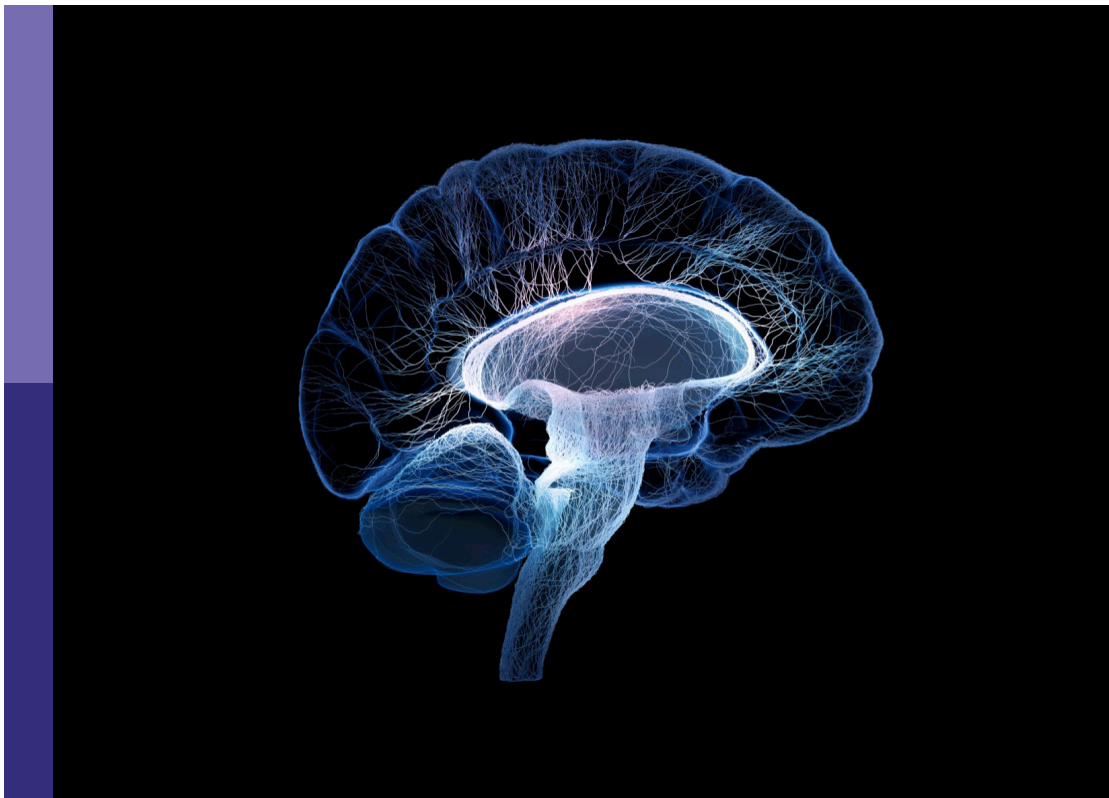
frontiers | Research Topics

Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches

Edited by
Pier Luigi Gentili, Siegfried Karg, Gyorgy Csaba and Konrad Szaciłowski

Published in
Frontiers in Neuroscience



ISSN 1664-8714
ISBN 978-2-8325-5637-5
DOI 10.3389/978-2-8325-5637-5

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches

Topic editors

Pier Luigi Gentili — Università degli Studi di Perugia, Italy
Siegfried Karg — IBM Research - Zurich, Switzerland
Gyorgy Csaba — Pázmány Péter Catholic University, Hungary
Konrad Szaciłowski — AGH University of Science and Technology, Poland

Citation

Gentili, P. L., Karg, S., Csaba, G., Szaciłowski, K., eds. (2024). *Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-5637-5

Topic editor Siegfried Karg is employed by IBM Research - Zurich.
Topic Editor Gyorgy Csaba receives financial support from Intel Corporation, for his involvement in a research project on oscillator-based computing.

Table of contents

frontiers | Frontiers in Neuroscience

TYPE Editorial
PUBLISHED 07 October 2024
DOI 10.3389/fnins.2024.1498684

OPEN ACCESS

EDITED AND REVIEWED BY
Bernabe Linares-Barranco,
Spanish National Research Council (CSIC), Spain

*CORRESPONDENCE
Pier Luigi Gentili
✉ pierluigi.gentili@unipg.it

RECEIVED 19 September 2024
ACCEPTED 23 September 2024
PUBLISHED 07 October 2024

CITATION
Gentili PL, Karg S, Csaba G and Szaciłowski K (2024) Editorial: Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches. *Front. Neurosci.* 18:1498684. doi: 10.3389/fnins.2024.1498684

Editorial: Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches

Pier Luigi Gentili[1]*, Siegfried Karg[2], Gyorgy Csaba[3] and Konrad Szaciłowski[4,5]

[1]Department of Chemistry, Biology, and Biotechnology, Università degli Studi di Perugia, Perugia, Italy, [2]IBM Research Europe (Zurich), Rüschlikon, Switzerland, [3]Faculty for Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary, [4]Academic Centre for Materials and Nanotechnology, AGH University of Krakow, Kraków, Poland, [5]Unconventional Computing Lab, University of the West of England, Bristol, United Kingdom

KEYWORDS

Natural Computing, Oscillatory Neural Networks, graphene-based memristive devices, phase-change materials, optoelectronic devices, chemical reaction networks, synthetic biology, nanofluidic iontronics

Editorial on the Research Topic
Reviews and perspectives in neuromorphic engineering: novel neuromorphic computing approaches

In the XXI century, humanity is spurred to face global challenges: climate changes, pollution, shortage of clean water, food and energy. These challenges regard Complex Systems, such as the intertwined human societies, the world economy, urban areas, natural ecosystems, and the climate of the Earth (UN General Assembly, 2015; Martin, 2007; Harari, 2018; Gentili, 2021; Gentili et al., 2022). Whenever we deal with Complex Systems, we experience some limitations in their description, and in understanding and predicting their behavior. Such limitations outline the so-called Epistemological Complexity (Gentili, 2023). A limitation is due to Computational Complexity (Goldreich, 2008): many computational problems involving Complex Systems are solvable but intractable. Examples are (1) Practical problems, such as scheduling and the traveling salesman problem; (2) Fundamental science problems, such as the Schrödinger equation and protein folding; (3) Pattern recognition problems faced through machine learning algorithms. They are all exponential problems that become intractable when they have large dimensions: it is impossible to determine their exact solutions in a reasonable time, even if we use the fastest supercomputers in the world. A promising strategy to face Epistemological Complexity and, hence, Computational Complexity is Natural Computing (Rozenberg et al., 2012; Gentili, 2023). Natural Computing is an interdisciplinary research line that draws inspiration from nature to formulate (a) new algorithms, propose (b) new materials and architectures to compute, and (c) new methods and models to understand Complex Systems. Wealthy sources of inspiration for new computing architectures and algorithms are the human and animal brains. Their imitation has sparked the burgeoning field of neuromorphic engineering that promises to outperform conventional Artificial Intelligence (AI) algorithms and high energy-demanding hardware,

offering a hopeful and optimistic outlook for the future of computing. Combining new algorithms, materials and architectures at the same time might be a complex task, but it may be the most promising route to Natural Computing (Maher et al., 2024). This Research Topic presents seven cutting-edge works in this field.

Among the many examples of analog computing, Rudner et al. highlight that Oscillatory Neural Networks (ONNs) are particularly alluring. Computing is carried out on the basis of the rich, complex, non-linear synchronization dynamics of an artificial neural network. Using the phase of oscillators enables a rich, robust, and parallel way of encoding of information, as it is often done in biological systems. Artificial ONNs often rely on some version of a Hebbian rule to define attractor states for the oscillators' phases. In their study, the authors, using computer simulations, demonstrate that a state-of-the-art machine learning method, namely Backpropagation Through Time, when applied to a circuit-level model of the ONN (based on resistively coupled ring oscillators), significantly enhances the computational power of the ONNs in recognizing various patterns.

Abernot et al. present possible algorithms and implementation of continual on-chip learning based on a digital ONN design for pattern recognition. They highlight that Hopfield Neural Network's unsupervised learning algorithms are compatible with ONN on-chip learning only if they satisfy two constraints on the weight matrix, the symmetry and the 0-diagonal, and two additional constraints on the learning algorithm, locality, and incrementality. The results of this work show that two unsupervised learning rules are compatible with ONN on-chip learning: Hebbian and Storkey. The proposed architecture takes advantage of a Processing System of a Zynq processor to implement the learning algorithms and Programmable Logic resources to implement the digital ONN.

Jiménez et al. describe an ONN implemented in a commercial CMOS technology to emulate the behavior of neural surrogates based on the phase-change VO_2 material. VO_2 undergoes metal-insulator transitions under given electrical stimuli. VO_2 devices stand out for their hysteresis in the characteristic I–V curve, which enables compact low-power relaxation oscillators. The declared purpose of this work is to study in-depth the synchronization dynamics of relaxation oscillators similar to those that can be performed with VO_2 devices. The fabricated circuit is very flexible since it allows programming the synapses to implement different ONNs, calibrating the frequency of the oscillators, or controlling their initialization. It uses differential oscillators and resistive synapses, equivalent to memristors. The ONN has been tested in its Associate Memory functionality.

Rajalekshmi et al. present a comprehensive analysis of the structural and design aspects of graphene-based Resistive Random Access Memory (RRAM) devices for their applications in in-memory and neural computing. Graphene-based RRAM devices are memristive systems with enhanced switching speed, retention time, endurance, and power consumption. Graphene assures additional performances, such as more substantial heat dissipation and chemical stability. Moreover, graphene provides more than two states to the memristive device, allowing the implementation of analog computing devices and storage.

El Srouji et al. proclaim that co-integrated photonic and electronic technologies are key to the future of neuromorphic computing. Biological neural networks are remarkably heterogeneous in terms of individual neuron dynamics and morphological structure. Such neural heterogeneity increases the sensitivity toward the complexity of behaviors and sensory modalities the brain must handle. An optoelectronic approach to neuromorphic computing is better suited to provide the interconnect bandwidths necessary to support the neuronal fan-in and fan-out required to model neural networks at biological scales while allowing for flexible and programmable neural dynamics.

Gentili et al. outline neuromorphic engineering in wetware, i.e., in a liquid solution, the peculiar phase supporting life. In wetware, three are the principal strategies to mimic some structural and functional features of the human brain. The first one, described also by Csizi and Lörtscher, relies on networks of chemical reactions: any solution containing reactive species can be compared to a neural network. Some reactions reproduce binary logic functions; others are appropriate for processing fuzzy logic. In the presence of strong non-linear interactions between the intermediate reactive species, some reactions exhibit bottom-up self-organization phenomena that reproduce the dynamics of real neurons and ONNs can be implemented. Such dynamic neural surrogates can communicate through electro-chemical and optical signals and become building blocks of feedforward and recurrent networks. When the molecules participating in the chemical reaction networks are biopolymers, such as DNA, RNA, and proteins, we enter the realm of synthetic biology, which constitutes the second strategy for developing neuromorphic engineering in wetware. The third strategy is nanofluidic iontronics, which represents the possibility of emulating neural networks through hybrid circuits made of solid nanochannels and electrically conductive ionic solutions.

Author contributions

PG: Writing – original draft, Writing – review & editing. SK: Writing – review & editing. GC: Writing – review & editing. KS: Writing – review & editing.

Acknowledgments

The authors thank the Content Specialist of Frontiers, Dr. Sara Murru, for her support in organizing and managing this Research Topic.

Conflict of interest

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

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

References

Gentili, P. L. (2021). Why is complexity science valuable for reaching the goals of the UN 2030 agenda? *Rend. Fis. Acc. Lincei.* 32, 117–134. doi: 10.1007/s12210-020-00972-0

Gentili, P. L. (2023). The relevant role that natural computing can play in the development of complexity science. *Int. J. Unconvent. Comp.* 18, 291–304.

Gentili, P. L., Cardinali, G., Dominici, P., Grohmann, D., Menconi, M. E., and Santi, C. (2022). The science of complex systems for preparing the new generation to tackle global challenges. *Int. Conf. High. Educ. Adv.* 2022, 959–966. doi: 10.4995/HEAd22.2022.14319

Goldreich, O. (2008). *Computational Complexity. A Conceptual Perspective*. Cambridge: Cambridge University Press.

Harari, Y. N. (2018). *21 Lessons for the 21st Century*. London: Penguin Random House.

Maher, O., Jiménez, M., Delacour, C., Harnack, N., Núñez, J., Avedillo, M. J., et al. (2024). A CMOS-compatible oscillation-based VO2 Ising machine solver. *Nat. Commun.* 15:3334. doi: 10.1038/s41467-024-47642-5

Martin, J. (2007). *The Meaning of the 21st Century: a Vital Blueprint for Ensuring our Future*. New York: Riverhead Books.

Rozenberg, G., Bäck, T., and Kok, J. N. (2012). *Handbook of Natural Computing*. Berlin: Springer. doi: 10.1007/978-3-540-92910-9

UN General Assembly (2015). *Transforming Our World: the 2030 Agenda for Sustainable Development. A/RES/70/1*. Available at: https://www.refworld.org/docid/57b6e3e44.html (accessed September 2, 2024).

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 15 June 2023
DOI 10.3389/fnins.2023.1196796

OPEN ACCESS

EDITED BY
Elisa Donati,
University of Zurich, Switzerland

REVIEWED BY
Nicole Sandra-Yaffa Dumont,
University of Waterloo, Canada
Filippo Moro,
University of Zurich, Switzerland

*CORRESPONDENCE
Aida Todri-Sanial
✉ a.todri.sanial@tue.nl

RECEIVED 30 March 2023
ACCEPTED 29 May 2023
PUBLISHED 15 June 2023

CITATION
Abernot M, Azemard N and Todri-Sanial A (2023) Oscillatory neural network learning for pattern recognition: an on-chip learning perspective and implementation. *Front. Neurosci.* 17:1196796. doi: 10.3389/fnins.2023.1196796

Oscillatory neural network learning for pattern recognition: an on-chip learning perspective and implementation

Madeleine Abernot[1], Nadine Azemard[1] and Aida Todri-Sanial[1,2]*

[1]Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), Department of Microelectroncis, University of Montpellier, CNRS, Montpellier, France, [2]Electrical Engineering Department, Eindhoven University of Technology, Eindhoven, Netherlands

In the human brain, learning is continuous, while currently in AI, learning algorithms are pre-trained, making the model non-evolutive and predetermined. However, even in AI models, environment and input data change over time. Thus, there is a need to study continual learning algorithms. In particular, there is a need to investigate how to implement such continual learning algorithms on-chip. In this work, we focus on Oscillatory Neural Networks (ONNs), a neuromorphic computing paradigm performing auto-associative memory tasks, like Hopfield Neural Networks (HNNs). We study the adaptability of the HNN unsupervised learning rules to on-chip learning with ONN. In addition, we propose a first solution to implement unsupervised on-chip learning using a digital ONN design. We show that the architecture enables efficient ONN on-chip learning with Hebbian and Storkey learning rules in hundreds of microseconds for networks with up to 35 fully-connected digital oscillators.

