrow \Omega'. \quad (1)$$

This transformation ability of the phase space fulfills the definition of a filter. The reservoir system (the reservoir itself and a trainable output layer) removes the unwanted part of the input data, thus generating the output. In the most general case, the input set u , which is a bounded subset of \mathbb{R} , is mapped into $\mathbb{U} \subset \mathbb{R}^n$, i.e. the filtration of the input data is carried out by increasing the dimensionality of the input dataset. Finally, a readout layer (e.g. trained linear perceptron) takes states of the selected nodes of the reservoir and generates the output. This is also called the output layer, by analogy to ANNs. These two terms can be used interchangeably. The term “output layer” is usually used in neural network domains—as the final layer, with a unique set of weights, which contributes to the final result prediction. In RC, the final layer is denoted as the “readout layer”—it translates the reservoir output (by means of linear transformation) into one of the desired class. The purpose of both layers is the same: mapping high-dimensional output into a much lower number of categories. The difference between these definitions is the fact that the readout layer can be an external part from the reservoir itself, whereas the output layer of a neural network consists typically of the same type of nodes and connections. In effect, the reservoir separates the input data into distinct categories that do not overlap by temporarily increasing the dimensionality of the data space (Fig. 1).⁶³ Therefore, the target projection space subset can be considered as a Tikhonov space—a topological space the elements of which can be separated by a continuous function, and a distance between elements of the space can be defined.⁶⁴ Therefore, the distance between categories is considered as an abstract measure of the performance of the reservoir.

Formally, the dynamic mapping of the input space in the reservoir states can be described by Eqs. (2) and (3):

$$x_n = \mathbf{F}(u_n, q_n) \quad (2)$$

$$x_n = (1 - \alpha)x_{n-1} + \alpha f(\beta \mathbf{W}_{\text{in}} u_n + \gamma \mathbf{W}_{\text{drive}} q_n + \delta \mathbf{W} x_{n-1}), \quad (3)$$

where f is a nonlinear activation function (e.g. Heaviside, or hyperbolic tangent functions), u_n is the input signal, and q_n is the external drive. The matrices \mathbf{W}_{in} and $\mathbf{W}_{\text{drive}}$ denote

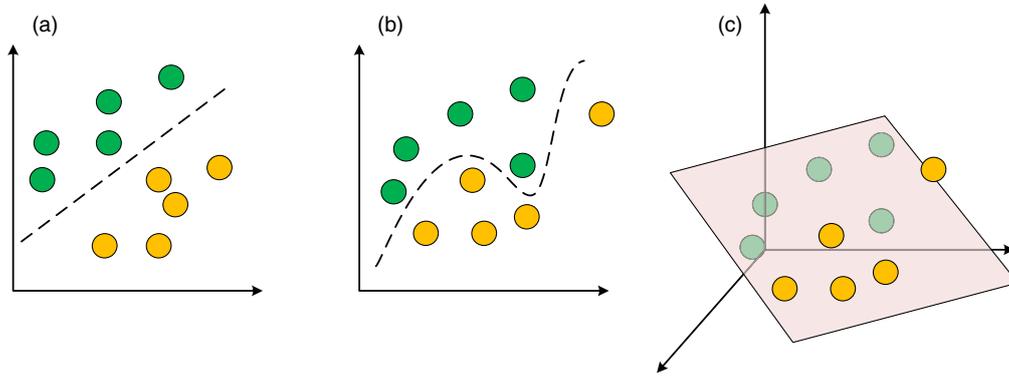


Fig. 1. A simple dataset in which linear separation by a perceptron is possible (a), compared with a complex data space that cannot be linearly separated (b). The transformation of the higher-dimensional input data space into the space by the reservoir enables linear separation of the dataset into the desired categories (c). Adapted from Ref. 63.

the weight matrices responsible for mapping the input and the drive signal into the reservoir. Matrix \mathbf{W} denotes the connection weight matrix controlling the internal dynamics of the reservoir. The scaling parameters α , β , γ , and δ describe the leakage of the previous state of the reservoir to the current one and the sensitivity to the input, drive, and reservoir history, respectively.

The computation output \mathfrak{R} in the simplest case may be given by the trained perceptron, taking a subspace of the reservoir state as input (4):

$$\mathfrak{R} = \mathbf{W}_{\text{out}} \cdot y_n, \quad (4)$$

where y_n is a well-defined subset of x_n , i.e. $y_n \subset x_n$, and \mathbf{W}_{out} is the trained weight matrix.

More details of the formal description of reservoirs are given in dedicated papers.^{39,65,66} Recently, more advanced approaches to RC move the nonlinear activation function from the reservoir to the output layer.⁶⁷ This is an important advancement, as it significantly accelerates with numerical calculations in the case of software implementation of RC. This is, however, not so relevant for physical systems, where the nonlinearity of the reservoir is embedded in the properties of the computations medium.⁶⁸ In this case, however, management of delayed feedback (and the introduction of multiple delays into a single physical system) is of crucial importance.^{66,69}

The performance of the reservoir can be preliminarily evaluated by spectral analysis of weight matrices. This can be done on various numerical models in which the dimensionality of the reservoir is known; however, it cannot be directly performed on physical reservoir systems. The spectral radii of the input and reservoir matrices measure the performance of the reservoir. Values that are too high [especially the spectral radius of \mathbf{W} , $\rho(\mathbf{W})$] result in chaotic dynamics of the system, whereas low values result in quickly fading oscillations. The same concern the input matrix—a too high spectral radius results in extreme separation and lack of categorization, as a pair of very similar input vectors is mapped into two distant points of the reservoir space.

Physical RC can be simply described as an echo state property of the physical system. In the most simplistic realization, any input applied to the system results in an out-of-equilibrium state, which evolves to an initial (or significantly different) equilibrium state. This case corresponds to reservoirs with very a low spectral radius of \mathbf{W} [i.e. $\rho(\mathbf{W}) \ll 1$], which, despite high sensitivity to the input, cannot present complex dynamic patterns [Fig. 2(a)]. In a more complex case, the reservoir is a dynamic system [$\rho(\mathbf{W}) \approx 1$], and a reservoir is a self-sustained oscillator, in which any input can change the characteristics dynamics [Figs. 2(b)–2(c)]. Too high values of the spectral radii may result, in turn, in an explosion of the phase space [Fig. 2(d)].

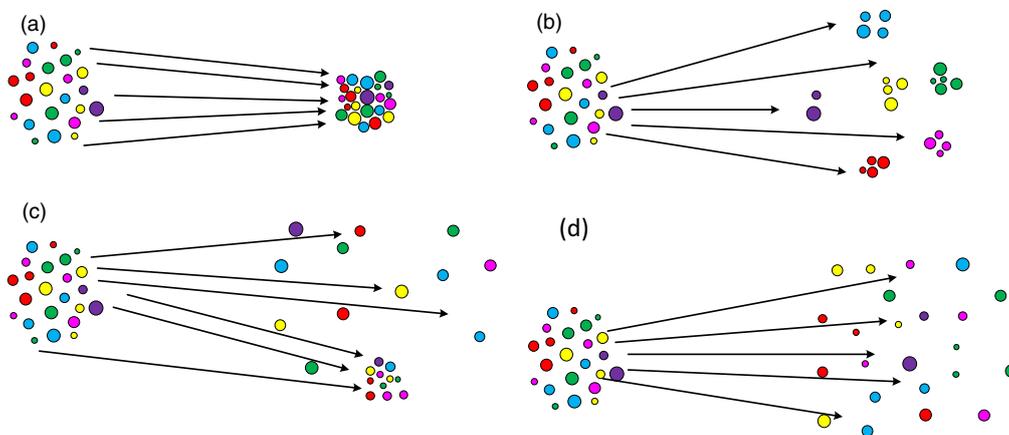


Fig. 2. Performance of a reservoir computer in terms of phase space transformation: too weak activity leads to the death of reservoir dynamics (a), efficient complete (b) and partial (c) separation of input data into selected categories, and an explosion of the phase space (d).



Fig. 3. “An old pond, frog jumps in, splash!”. Turbulence of water induced by a jumping frog (top). The full figure showing the frog in water (bottom). The upper panel illustrates the operational principle of RC: the input can be detected/classified/recognized according to the disturbances of the internal dynamic of the system under consideration.

In simple terms, RC can be related to a famous haiku by Matsuo Bashō:

古池や蛙飛びこむ水の音

(hiragana: ふるいけやかごずとびこむみずのおと;

romaji: *furuike ya kawazu tobikomu mizu no oto*; English: “An old pond, frog jumps in, splash!”). In this beautiful piece of Japanese poetry, an old pond (古池や) symbolizes the reservoir itself, in which the internal dynamics (水の音) can be disturbed by an input or external stimulus, depicted as a jumping frog (蛙飛びこむ). A graphical illustration of this process, as well as of Bashō’s haiku, is shown in Fig. 3.

Even with this simplistic description, the real power of RC in the classification of unknown inputs may be observed. Following Bashō’s analogy, each object thrown into the water (of different masses/shapes) would induce different patterns on the surface. This can be understood as a translation of key properties of objects under consideration into a new observable present in a different phase space and filtration of any irrelevant information (e.g. color in this particular case). The computation power of any reservoir is determined by its complexity: the simplest systems (such as a rock, which was quoted at the beginning of this paper) cannot perform much computation, but highly complex dynamic systems may have high memory volume and high computation capability.^{44,70} Time- and space-multiplexing may further boost their power.^{71,72}

3. Physical reservoir computing

If the reservoir parameters are properly set, RC brings forth several notable advantages. Training in RC is simplified, primarily focusing on the readout part, resulting in swift, stable learning while requiring fewer parameter adjustments than conventional methods. The framework excels in multitasking scenarios, ensuring that tasks can be performed concurrently without interference or the risk of forgetting previous learning.

Additionally, RC provides flexibility in choosing reservoirs, allowing the utilization of diverse dynamical systems. This versatility allows users to tailor the system to their specific needs and preferences. In the context of conventional RC, a stable software framework within RNNs is characterized by interconnected nodes.⁷³ RNNs have the potential to form a reservoir computer, where the weights of the recurrent network are initialized randomly and remain untrained. Meanwhile, the weights of a simple output layer undergo adjustments to train the network for a specific desired output. In contrast, a physical reservoir is a tangible structure, completely separate from neural networks. The former encapsulates a conceptual framework within the software domain, while the latter is associated with tangible real-world infrastructure, serving as a potential candidate for an unconventional computing paradigm. Appeltant and colleagues present a new architecture that minimizes the typically necessary multitude of elements to a single nonlinear node featuring delayed feedback that minimizes the number of vital elements in a conventional reservoir.⁷⁴ Research has actively explored electronic circuits for RC systems, with the aim of low-cost machine-learning devices. Although existing ANNs and neuromorphic circuits can serve as electronic reservoirs, efforts have focused on simpler configurations to enhance energy efficiency, computational speed, and robustness to hardware imperfections.

Achieving a cohesive perspective involves classifying physical reservoir architectures into three types: a single nonlinear node with delayed feedback, network structures, and an excitable continuous medium.⁷⁴ However, formulating a comprehensive design guide for each physical reservoir type remains challenging due to the intricate influence of factors like system architecture, physical attributes, and signal-processing methods on computational performance in physical RC (PRC) systems.

Implementing an entire RNN physically involves mapping the network’s architecture onto a physical system using electronic or photonic components to emulate the synaptic connections and dynamics of the neurons. While technically possible, this approach presents significant challenges due to the complexity and scalability of replicating the intricate connectivity and dynamic behavior of neural networks. Specialized hardware and precise engineering may be required, making it a resource-intensive task.

An alternative approach is to focus on implementing a single node or a small subset of nodes physically, while the rest are simulated in software. This reduces the complexity of the physical implementation while still harnessing the benefits of PRC. Although more feasible than replicating an entire RNN, challenges remain in achieving accurate emulation of the node’s behavior. Ensuring that the physical node exhibits the desired dynamic properties and interactions can be technically demanding.

In both cases, physical constraints, such as noise, non-idealities, and limitations in precision, need to be considered. Despite these challenges, PRC offers the advantage of utilizing the intrinsic dynamics of physical systems for computation, potentially providing unique capabilities compared to traditional digital implementations. Successful implementation requires careful consideration of the physical constraints and desired dynamic properties of the neural network.

The available physical reservoirs currently include diverse implementations, such as single-node reservoirs using analog circuits, RC systems with field-programmable gate arrays, designs for RC devices using very large-scale integrated circuits, ionic liquids,^{75,76)} soft robotics,⁷⁷⁾ sensory devices,⁷⁸⁾ photonics,⁷⁹⁾ spintronics,⁸⁰⁾ nanoscale materials and devices,⁸¹⁾ and quantum reservoirs.⁸²⁾

Certain physical reservoirs show potential to expedite data processing, in contrast to software-based RC. This hardware-centric approach addresses power consumption concerns in machine-learning devices and the RC framework proving adept at cost-effective computation. Edge computing emerges as a paradigm advantageous for real-time tasks, and the significance of machine-learning hardware, including RC hardware, is anticipated to rise, facilitating efficient edge computing for data processing. Despite this, optimizing physical RC systems presents challenges dictated by practical constraints. To effectively perform tasks, a reservoir must have key characteristics. First, it is required to transform nonlinear input signals into a high-dimensional state space using numerous interconnected reservoir nodes with recurrent nonlinear dynamics. Traditionally, hundreds or thousands of such nodes are used for optimal performance. Second, the reservoir's dynamics should exhibit fading memory, influenced by recent inputs rather than those from the distant past, which is crucial for processing temporal sequences.