KEYWORDS

oscillatory neural networks, on-chip learning, unsupervised learning, pattern recognition, FPGA implementation

1. Introduction

Current Artificial Intelligence (AI) models are mainly used for two functions, overcoming the human brain to solve a specific task, or replacing the human brain on more general purpose tasks (Pehlevan and Chklovskii, 2019). In both cases, AI models need to learn how to correctly solve a given task. However, while humans are capable of learning continuously through life to adapt to the changing environment and learn new tasks, current AI models are trained in advance for inference, making it impossible to learn from evolving environments and input data (Thrun and Mitchell, 1995; Ring, 1997). To adapt AI models to evolving environments and input data, continual learning is necessary, so there are ongoing efforts to develop continual learning algorithms for AI models (Thangarasa et al., 2020). In particular, efforts are concentrated first on supervised continual learning (De Lange et al., 2022; Mai et al., 2022) to improve the performance of classification models over time, and then on continual reinforcement learning to learn from the environment, for example in robotics (Lesort et al., 2020; Khetarpal et al., 2022).

Continual learning algorithms expect to learn novel data while avoiding catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999) of previously learned data, for example, considering bio-inspired synaptic plasticity, or reminding solutions (Hayes et al., 2020; De Lange et al., 2022; Jedlicka et al., 2022). Additionally, continual learning demands to be implemented on-chip for fast and efficient performances. However, to allow continual

on-chip learning, each synapse needs to be re-programmable in a real-time latency requiring additional space, and resources, and consuming more energy consumption than systems without on-chip learning.

Moreover, there are several ongoing works to propose hardware implementations of fast, low-resource, and power-efficient AI computing paradigms. In particular, neuromorphic computing (Christensen et al., 2022) takes inspiration from the human brain neural network for the AI models architectures, and for the data representation. The most widely used neuromorphic computing paradigm is called Spiking Neural Network (SNN; Maass, 1997) which takes inspiration from spikes transmitted among neurons through the brain synapses by encoding information in the latency between two spike signals. SNN has been widely explored in the last decades both in terms of network implementation, with the development of different SNN-based chips for edge AI computing (Davies et al., 2018; Pehle et al., 2022), and in terms of learning, in particular for continual learning (Wang et al., 2014; Lobo et al., 2019). In this paper, we focus on another neuromorphic paradigm, called the Oscillatory Neural Network (ONN), which is drawing attention as an alternative neuromorphic solution for edge AI computing.

ONN takes inspiration from the collective synchronization of human brain neurons through oscillations (Tognoli and Kelso, 2009). ONN is an analog-based computing paradigm built as a network of coupled oscillators (Izhikevich and Kuramoto, 2006; Schwemmer and Lewis, 2012; Raychowdhury et al., 2019; Csaba and Porod, 2020; Todri-Sanial et al., 2022) computing with the parallel phase synchronization of coupled oscillators, called phase computing. In phase computing, information is encoded in the phase relationship between oscillators which can potentially limit voltage amplitude and, therefore, reduce the energy consumption (Delacour et al., 2023a), making it attractive for edge computing. Currently, efforts are given on ONN implementation, from materials to devices, on ONN circuit architecture (Abernot et al., 2021; Delacour et al., 2023b), and on ONN applications with demonstrators of ONNs for image processing (Fernandes et al., 2004; Abernot and Todri-Sanial, 2023), robotic navigation (Abernot et al., 2022a), or optimization problems (Wang and Roychowdhury, 2019; Delacour et al., 2022). Yet, learning and continual learning algorithms for ONN are still to be investigated. Thus, this work focuses on ONN on-chip learning for pattern recognition.

In state-of-the-art, ONNs are often studied as a fully-connected recurrent architecture to perform pattern recognition similar to Hopfield Neural Networks (HNNs) (Hoppensteadt and Izhikevich, 1997; Nikonov et al., 2015; see Figure 1). While in the literature ONNs are typically trained with unsupervised learning rules that were first introduced for HNNs. To the best of our knowledge, learning rules specific to ONNs are yet to be developed. In this work, we present an adaptation of HNN unsupervised learning rules for ONNs while analyzing the different learning rules for continual on-chip learning. Recently, we introduced an on-chip learning architecture for a digital ONN implementation (Abernot et al., 2022b) with the Hebbian learning rule applied to a small 15-neuron ONN for a three-digit pattern recognition application. In this work, we go beyond by demonstrating that the ONN architecture is compatible with other learning rules than Hebbian by implementing the Storkey learning rule. Next, we analyze the scalability of the ONN architecture to provide a more complete evaluation of the system.

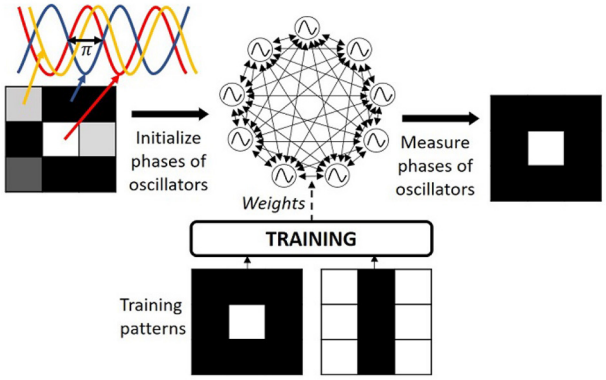


FIGURE 1
ONN computing paradigm configured for pattern recognition.

The main contributions of the paper are summarized as (i) adaptation of existing HNN unsupervised learning rules to ONNs, (ii) development of a continual on-chip learning algorithm on ONN with unsupervised learning rules, (iii) an implementation approach for on-chip learning on digital ONN for auto-associative memory tasks, and (iv) present a scalability analysis of our approach in terms of latency, precision and resource utilization.

First, Section 2.1 presents the ONN paradigm and its auto-associative memory capabilities. Then, Section 2.2 gives details on the various learning rules introduced for HNN and their compatibility with ONN for on-chip learning. After, Section 2.3 defines the proposed hardware implementation to perform on-chip learning with a digital ONN design. Section 3 shows results obtained with our on-chip learning solution for various ONN sizes, learning algorithms, and weight precision. Finally, Section 4 discusses the results compared to state-of-the-art and the advantages and limitations of our on-chip learning implementation.

2. Materials and methods

2.1. Oscillatory neural networks

In ONNs, each neuron is an oscillator coupled with synaptic elements representing weights between neurons (Delacour and Todri-Sanial, 2021), and information is represented in the phase relationship between oscillators such that ONN computes in phase using the weakly coupled oscillator dynamics (Schwemmer and Lewis, 2012). For example, for binary information, if an oscillator oscillates with a 0° phase difference from a reference oscillator, it will represent a binary "0" value, while if it oscillates with a 180° phase difference from a reference oscillator, it will represent a binary "1" value. Typically, one oscillator from the network is used as the reference oscillator. The inference process

starts with the initialization of each neuron phase as the input information, then, oscillators' phases evolve in parallel thanks to the dynamics of coupled oscillators (Schwemmer and Lewis, 2012) until stabilization to a final phase state, which represents the ONN inference output (see Figure 1). Phase computation can potentially reduce the voltage amplitude meanwhile it enables parallel computation, providing an attractive low-power edge computing paradigm (Delacour et al., 2023a).

The evolution of the phases during inference is associated with the minimization of an intrinsic parameter called the energy of the network. Note, it does not have any relationship with the power consumption of the system. The energy of the network is defined as follows:

$$E = \sum_i \sum_j W_{ij}\phi_i\phi_j \tag{1}$$

with ϕ_i the phase state of neuron i, ϕ_j the phase state of neuron j, and W_{ij} the coupling weight between neuron i and neuron j. Considering this intrinsic energy parameter, ONN learning consists of shaping the energy function, and more importantly, defining the minima of this energy function given a specific task (see Figure 2A). For example, ONN can solve graph optimization problems, like max-cut (Bashar et al., 2020; Delacour et al., 2022, 2023b; Vaidya et al., 2022), graph coloring (Wang and Roychowdhury, 2019), or traveling salesman problem (Landge et al., 2020), by mapping a graph to an ONN such that if you start the ONN with random phases, it will evolve to the optimal solutions represented by the minima of the energy function. More commonly, ONN is used to solve auto-associative memory, or pattern recognition tasks (Hoppensteadt and Izhikevich, 1997; Nikonov et al., 2015) using a fully-connected architecture as in HNNs (Hopfield, 1982; see Figure 1). Interestingly, the energy function is shaped such that training patterns are minima of the energy landscape (see Figure 2B), and when the network starts on corrupted information, it will evolve and stabilize to one of the training patterns. Note, for simplicity, we represent the energy function as a two-dimensional function, however, it is N-dimensional depending on the states of the N neurons.

2.2. ONN on-chip learning for pattern recognition

In this paper, we focus on auto-associative memory tasks or pattern recognition. The pattern recognition task is first defined, then the ONN learning is presented. Finally, we explain constraints, adaptation, and compatibility of unsupervised learning rules for use as on-chip learning on ONN.

2.2.1. Pattern recognition

In this work, we define the pattern recognition task, also called the auto-associative memory task, as the ability to learn patterns and retrieve them from corrupted input information. For example, considering images as patterns, a system configured for pattern recognition can memorize images and retrieve them from corrupted input with noisy or missing pixels. Classical HNNs are fully connected recurrent networks, also characterized by an energy function, which are state-of-the-art neural networks for solving pattern recognition (Hopfield, 1982). In classical HNN, each neuron follows a *sign* activation function, allowing two bipolar activation values $\{-1; 1\}$, where in the case of images, each neuron represents a pixel, and the neuron activation value $\{-1\}$ or $\{1\}$ represents the pixel color. Thus, classical HNN can treat and learn binary patterns, like images with black and white pixels. Recently, alternative HNNs are proposed to treat and learn multi-state or continuous patterns, such as the complex HNN using complex activation functions and complex weights (Muezzinoglu et al., 2003; Tanaka and Aihara, 2009), or the modern HNN considering a *softmax* activation function (Ramsauer et al., 2021). For ONNs, each neuron activation can take various phase values depending on the ONN design such as for the treatment of multi-state or continuous information, like gray-scale images.

For pattern recognition, the couplings among neurons represent the memory of the network. During the learning process, the training algorithm defines the coupling weight values such that learning patterns become the minima on the energy landscape. Learning does not ensure that all local minima are training patterns, and in some cases, local minima can become stable phase states while it does not correspond to any learning pattern, which is also labeled as a spurious pattern (see Figure 2B). During the inference process, one input pattern is applied to the network by initializing the oscillators' phases with the corresponding input information. Then, phases evolve thanks to the inherent phase interaction between coupled oscillators until they stabilize and the final phase state represents the ONN output pattern (see Figure 1).

2.2.2. ONN learning for pattern recognition

Existing learning algorithms to train an ONN for pattern recognition are mainly unsupervised learning rules, which were first introduced for HNNs. Unsupervised learning algorithms only use learning patterns to compute coupling weights, without additional feedback, unlike supervised learning algorithms, and are mainly used to solve clustering problems. In pattern recognition, each pattern becomes the point of attraction of various clusters created from the energy landscape (see Figure 2A). In this section, we discuss how to adapt HNN-based unsupervised learning algorithms for ONN.

Adapting HNN unsupervised learning rules to ONN requires weight matrix symmetry and zero diagonal values to avoid self-coupling. Originally, in HNN, the weight matrix is symmetric, meaning weights between two neurons in both directions have the same values, and the weight matrix diagonal has zero values to avoid self-coupling. Later, to improve precision and capacity, novel unsupervised learning algorithms were introduced allowing asymmetric weight matrix (Diederich and Opper, 1987; Krauth and Mezard, 1987; Gardner, 1988) and self-coupling (Gosti et al., 2019). However, most ONN implementations, in particular analog ONN implementations, do not support self-coupling and non-symmetric weights as the coupling is often implemented with discrete analog components like resistors or capacitors (Delacour and Todri-Sanial, 2021). Consequently, even if the digital ONN

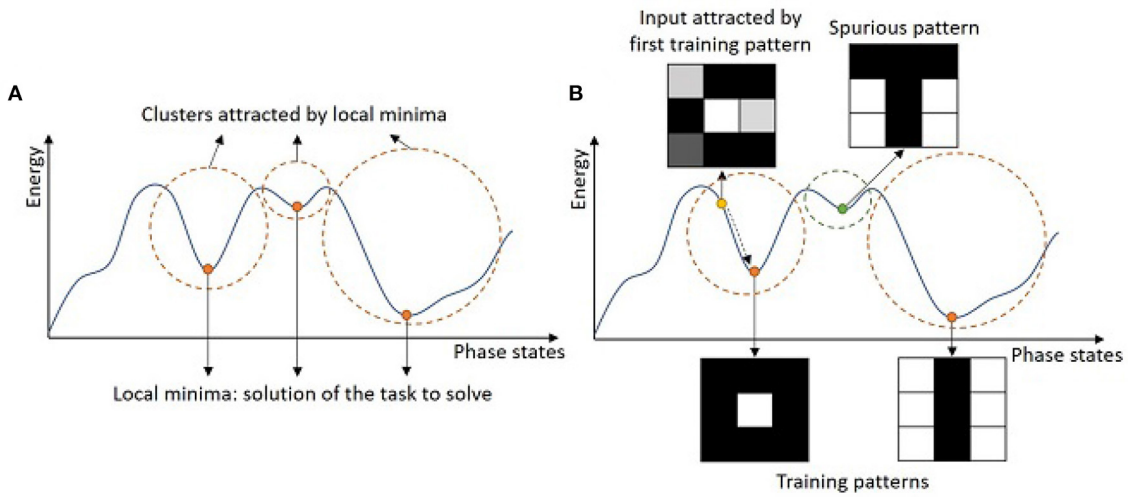


FIGURE 2
Simplified representation of an energy landscape for **(A)** a global interpretation, and **(B)** an interpretation in the case of pattern recognition.

supports non-symmetric weights and self-coupling, there are ongoing efforts to develop alternative analog ONN designs to allow self-coupling and non-symmetric weights (Delacour et al., 2023b). Most unsupervised learning algorithms introduced for HNN can be modified to be used with ONNs by adding constraints on the weight matrix. However, it was shown to impact negatively the HNN precision and memory capacity (Tolmachev and Manton, 2020). We provide a classification of the unsupervised learning rules respecting weights symmetry and 0-diagonal in Section 3. Moreover, using unsupervised learning algorithms introduced for classical HNN limits patterns to binary information while ONN with its continuous phase values could, in principle, stabilize to non-binary patterns e.g., any phase between 0° and 360°. However, to the best of our knowledge, there exist no unsupervised learning rules for pattern recognition adapted to ONN capable of learning non-binary patterns.