Well-known models of nonlinear dynamical systems, such as delayed dynamical systems, cellular automata, and coupled oscillators, are generally considered software-based models. They are mathematical abstractions implemented through software or mathematical algorithms. The single-node reservoir by Appeltant et al. was implemented and tested through software simulations, showcasing innovation in architectural design and the exploitation of dynamic properties in a software environment.⁷⁴⁾

Breakthroughs in tasks such as pattern recognition leverage abundant data and high-performance computing devices. Current ANNs lack an ideal hardware platform that fully implements physical neurons and synaptic links, hindering speed and energy efficiency. Ongoing efforts seek novel hardware substrates for ANN concepts that match the efficiency of the human brain in learning and information processing.

Since 2011, electronic nonlinear delay systems, optoelectronic systems, and all-optical nonlinear delay systems have demonstrated fully implemented analog reservoirs, enabling the physical realization of ANNs with a large number of neurons.^{79,83,84)} Photonic delay systems have revolutionized hardware implementation, aligning with the fundamental principles of RC and offering significant degrees of freedom for information processing in analog systems.⁷⁹⁾

The implementation of RC into physical substrates required exploring spatial dimensions for the RNN component. A demonstration using water tank waves showed limited processing performance, while a numerical demonstration with semiconductor optical amplifiers followed.⁷⁹⁾ However, progress in spatio-temporal systems was slow, and RC implementation based on delay systems gained popularity more rapidly.

Nonlinear delay systems have garnered attention for their dual impact on dynamic behavior—either stabilizing or destabilizing systems through parameters like feedback strength.⁷⁴⁾ Previously considered a nuisance, these systems are now recognized as valuable resources, as seen in configurations like semiconductor lasers with delayed feedback. This article explores the use of the rich dynamics of delay systems for processing time-dependent signals, and introduces modifications to RC. RC, inspired by the brain's information processing, excels in challenging tasks like chaotic time-series prediction and speech recognition by mimicking neuronal networks. Its hardware realization, such as fixed connections, enables training difficulties to be overcome, allowing generalization in processing unseen inputs or classifying them based on learned patterns.

PRC offers potential as a computing system if it outperforms traditional computers in terms of speed or energy efficiency. Micro-mechanical oscillators with PRC, especially, could serve dual purposes in sensing and computing, creating innovative devices for distributed sensing or control applications.⁸⁵⁾ Dion et al. present a microfabricated silicon beam with nonlinear dynamics, coupled with a feedback mechanism, demonstrating a micro-electrical-mechanical system reservoir computer trained to process bit streams and classify spoken words. This small and energy-efficient computing device has the potential to function as both a sensor and a computer, encoding information in the mechanical domain.

3.1. Spintronic reservoir computing

In the context of spintronics, various materials and devices play crucial roles in the advancement of computing capabilities. Ferromagnetic metallic films, governed by magnetic anisotropy energy, exhibit nonvolatility, making them suitable for magnetic random access memory devices.⁸⁰⁾ This property is used in PRC within spintronics, particularly through magnetic tunnel junction devices, which detect changes in magnetization via the tunneling magnetoresistance effect. These devices, composed of ferromagnetic and dielectric thin films, provide electrically controllable, nonvolatile, and high-density memory. Their precessional magnetization dynamics offer short-term memory effects and nonlinear characteristics, making them valuable for computational tasks.

Spintronics-based reservoirs, especially those that incorporate skyrmions, present opportunities for integration into existing complementary metal-oxide-semiconductor (CMOS) devices with efficient low-power responses and tunable properties.⁸⁶⁾ Msiska et al. introduced a nanosecond-scale multichannel skyrmion reservoir for pattern recognition, achieving outstanding accuracy in spoken digit speech recognition.⁸⁷⁾ Similarly, Liu et al. proposed a stackable reservoir system using ferroelectric α - In_2Se_3 devices, demonstrating impressive memory capacity and computing capability for advanced neuromorphic computing with emerging two-dimensional materials.⁸⁸⁾ Nakane et al. delved into spin-wave-based RC, emphasizing its potential for edge-computing applications (Fig. 4).⁸⁹⁾ Their exploration involved spin-wave dynamics in a continuous magnetic garnet film with a stripe domain structure, showcasing the interplay between spin-wave behavior and RC capabilities. Spatially arranged electrodes detect spin-vector outputs with diverse

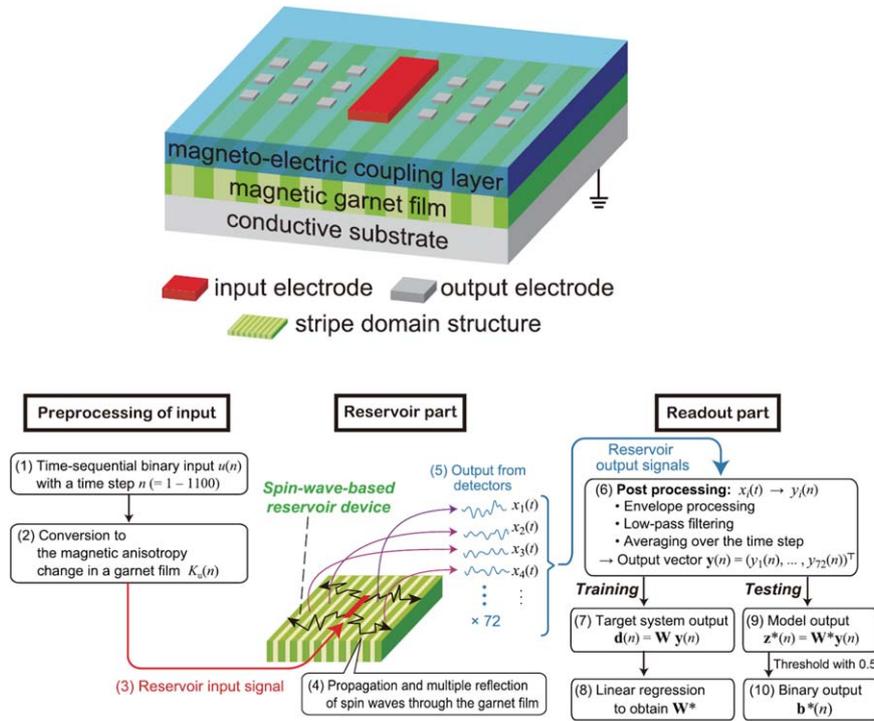


Fig. 4. Structure of a spin-wave-based RC device with a magnetic garnet film, stripe domain structure, and input/output electrodes. The input voltage excites spin waves in the film, propagating to the output electrodes, detected through the output voltages (a). A spin-wave-based RC system with input preprocessing, a reservoir, and a readout. Training follows in Sects. (1)–(8) and testing follows in Sects. (3)–(6) and Sects. (9)–(10). The reservoir includes an input exciter, a magnetic stripe domain structure, and expected spin-wave propagation. The blue curves represent time-series output waveforms at 72 detectors (b). Reproduced from Ref. 89 with permission.

nonlinear characteristics, achieving remarkable performance in temporal exclusive-OR problems.

Within the realm of spintronics, a broad spectrum of materials and devices contribute to the advancement of computational capabilities. From ferromagnetic metallic films with magnetic anisotropy energy to emerging technologies like skyrmion reservoirs and ferroelectric devices, the diversity of materials in spintronics has expanded from traditional solid-state components to innovative liquid-based systems. For example, the utilization of a Fe_3O_4 water-based ferrofluid (FF) by Crepaldi et al.⁹⁰ introduces a dynamic element into this landscape, drawing parallels with solid-state memristors (Fig. 5). This FF exhibits intricate behavior influenced by Brownian motions and electrical polarizability of surfactant molecules, offering both fading memory and long-term plasticity. Despite these advantages, challenges in maintaining its dynamics over time are addressed through a specific “reset” sequence, which may also be considered as a kind of system programming or training. This tailored approach mitigates variations and maintains stability in the FF’s behavior, thereby contributing to the exploration of novel computing elements in the evolving field of spintronics. These diverse materials, from solids to liquids, showcase the versatility and potential of spintronics in shaping the future of computational technology.

3.2. Ionic liquids

Ionic liquids (ILs), renowned for their expansive potential window and tunable properties, represent a compelling avenue for systematic investigations in various scientific domains. In the realm of spintronics, Matsuo et al.⁷⁶ have introduced a groundbreaking physical reservoir device that harnesses faradaic currents arising from the redox reactions

of metal ions within ionic liquids. Through the application of triangular voltage pulses, which symbolize binary sequences, the study systematically evaluates the impact of faradaic current on short-term memory and parity check tasks. The findings not only underscore the advantages of faradaic current for short-term memory, but also emphasize its role in nonlinear conversion within physical reservoir devices. This research provides valuable information for the design and control of such systems, as illustrated in Fig. 6.

In a related study by Sato et al.,⁷⁵ the efficiency of information processing in machine learning for the classification of electrocardiogram signal waves was enhanced using 1-butyl-3-methylimidazolium bis(trifluoromethylsulfonyl) amide ([BMIM][Tf₂N]) containing 0.4 M $\text{Cu}(\text{Tf}_2\text{N})_2$. The evaluation processes for short-term memory (STM) and nonlinear auto-regressive moving average (NARMA2) tasks within IL-based physical reservoir devices (IL-PRDs) are meticulously depicted in Fig. 7. This work not only contributes to the understanding of information-processing mechanisms in IL-based reservoir devices but also exemplifies the potential of ILs, such as [BMIM][Tf₂N], to optimize machine-learning tasks through their unique electrochemical properties.

3.3. Memristors

Memristors, known for their resistive switching properties, have made significant strides in integrating ANNs within RC systems, using their dynamic and nonlinear characteristics. Challenges persist in fine-tuning reservoir states and minimizing additional read operations for enhanced system performance and speed. Reservoir computers comprise a dynamic reservoir and a static readout. Tong et al.⁹¹ applied two parallel memristive devices as a reservoir component in

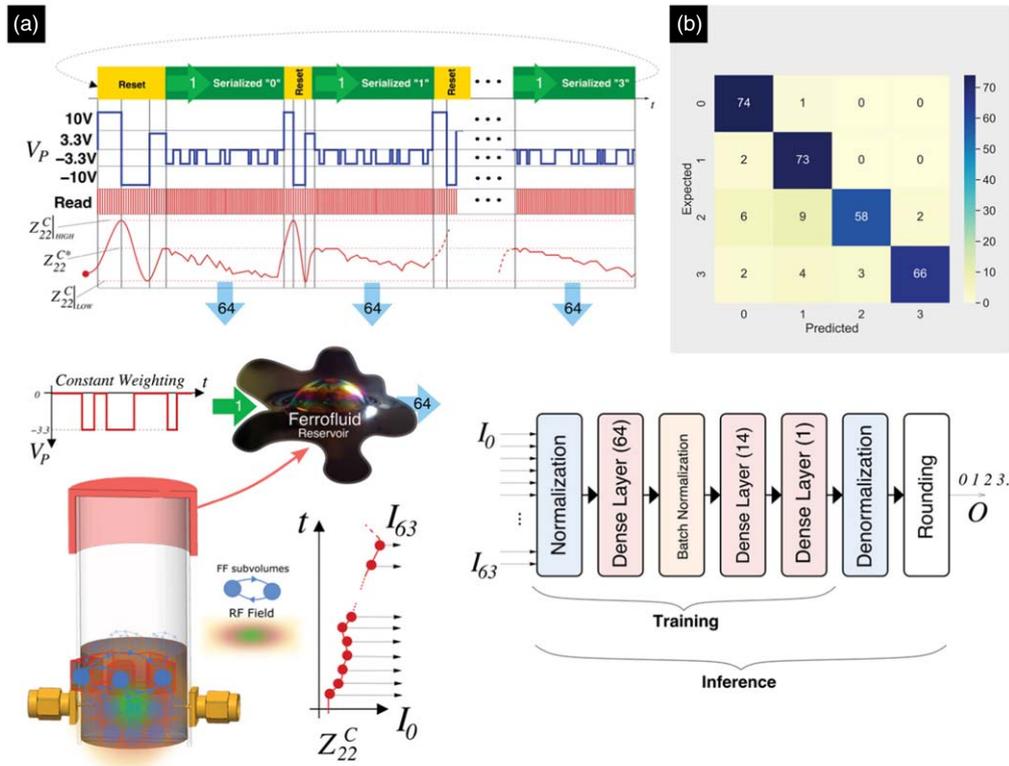


Fig. 5. The stimulus setup for the PRC tests includes parallelized outputs in the readout neural network layer, detailing the impact of each pixel value. It also covers information about the neural network layer and the conceptual liquid reservoir (a). The confusion map shows the real-time classification testing of digits 0 to 3 using the trained neural network (b). Reproduced from Ref. 90 with permission.