2.2.3. ONN on-chip learning adaptation

In this work, we define ONN on-chip learning for pattern recognition as the ability of an ONN-computing system to learn new patterns by updating ONN coupling weights meanwhile avoiding catastrophic forgetting of previously memorized patterns.

There exist mainly two features to categorize unsupervised learning rules for pattern recognition: locality which means that the update of the coupling weight between neuron i and neuron j only depends on activation values of neurons i and j on both sides of the synapse, and incrementality, which means that the update of the weights can be done pattern by pattern without forgetting previously learned patterns. The locality feature is important for on-chip learning because the update of the weights can be implemented by using limited additional resources in each synapse. Though locality is not mandatory as the update of the weights is not always integrated and implemented at the synapse level. The incrementality feature is also important to be able to learn patterns one at a time. For efficient incremental learning, previously learned patterns are memorized in the weight matrix of the network to avoid learning them again. To avoid catastrophic forgetting, some algorithms require repetitive learning of previous and novel patterns but it is not optimal for on-chip learning as it requires additional computing, and memory (Personnaz et al., 1986; Diederich and Opper, 1987; Krauth and Mezard, 1987; Gardner, 1988). Adding learning capacity to every synapse can be costly in terms of resources, so it is important to also consider sparsity and weight precision in the weight matrix. In this work, we study the impact of weight precision on HNN and ONN performances.

2.3. On-chip learning architecture

Here, we propose an architecture to perform ONN on-chip learning for pattern recognition. In particular, we consider a digital ONN implementation on FPGA, introduced in Abernot et al. (2021) and we explore its capability for on-chip learning. The on-chip learning architecture was first introduced in Abernot et al. (2022b) for a small-size ONN with 15 neurons, however, in this work, we study architecture scalability for different ONN sizes, learning rules, and weight precision. Here, we present the digital ONN design implementation for pattern recognition, its adaptation to on-chip learning, and our evaluation methods.

2.3.1. Digital ONN design

ONNs with their phase dynamics are intrinsically analog in nature and implemented with analog computing for low-power implementations (Delacour et al., 2023a). However, digital ONNs are attractive implementations for studying various applications, fast demonstration, and investigating scalability (Moy et al., 2022; Lo et al., 2023). In particular, a digital ONN implementation on FPGA was introduced in Abernot et al. (2021) to explore novel ONN architectures, learning algorithms, and applications. The

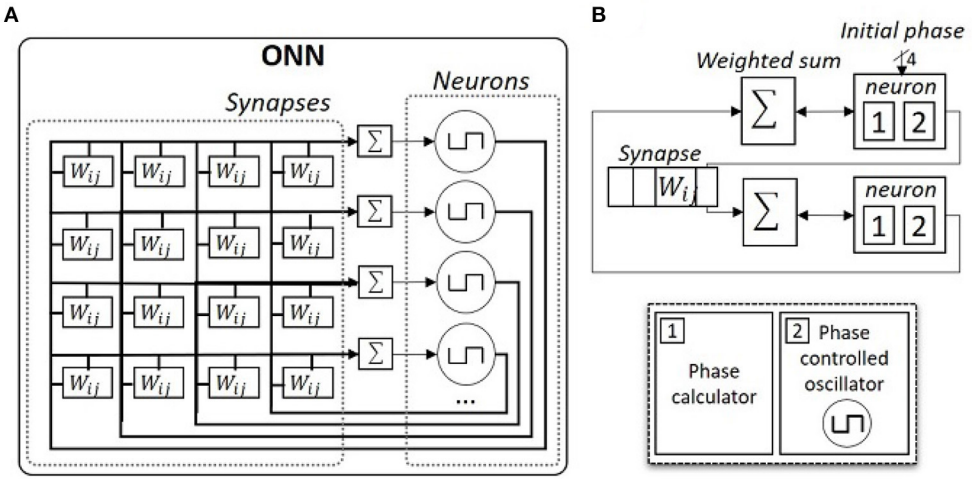


FIGURE 3
Schematics of ONN digital design. **(A)** Schematic of a fully-connected digital ONN. **(B)** Detailed schematic of a two-neuron digital ONN.

digital ONN on FPGA showcased fast and efficient computation for edge applications, for example performing obstacle avoidance on mobile robots by reading proximity sensor information (Abernot et al., 2022a), replacing convolution filters for image edge detection (Abernot and Todri-Sanial, 2023), or even accelerating the SIFT feature detection algorithm (Abernot et al., 2023a). We believe ONN implementation on FPGA is attractive for real-time applications for which providing on-chip learning is important.

Hence, we focus on the digital ONN implementation on FPGA as introduced in Abernot et al. (2021). In the digital design, each neuron is a 16-stage phase-controlled digital oscillator that can represent phases between 0 and 180° with a precision of 22.5° and each synapse is implemented using signed registers (see Figure 3). Originally, synapses are fixed to 5-bit signed registers, but in this work, we study the impact of weight precision on resource utilization, precision, and latency of the ONN on-chip learning architecture. We especially test three weight precision, with 3-, 4-, and 5-bit signed register implementations. Note, the digital design allows the implementation of non-symmetric weights with self-coupling (non-zero diagonal). However, in this work, we only consider symmetric weights without self-coupling to be coherent and compatible with other ONN implementations, for example, analog ONN designs (Jackson et al., 2018; Moy et al., 2022).

2.3.2. Architecture for on-chip learning

In this work, we perform ONN on-chip-learning using the digital ONN design in an architecture implemented on the Zybo-Z7 development board (Digilent, 2018), which is based on a ZYNQ processor (Xilinx, 2011). The ZYNQ processor is equipped with a Processing System (PS), a dual-core Cortex-A9 processor, and Programmable Logic (PL) resources equivalent to an Artix-7 FPGA. First, for the ONN on-chip learning architecture, ONN digital design is implemented using PL resources as in Abernot et al. (2021) and is controlled by PS to allow the integration of learning algorithms in PS (see Figure 4).

Communication between PS and PL uses the AXI4-Light parallel communication protocol. We use PS as master and PL as slave such that when PS receives external pattern and command, it controls the digital ONN in PL. If PS receives an external learning command, the Master updates weights following the learning rule and sends weights to the digital ONN in PL. If PS receives an external inference command, PS sends the pattern to the ONN and receives the ONN output after inference. AXI4-Light communication accesses four 32-bit AXI4 registers to send and receive information. The latency of weights transmission, for a given ONN size, depends on the weight precision and the number of weights to fit in a 32-bit register.

The learning process starts when PS receives an external learning command in parallel with an input pattern. It engages the update of the weights on PS following the implemented learning rule before sending the updated weights to the digital ONN in PL through the AXI4-light bus. Note, during weight update, ONN is in reset mode. Once the weight update is over, ONN comes back in inference mode and informs PS that the weight update is done. The inference process starts when PS receives an input pattern with an inference command, such that PS transmits the input pattern through AXI4-Light to the digital ONN in PL, the digital ONN infers, and it sends back its output pattern to PS through the AXI4-Light. Note, an additional command performs a reset of the weights to zeros if necessary.

2.3.3. Evaluation

Here, we study the compatibility of HNN learning rules to ONN on-chip learning for pattern recognition and implement the compatible learning rules in our digital ONN on-chip learning architecture. We evaluate the performances of our architecture with the implemented learning rules through three metrics, resource utilization, capacity, and latency.

We analyze the resource utilization of our ONN on-chip learning architecture as it determines the cost of implementation of

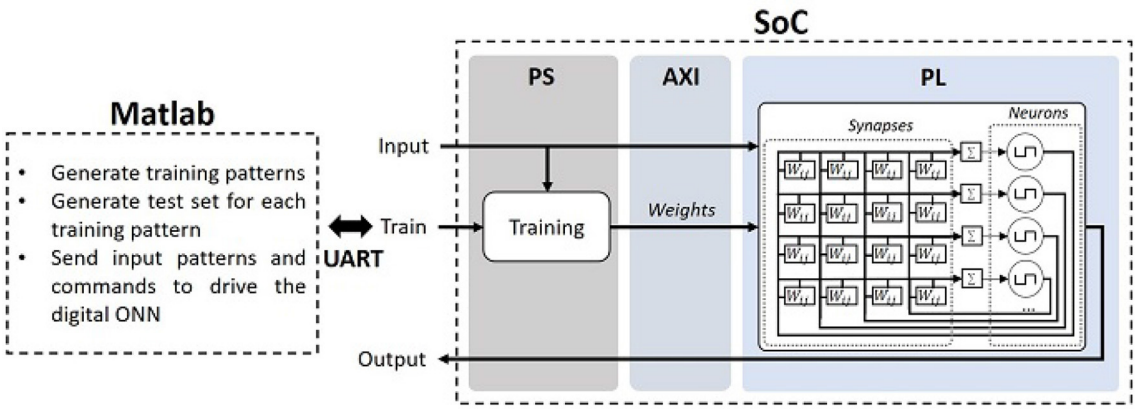


FIGURE 4
Architecture for ONN on-chip learning.

our solution in hardware. In Abernot et al. (2022b), authors showed that resource utilization increases drastically from off-chip to on-chip learning for a 15-neuron ONN. In this work, we go beyond and study the scalability of the on-chip learning architecture for larger ONN sizes.

Memory capacity is defined by the number of patterns a network (HNN or ONN) can correctly learn and retrieve. It can be evaluated by learning patterns in the network and verifying if the network retrieves the correct training pattern when one of the training patterns is presented. However, we believe it is also necessary to verify if the network can retrieve the correct training pattern from corrupted input information, corresponding to none of the training patterns, to evaluate the robustness to noise. In this work, we evaluate the capacity of N-neuron HNN and ONN networks trained with up to N random training patterns, by testing with corrupted input patterns generated from training patterns with up to $N/2$ flipped pixels, represented by the hamming distance. Note, an inference cycle is performed for each input pattern. Also note, the size of the network, as well as the correlation between the training patterns, impact the capacity of the network, so we perform 100 trials for each configuration. We first evaluate HNN capacity on Matlab to validate Hebbian and Storkey learning rules for three HNN sizes (25, 50, and 100 neurons), then we implement Storkey and Hebbian in the on-chip learning architecture to extract the real capacity metric for a 25-neuron ONN because the resource utilization limits the ONN size. A test flow is set up and automatized for testing the digital ONN on-chip learning architecture using Matlab to send commands and patterns to the system through a UART communication protocol (see Figure 4).

We measure the latency of the 25-neuron ONN for on-chip learning. The latency is divided into three parts, the ONN computation latency, the weight computation latency, and the transmission latency. The ONN computation latency is by default stable no matter the weights and size of the network, so we expect it to stay stable. The weight computation latency mainly depends on the learning rule and computation complexity of the learning rule. And the transmission latency depends on the weight precision and the network size.

3. Results

This section presents results obtained with both HNN on Matlab and ONN on FPGA. First, we explain the choice of the most suitable learning rules to implement for ONN on-chip learning. Then, we test the learning rules with ONN on-chip learning constraints in Matlab to study the impact of the weight precision on the HNN capacity and decide which weight precision to apply to the digital ONN design. After, we implement the learning rules in our digital ONN on-chip learning architecture and report on resource utilization, capacity, and latency of our solution for various weight precision.

3.1. Learning rules for ONN on-chip learning

In this work, we focus on local and incremental unsupervised learning algorithms introduced for HNNs to be compatible with other ONN implementations. In particular, Tolmachev and Manton (2020) recently surveyed HNN unsupervised learning rules for pattern recognition and studied the impact of weight symmetry, 0-diagonal, and incrementality on HNN pattern recognition capacity. In this work, we consider the various learning rules from Tolmachev and Manton (2020) as potential candidates for ONN on-chip learning and investigate which ones are best suited for ONN on-chip learning (see Table 1). In Tolmachev and Manton (2020), authors show that iterative rules, requiring learning each pattern for more than one iteration (Diederich and Opper, 1987; Krauth and Mezard, 1987; Gardner, 1988) have better precision than other non-iterative learning rules, however, they are often not incremental, making them not suitable for on-chip learning implementation, as shown in Table 1. Table 1 highlights that, based on the learning rules from Tolmachev and Manton (2020), there are only two unsupervised learning rules which satisfy the ONN on-chip learning constraints, Hebbian and Storkey. Storkey learning rule is known to have better capacity than Hebbian, while requiring more computation. The weights update computation W_{ij} between

10.3389/fnins.2023.1196796

TABLE 1 HNN learning rules features.

Learning rules	Weight symmetry	Zero-diagonal	Local	Incremental
Hebbian	x	x	x	x
Storkey	x	x	x	x
Diederich Opper I		x	x	
Diederich Opper II			x	
Gardner		x	x	
Krauth Mezard			x	
Pseudo-Inverse	x	x		

neuron i and neuron j, in a network of N neurons to learn a novel pattern ϕ with Hebbian learning rule is

$$W_{ij} = W_{ij} + \frac{1}{N}\phi_i\phi_j \tag{2}$$

And with Storkey learning rule is

$$W_{ij} = W_{ij} + \frac{1}{N}(\phi_i\phi_j - \phi_i h_{ji} - h_{ij}\phi_j) \tag{3}$$

with h_{ij} a local field computed with

$$h_{ij} = \sum_{k=1}^{N} W_{ik}\phi_k \tag{4}$$

For the rest of the paper, we implement both Hebbian and Storkey learning rules in our digital ONN on-chip learning architecture.

3.2. Incremental learning with HNN on Matlab

We study the impact of weight precision on HNN accuracy for various HNN sizes. In particular, we analyze the capacity of HNN trained with Hebbian and Storkey for three HNN sizes, 25, 50, and 100 neurons, as well as for five weight precision, 2, 3, 4, 5 bits, and full precision.

Figure 5 shows the HNN capacity for a 100-neuron HNN trained with Storkey with 1 up to 100 training patterns and tested for 100 trials with corrupted input patterns with 1 up to 50 hamming distance. A black pixel represents that over the 100 trials, for a given configuration, all tests were successful, while a white pixel points out that none of the tests were successful. The capacity lines highlight, for each number of training patterns, the maximum hamming distance of corrupted input patterns supported by the network, such that the network successfully associates the corrupted input pattern with a training pattern for at least θ trials over 100, with $\theta = \{85; 90; 85; 100\}$. Then, to simplify the readability of our results, we choose to represent only the capacity lines for one value of θ. We choose $\theta = 90$ to have results representative of a majority of cases and to allow some error tolerance.