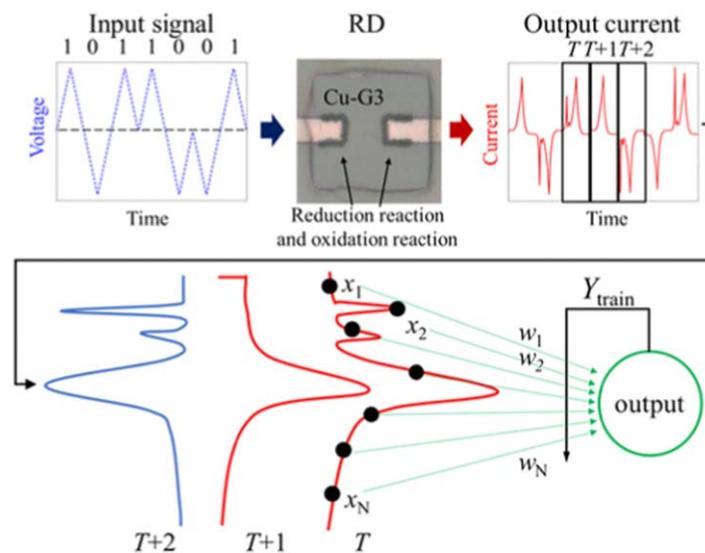


Fig. 6. Illustrated signal-processing flow for physical reservoir computation. Using a triangular pulse as the input signal, output current values (x_1, x_2, \dots, x_N) at each time step were generated by redox reactions at the copper-2,5,8,11-tetraoxadodecane (Cu-G3)/electrode interface. Then, these values were input to N virtual nodes. The learning process, determining the weights (w_1, w_2, \dots, w_N), was performed through linear regression using Y_{train} as the training data. Reproduced from Ref. 76 with permission on the CC-BY 4.0 licence.

waveform and electrocardiogram classification tasks. Their study proposes a basic circuit configuration for pattern classification by physical reservoir, emphasizing the potential benefits of memristive systems with intrinsic nonlinearity. The nonlinearity and input history dependence are explored by replacing memristors with normal resistors, revealing the superior performance of memristors in achieving high

computational accuracy. The study also investigates the impact of using a single memristor, indicating that at least two memristors with different responsive characteristics are essential for optimal performance in pattern classification tasks. The memristive circuit reservoir, comprising two memristors and three additional circuit elements (a resistor, an inductor, and a capacitor), is identified as a minimal but

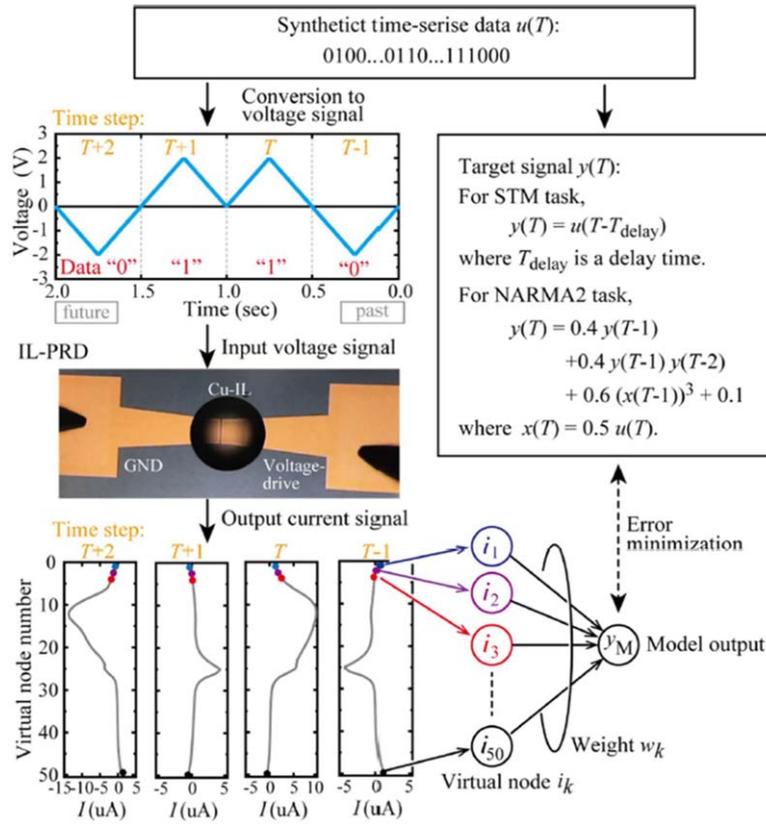


Fig. 7. Evaluation processes for STM and NARMA2 tasks involve applying input voltage to IL-PRDs and conducting linear regression analysis with output current values. The binary data (0 and 1) is transformed into a triangular voltage pulse train and applied to the IL-PRD. The input dataset for linear regression is generated using a virtual node method with a node number $k = 50$. Reproduced from Ref. 75 with permission on the CC-BY 4.0 licence.

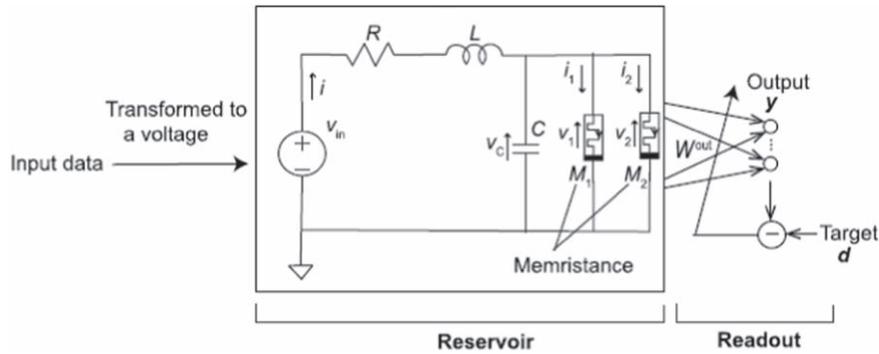


Fig. 8. Schematic representation of the circuit of an RC system with two parallel memristors, a resistor, an inductor, and a capacitor. The reading part is a simple linear classifier. Reprinted from Ref. 91 according to the CC-BY-NC 4.0 licence.

effective configuration (Fig. 8). The description of the memristive circuit reservoir depicted in Fig. 8 is depicted through the following equations:

$$C \frac{dv_C(t)}{dt} = i(t) - i_1(t) - i_2(t) \quad (5)$$

$$L \frac{di(t)}{dt} = -Ri(t) - v_C(t) + v_{in}(t) \quad (6)$$

$$v_C(t) = v_1(t) = v_2(t). \quad (7)$$

In the given context, where C represents the capacitance of the capacitor, $v_C(t)$ signifies the capacitor voltage, $i(t)$ denotes the current at the voltage source, $i_m(t)$ represents the electric current at the m th memristor (where $m = 1, 2$), L is the

inductance of the inductor, R is the resistance of the resistor, $v_m(t)$ [equal to $v_C(t)$] stands for the voltage at the m th memristor (for $m = 1, 2$), and $v_{in}(t)$ is the input voltage at the voltage source. The correlation between $i_m(t)$ and $v_m(t)$ relies on the intrinsic characteristics of the m th memristor.

During the readout process, the voltage $v_m(t)$ and current $i_m(t)$ are measured within the time interval $\langle 0, T \rangle$ for $m = 1, 2$. The evolution of memristance, denoted as

$$M_m(t) = \frac{v_m(t)}{i_m(t)} \quad M_m(t) = \frac{v_m(t)}{i_m(t)} \quad \text{for } t \in \langle 0, T \rangle, \quad (8)$$

is subsequently computed and converted into discrete-time signals $\{x_m(n) \mid n = 1, \dots, L\}$ through sampling. These discrete-time signals are then linearly combined with output weights $\mathbf{W}_{\text{out}} \in \mathbb{R}^{N_y \times 2}$ to generate the system output, which

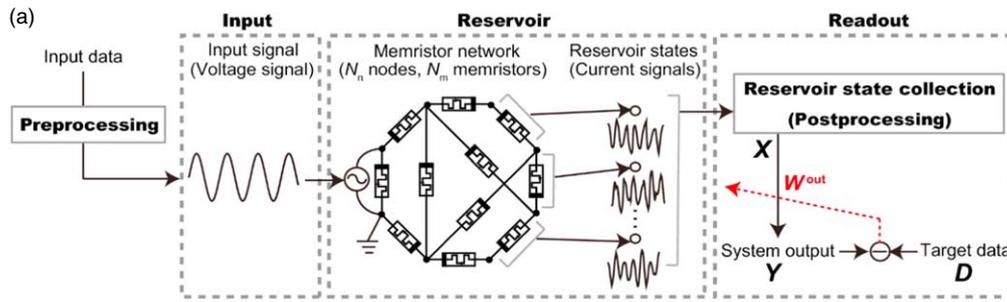


Fig. 9. The RC system utilizes a memristor network with preprocessing, input, reservoir, and readout components. It processes time-series data, measures current signals in the reservoir, and optimizes the output weight matrix W_{out} through linear regression in training. Reproduced from Ref. 92 with permission.

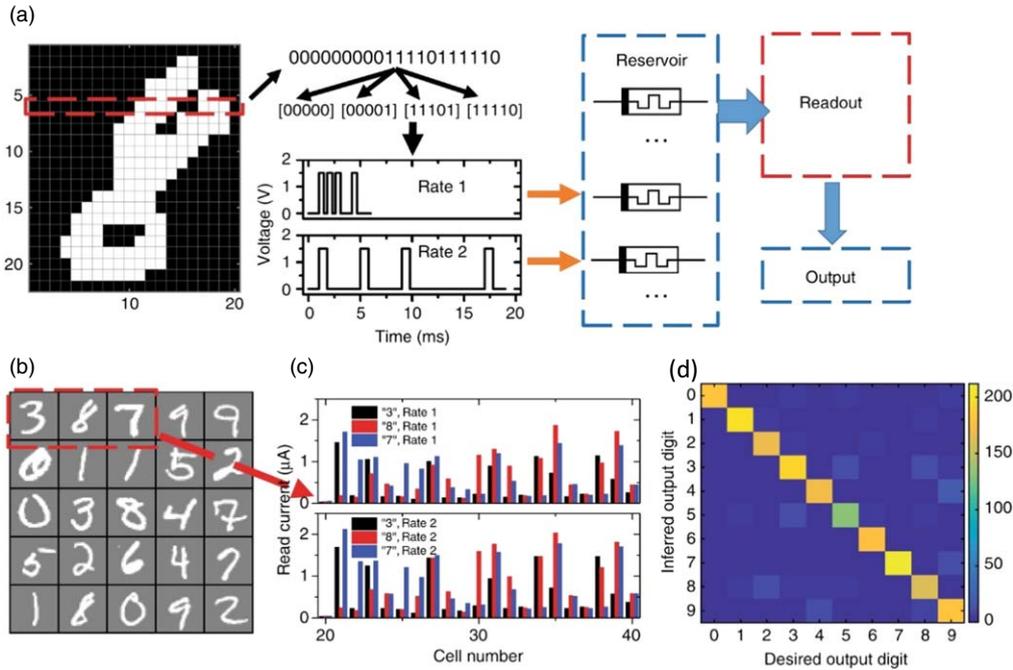


Fig. 10. Handwritten digit recognition using a memristor array RC system involves converting images into pulse streams and feeding them into the reservoir at different rates (a). Examples from the modified National Institute of Standards and Technology (MNIST) database are shown (b). The reservoir states corresponding to these examples at two input rates reveal significant differences (c). The false color confusion matrix demonstrates a recognition accuracy of 88.1% from the 88-memristor reservoir (d). Reprinted from Ref. 93 with permission.

can be expressed as follows:

$$y(n) = \mathbf{W}_{out}x(n) \quad y(n) = \mathbf{W}_{out}x(n) \text{ for } n = 1, \dots, L. \quad (9)$$

Here, $y(n)$ denotes the output vector $(y_1(n), \dots, y_{N_y}(n))^T$ and $x(n) = (x_1(n), \dots, x_{N_x}(n))^T$ is the reservoir state vector.

Future work involves experimental validation, assessment of energy efficiency, and further improvement of classification accuracy through the exploration of different memristor models.

Memristors show advantages, including nonlinearity and input history-dependent reactions, rendering them highly suitable for tackling challenges associated with linearly inseparable problems in time series data analysis.⁹² Previous studies have underlined the promising capabilities of memristive reservoirs in temporal pattern recognition. This potential manifests itself in two primary categories: memristor networks (Fig. 9) and memristor arrays (Fig. 10).

Przyczyna et al. introduced a network of nanodevices comprising four memristors and a differential amplifier for the detection of epileptic seizures (Fig. 11). Feedback loop

evolution enhances classification accuracy, and signal transformation alters complexity parameters, contributing to improved classification scores.⁹⁴

Among memristor-based RC systems, a variety of possible solutions, based on metal oxides, have been proposed through the literature. This is mostly because nonstoichiometric oxides (from TiO_2 , WO_2 , HfO_2) have long been known to demonstrate memristive properties.^{95,96} Additionally, they are characterized not only by a nonlinear response but, as their mechanism is based on charge carrier migration, they possess the volatility necessary for RC systems.

In some cases, cross-points (memristors) are made from a multilayer oxide configuration—such as $Ti/TiO_x/Pt$.⁹⁷ The RC system is denoted as a dynamic one—the signal is incorporated as a temporal sequence. In tasks such as waveform classification, the working configuration consists of several parallel memristor-based reservoirs, controlled by a mask process that tunes critical parameters. The system achieves good performance, with a low word error rate of 0.4% in spoken digit recognition and a normalized root mean square error of 0.046 in time-

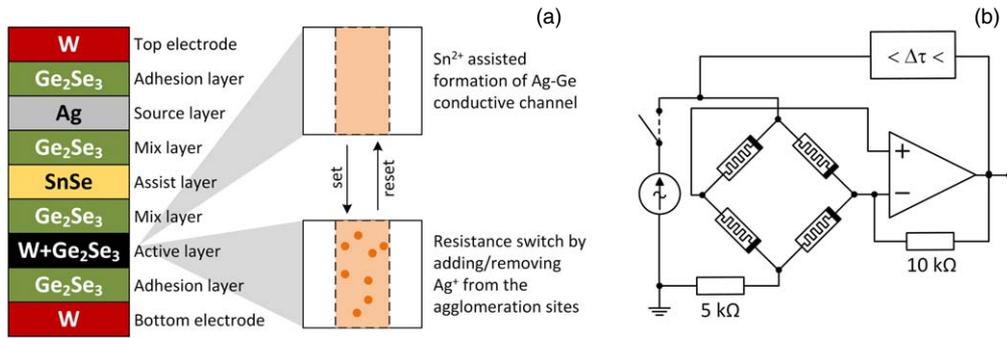


Fig. 11. Structure and switching mechanism of KNOWM memristive devices (a) and a network of memristors applied in RC (b). Adapted from Ref. 94.

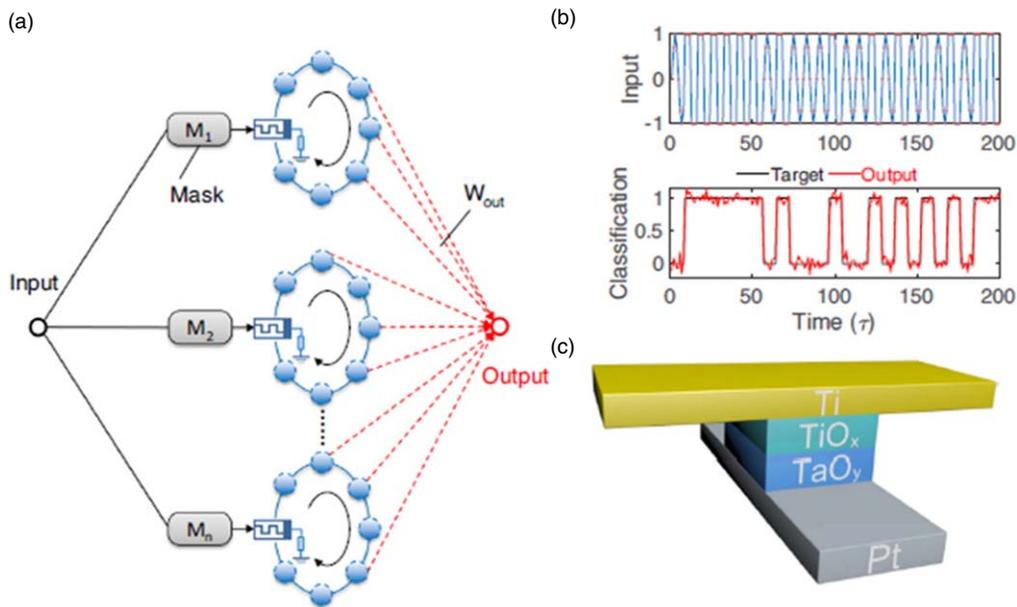


Fig. 12. Schematic of a dynamic memristor-based parallel RC system, where the mask sequences are different for every single memristor RC unit. The output is the linear combination of all states of the reservoir. In our experiment, this parallel RC system is realized by testing a single memristor in multiple cycles (a). An exemplary input signal alongside results; signal classification to one of the two categories—either sine or square waves. Proper results required optimization of mask length and the number of reservoirs that will run in parallel. The output of all reservoir states is taken from their linear combination (b). Device cross-point structure (c). Reproduced from Ref. 97 with permission.

series prediction of the Hénon map for time-series prediction tasks, surpassing most previously reported hardware and software-based reservoir solutions. The schematic in Fig. 12 depicts the configuration of a dynamic memristor-based parallel RC system and also one of the results for the waveform classification task.

3.4. Nanotube/nanowire reservoirs

Realization of a brain-like connection type is straightforward in nanowire networks as they already consist of interconnected nonlinear processing nodes. Such a blueprint of design allows for effective projection of input signals to a higher-dimensional feature space in case processing centers experience fading memory property. This in turn allows for the use of these physical reservoirs in an RC approach to machine learning. What is unique is the static part of the system, inseparably connected with its design and connectivity map between the nodes, although the connection weight updates possess a dynamic characteristic.

Zhang et al. delve into the development of a memristor-based PRC, a framework inspired by the brain's computational principles.⁹⁸ The key strategy involves constructing

neural network-like random networks to enable efficient and energy-saving information transfer. The study explores the feasibility of *in materia* PRC systems through the demonstration of physical systems such as silicon-based photonic chips^{99,100} and atomic switch networks (ASNs).^{101,102} A physical reservoir incorporating ASNs is considered a highly promising framework for the implementation of hardware-based RC. In such a setup, the emergent dynamics are generated by the entire network system rather than by individually tuning its elements. To achieve this, a typical approach involves coating a self-assembled metallic nanowire network with memristive materials. Consequently, each junction within the network functions as an atomic switch, similar to a biological synapse. Challenges arise in creating highly random, interpenetrating networks with nanoscale switching properties, necessitating unconventional processing methods for self-assembly and self-organization at nanoscales. The application of nanowire networks coated with organic and inorganic materials (such as Ag₂S, polyoxometalate (POM), porphyrin-POM, and polyvinylpyrrolidone) in RC, considering factors like activation modes and

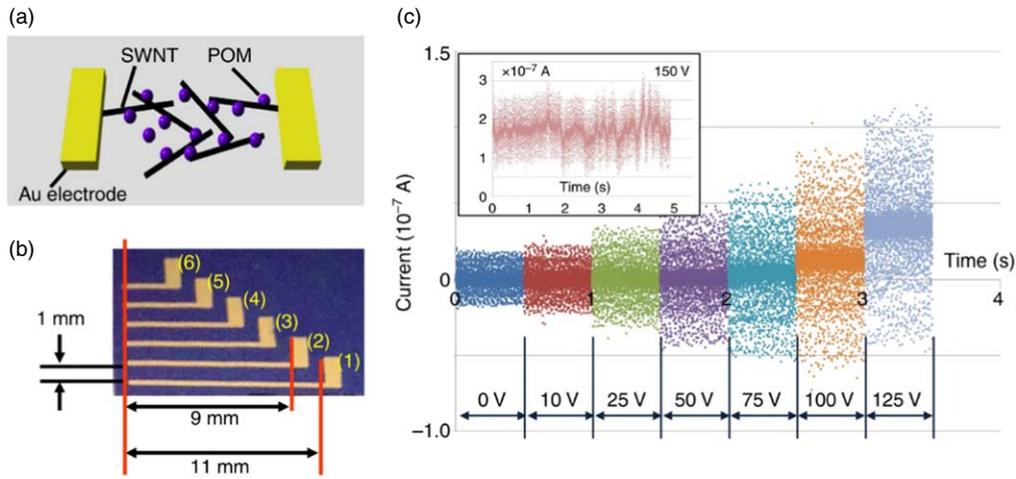


Fig. 13. Enhancement and noise generation in the POM-SWCNT network are illustrated in the experimental setup. (a) The network schematic depicts terminal electrodes (depicted as yellow cuboids), SWCNTs (represented by black tubes), and POM particles (depicted as purple spheres). (b) A photograph of the substrate featuring six terminal electrodes is included, with the entire substrate coated with the POM-SWCNT complex. (c) The sampled current density over time, indicating the current magnitude distributions, is presented in a plot where the bias voltage increases incrementally from 0 to 125 V across the electrodes in sample A. Reproduced from Ref. 112 with permission.

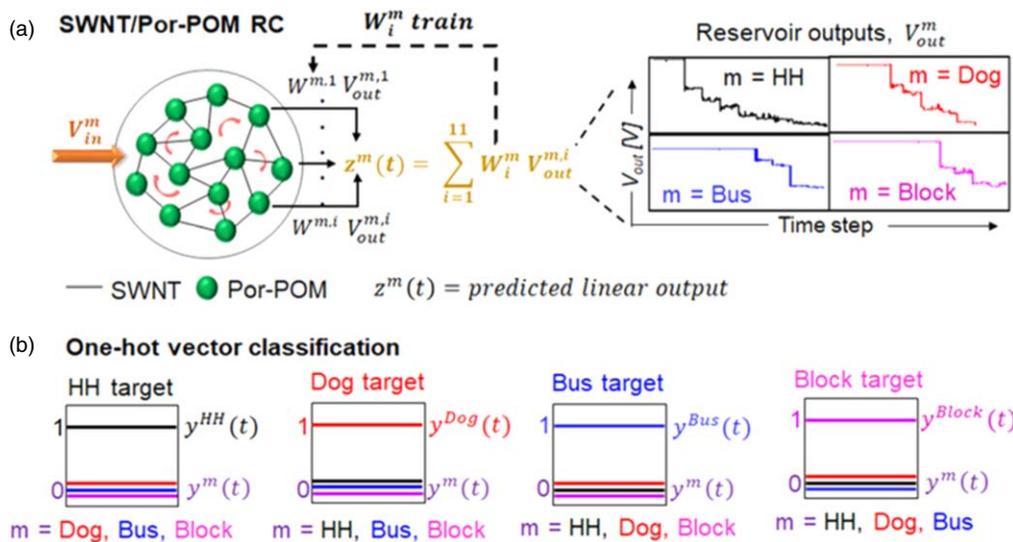


Fig. 14. (a) Time-series inputs from various objects (hedgehog, dog, bus, and block) are individually introduced into the SWCNT (black line)/Por-POM (green circles) reservoir featuring recurrent connections (red arrows, left). The voltage readouts of the i output pads for each object are gathered, as illustrated on the right (result for one electrode pad). (b) One-hot vector encoding is used for binary classification. Each square box with the lines inside represents the target signal. The one-hot vector, as the object is truly predicted, is given a vector value 1 while the others are given a vector value 0. Reproduced from Ref. 113 with permission.

network adaptability between ordered and chaotic dynamics, has been widely investigated. ^{18,60,103–108}

Moreover, the integration of POM-decorated single-walled carbon nanotubes (POM-SWCNTs) in neuromorphic devices for computing has demonstrated the use of multiple redox states of POMs and electrochemical reactions to generate spontaneous spikes and noise, which can result in improved performance of PRC (Fig. 13). ^{109–112}

Another of the examples, this time by Tanaka lab, is also a system that constitutes a real RC platform. ¹¹³ It is a recurrent network formed by SWCNT-porphyrin POM (Por-POM) complexes. This architecture demonstrates properties such as nonlinearity and higher harmonic generation. The system can also process information in an “edge of chaos” manner. The RC system is benchmarked against a supervised object classification task using tactile sensory input datasets from

the Toyota HSR. The sensor measured the change in gripper angle (θ) as a function of applied forces, providing tactile information to classify mixed hardness and softness, including a bus, block, dog, and hedgehog [cf Fig. 14(a)]. Objects were sequentially presented to the device, and the resulting outputs of all electrode pads were collected for training and testing, utilizing a supervised regression model in Python with a hot target vector [cf Fig. 14(b)].