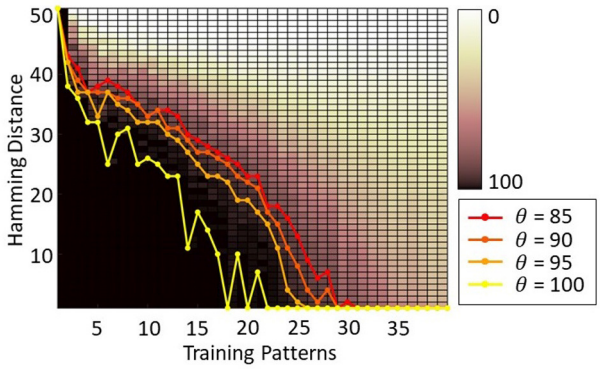


FIGURE 5
Capacity of a 100-neuron HNN trained with Storkey with 100 training patterns tested with corrupted input patterns with different hamming distances (1 up to 50 flipped pixels) with the training patterns. The capacity lines represent for each number of training patterns the maximum hamming distance of corrupted input patterns supported by the network, such that the network successfully associates the corrupted input pattern with a training pattern for at least θ trials over 100.

Figure 6 shows the HNN capacity lines for $\theta = 90$ for the Hebbian and Storkey learning rules for the different weight precision and network size. Figure 6 also plots the error bounds for each weight precision configuration. Figure 6 first highlights the difference in precision and capacity between Storkey and Hebbian learning rules. HNN trained with Storkey can retrieve a larger number of training patterns when initialized with more corrupted input patterns (patterns with larger hamming distances), thus HNN trained with Storkey shows better capacity than HNN trained with Hebbian for all weight precision configurations. Then, Figure 6 displays that for Storkey learning, using 5-bit weight precision, HNN obtains a similar capacity than considering full weights precision. Note, the impact of reducing weight precision to 4-, 3-, or 2-bit precision depends on the network size. The larger the network is, the more impact the reduction of the weight precision has on the network capacity.

3.3. On-chip learning with digital ONN on FPGA

After selecting suitable learning rules and studying their efficiency for HNN on Matlab, we implement Hebbian and Storkey learning rules in our digital ONN on-chip learning architecture and consider three weight precision with 3-, 4-, and 5-bit precision to study the impact on the resource utilization, capacity, latency, and power consumption.

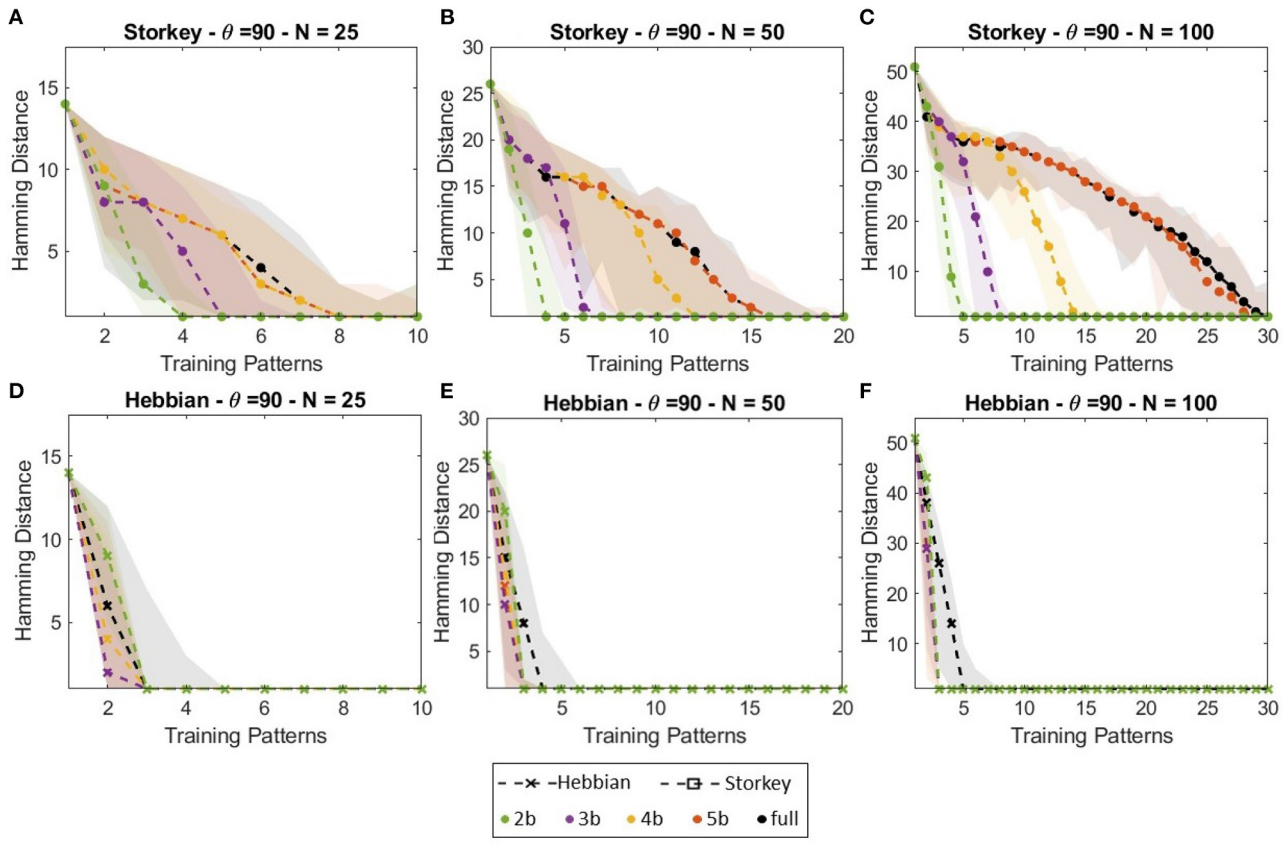


FIGURE 6
Capacity of HNN networks of **(A, D)** 25 neurons, **(B, E)** 50 neurons, and **(C, F)** 100 neurons trained with **(A–C)** Storkey, or **(D–F)** Hebbian with various weight precision. The capacity is represented, for each network size, for each learning rule, and each number of training patterns, by the maximum hamming distance of corrupted input patterns supported by the network, such that the network successfully associates the corrupted input pattern with a training pattern for at least θ = 90 trials over 100 (90%).

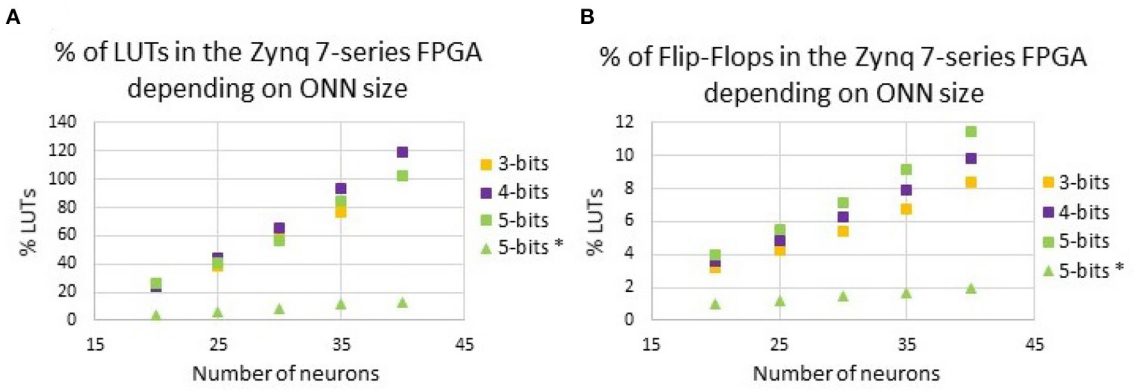


FIGURE 7
Resource utilization for various ONN sizes for various weight precision with **(A)** LUTs, and **(B)** Flip-Flops. We compare with the previous digital ONN with random hard-coded weights in a 5-bit precision (5 bits*).

3.3.1. Resource utilization

First, we report on ONN resource utilization. From Abernot et al. (2022b), we know that for a small 15-neuron scale ONN, re-programmable synapses utilize a large number of resources, in particular Look-Up-Tables (LUTs). In the proposed architecture, a large number of LUTs are used as reconfigurable memory of the weight matrix, so due to the fully-connected ONN architecture, the number of synapses increases following $N(N-1)$ for N neurons, and so the number LUTs also increases. Figure 7 highlights the LUTs and Flip-Flops utilization

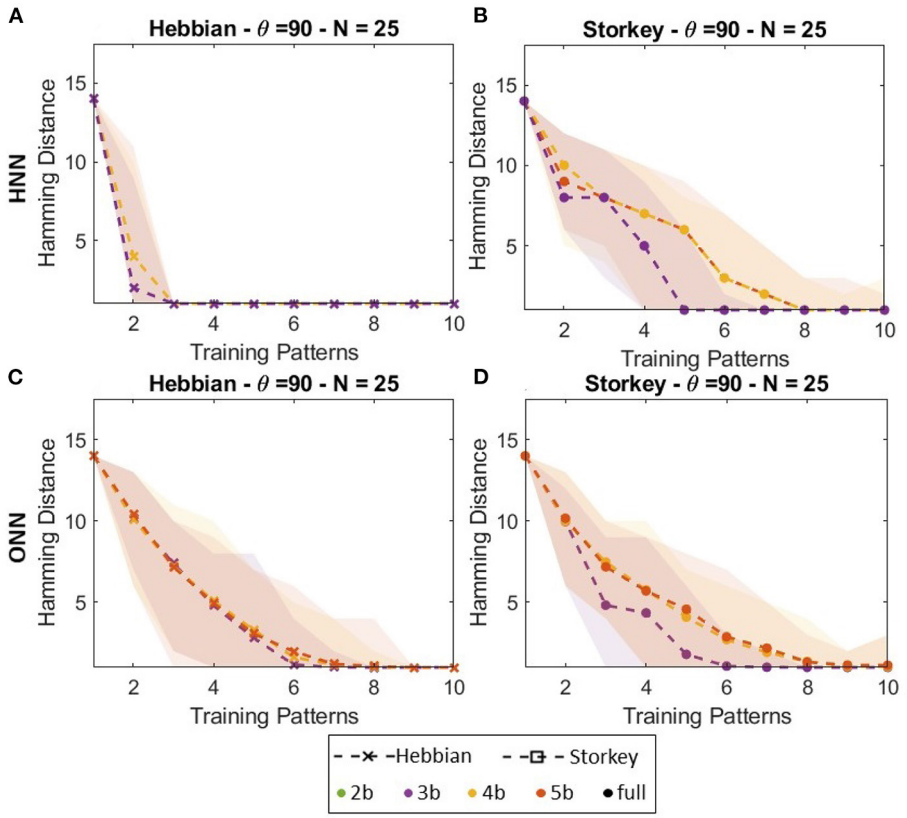


FIGURE 8
Capacity of **(A, B)** HNN and **(C, D)** ONN networks of 25 neurons trained with **(A, C)** Hebbian or **(B, D)** Storkey.

for ONN with 20 up to 40 neurons with and without on-chip learning.

To limit the impact of re-programmable synapses, we analyze the impact of reducing the weight precision on resource utilization. In Figure 7, we report on the number of Look-Up-Tables (LUTs), as well as the number of Flip-Flops (FFs) necessary for our digital ONN implementation, for 3-, 4-, and 5-bit precision. As mentioned previously, in the proposed architecture, a large number of LUTs are used as reconfigurable memory of the weight matrix. Thus, we expect the reduction of the weight precision to also reduce LUTs utilization. However, Figure 7 indicates that for some ONN sizes, reducing the weight precision does not reduce the number of LUTs. For example, for the 35-neuron ONN, the number of LUTs is larger for the 4-bit precision than for the 5-bit precision. We believe it depends on the configuration of the FPGA, which provides fixed-size LUTs. Additionally, the reduction of the weight precision from 5 to 3 bits does not significantly reduce the resource utilization as expected, limiting the ONN size for on-chip learning implementation. With our solution, we can implement an ONN with up to 35 fully-connected neurons with re-programmable synapses. Next, we consider a 25-neuron ONN to report on its capacity and latency.

3.3.2. Capacity

Figure 8 presents capacity lines obtained for a 25-neuron digital ONN trained on-chip with both Hebbian or Storkey for three different weight precision (3, 4, and 5 bits) compared with HNN trained with the same configuration. Figures 6B, D show that for Storkey on-chip learning, HNN and ONN have similar capacities. However, considering Hebbian learning, Figures 8A, C demonstrate ONN has a better capacity than HNN. Figure 8 also shows less ONN capacity variations depending on the weight precision than HNN capacity. These are unexpected as were not observed in previous configurations, but this is, to the best of our knowledge, the first large-scale capacity tests performed with the digital ONN. We believe the difference between HNN and ONN trained with the Hebbian learning rule might come from the difference in the system dynamics between HNN and ONN. Classical HNN can only take two state values, $-1;1$, because of the sign activation function. However, the ONN activation function allows it to take multi-state or continuous values during dynamical evolution. Thus, even if an ONN trained with binary patterns will stabilize to binary phase states $0°;180°$, the activation function, which is difficult to derive, allows non-binary phase states during phase dynamics. We believe that the phase dynamics of the ONN evolve slowly from a corrupted input pattern to the correct training pattern, while the sharp HNN activation function may evolve too fast, reaching a wrong training pattern. HNN may require more precise weights, as with Storkey, to take the correct decision, while the ONN can still evolve to a correct training pattern even with less precise weights. However, we believe it requires additional investigation to draw conclusions. It is important to note that our architecture enables incremental

TABLE 2 Measurements of latency for ONN training and inference with ONN oscillation frequency $F_{onn} = 97.7KHz$ and PS clock frequency $F_{PS} = 667MHz$.

	This work			Abernot et al. (2022b)
Weights	3 bits (μs)	4 bits (μs)	5 bits (μs)	5 bits
Training				
Hebbian learning	55			33 μs
Storkey learning	210			77 μs
Weight precision	140			NA
Weight transmission	18	71	175	86 μs
Total Hebbian	213	266	370	119 μs
Total Storkey	368	421	525	163 μs
Inference				
Input transmission	9			NA
ONN computation	17			NA
Output transmission	18			NA
Total	44			NA

on-chip learning of a digital ONN design with two different learning rules, Hebbian and Storkey, for pattern recognition tasks.

3.3.3. Latency

Finally, we report on training and inference latency for a 25-neuron ONN working at $F_{osc} = 187.5$ KHz. Concerning inference, we measure input pattern transmission latency from PS to PL, ONN computation latency in PL, and ONN output transmission latency from PL to PS. Table 2 shows that ONN inference takes around two to three oscillation cycles to compute, similar to the solution with off-chip learning (Abernot et al., 2021). Then, the transmission of ONN input and output takes 27μs which is 1.5 times higher than the ONN computation. Note, increasing the ONN size will also increase the transmission latency as the information to transmit will be larger, while the ONN computation should stay stable. Thus, the architecture increases the inference latency compared to off-chip learning solutions because of information transmission from PS to PL, and reversely.