The dominance in the literature of nanowire networks derives directly from their constitution—it is relatively easy to just add linear classificatory in place of device output. Zhu et al. ¹⁰⁶ were able to implement an RC system, consisting of a metallic nanowire network, where at the cross-point junctions electrochemical metallization created nanofilaments, asserting memristive switching abilities of the system. The physical form of the device was

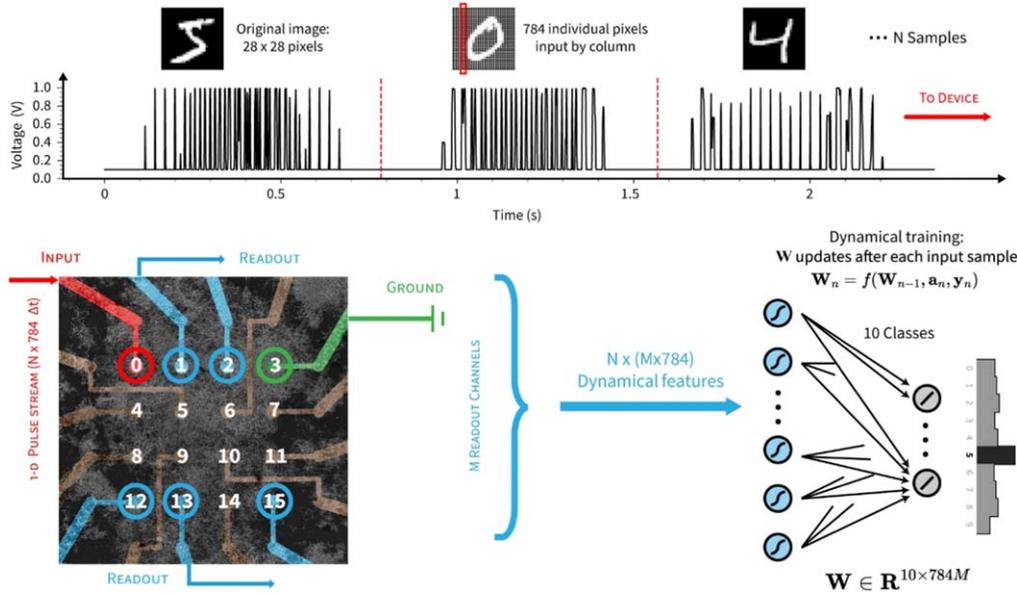


Fig. 15. An RC device design and the principle of operation. The nanowires used in the device are Ag_2Se . A scanning electron microscope (SEM) image of the 16-electrode device features different electrodes: channel 0 (red), drain, channel 3 (green), readout electrodes, channel 1, 2, 12, 13, 15 (blue), and unused electrodes. On the right, the readout voltages ($N \times M \times 784$ dynamical characteristics) are fed into an external linear classifier, where the weight matrix (W_n) for $M \times 784$ features per digit sample is updated after each sample, with the corresponding class as the target output (digit 5 displayed as an example of the classification result). Reproduced from Ref. 106 with permission.

multielectrode arrays (shown in Fig. 15), patterned with Cr/Ti (5 nm) and Pt (150 nm). Then the actual functional material was made from synthesized Ag_2Se nanowires. This study showcases the effectiveness of online learning with nanowire networks, achieving a 93.4% accuracy in image classification tasks, highlighting the benefits of recursive least squares for faster convergence and reduced numerical errors. The computing system optimally utilizes the rich dynamical features of recurrent networks, changing the focus of the training strategy to a linear output layer only.

Ag_2Se nanowires also constitute other types of physical systems, such as those presented by Kotooka et al., yet they are not always recognized by name as RC systems.¹¹⁴ This happens even though the reported nonlinear and high-dimensional properties are shown and the device executes classification tasks.

Another physical realization—a self-assembled metal (Ag) nanowire network with an MNIST type of test—was presented by Milano et al. The network, as with previous systems, features random connections among multiple nonlinear memristive elements.¹⁰² The reservoir layer is constructed with a low-cost bottom-up approach using an interconnected memristive network. It has both nonlinear dynamics and fading memory properties and enables spatio-temporal processing of multiterminal inputs through functional synaptic connectivity. The readout layer is implemented by mapping synaptic weights associated with each output neuron using an array of TaO_x resistive random-access memory (ReRAM) cross-point junction devices, where resistive switching relies on the formation/rupture of a filament. This structure is presented in Fig. 16. In conclusion, this study introduces a fully memristive RC architecture: self-organizing (bottom-up) nanowire (NW) networks combined with top-down ReRAM devices for general purpose intelligent systems. The exemplary computing capabilities of the

physical reservoir of the NW network were demonstrated on the basis of MNIST digits classification.

Despite superior computing merits, the application of NW-based RC to real-life applications requires continuous efforts due to existing challenges such as material compatibility, the dynamic complexity of traditional CMOS technology, and achieving optimal performance.¹⁸

3.5. Reservoirs from nanoparticles

Other types of nanostructure, nanoparticles (NPs), sometimes still with the support of elongated connectors (NWs), can be implemented in physical RC systems. Some of the recent examples are of Ag- Ag_2S core-shell NPs.¹¹⁵ In this study, the examined system utilizes the dynamics between constituting NPs, generating in turn high dimensionality within the echo state property. These elevate the accuracy of target waveforms up to 99%.

The same group also implemented $\alpha\text{-Fe}_2\text{O}_3$ /titanium bismuth oxide compounds using sol-gel synthetic procedures.¹¹⁶ The computational efficiency of this RC device was evaluated by introducing varying levels of noise injection. Characterization and identification of functional structures indicated the presence of $\alpha\text{-Fe}_2\text{O}_3$, TiO_2 , as well as $\text{Bi}_4\text{Ti}_3\text{O}_{12}$ NPs. The accuracy of the predicted results was enhanced by additional white noise of a small voltage intensity, added to the input. The system had desired nonlinearity and allowed for waveform generation tasks/tests with 87% accuracy. Noise injection is similar to the phenomenon observed in the biological brain—fluctuations of the external world.

Universal concepts of device design for both of the above solutions are presented in Fig. 17.

4. Enhancing dynamics: reservoirs with drive

All recently reported RC systems utilize the simplified approach—the only stimulus reaching the reservoir is the input data. Initial considerations, as well as numerous

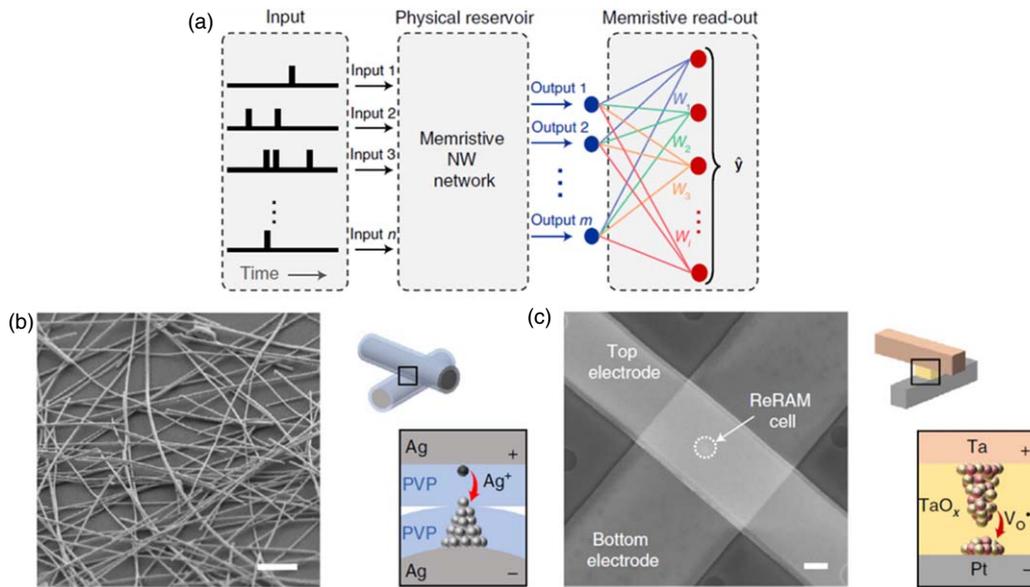


Fig. 16. Schematics of the device. The input is encoded as pulse streams, and, secondly, is passed to the NW network physical reservoir, and in the end is classified by the hardware (resistive switching based) readout neural network. (b) SEM image and schematic drawing of a reservoir layer. In place of NW junctions, the formation and rupture processes of metallic Ag filaments occur. (c) SEM image and schematic drawing of a TaO_x resistive switching cell at the metal electrode cross-point in place of a junction formation, and rupture processes of the vacancy-based conductive filament occur. Reproduced from Ref. 102 with permission.

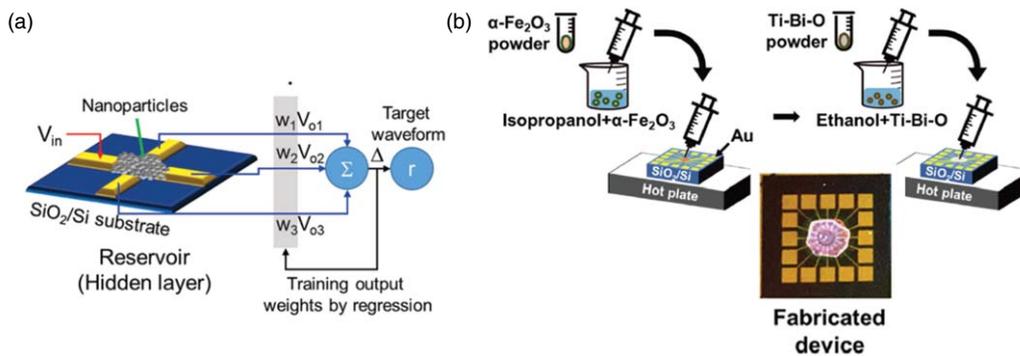


Fig. 17. Schematic of the NP-based RC device with the connection to the trained output layer for a specific task (a). Processing of the NPs at the electrode-patterned substrate surface alongside an image of the fabricated device (b). Reproduced from Ref. 116 with permission.

theoretical works, emphasize the importance of the drive—an additional signal that influences the reservoir along with the input data. This approach has been developed mainly for sensing applications: drive stimulates the reservoir using different physical channels, so direct interference is avoided; e.g. metal ions were considered as an input, whereas light (or voltage in some cases) pulses were considered as a drive.^{36–39,117,118} To date, this approach has not been applied in any physical implementations of RC bar one: the FF-based computing systems reported in 2023 by Crepaldi et al.⁹⁰ In this particular case, however, the drive has been applied not along with the data to be processed, but before, in order to prepare the device for a particular task. This can be considered as a first step towards programmable reservoir computers, in which not only the readout layer is trained, but also the dynamics of the reservoir can be put into a desired state prior to computation.

A recent paper by Shibata et al. uses the term “physical masking,” which in principle is the application of a drive to a physical reservoir.¹¹⁹ The device under study is a redox/ion-driven transistor with a LiCoO₂ channel, Li₃PO₄ gate dielectric, and two independent drain electrodes [Fig. 18(a)].

An input signal is supplied to the gate electrode, whereas the second drain can be used as a drive electrode (referenced by the authors as physical masking), as shown in Fig. 18(b). The absence of the drive input application of voltage pulses results in relaxation-type dynamics [Fig. 18(b)], which is, however, rich enough to be engaged in computation, e.g. the prediction of waveforms following the second-order dynamic equations. Application of the saw tooth-like drive results in apparently uniform, featureless output [Fig. 18(c)], the dynamics of which, however, are much richer and present higher variability, as reflected by the time course of the reservoir state evolution [Fig. 18(d)]. This simple example demonstrates the importance of the drive, which enriches the reservoir dynamics and contributes to the increased dimensionality of the phase space of reservoir states. This in turn directly translates to the separability performance and may also affect the generalization features of any RC system.

However, application of drive signals is not a trivial task, as it directly affects the reservoir phase space and may move the systems into an explosive mode [cf Fig. 2(d)] if either the sensitivity of the reservoir to the drive is too high or the

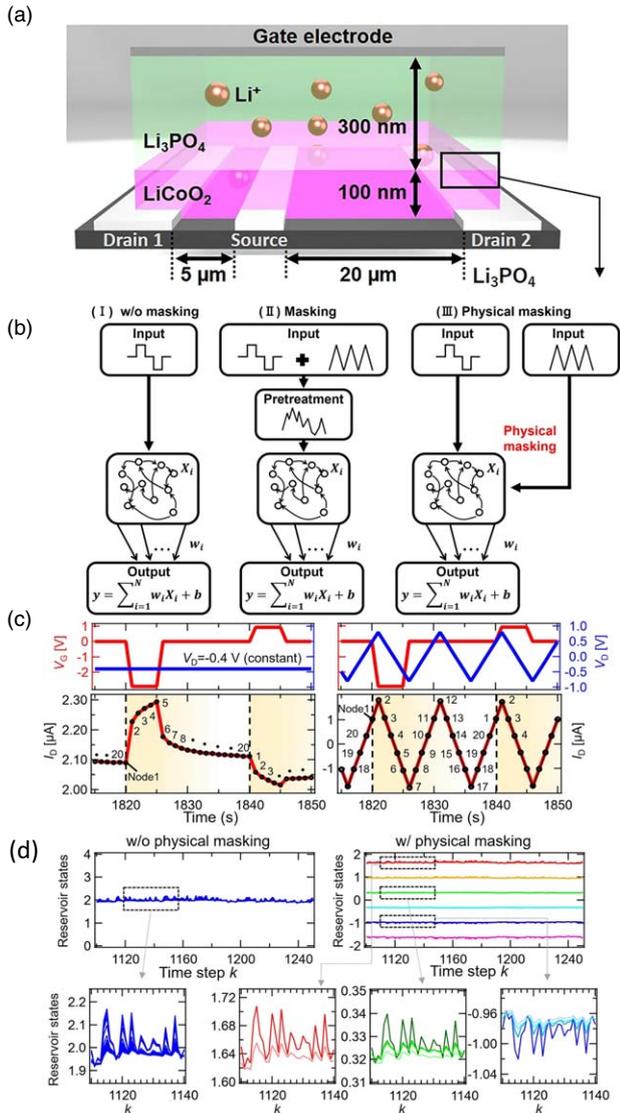


Fig. 18. Schematic image of the LiCoO_2 -based redox ion gate reservoir (a). General scheme of the unmasking reservoir, with digital masking applied to the input data and the unmasking reservoir with physical masking (or a drive) (b). Gate voltages and drain currents of devices without (left) and with physical masking (right) (c). Time series corresponding to reservoir states without (left) and with (right) physical masking/driving (d). Reproduced from Ref. 119 with permission.

spectral radii of reservoir matrices are too high. So far, there are no good universal search algorithms for drive signals; however, Athanasiou and Konkoli derived periodic drive signals for memristive reservoirs that showed good performance in classification of electrocardiography (ECG) signals.¹¹⁷

In light of Eq. (3) it can be seen that the input and the drive can interact both indirectly (like in photoelectrochemical sensors) or, assuming relations (10–11) applied to Eq. (3):

$$\beta \approx \gamma \quad (10)$$

$$\mathbf{W}_{\text{in}} \cong \mathbf{W}_{\text{drive}}. \quad (11)$$

The input and the drive may be undistinguishable in the extreme case.

The main obstacle in the application of drive-operated RC systems is the difficulty in finding the appropriate drive signal. However, depending on the task, some solutions can

be found. For example, a search for a well-defined pattern can be performed by applying a drive, which is related to the patterns in question by some symmetry rules. In the simplest case, the drive that will be a negative of the searched signal will be suitable for the task. Then, any small difference between the target patterns and the input will be detected, and the reservoir will effectively measure the differences between the input and the target pattern. In this particular case, further simplification may be achieved; just observation of the reservoir dynamics, without a trained output layer, may be sufficient for the task. This may be useful in speech recognition, in particular for classification phonemes. This will, however, require either sequential operation of the same reservoir with different drives (time consuming) or operation of numerous reservoirs in parallel (or complex reservoirs with numerous inputs and drives, which automatically can be translated to parallel operation of simpler reservoirs).^{69,120} Another alternative would be exploitation of reservoir evolution in time, as in single node echo state machines and hierarchical structures based there [Fig. 19].^{121–123} This approach has already found a couple of physical implementations^{94,124,125} and is pretty common in photonic systems.^{99,126–133} It is applicable to chemical sensing as well, even in very simplistic cases, as minute changes in the impedance of the layer at the electrode are translated into differences in signal evolution.^{36,40} Most of these systems operate, however, without a drive signal. This hierarchy was also implemented in thin-layer memristor-based reservoirs for signal amplitude discrimination.^{124,125}

Some time ago, numerical simulations of the analysis of musical harmony with single node echo state machines was reported.^{134,135} In these studies, two sine wave signals were applied to the reservoir feedback loop, and their evolution

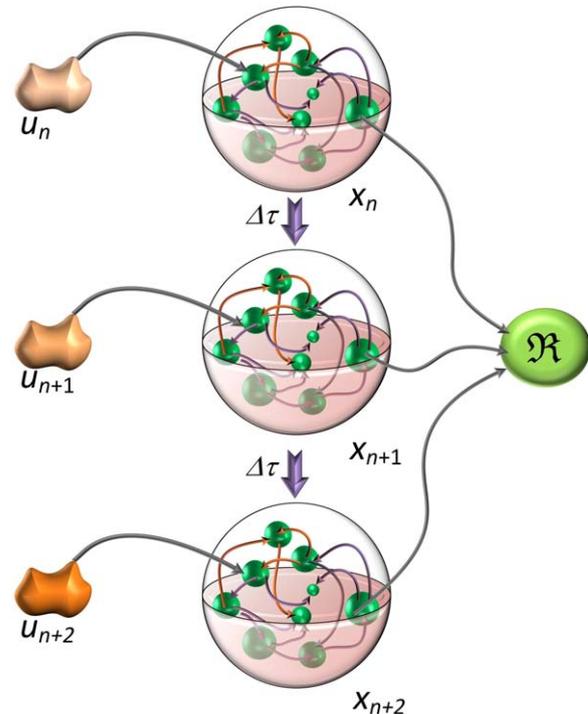


Fig. 19. Concept of a hierarchical reservoir. States of the same physical system at different moments in time (and upon stimulation with the same or different inputs) are considered as different computational devices. The cost of device complexity is paid by the time required for computation.

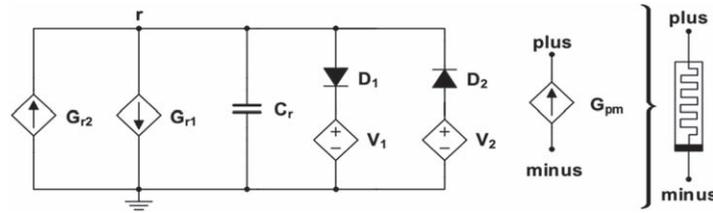


Fig. 20. An equivalent circuit of the memristor model used in Vourkas and Sirakoulis's study. Adapted from Ref. 137.

was followed. In the study of musical harmony, both signals were considered as inputs, but formally one of them could be considered as a drive. Therefore, here we present preliminary results on numerical simulations of a closely related circuit (memristive Wien bridge¹³⁶ with operational amplified and capacitive circuits). The simulator, implemented as a SPICE object in Multisim, is based on a memristor model by Vourkas and Sirakoulis (Fig. 20 and Table I).¹³⁷

The full circuit of the bridge synapse, with an internal capacitor and capacitive coupling to the operational amplifier, is shown in Fig. 21(a). An ideal amplifier with an internal input resistance of 10 MΩ and output resistance of 10 Ω has been used. The variable resistor was set to 2 kΩ for the highest nonlinearity of the response, evaluated on the basis of Fourier analysis of the output with sine input. No feedback resistor was used for the amplifier, but despite that the whole circuit performed both partial integration (charge storage at the central

capacitor) and differentiation (capacitive coupling of the amplifier). The performance of the circuit was tested for a single-frequency sine input within 30–120 Hz. It was found that this type of input induces a series of fading self-oscillations, with subsequent echoes of a heavily distorted sine character.

The application of a two-sine function generator with proper output amplitude, operating at the same and at different frequencies, had no effect on the behavior of the circuit: fading echoes have been observed for almost all frequency combinations [Fig. 21(b)]. OriginPro's envelope function has been used to compare fading profiles [Fig. 21(c)]. It was found that the only input combination that can be differentiated from the others on the basis of fading profiles is the situation in which two input signals fulfill condition (12):

$$f_2 = 2f_1, \tag{12}$$

i.e. they form an octave. This observation is related to the previous report, in which full characterization of musical harmony was performed.¹³⁴ Signals that constitute an octave exhibited much longer persistence in the feedback loop, generating up to 12–15 high-intensity echoes, and then fading abruptly out [Figs. 21(c), 21(d)]. Any other combination of input/drive frequencies resulted in a rapid decrease in echo intensity [Fig. 21(c)].

The persistence has been observed for any pair of signals fulfilling condition (7) within 40–100 Hz. A narrow frequency window is a consequence of the memristor model, which is optimized for low frequency applications.

With this approach, frequency deviations of ca. 0.5 to 1.5 Hz were detectable within the given frequency window. Therefore, in light of previous considerations of a proportional integral derivative controller, which also combines integrating and differentiating components and can be regarded as a primitive form of reservoirs,¹³⁸ we attempted to perform phase separation of sine signals as well as waveform discrimination, which was also demonstrated in an *in materia* device, but without delayed feedback.^{139,140} Therefore, a pair of signals (100 Hz input and 50 Hz drive) were applied to the reservoir, and the input signal was subjected to phase shifts from 1 to 180 degrees [Figs. 21(d), 21(e)].

The signal with a small phase shift ($\varphi < 3^\circ$) resulted in significant persistence of the signal [Fig. 21(e)], while larger phase shifts resulted in rapid decay of the signal intensity. This observation indicated a very high sensitivity of a relatively simple reservoir circuit to even subtle changes in the input signal. It also demonstrates the power of a drive; application of a properly designed signal enables even readout-free operation of the reservoir and one-hot classification of inputs on the basis of output amplitude after a given evolution time.

Table I. A memristor description in PSSpice syntax according to Ref. 137.

```
.SUBCKT mem1 plus minus
.SYNTAX PSSpice
*Parameters' values
.param rmin = 100
.param rmax = 390
.param rinit = 390
.param alpha = 40000
.param beta = 10
.param gamma = 0.2
.param VtR = 1.5
.param VtL = -1.5
.param yo = 0.0001
.param m = 82
.param fo = 310
.param Lo = 4
.param Dbreak = 1
.param Dbreak = 1
Gr1 0 r value = {Dr_dt(V(plus)-V(minus))*st_f(-(V(plus)-V(minus)))}
Gr2 0 r value = {Dr_dt(V(plus)-V(minus))*st_f(V(plus)-V(minus))}
D1 k r {Dbreak}
V1 k 0 {rmin}
D2 r g {Dbreak}
V2 g 0 {rmax}
Cr r 0 1 IC = {rinit}
*Current equation Imem = V/R(L)
Gpm plus minus value = {(V(plus)-V(minus))/(fo*exp(2*L(V(r))))/L(V(r))}
*Func. for nonlinear threshold-based behavior
func Dr_dt(y) = {-alpha*((y-VtL)/(gamma+abs(y-VtL)))*st_f(-y+VtL)-beta*y*st_f(y-VtL)*st_f(-y+VtR)-alpha*((y-VtR)/(gamma+abs(y-VtR)))*st_f(y-VtR)}
*smoothing function
.func st_f(y) = {1/(exp(-y/yo)+1)}
*L(V) function
.func L(y) = {Lo-Lo*m/y}
.ends mem1
```

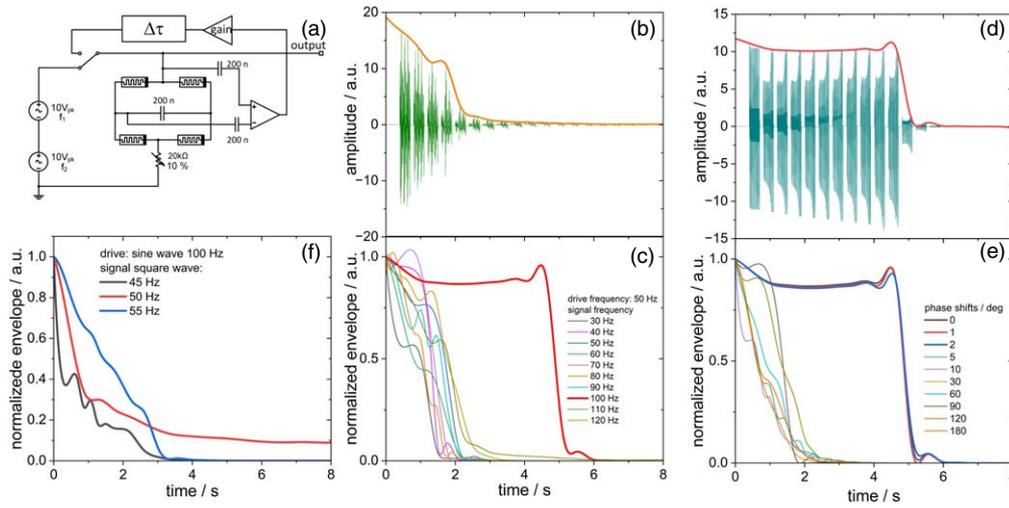


Fig. 21. Single-node echo state machine implemented on the Multisim platform. An appropriate gain was set to eliminate premature fading of the signal in higher epochs (a). Evolution of the input signal in a reservoir feedback loop based on bridge synapse (b) and normalized output envelopes recorded for various sets of input frequencies (c) for different combinations of input frequencies. Evolution of the input signal in a reservoir feedback loop for a 100 Hz sine input and 50 Hz sine drive (d) and normalized output envelopes recorded for various phase shifts of the input (e). Normalized output envelopes recorded for a combination of sine and square waveforms of different frequencies (f).

Finally, the same circuit has been subjected to different waveforms to check how the octave-detection procedure behaves in the presence of higher harmonics (a square wave may be considered as a sum of sine harmonic components). Surprisingly, the persistence of an octave was significantly higher than other tone combinations; however, the effect was not as clear, as in the case of pure sine functions [Fig. 21(f)].

5. Conclusions

In order to implement RC concepts in a form of physical devices, several types of internal processing nodes need to be considered—optoelectronic, spintronic, and memristive. The latter were implemented in thin layers, assemblies of nanowires, and NPs. Despite various compositions and different topologies, there is no clear relation between the physical form of the material (bulk or nanostructures of different dimensionalities). Due to a growing number of different examples, and no simple classification system, the final distinction of RC system types can be made based either on their structure or application.