Concerning training, we differentiate the latency into three steps, one to perform the training algorithm in PS, another to rescale weights to the corresponding weight precision, and finally to transfer weights from PS to the ONN in PL. Table 2 highlights that Storkey requires more computation time than Hebbian. This is because Storkey requires more computation than Hebbian, see Equations (2) and (3), and PS performs sequential processing. Then, weight transmission increases drastically with the increase of the weight precision and the number of neurons. Reducing the weight precision has an important impact to reduce transmission latency because we use AXI4-Lite with 32-bit parallel transmission.

Our solution, for a network of 25 neurons, allows computing Hebbian in 55 μs, and Storkey in 210 μs. Additionally, to allow reducing weight precision to 3, 4, or 5 bits, additional treatment is necessary, taking 140 μs. Then, transmission time depends on the weight precision taking between 18 and 175 μs. In total, training a fully-connected ONN, configured for 5-bits signed synapses, with a novel training pattern takes 370 μs with the Hebbian learning algorithm and 525 μs with the Storkey learning algorithm. Thus, because Hebbian and Storkey have similar precision in the digital ONN design, it can be more of interest for a system with high time constraints to implement Hebbian rather than Storkey on-chip learning.

3.3.4. Power consumption

We extract the estimated post-place and route power consumption of our digital ONN with re-programmable synapses on Vivado considering the xc7z020-1clg400c target, and we compare it with the digital ONN implementation without the re-programmable synapses (Abernot et al., 2021) and with other fully-connected ONN implementations (Jackson et al., 2018; Bashar et al., 2021; Delacour et al., 2023b). We compute the energy per neuron per oscillation by considering an ONN computation time of three oscillation cycles. Table 3 highlights that the digital ONN with re-programmable synapses requires slightly more energy per oscillation than the digital ONN without re-programmable synapses (Abernot et al., 2021), certainly because of the additional LUTs resources necessary for the on-chip learning. Also, both digital ONNs are in the same energy per oscillation range as the analog ONN implementation in Bashar et al. (2021) as they operate at a lower frequency than the other implementations (Jackson et al., 2018; Delacour et al., 2023b). Using a higher ONN frequency could reduce the computation time, ultimately reducing the energy per computation and oscillation, however, the digital ONN frequency is currently limited by the FPGA.

TABLE 3 Comparison of the digital ONN with re-programmable synapses with other fully-connected ONN implementations.

	Jackson et al. (2018)	Bashar et al. (2021)	Delacour et al. (2023b)	Abernot et al. (2021)	This work
Neurons	100	30	16	60	25
Power	303 mW	1.76 mW	160 μW	20 mW	10 mW
Frequency	1 GHz	45 kHz	1 MHz	187.5 kHz	187.5 kHz
Energy/osc	0.3 pJ	1.3 nJ	10 pJ	1.78 nJ	2.13 nJ

4. Discussion

This paper studies possible algorithms and provides an implementation to perform continual on-chip learning with a digital ONN design for pattern recognition. It highlights that HNN unsupervised learning algorithms are compatible with ONN on-chip learning only if they satisfy two constraints on the weight matrix, the symmetry and the 0-diagonal, and two additional constraints on the learning algorithm, locality, and incrementality. This work evaluated seven state-of-the-art unsupervised learning rules developed for HNN (Tolmachev and Manton, 2020) and defined two of them to be compatible with ONN on-chip learning, Hebbian and Storkey. Both Hebbian and Storkey learning rules exhibit similar capacity results when implemented in the proposed architecture to perform on-chip learning on a 25-neuron ONN, making them both suitable for continual ONN on-chip learning.

The proposed architecture takes advantage of a Zynq processor (Xilinx, 2011) equipped with both PS and PL resources to implement a fully-connected digital ONN introduced in Abernot et al. (2021) with re-programmable synapses in PL, and execute the unsupervised Hebbian and Storkey learning algorithms in PS. The architecture was first introduced in Abernot et al. (2022b) for a small-size ONN with 15 neurons, while this work evaluates the scalability of the architecture. First, it is important to highlight that the solution does not require many changes from the first digital ONN design, making it easy to adapt and install. The main scalability limitation of the architecture is due to the digital ONN re-programmable synapses which demand a large number of LUTs, even with reduced weight precision, limiting the ONN size up to 35 fully-connected oscillators while the digital ONN without re-programmable synapses could reach hundreds of fully-connected neurons (Abernot et al., 2021). Another limitation of the architecture is the latency induced by the separation between ONN learning and computation in PS and PL. On one side, PS allows to implement and compute a large panel of unsupervised learning algorithms, executing them sequentially with a fast frequency of $F_{ps} = 666$ MHz. On the other side, it generates latency to transmit the weights from PS to PL, increasing with the ONN size. An alternative solution is to implement the training algorithms using the parallel properties of PL resources to provide fast training and remove the transmission latency. However, we believe it would utilize additional PL resources, including LUTs, which are already limited. Another solution is to use other communication protocols than AXI-Lite between PS and PL, such as AXI-stream which provides more parallel transmission. Overall, our solution permits to train a 25-neuron ONN in hundreds of microseconds, between 350 and 550 μs which is the first solution to perform ONN on-chip learning.

Future work will first explore alternative solutions to try to overcome the current limitations of the ONN on-chip learning architecture. Furthermore, the next developments will focus on possible applications with the ONN on-chip learning architecture. The digital ONN design has already been used for sensor data treatment in various applications, like interfacing with a camera for image recognition (Abernot et al., 2021) or using proximity sensor data to perform obstacle avoidance (Abernot et al., 2022a), so we are confident on the integration of our architecture with different sensors. Possible applications for the digital ONN on-chip learning architecture could be in the robotics domain where real-time continual learning is often necessary, and where the digital ONN design already showcased good performances (Abernot et al., 2022a). For example, navigation, in the context of mobile robots, is a complex task depending on the environment, where continuous learning is necessary to adapt to evolving situations. A first proof of concept of two pre-trained cascaded ONNs performing obstacle avoidance from proximity sensors was shown in Abernot et al. (2022a). Though in Abernot et al. (2022a), the pre-trained ONNs are capable of finding a novel direction using information from 15 proximity sensors whose configurations are used to define the training patterns. However, if we consider an obstacle avoidance application using more sensor information than 15 proximity sensors, it becomes impossible to define all possible training patterns before inference. Using ONN on-chip learning allows training the ONN continuously through time depending on the environmental configuration given by the sensory information. Thus, we believe that using the ONN on-chip learning architecture can be beneficial in the case of applications with large-scale inputs where all possible configurations can not be anticipated. A first idea was proposed recently to perform real-time ONN on-chip learning for an obstacle avoidance application using the proposed architecture (Abernot et al., 2023b), however, a demonstrator is yet to be developed.

5. Conclusion

This work analyses unsupervised learning rules for Oscillatory Neural Network (ONN) learning for pattern recognition tasks, and in particular for continual ONN on-chip learning. We evaluate the adaption of unsupervised learning rules developed for Hopfield Neural Networks (HNNs) for ONN on-chip learning and show that Hebbian and Storkey learning rules are both suitable for ONN on-chip learning. Additionally, we propose an architecture capable of performing ONN on-chip learning using a digital ONN implementation with various unsupervised learning algorithms. It uses a Processing System (PS) of a Zynq processor to implement

the learning algorithms and Programmable Logic (PL) resources to implement the digital ONN. We point out that the architecture limits the network in size, with up to 35 neurons, due to the large resource utilization. Also, with the proposed architecture, learning and inference latency increase with the network size, which can become a limitation for time-constrained systems. Our current solution can train a 25-neuron ONN on-chip in hundreds of micro-seconds, between 350 and 550 μs. This is, to the best of our knowledge, the first solution to perform ONN on-chip learning with unsupervised learning algorithms for pattern recognition. We believe it can be useful for investigating novel ONN learning algorithms and applications such as reinforcement learning for robotic applications.

Data availability statement

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

Author contributions

AT-S motivated the project and experiments. MA performed the survey on the learning rules, performed the HNN tests, developed the on-chip learning architecture, and performed the measurements. AT-S and NA were involved in the discussion and editing of the manuscript and provided valuable inputs at multiple stages of this work. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the European Union's Horizon 2020 research and innovation program, EU H2020 NEURONN project under Grant No. 871501 (www.neuronn.eu).

Acknowledgments

We thank T. Gil (LIRMM—CNRS), M. Jimenez, J. Nunez, and M. J. Avedillo (IMSE—CSIC) for fruitful exchanges and constructive suggestions on this research work.

Conflict of interest

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

Publisher's note

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

References

Abernot, M., Gauthier, S., Gonos, T., and Todri-Sanial, A. (2023a). "Sift-ONN: sift feature detection algorithm employing ONNs for edge detection," in *Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference, NICE '23* (New York, NY: Association for Computing Machinery), 100–107. doi: 10.1145/3584954.3584999

Abernot, M., Gil, T., Jiménez, M., Núñez, J., Avellido, M. J., Linares-Barranco, B., et al. (2021). Digital implementation of oscillatory neural network for image recognition applications. *Front. Neurosci.* 15:713054. doi: 10.3389/fnins.2021.713054

Abernot, M., Gil, T., Kurylin, E., Hardelin, T., Magueresse, A., Gonos, T., Jimenez, M., et al. (2022a). "Oscillatory neural networks for obstacle avoidance on mobile surveillance robot E4," in *2022 International Joint Conference on Neural Networks (IJCNN)* (Padova), 1–8. doi: 10.1109/IJCNN55064.2022.9891923

Abernot, M., Gil, T., and Todri-Sanial, A. (2022b). "On-chip learning with a 15-neuron digital oscillatory neural network implemented on zynq processor," in *Proceedings of the International Conference on Neuromorphic Systems 2022, ICONS '22* (New York, NY: Association for Computing Machinery), 1–4. doi: 10.1145/3546790.3546822

Abernot, M., Gil, T., and Todri-Sanial, A. (2023b). "ONN-based on-chip learning for obstacle avoidance on mobile robot," in *Smart System Integration* (Bruges). Available online at: https://hal.science/hal-04007886/document

Abernot, M., and Todri-Sanial, A. (2023). Simulation and implementation of two-layer oscillatory neural networks for image edge detection: bidirectional and feedforward architectures. *Neuromorph. Comput. Eng.* 3:014006. doi: 10.1088/2634-4386/acb2ef

Bashar, M. K., Mallick, A., and Shukla, N. (2021). Experimental investigation of the dynamics of coupled oscillators as ising machines. *IEEE Access* 9, 148184–148190. doi: 10.1109/ACCESS.2021.3124808

Bashar, M. K., Mallick, A., Truesdell, D. S., Calhoun, B. H., Joshi, S., and Shukla, N. (2020). Experimental demonstration of a reconfigurable coupled oscillator platform to solve the max-cut problem. *IEEE J. Explor. Solid State Comput. Devices Circuits* 6, 116–121. doi: 10.1109/JXCDC.2020.3025994

Christensen, D. V., Dittmann, R., and Linares-Barranco, B. (2022). 2022 roadmap on neuromorphic computing and engineering. *Neuromorph. Comput. Eng.* 2:022501. doi: 10.1088/2634-4386/ac4a83

Csaba, G., and Porod, W. (2020). Coupled oscillators for computing: a review and perspective. *Appl. Phys. Rev.* 7:011302. doi: 10.1063/1.5120412

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., et al. (2022). A continual learning survey: defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 3366–3385. doi: 10.1109/TPAMI.2021.3057446

Delacour, C., Carapezzi, S., Abernot, M., and Todri-Sanial, A. (2023a). Energy-performance assessment of oscillatory neural networks based on vo $_2$ devices for future edge AI computing. *IEEE Trans. Neural Netw. Learn. Syst.* 1–14. doi: 10.1109/TNNLS.2023.3238473

Delacour, C., Carapezzi, S., Boschetto, G., Abernot, M., Gil, T., and Todri-Sanial, A. (2022). "VO2-based oscillatory ising machine: the role of external temperature on performance," in *IEEE NANO 2022 - 22nd IEEE International Conference on Nanotechnology* (Palma de Mallorca).

Delacour, C., Carapezzi, S., Boschetto, G., Abernot, M., Gil, T., and Todri-Sanial, A. (2023b). A mixed-signal oscillatory neural network for scalable analog computations in phase domain. *[Preprint]*.

Delacour, C., and Todri-Sanial, A. (2021). Mapping hebbian learning rules to coupling resistances for oscillatory neural networks. *Front. Neurosci.* 15:694549 doi: 10.3389/fnins.2021.694549

Diederich, S., and Opper, M. (1987). Learning of correlated patterns in spin-glass networks by local learning rules. *Phys. Rev. Lett.* 58, 949–952. doi: 10.1103/PhysRevLett.58.949

Digilent (2018). *Zybo z7 reference manual*. Available online at: https://digilent.com/reference/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf

Fernandes, D., Olivier, P., and Navaux, A. (2004). "A low complexity digital oscillatory neural network for image segmentation," in *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004* (Rome), 365–368. doi: 10.1109/ISSPIT.2004.1433795

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci*. 3, 128–135. doi: 10.1016/S1364-6613(99)01294-2

Gardner, E. (1988). The space of interactions in neural network models. *J. Phys. A* 21:257. doi: 10.1088/0305-4470/21/1/030

Gosti, G., Folli, V., Leonetti, M., and Ruocco, G. (2019). Beyond the maximum storage capacity limit in hopfield recurrent neural networks. *Entropy* 21:726. doi: 10.3390/e21080726

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. (2020). "REMIND your neural network to prevent catastrophic forgetting," in *Computer Vision-ECCV 2020*, eds A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm (Springer International Publishing), 466–483. doi: 10.1007/978-3-030-58598-3_28

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A*. 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hoppensteadt, F., and Izhikevich, E. (1997). "Associative memory of weakly connected oscillators," in *Proceedings of International Conference on Neural Networks (ICNN'97)*, *Vol. 2* (Houston, TX), 1135–1138. doi: 10.1109/ICNN.1997.616190

Izhikevich, E., and Kuramoto, Y. (2006). "Weakly coupled oscillators," in *Encyclopedia of Mathematical Physics*, eds J. P. Françoise, G. L. Naber, and T. S. Tsun (Oxford: Academic Press), 448–453. doi: 10.1016/B0-12-512666-2/00106-1

Jackson, T., Pagliarini, S., and Pileggi, L. (2018). "An oscillatory neural network with programmable resistive synapses in 28 Nm CMOS," in *2018 IEEE International Conference on Rebooting Computing (ICRC)* (McLean, VA), 1–7. doi: 10.1109/ICRC.2018.8638600

Jedlicka, P., Tomko, M., Robins, A., and Abraham, W. C. (2022). Contributions by metaplasticity to solving the catastrophic forgetting problem. *Trends Neurosci*. 45, 656–666. doi: 10.1016/j.tins.2022.06.002

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. (2022). Towards continual reinforcement learning: a review and perspectives. *J. Artif. Intell. Res*. 75, 1401–1476. doi: 10.1613/jair.1.13673