Although typical solutions, such as software RNNs, have high complexity, RC physical systems avoid all of this, as their constitution relies in most cases on nanomaterials, leaving only a relatively simple output layer to be “programmable” by users. Among the main material requirements is for it to be characterized by a nonlinear response and complex internal dynamics.

For typical material-based (*in materia*) types of systems, there is a discrepancy regarding software simulations, as these cannot take into consideration all of the intrinsic phenomena. That is why, despite its growing popularity, the approach to model just one physical entity (either a single memristor or a single information-processing node) that as yet present the whole network only *in silico* should be treated as an approximation.

In order to describe these systems, one should follow the interface–input–reservoir–output distinction. In this way, the

explanation of the principle of work for the system is made easier. The end user communicates with the system through an intrinsic interface, which allows the input signal to be incorporated further. Depending on the choice of solutions, the signal is then modified within the framework of the actual RC system. In order to utilize the full classification functionality, output signal is generated and the final signal classification is executed. Typically, there exists some kind of readout layer implementation; however, it is possible to find approaches, where the readout layer is omitted, relying only on specially designed postprocessing algorithms or procedures—see some examples from our group.^{125,141)}

The difficulties related to postprocessing of the reservoir output (via application of readout perceptron, other neural networks, and complex mathematical treatment) are some designated cases and for specific computational tasks can be greatly simplified by the application of drive signals. Drive application has already been demonstrated in chemical sensing and analysis of acoustic signals. These RC systems operated without a proper readout layer, but were still capable of performing one-hot classification of inputs or yielded high-quality analytical signals.

There is another difficulty in physical implementations of RC. The input must be appropriately converted in order to interact with the reservoir—no relevant data should be lost in this translation. Furthermore, as we demand universality of computing systems, then RC should operate in a task-independent way. For physical systems, this may be very difficult to implement. The translation problem mentioned above is not trivial—some input data may be naturally and intuitively translated, but it is not always true. The RC systems described so far are designed for one, well-defined computational task, so achieving universality in the context of hardware implementation is still an unsolved problem.

Intuitively, RC systems seem to be similar to oracle machines. An oracle machine (commonly abbreviated to o-machine) is a variation of a Turing machine, which can solve any computable problem is a single step. In general, an oracle

can provide a solution to any decision-making or functional problem. Such a problem does not have to be algorithmically decidable; the oracle can represent answers to any mathematically defined set of problems.^{142,143} Complex reservoirs, in principle, can perform any classification task, but not in a single step (well, at least delayed feedback devices require finite time to establish a new internal dynamic setup, or require given time for evolution into a final step). If we associate this feature with the problem of the drive (an additional input, which can also be understood as an equivalent of software for the reservoir), we are approaching an interesting problem. A reservoir (with internal complexity suitable for a specific set of problems) may operate as a time-delayed oracle machine. The only problem is the communication language: appropriate drive (command) is not known (or rather the language in which it should be given is unknown). Furthermore, the output, understood as a modification of internal dynamics of the reservoir, may be considered as an answer given in an unknown language. To make the story even more confusing, one cannot assume *a priori* that these two languages are identical. Then, communication with complex reservoirs may be as complex as attempts to understand the famous Cthulhian incantation “Ph'nglui mglw'nafh Cthulhu R'l'yeh wgah'nagl fhtagn”,¹⁴ the Voynich's manuscript,^{144,145} *Codex Seraphinianus*,¹⁴⁶ or fungal languages.¹⁴⁷ It has been demonstrated, however, that, for special classes of problems, the formal description of RC simplifies¹³⁸ and therefore the search for appropriate drives is not in vain.¹¹⁷

Therefore, *in materia* (or physical) RC constitutes a universal information-processing platform that, however, needs further development, not only from the material point of view (higher nonlinearity, robustness, and memory capacity are still to be developed) but also in the design of reservoirs and their operation protocols.

Acknowledgments

The authors acknowledge the financial support of the Polish National Science Centre within the OPUS program (Grant Agreement No. UMO-2020/37/B/ST5/00663). This research was partially supported by the program “Excellence Initiative-Research University” for the AGH University of Science and Technology and by the NAWA program (contract no. PPN/BIN/2019/1/00100).

- 1) N. Kasabov, *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*. (Springer-verlag GmbH, Berlin, 2019).
- 2) N. Jouppi, C. Young, N. Patil, and D. Patterson, *Commun. ACM* **61**, 50 (2018).
- 3) J. J. Hopfield, *IEEE Circuits Devices* **4**, 3 (1988).
- 4) L. Yin, R. Cheng, Y. Wen, C. Liu, and J. He, *Adv. Mater.* **33**, 2007081 (2021).
- 5) P. Mannocci, M. Farronato, N. Lepri, L. Cattaneo, A. Glukhov, Z. Sun, and D. Ielmini, *APL Mach. Learn.* **1**, 010902 (2023).
- 6) X. Zou, S. Xu, X. Chen, L. Yan, and Y. Han, *Sci. China Inf. Sci.* **64**, 160404 (2021).
- 7) G. W. Burr et al., *Adv. Phys.: X* **2**, 89 (2017).
- 8) H. Jaeger, B. Noheda, and W. G. van der Wiel, *Nat. Commun.* **14**, 4911 (2023).
- 9) H. Jaeger, *Neuromorph. Comput. Eng.* **1**, 012002 (2021).
- 10) A. M. Turing, *Proc. Lond. Math. Soc.* **s2-42**, 230 (1937).
- 11) J. Backus, *Commun. ACM* **21**, 613 (1978).
- 12) S. Kundu, P. B. Ganganaik, J. Louis, H. Chalamalasetty, and B. P. Rao, *IEEE Trans. VLSI Syst.* **30**, 755 (2022).

- 13) U. Axelson and I. Makarov, *London School of Economics Discussion Papers* **754**, 1 (2016).
- 14) H. P. Lovecraft, *Weird Tales* **11**, 159 (1928).
- 15) H. Jaeger, W. Maass, and J. Principe, *Neural Netw.* **20**, 287 (2007).
- 16) H. Jaeger, German National Research Center for Information Technology GMD Technical Report **148**, 1 (2001).
- 17) W. Maass, T. Natschläger, and H. Markram, *Neural Comput.* **14**, 2531 (2002).
- 18) R. Fang, W. Zhang, K. Ren, P. Zhang, X. Xu, Z. Wang, and D. Shang, *Mater. Futures* **2**, 022701 (2023).
- 19) P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, *Neural Netw.* **126**, 191 (2020).
- 20) S. Dreyfus, *J. Math. Anal. Appl.* **5**, 30 (1962).
- 21) P. J. Werbos, *Proc. IEEE* **78**, 1550 (1990).
- 22) P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **474**, 20170844 (2018).
- 23) D. H. Wolpert and W. G. Macready, *IEEE Trans. Evol. Comput.* **1**, 67 (1997).
- 24) H. Jaeger, *Scholarpedia* **2**, 2330 (2007).
- 25) M. Lukoševičius and H. Jaeger, *Comput. Sci. Rev.* **3**, 127 (2009).
- 26) D. H. Wolpert, *Neural Comput.* **8**, 1341 (1996).
- 27) G. Ballarin, P. Dellaportas, L. Grigoryeva, M. Hirt, S. van Huellen, and J.-P. Ortega, *International Journal of Forecasting* (2023), [10.1016/j.ijforecast.2023.10.009](https://doi.org/10.1016/j.ijforecast.2023.10.009) (in press).
- 28) R. Budhiraja, M. Kumar, M. K. Das, A. S. Bafila, and S. Singh, *PLoS One* **16**, e0246737 (2021).
- 29) Y. Horii, K. Inoue, S. Nishikawa, K. Nakajima, R. Niiyama, and Y. Kuniyoshi, in *ALIFE 2021: The 2021 Conf. on Artificial Life*, 2021, 92, [10.1162/isaal_a_00426](https://doi.org/10.1162/isaal_a_00426).
- 30) P. Baldini, (2022), 2206.11222.
- 31) P. Antonik, M. Gulina, J. Pauwels, and S. Massar, *Phys. Rev. E* **98**, 012215 (2018).
- 32) Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, *Sci. Rep.* **5**, 12785 (2015).
- 33) M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, *IEEE Trans. Comput.* **52**, 403 (2003).
- 34) J. Sweller, *Educ. Psychol. Rev.* **21**, 11 (2009).
- 35) K. E. Jennings, *Minds Mach.* **20**, 489 (2010).
- 36) D. Przychyna, S. Pecqueur, D. Vuillaume, and K. Szaciłowski, *Int. J. Unconv. Comput.* **14**, 267 (2019).
- 37) Z. Konkoli, *Int. J. Parallel Emergent Distrib. Syst.* **33**, 121 (2018).
- 38) V. Athanasiou and Z. Konkoli, *Int. J. Parallel Emergent Distrib. Syst.* **33**, 367 (2018).
- 39) G. Abdi, L. Alluhaibi, E. Kowalewska, T. Mazur, K. Mech, A. Podborska, A. Sławek, H. Tanaka, and K. Szaciłowski, *Coord. Chem. Rev.* **487**, 215155 (2023).
- 40) K. Pilarczyk, E. Wlazlak, D. Przychyna, A. Blachecki, A. Podborska, V. Athanasiou, Z. Konkoli, and K. Szaciłowski, *Coord. Chem. Rev.* **365**, 23 (2018).
- 41) J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, *Sci. Rep.* **2**, 514 (2012).
- 42) K. Yang, J. Joshua Yang, R. Huang, and Y. Yang, *Small Sci.* **2**, 2100049 (2022).
- 43) H. Jaeger and H. Haas, *Science* **304**, 78 (2004).
- 44) M. Inubushi and K. Yoshimura, *Sci. Rep.* **7**, 10199 (2017).
- 45) Z. Konkoli, S. Nichele, M. Dale, and S. Stepney, in *Computational Matter*, ed. S. Stepney et al., (Springer, Cham, Switzerland, 2018).
- 46) Z. Konkoli, S. Stepney, H. Broersma, P. Dini, C. L. Nehaniv, and S. Nichele, in *Computational Matter*, ed. S. Stepney et al., (Springer, Cham, Switzerland, 2018).
- 47) M. Xu, X. Chen, Y. Guo, Y. Wang, D. Qiu, X. Du, Y. Cui, X. Wang, and J. Xiong, *Adv. Mater.* **35**, 2301063 (2023).
- 48) R. Khan, N. U. Rehman, S. Iqbal, S. Abdullaev, and H. M. Aldosari, *ACS Appl. Electron. Mater.* **6**, 73 (2024).
- 49) H. Chen, H. Li, T. Ma, S. Han, and Q. Zhao, *Sci. Technol. Adv. Mat.* **24**, 2183712 (2023).
- 50) S. Choi, J. Yang, and G. Wang, *Adv. Mater.* **32**, 2004659 (2020).
- 51) Z. Li, W. Tang, B. Zhang, R. Yang, and X. Miao, *Sci. Technol. Adv. Mater.* **24**, 2188878 (2023).
- 52) H. Liu et al., *Adv. Mater.* **35**, 2205047 (2023).
- 53) K. He, C. Wang, Y. He, J. Su, and X. Chen, *Chem. Rev.* **123**, 13796 (2023).
- 54) X. Liu, D. Li, Y. Wang, D. Yang, and X. Pi, *APL Mach. Learn.* **1**, 031501 (2023).
- 55) H. Zhou, J. Chen, J. Li, L. Yang, Y. Li, and X. Miao, *APL Mach. Learn.* **1**, 040901 (2023).

- 56) A. Mehonic and J. Eshraghian, *APL Mach. Learn.* **1**, 020401 (2023).
- 57) M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, and J. Sun, *Nat. Commun.* **15**, 2056 (2024).
- 58) F. Damicelli, C. C. Hilgetag, and A. Goulas, *PLoS Comput. Biol.* **18**, e1010639 (2022).
- 59) L. E. Suárez, A. Mihalik, F. Milisav, K. Marshall, M. Li, P. E. Vértes, G. Lajoie, and B. Misic, *Nat. Commun.* **15**, 656 (2024).
- 60) G. Milano, E. Miranda, and C. Ricciardi, *Neural Netw.* **150**, 137 (2022).
- 61) J. Morra and M. Daley, 2022 Int. Joint Conf. on Neural Networks (IJCNN) 2022, p. 1, [10.1109/IJCNN55064.2022.9892629](https://doi.org/10.1109/IJCNN55064.2022.9892629).
- 62) Y. H. Jang, S. H. Lee, J. Han, W. Kim, S. K. Shim, S. Cheong, K. S. Woo, J.-K. Han, and C. S. Hwang, *Adv. Mater.* **36**, 2309314 (2024).
- 63) M. Cucchi, S. Abreu, G. Ciccone, D. Brunner, and H. Kleemann, *Neuromorph. Comput. Eng.* **2**, 032002 (2022).
- 64) N. Bourbaki, *General topology* (Addison-Wesley Pub. Co, Reading MA, 1966).
- 65) G. Manjunath and J.-P. Ortega, *Phys. D* **449**, 133744 (2023).
- 66) X.-Y. Duan, X. Ying, S.-Y. Leng, J. Kurths, W. Lin, and H.-F. Ma, *Phys. Rev. Res.* **5**, L022041 (2023).
- 67) D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, *Nat. Commun.* **12**, 5564 (2021).
- 68) G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, *Neural Netw.* **115**, 100 (2019).
- 69) S. Kan, K. Nakajima, Y. Takeshima, T. Asai, Y. Kuwahara, and M. Akai-Kasaya, *Phys. Rev. Appl.* **15**, 024030 (2021).
- 70) M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer, *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **475**, 20180723 (2019).
- 71) K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, *Phys. Rev. Appl.* **11**, 034021 (2019).
- 72) M. Borghi, S. Biasi, and L. Pavesi, *Sci. Rep.* **11**, 15642 (2021).
- 73) D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (Wiley, Chichester, 2001).
- 74) L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, *Nat. Commun.* **2**, 468 (2011).
- 75) D. Sato et al., *ACS Appl. Mater. Interfaces* **15**, 49712 (2023).
- 76) T. Matsuo, D. Sato, S.-G. Koh, H. Shima, Y. Naitoh, H. Akinaga, T. Itoh, T. Nokami, M. Kobayashi, and K. Kinoshita, *ACS Appl. Mater. Interfaces* **14**, 36890 (2022).
- 77) K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, *Sci. Rep.* **5**, 10487 (2015).
- 78) K. Nakajima, *Jpn. J. Appl. Phys.* **59**, 060501 (2020).
- 79) D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, *J. Appl. Phys.* **124**, 152004 (2018).
- 80) T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki, and S. Miwa, *Phys. Rev. Appl.* **10**, 034063 (2018).
- 81) Z. Qi, L. Mi, H. Qian, W. Zheng, Y. Guo, and Y. Chai, *Adv. Funct. Mater.* **33**, 2306149 (2023).
- 82) Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto, *Sci. Rep.* **12**, 1353 (2022).
- 83) D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, *Nat. Commun.* **4**, 1364 (2013).
- 84) L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, *Opt. Express* **20**, 3241 (2012).
- 85) G. Dion, S. Mejaouri, and J. Sylvestre, *J. Appl. Phys.* **124**, 152132 (2018).
- 86) T. Taniguchi, S. Tsunegi, S. Miwa, K. Fujii, H. Kubota, and K. Nakajima, in *Reservoir Computing: Theory, Physical Implementations, and Applications*, ed. K. Nakajima and I. Fischer, (Springer Singapore, Singapore, 2021) p. 331, [10.1007/978-981-13-1687-6_14](https://doi.org/10.1007/978-981-13-1687-6_14).
- 87) R. Msiska, J. Love, J. Mulkers, J. Leliaert, and K. Everschor-Sitte, *Adv. Intell. Syst.* **5**, 2200388 (2023).
- 88) K. Liu, B. Dang, T. Zhang, Z. Yang, L. Bao, L. Xu, C. Cheng, R. Huang, and Y. Yang, *Adv. Mater.* **34**, 2270333 (2022).
- 89) R. Nakane, A. Hirose, and G. Tanaka, *Phys. Rev. Res.* **3**, 033243 (2021).
- 90) M. Crepaldi, C. Mohan, E. Garofalo, A. Adamatzky, K. Szaciłowski, and A. Chiolerio, *Adv. Mater.* **35**, 2211406 (2023).
- 91) Z. Tong, R. Nakane, A. Hirose, and G. Tanaka, *Int. J. Bifurcation Chaos* **32**, 2250141 (2022).
- 92) G. Tanaka and R. Nakane, *Sci. Rep.* **12**, 9868 (2022).
- 93) C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, *Nat. Commun.* **20178** (2204).
- 94) D. Przychyna, G. Hess, and K. Szaciłowski, *Int. J. Parallel Emergent Distrib. Syst.* **37**, 512 (2022).
- 95) D. Kim, J. Shin, and S. Kim, *Appl. Surf. Sci.* **599**, 153876 (2022).
- 96) H. Ryu and S. Kim, *Chaos, Solitons Fractals* **150**, 111223 (2021).
- 97) Y. Zhong, J. Tang, X. Li, B. Gao, H. Qian, and H. Wu, *Nat. Commun.* **12**, 408 (2021).
- 98) G. Zhang, J. Qin, Y. Zhang, G. Gong, Z.-Y. Xiong, X. Ma, Z. Lv, Y. Zhou, and S.-T. Han, *Adv. Funct. Mater.* **33**, 2302929 (2023).
- 99) M. Nakajima, K. Tanaka, and T. Hashimoto, *Commun. Phys.* **4**, 20 (2021).
- 100) B. Xu, Y. Huang, Y. Fang, Z. Wang, S. Yu, and R. Xu, *Photonics* **9**, 698 (2022).
- 101) H. O. Sillin, R. Aguilera, H.-H. Shieh, A. V. Avizienis, M. Aono, A. Z. Stieg, and J. K. Gimzewski, *Nanotechnology* **24**, 384004 (2013).
- 102) G. Milano, G. Pedretti, K. Montano, S. Ricci, S. Hashemkhani, L. Boarino, D. Ielmini, and C. Ricciardi, *Nat. Mater.* **21**, 195 (2022).
- 103) Q. Li, A. Diaz-Alvarez, R. Iguchi, J. Hochstetter, A. Loeffler, R. Zhu, Y. Shingaya, Z. Kuncic, K.-I. Uchida, and T. Nakayama, *Adv. Funct. Mater.* **30**, 2003679 (2020).
- 104) L. F. C. Pereira, C. G. Rocha, A. Latgé, and M. S. Ferreira, *J. Appl. Phys.* **108**, 103720 (2010).
- 105) O. Srikimkaew, D. Banerjee, S. Azhari, Y. Usami, and H. Tanaka, *ACS Appl. Electron. Mater.* **6**, 688 (2024).
- 106) R. Zhu, S. Lilak, A. Loeffler, J. Lizier, A. Stieg, J. Gimzewski, and Z. Kuncic, *Nat. Commun.* **14**, 6697 (2023).
- 107) G. Milano, K. Montano, and C. Ricciardi, *J. Phys. D* **56**, 084005 (2023).
- 108) S. Lilak, W. Woods, K. Scharnhorst, C. Dunham, C. Teuscher, A. Z. Stieg, and J. K. Gimzewski, *Front. Nanotechnol.* **3**, 675792 (2021).
- 109) M. D. Pike, S. K. Bose, J. B. Mallinson, S. K. Acharya, S. Shirai, E. Galli, S. J. Weddell, P. J. Bones, M. D. Arnold, and S. A. Brown, *Nano Lett.* **20**, 3935 (2020).
- 110) S. K. Bose, S. Shirai, J. B. Mallinson, and S. A. Brown, *Faraday Discuss.* **213**, 471 (2019).
- 111) S. Kan, K. Nakajima, T. Asai, and M. Akai-Kasaya, *Adv. Sci.* **9**, 2104076 (2022).
- 112) H. Tanaka, M. Akai-Kasaya, A. TermeHyousefi, L. Hong, L. Fu, H. Tamukoh, D. Tanaka, T. Asai, and T. Ogawa, *Nat. Commun.* **9**, 2693 (2018).
- 113) D. Banerjee, T. Kotooka, S. Azhari, Y. Usami, T. Ogawa, J. K. Gimzewski, H. Tamukoh, and H. Tanaka, *Adv. Intellig. Syst.* **4**, 2100145 (2022).
- 114) T. Kotooka, Y. Tanaka, H. Tamukoh, Y. Usami, and H. Tanaka, *Appl. Phys. Expr.* **16**, 014002 (2023).
- 115) Hadiyawardman, Y. Usami, T. Kotooka, S. Azhari, M. Eguchi, and H. Tanaka, *Jpn. J. Appl. Phys.* **60**, SCCF02 (2021).
- 116) M. Hakoshima, Y. Usami, T. Kotooka, and H. Tanaka, *Jpn. J. Appl. Phys.* **62**, SG1042 (2023).
- 117) V. Athanasiou and Z. Konkoli, *Sci. Rep.* **10**, 9191 (2020).
- 118) Z. Konkoli in *Unconventional Computing, Arts, Philosophy*, ed. A. Adamatzky, (World Scientific, Singapore, 2022) 2, p. 281.
- 119) K. Shibata, D. Nishioka, W. Namiki, T. Tsuchiya, T. Higuchi, and K. Terabe, *Sci. Rep.* **13**, 21060 (2023).
- 120) X. Liang et al., *Nat. Commun.* **13**, 1549 (2022).
- 121) L. Manneschi, M. O. A. Ellis, G. Gigante, A. C. Lin, P. Del Giudice, and E. Vasilaki, *Frontiers in Applied Mathematics and Statistics* **6**, 616658 (2021).
- 122) A. Hart, J. Hook, and J. Dawes, *Neural Netw.* **128**, 234 (2020).
- 123) L. F. Seoane, *Phil. Trans. B* **374**, 20180377 (2019).
- 124) E. Właźlak, P. Zawal, and K. Szaciłowski, *ACS Appl. Electron. Mater.* **2**, 329 (2020).
- 125) E. Właźlak, M. Marzec, P. Zawal, and K. Szaciłowski, *ACS Appl. Mater. Interfaces* **11**, 17009 (2019).
- 126) P. Antonik, M. Hermans, M. Haelterman, and S. Massar, in *IEEE Int. Joint Conf. on Neural Networks 2017, IEEE, 2017*, p. 2407.
- 127) S. Guy Van der, B. Daniel, and C. S. Miguel, *Nanophotonics* **6**, 561 (2017).
- 128) L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, *Phys. Rev. X* **7**, 011015 (2017).
- 129) G. Van der Sande, D. Brunner, and M. C. Soriano, *Nanophotonics* **6**, 561 (2017).
- 130) D. Brunner, M. C. Soriano, and G. Van der Sande, *Photonic Reservoir Computing* (De Gruyter, Berlin, 2019).
- 131) C. Rios, N. Youngblood, Z. Cheng, M. Le Gallo, W. H. P. Pernice, C. D. Wright, A. Sebastian, and H. Bhaskaran, *Sci. Adv.* **5**, eaau5759 (2019).
- 132) I. Bauwens, G. Van der Sande, P. Bienstman, and G. Verschaffelt, *Front. Phys.* **10**, 1051941 (2022).
- 133) D. Przychyna, K. Pilarczyk, A. Blachecki, V. Gorokh, and K. Szaciłowski, in *2019 Int. Semiconductor Conf. (CAS), 2019, 23*, [10.1109/SMICND.2019.8923800](https://doi.org/10.1109/SMICND.2019.8923800).
- 134) D. Przychyna, M. Szaciłowska, M. Przybylski, M. Strzelecki, and K. Szaciłowski, *Int. J. Unconv. Comput.* **17**, 81 (2022).
- 135) D. Przychyna, P. Zawal, T. Mazur, M. Strzelecki, P. L. Gentili, and K. Szaciłowski, *Jpn. J. Appl. Phys.* **59**, 050504 (2020).

- 136) H. Kim, M. P. Sah, C. Yang, T. Roska, and L. O. Chua, *Proc. IEEE* **100**, 2061 (2012).
- 137) I. Vourkas and G. C. Sirakoulis, *Memristor-Based Nanoelectronic Computing Circuits and Architectures* (Springer International Publishing, Heidelberg, 2016).
- 138) M. Lis, S. Onuma, D. Przychyna, P. Zawal, T. Mazur, K. Pilarczyk, P. L. Gentili, S. Kasai, and K. Szaciłowski, in *Handbook of Unconventional Computing* (World Scientific, Singapore, 2021) p. 1.
- 139) A. Adamatzky, K. Szaciłowski, Z. Konkoli, L. C. Werner, D. Przychyna, and G. C. Sirakoulis in *From Astrophysics to Unconventional Computation*, ed. A. Adamatzky and V. Kendon, (Springer Nature Switzerland AG, Cham, Switzerland, 2020).
- 140) D. Przychyna, M. Suchecki, A. Adamatzky, and K. Szaciłowski, *Materials* **14**, 1724 (2021).
- 141) D. Przychyna, M. Lis, K. Pilarczyk, and K. Szaciłowski, *Molecules* **24**, 2738 (2019).
- 142) R. I. Soare, *Ann. Pure Appl. Log.* **160**, 368 (2009).
- 143) B. J. Copeland, *Analysis* **58**, 128 (1998).
- 144) A. O. Tucker and J. Janick, *Unraveling the Voynich Codex* (Springer International Publishing, Cham, 2018).
- 145) A. O. Tucker and J. Janick, *Flora of the Voynich Codex: An Exploration of Aztec Plants* (Springer International Publishing, Cham, 2019).
- 146) L. Serafini, *Codex Seraphinianus* (Rizzoli, New York, 2013).
- 147) A. Adamatzky, *R. Soc. Open Sci.* **9**, 211926 (2022).