Krauth, W., and Mezard, M. (1987). Learning algorithms with optimal stability in neural networks. *J. Phys. A Math. Gen*. 20:L745. doi: 10.1088/0305-4470/20/11/013

Landge, S., Saraswat, V., Singh, S. F., and Ganguly, U. (2020). "n-oscillator neural network based efficient cost function for n-city traveling salesman problem," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow), 1–8. doi: 10.1109/IJCNN48605.2020.9206856

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., and Díaz-Rodríguez, N. (2020). Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. *Inform. Fus*. 58, 52–68. doi: 10.1016/j.inffus.2019.12.004

Lo, H., Moy, W., Yu, H., Sapatnekar, S., and Kim, C. H. (2023). A 48-node all-to-all connected coupled ring oscillator ising solver chip. *[Preprint]*. doi: 10.21203/rs.3.rs-2395566/v1

Lobo, J. L., Del Ser, J., Bifet, A., and Kasabov, N. (2019). Spiking neural networks and online learning: an overview and perspectives. *arXiv*. *arXiv:1908.08019 [cs]*. doi: 10.1016/j.neunet.2019.09.004

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw*. 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7

Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., and Sanner, S. (2022). Online continual learning in image classification: an empirical survey. *Neurocomputing* 469, 28–51. doi: 10.1016/j.neucom.2021.10.021

McCloskey, M., and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. *Psychol. Learning Motiv*. 24, 109–165. doi: 10.1016/S0079-7421(08)60536-8

Moy, W., Ahmed, I., and Chiu, P. (2022). A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nat. Electron*. 5, 310–317. doi: 10.1038/s41928-022-00749-3

Muezzinoglu, M., Guzelis, C., and Zurada, J. (2003). A new design method for the complex-valued multistate Hopfield associative memory. *IEEE Trans. Neural Netw*. 14, 891–899. doi: 10.1109/TNN.2003.813844

Nikonov, D. E., Csaba, G., Porod, W., Shibata, T., Voils, D., Hammerstrom, D., et al. (2015). Coupled-oscillator associative memory array operation for pattern recognition. *IEEE J. Explor. Solid-State Comput. Devices Circuits* 1, 85–93. doi: 10.1109/JXCDC.2015.2504049

Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., et al. (2022). The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci*. 16:795876. doi: 10.3389/fnins.2022.795876

Pehlevan, C., and Chklovskii, D. B. (2019). Neuroscience-inspired online unsupervised learning algorithms: artificial neural networks. *IEEE Signal Process. Mag*. 36, 88–96. doi: 10.1109/MSP.2019.2933846

Personnaz, L., Guyon, I., and Dreyfus, G. (1986). Collective computational properties of neural networks: New learning mechanisms. *Phys. Rev. A* 34, 4217–4228. doi: 10.1103/PhysRevA.34.4217

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., et al. (2021). Hopfield networks is all you need. *[Preprint]*. Available online at: https://arxiv.org/pdf/2008.02217.pdf

Raychowdhury, A., Parihar, A., Smith, G. H., Narayanan, V., Csaba, G., Jerry, M., et al. (2019). Computing with networks of oscillatory dynamical systems. *Proc. IEEE* 107, 73–89. doi: 10.1109/JPROC.2018.2878854

Ring, M. B. (1997). CHILD: a first step towards continual learning. *Mach. Learn*. 28, 77–104. doi: 10.1023/A:1007331723572

Schwemmer, M. A., and Lewis, T. J. (2012). *The Theory of Weakly Coupled Oscillators*. New York, NY: Springer New York, 3–31. doi: 10.1007/978-1-4614-0739-3_1

Tanaka, G., and Aihara, K. (2009). Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction. *IEEE Trans. Neural Netw*. 20, 1463–1473. doi: 10.1109/TNN.2009.2025500

Thangarasa, V., Miconi, T., and Taylor, G. W. (2020). Enabling continual learning with differentiable hebbian plasticity. *arXiv*. doi: 10.1109/IJCNN48605.2020.9206764

Thrun, S., and Mitchell, T. M. (1995). Lifelong robot learning. *Robot. Auton. Syst*. 15, 25–46. doi: 10.1016/0921-8890(95)00004-Y

Todri-Sanial, A., Carapezzi, S., Delacour, C., Abernot, M., Gil, T., Corti, E., et al. (2022). How frequency injection locking can train oscillatory neural networks to compute in phase. *IEEE Trans. Neural Netw. Learn. Syst*. 33, 1996–2009. doi: 10.1109/TNNLS.2021.3107771

Tognoli, E., and Kelso, J. A. S. (2009). Brain coordination dynamics: true and false faces of phase synchrony and metastability. *Prog. Neurobiol*. 87, 31–40. doi: 10.1016/j.pneurobio.2008.09.014

Tolmachev, P., and Manton, J. H. (2020). "New insights on learning rules for hopfield networks: memory and objective function minimization," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow), 1–8. doi: 10.1109/IJCNN48605.2020.9207405

Vaidya, J., Surya Kanthi, R. S., and Shukla, N. (2022). Creating electronic oscillator-based ising machines without external injection locking. *Sci. Rep*. 12:981. doi: 10.1038/s41598-021-04057-2

Wang, J., Belatreche, A., Maguire, L., and McGinnity, T. M. (2014). An online supervised learning method for spiking neural networks with adaptive structure. *Neurocomputing* 144, 526–536. doi: 10.1016/j.neucom.2014.04.017

Wang, T., and Roychowdhury, J. (2019). "OIM: oscillator-based ising machines for solving combinatorial optimisation problems," in *Unconventional Computation and Natural Computation*, eds I. McQuillan and S. Seki (Tokyo: Springer International Publishing), 232–256. doi: 10.1007/978-3-030-19311-9_19

Xilinx, A. (2011). *Zynq: Socs With Hardware and Software Programmability*. Available online at: https://docs.xilinx.com/v/u/en-US/ds190-Zynq-7000-Overview

frontiers | Frontiers in Neuroscience

TYPE Review
PUBLISHED 05 October 2023
DOI 10.3389/fnins.2023.1253075

OPEN ACCESS

EDITED BY
Konrad Szaciłowski,
AGH University of Science and Technology, Poland

REVIEWED BY
Seiya Kasai,
Hokkaido University, Japan
Tapobrata Som,
Institute of Physics, Bhubaneswar, India

*CORRESPONDENCE
Alex James
✉ apj@ieee.org

RECEIVED 04 July 2023
ACCEPTED 13 September 2023
PUBLISHED 05 October 2023

CITATION
R RT, Das RR, Reghuvaran C and James A (2023) Graphene-based RRAM devices for neural computing. *Front. Neurosci.* 17:1253075. doi: 10.3389/fnins.2023.1253075

Graphene-based RRAM devices for neural computing

Rajalekshmi T. R, Rinku Rani Das, Chithra Reghuvaran and Alex James*

Digital University, Thiruvananthapuram, Kerala, India

Resistive random access memory is very well known for its potential application in in-memory and neural computing. However, they often have different types of device-to-device and cycle-to-cycle variability. This makes it harder to build highly accurate crossbar arrays. Traditional RRAM designs make use of various filament-based oxide materials for creating a channel that is sandwiched between two electrodes to form a two-terminal structure. They are often subjected to mechanical and electrical stress over repeated read-and-write cycles. The behavior of these devices often varies in practice across wafer arrays over these stresses when fabricated. The use of emerging 2D materials is explored to improve electrical endurance, long retention time, high switching speed, and fewer power losses. This study provides an in-depth exploration of neuro-memristive computing and its potential applications, focusing specifically on the utilization of graphene and 2D materials in RRAM for neural computing. The study presents a comprehensive analysis of the structural and design aspects of graphene-based RRAM, along with a thorough examination of commercially available RRAM models and their fabrication techniques. Furthermore, the study investigates the diverse range of applications that can benefit from graphene-based RRAM devices.

KEYWORDS

chemical vapor deposition (CVD), cryptography, graphene, neuromorphic computing, resistive random access memory (RRAM)

1. Introduction

Graphene-based resistive random access memory (RRAM) devices have gained significant attention in recent years for their potential applications in neural computing. Graphene, a two-dimensional carbon material, has exceptional electrical and mechanical properties, making it an attractive candidate for RRAM devices. RRAM is considered one of the most promising emerging non-volatile memory, a potentially universal memory device that comes under the broad category of memristive systems (Meena et al., 2014). The advantage of RRAM is attributed to the ease of fabrication of a two-terminal structure that can be used to create efficient crossbar arrays, high read speeds, and low area overheads. The RRAMs in the crossbar can emulate multiply and accumulate (MAC) computations that are universal operations essential for implementing neural computations.

RRAM is a memory based on a resistive switching mechanism where the conducting filament is created and broken due to a change of external voltage (Yu et al., 2011a). The binary RRAMs operate in two states: low resistance state (LRS) and high resistance state (HRS). Various types of electrodes and metal oxides can be used for RRAM structure. Titanium, hafnium, silicon, germanium, and nickel are the most common oxide materials, whereas silicon, silver, indium, and tantalum are familiar electrode materials used in RRAM memory devices.

Unfortunately, RRAM memory devices face various limitations with the aforementioned electrode and oxide materials (Zhu et al., 2015). For accomplishing the resistive switching property, the electrode, and conducting filament can be modified with a wide variety of materials. The electrode materials used for RRAM are divided into the following five categories: (i) elementary substance electrodes, (ii) silicon-based electrodes, (iii) alloy electrodes, (iv) oxide electrodes, and (v) nitride-based electrodes (Zahoor et al., 2020). Depending on the electrode material, the number of possible states in the RRAM varies (Prakash and Hwang, 2019). As the number of states increases, the device finds application as an analog data storage device.

In RRAM, the graphene-related materials have been incorporated to increase the switching speed, retention time, endurance, and power consumption to improve the performance as a non-volatile memory (Rehman et al., 2020). Graphene provides additional properties such as transparency, flexibility, enhanced heat dissipation due to the high thermal conductivity of graphene, and chemical stability. Other than these properties, as a two-dimensional system, graphene can provide more than two states for the memristive device in implementing synapses for neuromorphic computing. It is reported that till now more than 16 states are possible with graphene in the memristive system (Schranghamer et al., 2020). Building more than two stable states in RRAMs to form analog computing systems or using them for analog storage is a open problem in RRAM-based systems.

With graphene-enabled RRAMs, it is expected that the higher number of states can improve the storage density and improve the reliability of the device. Graphene-enhanced RRAM exhibits faster switching speeds and enduring performance due to high carrier mobility, and the unique two-dimensional structure minimizes filament variability, ensuring stable set/reset processes in RRAM devices. Exceptional thermal and mechanical stability of graphene boosts RRAM features by optimizing performance across varying conditions (Galashev and Rakhmanova, 2014; Pan et al., 2017; Rehman et al., 2020). It is reported that RRAM devices offer a switching speed of less than 10 ns, power losses of about 10 pJ, lower threshold voltage of less than 1V, long retention time of greater than 10 years, high electrical endurance with more than 10^8 voltage cycles, and extended mechanical robustness of 500 bending cycles. These advantages are complemented by its ability to tolerate high-temperature variations. Graphene as an interface layer acts as a resistive switching medium which help to minimize power dissipation with low contact resistance. Graphene helps to optimize the surface effect such as photodesorption and chemisorption which are varied due to the increase and decrease of the temperature.

This review starts with an overview of neuro-memristive computing, graphene, and its synthesis techniques. Furthermore, the RRAM, working principle, and the resistive switching mechanism are discussed. The incorporation of graphene and graphene oxide in RRAM as an electrode, and the middle layer is elaborated in detail. The role of graphene in RRAM, to enhance the properties such as endurance, and retention is analyzed, and the enhancement in flexibility and transparency is discussed. The progress of multilevel cell storage in RRAM is reviewed in detail. Furthermore, the commercially available RRAM models and their fabrication methods, complementary metal-oxide-semiconductor (CMOS) compatibility with RRAM are also discussed.

2. Neuro-memristive computing

2.1. Memristive devices and neural dynamics

Memristive devices have been studied for their potential to create artificial neural networks that can learn and adapt in a manner similar to biological neural networks (Huang et al., 2020). These devices can be used to build artificial synapses that can modify their strength based on the pattern of electrical signals they receive. This is similar to how biological synapses modify their strength in response to the timing and frequency of incoming electrical signals (Zhang et al., 2023). Based on this, one potential application of memristive devices in neural dynamics is in the development of neuromorphic computing systems (Ma et al., 2018). These systems are designed to mimic the way the brain processes information, and memristive devices could provide a way to build artificial neural networks that are more efficient and flexible than traditional computing systems (Shehab et al., 2022). This section will cover the details of different kinds of memristive devices, their working, and their viability for application in neuromorphic computing systems.

Memristor is one kind of two-terminal device, considered a new-generation non-volatile memory (NVM) device. This new computing system proposed by Sano et al. (2013) can store information by changing the resistance of a material, whereas conventional memory devices program data by change of capacitance (Im et al., 2020). A pinched hysteresis loop is a characteristic feature of a memristor. The loop represents the behavior of the memristor as the voltage or current applied to it is varied as shown in Figure 1. The pinched hysteresis loop is a distinctive characteristic of memristors and distinguishes them from other electronic devices such as resistors, capacitors, and inductors. The pinched hysteresis loop arises due to the inherent properties of the memristor's material and structure, which allow it to exhibit memory and resistance variations based on the history of applied voltage or current. The exact shape and characteristics of the loop depend on the specific properties of the memristor, including its materials, fabrication methods, and operating conditions. The pinched hysteresis loop of a memristor has significant implications for applications in areas such as memory devices, neuromorphic computing, and analog signal processing. It enables the memristor to store information based on its resistance state and offers unique opportunities for non-volatile memory and computing architectures. The conventional memristor model and its symbol are shown in Figures 2A, B.

These devices offer several advantages over conventional memory technologies such as flash, dynamic random access memory (DRAM), and static random access memory (SRAM), including high density, low power consumption, and fast switching speeds (Yang and Williams, 2013). The combination of metal electrodes and insulators constructs a memristor configuration. The schematic diagram of the cross-point device, showing metallic

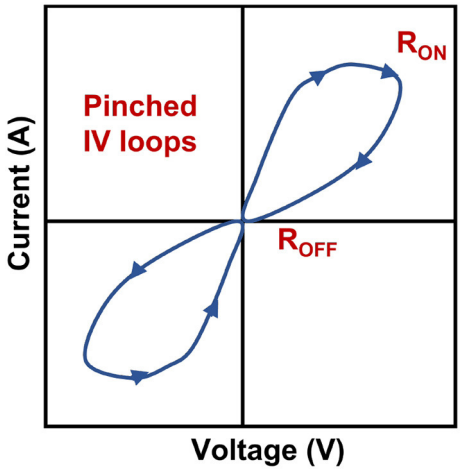


FIGURE 1
Example of pinched hysteresis loop of memristor.

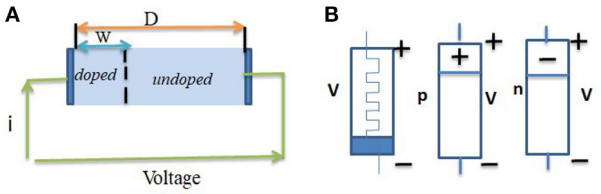


FIGURE 2
(A) Memristor model according to Strukov et al. (2008). **(B)** Traditional symbol, p-type and n-type memristors (copyright Starzyk et al., 2014).

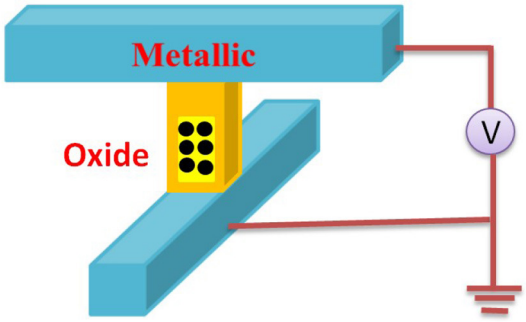


FIGURE 3
Schematic of the cross-point device, showing metallic top and bottom electrodes, and switching oxide (Yang and Williams, 2013).

top and bottom electrodes and switching oxide is shown in Figure 3. Resistive switching, phase change, spintronics ferroelectric, etc. are the various kinds of properties of memristor devices that are contributing to the development of emerging electronic technologies. Among them, a resistive switching memristor (RSM) is the most common memristive device which has low power consumption, high endurance, and potential for use in neuromorphic computing (Prodromakis and Toumazou, 2010; Yu et al., 2018). The applied voltage to the electrodes in the RSM device creates an electric field across the metal oxide layer, causing a change in the oxidation state of the material. This oxidation state changes the resistance of the material which can be detected and used to store data. Phase change element based phase change memory (PCM) is another type of memristive device that uses a material to change its physical state between a crystalline phase (low resistance) and an amorphous phase (high resistance) in response to heat or electric current. Spintronics memristors are a new type of magnetic RAM (MRAM) that works on magnetic tunnel junction (MTJ) (Xue et al., 2011) and offers high speed and high endurance performance. The resistance value has changed due to the spin of the electron and the storage of the data. Two ferromagnetic layers (FM) of these devices are separated by a non-magnetic (NM) layer. When an electric current is applied to the device, the spin of electrons in the magnetic layers is affected, causing a change in the resistance of the device. Ferroelectric tunnel junction (FTJ) (Ambriz-Vargas et al., 2017) is the most significant ferroelectric memory device for neuromorphic computation, having an insulating layer in between two metal electrodes. This ferroic nanostructure is comprised of an ultra-thin ferroelectric barrier, and its dominant mechanism is quantum electron tunneling. In this structure, electrons are able to penetrate through the potential barrier of the ultra-thin insulator. As research in this field continues to progress, memristive devices are expected to play an increasingly important role in the development of advanced computing and memory technologies.

Memristive devices are of great interest in the field of neuromorphic computing because they can be used to emulate the synaptic connections between neurons in the brain. The neural dynamics of memristive devices refers to the behavior of these devices when they are used to implement neural networks. When memristive devices are used as synapses in a neural network, their resistance values change over time in response to the input signals that they receive (Boybat et al., 2018). This behavior can be used to implement learning in the neural network, allowing it to adapt to new inputs and improve its performance over time. The dynamics of memristive devices in neural networks are highly non-linear and can be difficult to predict (Brivio et al., 2021). However, researchers have developed models and simulations to study the behavior of these devices in neural networks.

2.2. Memristors in crossbar

Memristors in crossbar arrays are a type of non-volatile memory technology that holds promise for high-density, low-power, and high-speed computing applications (Xia and Yang, 2019). In a crossbar array, memristors are arranged in a grid pattern, with one set of wires running vertically and another set of wires running horizontally, forming a series of intersecting points. At each cross-point, a memristor can be programmed to either a high or low resistance state, representing a binary 1 or 0, respectively. By applying voltage to the appropriate sets of wires, the resistance state of the memristor can be read or written. This allows for parallel access to multiple memory cells, making crossbar arrays

a potential solution for memory-intensive tasks such as machine learning and artificial intelligence.

A single memristor or one-transistor/one-resistor (1T, 1R) memristor array typically refers to a configuration where memristors are organized in a regular grid pattern. The purpose of a single memristor array is to enable the simultaneous operation and interconnection of multiple memristors (Xu et al., 2021). In a 1T, 1R memristor array, each memristor is paired with a transistor. The transistor serves as the access device or switch, allowing individual memristors within the array to be addressed and read or written to Kim et al. (2012). The key advantage of a 1T, 1R memristor array is its high density and potential for low-power operation. By combining the storage element (memristor) and the access device (transistor) into a single unit, the overall footprint of the memory array can be reduced. There are various ways to arrange the memristors, depending on the desired application and circuit design (Lu et al., 2022). The two-memristor crossbar array is a grid-like structure where the two memristors are positioned at the intersection of a row and a column. The rows and columns are connected to input and output nodes or other circuit elements. This configuration is commonly used in memristive crossbar arrays, where the resistance states of the memristors can be manipulated to enable or disable the connections between rows and columns (Vourkas et al., 2016). Crossbar arrays are particularly relevant in applications such as memory arrays, neural networks, and digital logic circuits (Li et al., 2021). In a bridge memristive crossbar array, two memristors are connected in series between two nodes, forming a bridge structure. The nodes can represent inputs, outputs, or intermediate connections in a larger circuit. The bridge configuration allows for specific control over the flow of current or signals through the array. By adjusting the resistance states of the individual memristors in the bridge, it is possible to selectively enable or disable the connection between the two nodes. This can be achieved by applying appropriate voltage or current across the bridge.

Memristors in crossbar arrays also have the potential for use in neuromorphic computing, which seeks to emulate the structure and function of the human brain (Xia and Yang, 2019). Memristor-based crossbar arrays can potentially perform tasks such as pattern recognition and decision-making in a highly efficient and parallelized manner. Starzyk et al. (2014) developed a novel neural network architecture that utilizes a compact crossbar layout of memristors, which allows us to preserve a high density of synaptic connections. Yakopcic et al. (2019) studied a memristor-based neuromorphic system for ex-situ training of multi-layer perceptron algorithms. This technique facilitates the direct translation of neural algorithm weights onto the resistive grid of a memristor crossbar. It is observed that a parallel crossbar improves the speed and power dissipation. Hu et al. (2012b) proposed a memristive crossbar array for high-speed image processing. It exhibits automatic memory, continuous output, and high-speed parallel computation, making it well suited for implementation in VLSI (very large-scale integration) technology. Huang et al. (2021) developed a vertical crossbar MIM (metal insulator metal) RRAM device for neuromorphic computing that is based on the 2D material $ReSe_2$. This design has been shown to exhibit improved accuracy when used in brain-inspired neuromorphic computing systems.

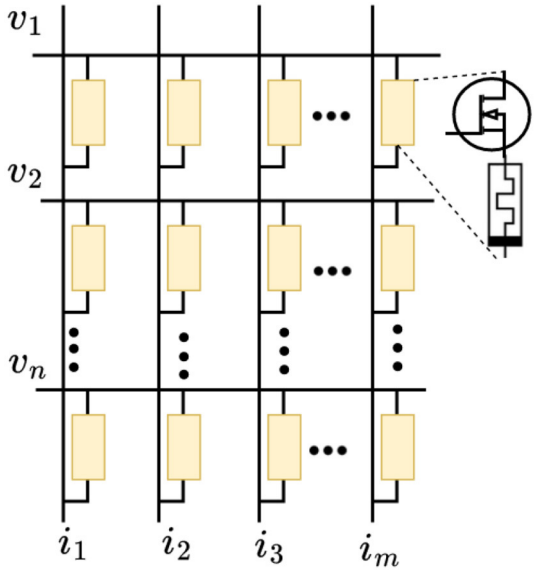


FIGURE 4
ITIM configuration for implementing DNN neural network.

2.3. Neuro-memristive architectures

The memristive circuits and computing architectures are one of the promising solutions for implementing neuromorphic computing. The memristor implementations provide various advantages such as scalability, on-chip area and power reduction, efficiency, and adaptability, especially for device scale-up architectures. There are existing different memristive neuromorphic architectures in the literature used for edge computing applications. The section reviews the most popular neural architectures for edge computing applications.

2.3.1. Deep neural network (DNN)

The DNN is implemented using memristor crossbar arrays. Each DNN layer is implemented using one transistor/memristor (1T 1M) configuration as in Figure 4. Each layer consists of M word lines (WLs) and N bit lines (BLs). The transistor switch enables or disables the column-wise memristor nodes. In Figure 4, v_1, v_2, ... v_n from the inputs, conductance $g_{i,j}$ of memristors as weights and columns current $i_1, i_2, ... i_m$ as outputs, where i, j are the coordinates of the crossbar node. The output currents indicate the weighted summation of input voltages. The bias is included as an additional input line.

2.3.2. Convolutional neural network (CNN)

There are several analog memristive crossbar implementations of CNN architecture (R et al., 2022). Figure 5 shows the hardware

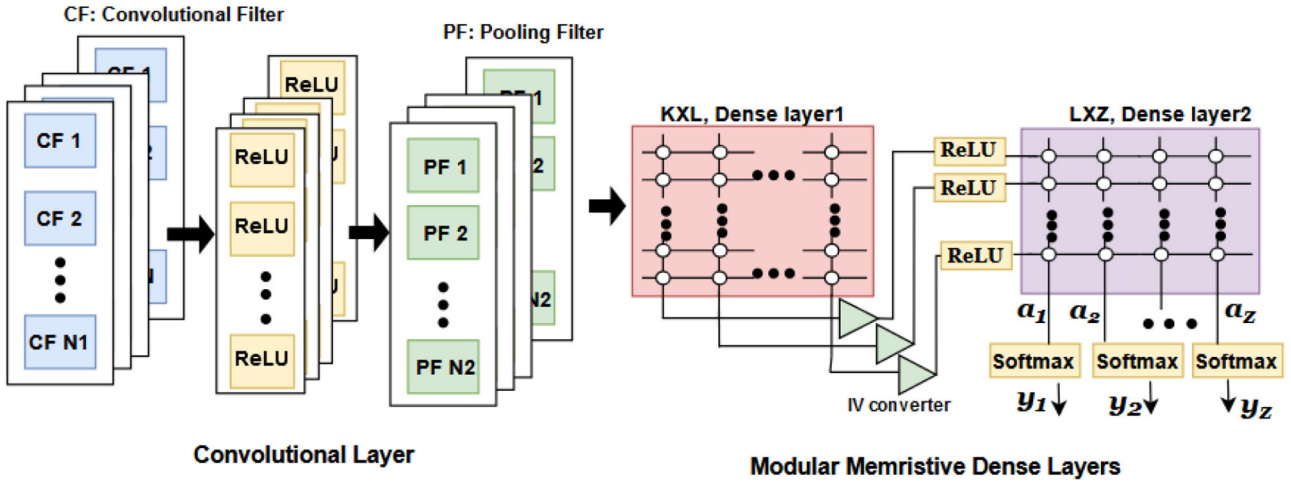


FIGURE 5
CNN implementation using memristor crossbar arrays (R et al., 2022).

implementation of CNN consisting of a convolution layer, mean pooling layer, and dense layers. The convolution filters are realized as memristive crossbars. The conductance of memristive devices is the trained weights of the convolutional filter (CF). The number of memristors in each layer is determined by the required feature maps. The features are then fed to the pooling layer circuit. The pooling layer reduces the dimensionality by performing mean-pooling operation (R et al., 2022). The output of the mean-pool operation is flattened and is connected to dense layers for classification. The current-to-voltage (IV) converter block is used to convert currents to corresponding voltages. The activation functions used are ReLU (rectified linear unit) and softmax.

2.3.3. Cellular neural network (CeNN)

The CeNN is developed by Chua and Yang by mimicking the features of neural networks and cellular automata finds applications in the area of image processing (Chua and Yang, 1988a,b). The CeNN network in Figure 6 consists of $I \times J$ cells. Each cell is connected only to its neighboring cells. The connections from each cell $C(i,j)$ to its neighbors is defined by cloning templates, $A(i,j;k,l)$ and $B(i,j;k,l)$, for feedback and feedforward connections (Chua and Yang, 1988b; Duan et al., 2015). The input signal U is connected to $C(i,j)$ through the feedforward weights $B(i,j;k,l)$. The output of the cell $y_{k,l}$ is fed to $C(i,j)$ through the feedback weights $A(i,j;k,l)$. The state equation can be mathematically expressed as Chua and Yang (1988b).

$$\frac{dx_{i,j}}{dt} = -x_{i,j} + \sum_{c_{k,l}} A(i,j;k,l)y_{k,l} + \sum_{c_{k,l}} B(i,j;k,l)u_{k,l} + I_b, \quad (1)$$

where I_b is the bias current, $x_{i,j}$ is the cell state, and $y_{i,j}$ is the output, respectively. There are various memristive implementations of CeNN in the literature (Duan et al., 2015; Hu et al., 2016). In Figure 6, the feedback and feedforward connections in the CeNN network are implemented using memristor crossbar arrays.

2.3.4. Recurrent neural network (RNN)

The recurrent neural networks-based methods demonstrated outstanding ability in prediction tasks using time-series data by combining large dynamical memory and adaptable computational capabilities. Long short-term memory (LSTM), the special configuration of RNN, is aimed at overcoming the vanishing gradient problems in conventional RNN (Adam et al., 2018). The memristive hardware implementation is presented in Figure 7A (Adam et al., 2018). The input data to the network is the concatenation of input data x_t, data from previous cell h_{t-1} and b_t. The input is multiplied by a weight matrix which is the programmed conductance value of the memristor crossbar array. The crossbar outputs are the input to the activation functions (either sigmoid or hyperbolic tangent) to get the gate values. f_t is the output value of forget gate, $i(t)$ is the output of input/update gate, $o(t)$ denotes the output from the output gate, and $c(t)$ denotes the cell state.

The calculation time in LSTM is very heavy and time-consuming. Echo state network (ESN), a reservoir computing architecture, has emerged as an alternative to the gradient descent training method for RNN (Yu et al., 2022). ESN consists of an input layer where the inputs are associated with a weight matrix w_{in}, followed by a recurrent and sparsely connected reservoir using weight matrix w_{res} and finally, a readout layer associated with a weight matrix w_{out}. The memristive architecture of the ESN reservoir layer is shown in Figure 7B. In ESN, the output readout layer is only trained, and the input and reservoir weight matrices are randomly generated and fixed throughout. The input weights are sampled from a uniform distribution $u(-a,a)$, using a scaling factor a and not trained. The weights of the reservoir are sampled from $u(-1,1)$. Hence, the ESNs are conceptually simple and practically easy to implement.

2.3.5. Spiking neural network (SNN)

The main advantage of SNN hardware implementation is reduced power dissipation in comparison with the pulse-based

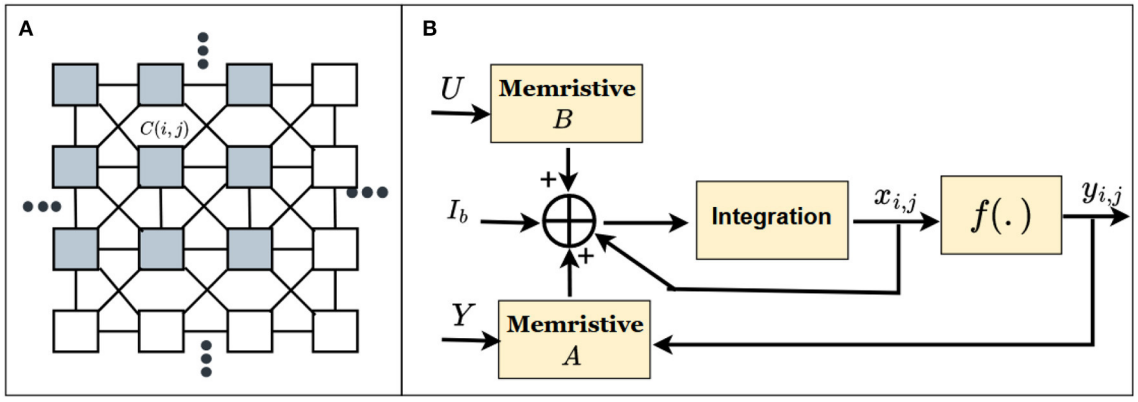


FIGURE 6
(A) Structure of CeNN and **(B)** CeNN implementation using memristor crossbar array (Hu et al., 2016).

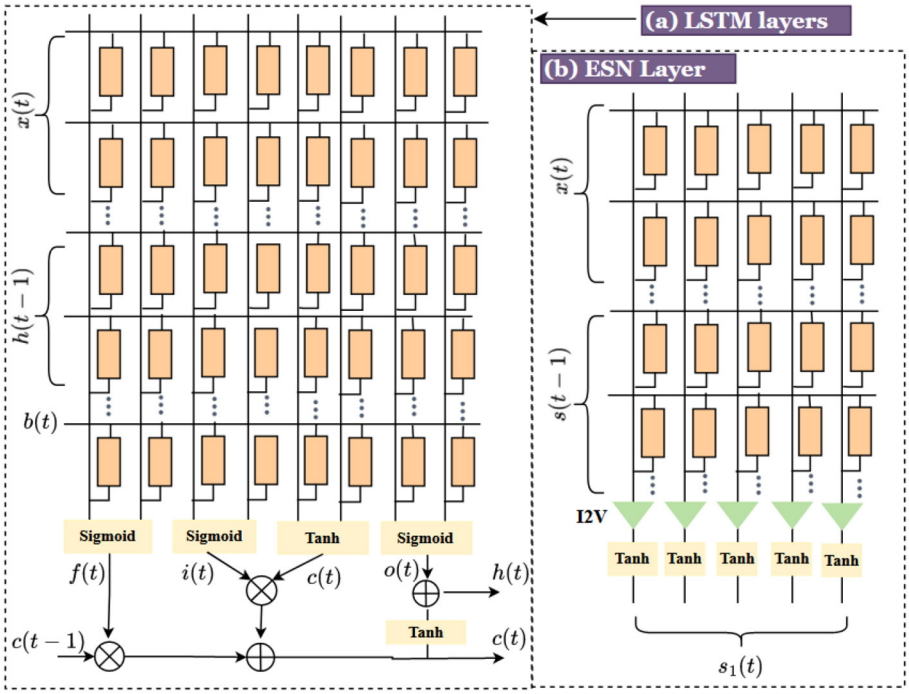


FIGURE 7
(a) Memristive crossbar LSTM architecture (Adam et al., 2018), **(b)** ESN architecture.

systems. The data signals are transmitted as spikes in SNN. The SNN is based on the emulation of brain processing using particular spike events represented by spike-timing-dependent plasticity (STDP). STDP is based on presynaptic and postsynaptic impulses. The implementation of SNN architectures with STDP using memristive crossbar arrays is presented in Figure 8. The architecture consists of presynaptic and postsynaptic neurons connected through memristor crossbar arrays. Most cases use a winner-takes-all (WTA) approach for implementation (Wu et al., 2015b). Recent studies introduce stochasticity by adding noise to WTA architecture (Bill and Legenstein, 2014; Krestinskaya et al., 2020). Stochasticity introduces the biological concept of the probabilistic behavior of neurons in the brain.

As discussed in the section, the field of neuromorphic computing using memristor crossbar arrays is advancing and the exploration of novel materials and devices for in-memory computing is required to improve efficiency and scalability. The RRAM devices are promising candidates for synapses and neurons in neuromorphic circuits. The analog tunable capability of RRAM devices enables novel computing functions for the realization of neuromorphic computing. The material class for RRAM devices is from magnetic alloys, metal oxides, 2D materials, and organic materials. Existing studies in the literature report

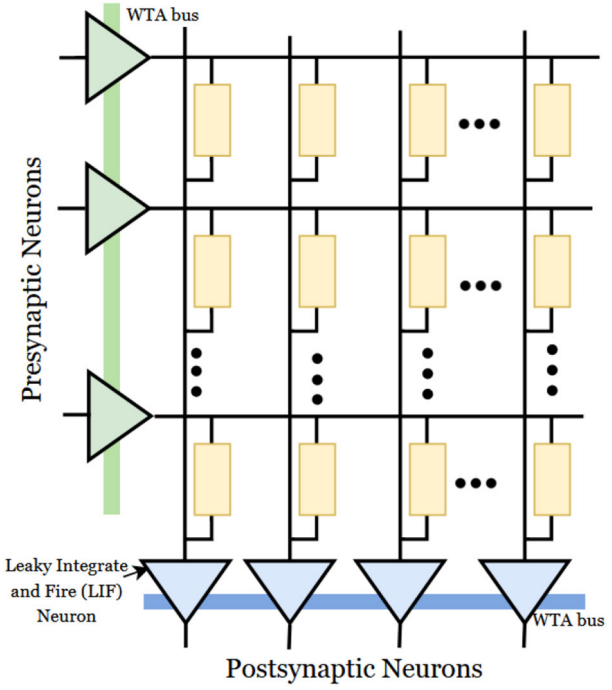


FIGURE 8
Memristive spiking neural network (Wu et al., 2015b).

that 2D material-based RRAM devices have better properties compared to conventional electrode materials which enhances the characteristics of RRAM in such a way to improve its application in neural computing. The coming section reviews the mechanism of the working principle of RRAM and the use of 2D materials for enhancing the properties are discussed in detail.

3. Graphene and 2D materials based RRAM for neural computing

Graphene and other 2D materials have the potential to revolutionize neural computing due to their unique electrical, mechanical, and optical properties. Graphene is a single layer of carbon atoms arranged in a hexagonal lattice, and it is a highly conductive and transparent material. Other 2D materials, such as transition metal dichalcogenides (TMDs) and hexagonal boron nitride (h-BN), also exhibit interesting properties that make them promising for use in neural computing (Zhang et al., 2022).

TMDs have gained significant attention in recent years due to their unique properties and potential applications in various fields, including neural computing. TMDs are a class of materials composed of transition metals (such as molybdenum or tungsten) and chalcogen elements (such as sulfur or selenium). TMDs can be used to create synaptic devices, which are fundamental building blocks of artificial neural networks (Cao et al., 2021). TMDs exhibit excellent electrochemical properties, allowing them to function as efficient and reliable synapses. By controlling the electrical current through TMD-based synaptic devices, the strength of synaptic connections can be modulated, mimicking the synaptic plasticity observed in biological neural networks (Sung et al., 2022). TMDs can also be utilized in the development of neuromorphic computing systems. These systems offer advantages such as parallel processing, low power consumption, and efficient data processing (Lu et al., 2023). TMD-based devices can be integrated into neuromorphic architectures to perform tasks such as pattern recognition, data analysis, and decision-making (Ko et al., 2020).

Another 2D material suitable for neural computing is h-BN (Xie et al., 2022). h-BN is a two-dimensional material, similar to graphene, but with insulating properties. It can serve as a platform for fabricating electronic components, such as transistors, interconnects, resistive memory, and sensors, with potential applications in neural computing. h-BN has been explored as a material for developing neuromorphic devices that can emulate the behavior of biological neurons. The two-dimensional nature of h-BN allows for the integration of multiple components into compact and efficient architectures.

Graphene-based electrodes have been shown to be biocompatible and capable of recording neural signals with high resolution and sensitivity. Additionally, graphene-based transistors have demonstrated fast switching speeds and low power consumption, making them suitable for use in neural signal processing. Another potential application is in the development of neuromorphic computing, which aims to mimic the structure and function of the human brain (Schranghamer et al., 2020). Graphene and other 2D materials can be used to create artificial synapses, which are the connections between neurons that allow them to communicate with each other. The details of fabrication techniques and applications of 2D materials are shown in Table 1.

Overall, graphene and other 2D materials and their combinations hold a great promise for advancing the field of neural computing and could lead to the development of more efficient and powerful neural interfaces and neuromorphic computing systems. Among the 2D materials, the present review focuses mainly on the role of graphene and graphene oxide for RRAM for application in neural computing. There are still many challenges to overcome, such as improving the scalability and reproducibility of these materials and devices, before they can be widely adopted in practical applications (Lin et al., 2016). In this section, the importance and synthesis methods of graphene are discussed in brief and a detailed analysis on the structure and working principles of RRAM is included for a better understanding of the applications of graphene-based RRAM in neural computing.

3.1. Properties of graphene and the different methods for its synthesis

Graphene is a 2D material made up of a single layer of sp^2 hybridized carbon atoms, arranged in a hexagonal lattice. The one atomic layer thickness makes graphene lightweight and flexible. The strong atomic bonding with the nearest carbon atoms provides high mechanical strength to the system, greater than that of steel. Many of these properties vary based on the quality of graphene synthesized. Figure 9 shows the classification of graphene synthesis methods prevalent today. The most popular approaches include those as follows:

TABLE 1 Review on 2D materials for neuromorphic computing applications.

Sl no.	References	2D Material	Fabrication method	Target application	switching voltage
1	Schranghamer et al. (2020)	Graphene	Chemical vapor deposition (CVD)	High precision neuromorphic computing	5.5 V
2	Qian et al. (2016)	h-BN	CVD	Resistive memory	0.72 V
3	Xu et al. (2019a)	MoS_2	MOCVD	Synapse	0.2 V
4	Kumar et al. (2019)	WS_2	RF sputtering	Memristors	1.6 V
5	Krishnaprasad et al. (2019)	MoS_2/ Graphene	CVD	Synapse	1V
6	Liu et al. (2012)	MoS_2/r-Graphene oxide	Liquid exfoliation	Resistive memory	3.5 V

1. Chemical vapor deposition (CVD) - The copper or nickel substrate is heated in a reactor chamber while introducing a hydrocarbon gas (such as methane) to the chamber. These hydrocarbons react with the substrate to form graphene.
2. Epitaxial growth - Substrates similar to crystal structure of graphene [e.g., silicon carbide (SiC) or hexagonal boron nitride (h-BN)] can be used to grow graphene for obtaining epitaxial growth via CVD process.
3. Mechanical exfoliation - The bulk crystal graphite consists of multiple layers of graphene. These layers are peeled off using tape or a sharp object.
4. Electrochemical exfoliation - The electrolyte solution is used to exfoliate graphene from graphite.
5. Solvothermal synthesis - The exfoliation of graphene from a bulk crystal of graphite is done in an autoclave having high pressure and temperature.
6. Thermal reduction of graphene oxide - The repeated reduction of graphene oxide by heating in a hydrogen gas environment can result in graphene formation.

The discovery of graphene was through the mechanical exfoliation (Novoselov et al., 2004) of graphite. Different exfoliation techniques such as mechanical exfoliation, liquid-phase exfoliation (Nicolosi et al., 2013; Farajian et al., 2019), and electrochemical exfoliation (Chen et al., 2019; Ejigu et al., 2019) are used for the synthesis of graphene. In the case of mechanical exfoliation of graphene, highly ordered pyrolytic Graphite (HOPG) is used. The simplest method to exfoliate is by using a scotch tape, and the graphene layer is transferred to the required substrate by sticking the tape on it. However, large-scale synthesis of graphene through this approach is time-consuming, expensive, and not practical. In practice, the use of CVD is more commonly used to obtain high-quality graphene films (Fujita et al., 2017). In the CVD process, the gaseous reactants combine to produce the graphene layer on the substrate surface. Depending on the substrate temperature, the formation process of the sample can be controlled. With the CVD process, relatively high-quality graphene can be produced. The modern CVD techniques can be classified into LPCVD (low-pressure CVD) and UHVCVD (ultrahigh vacuum CVD) (replace with PECVD, hot wall, and cold wall) (Mueller et al., 2014; Sharma et al., 2020).

In CVD, the deposition of a monolayer graphene on the surface of a metal substrate is relatively easy and has a large area scalability potential. Several other growth techniques have been reported for graphene synthesis toward RRAM applications including atomic layer deposition (ALD) (Zhang et al., 2014a), solution deposition techniques (Zhong et al., 2015), plasma-assisted techniques, reduction of graphene oxide (Kurian, 2021), arc discharge (Li et al., 2011). Solution coating methods such as spin coating (Long et al., 2019), dip coating (Kim et al., 2019), and drop coating (Puah et al., 2020) offer attractive platforms for obtaining high-quality graphene films due to their low-cost and large area processability. Laser scribing technology can be used to convert GO to rGO using laser, and RRAM realized using laser scribed reduced graphene oxide was reported in Li et al. (2016). CO_2 laser-induced graphene (LIG) can be used for the fabrication of RRAM, where the graphene is transferred to polydimethylsiloxane (PDMS) from polyimide (PI) (Jung et al., 2021) and SnO_2 is deposited on it. This will provide a flexible RRAM device. Graphene is the thinnest material discovered to date, and properties such as transparency, and flexibility make this suitable for various electronic device applications.

3.2. Features and working mechanisms of RRAM

RRAM is a non-volatile memory that makes use of a material sandwiched between two metal electrodes that have resistive switching properties. The resistance of the RRAM changes depending on the voltage applied across it.

The popular resistive switching material such as titanium dioxide (TiO_2) resistance can be changed by the application of electrical current to the RRAM. The change in resistance to a high or low resistance is mapped to binary states of "0" and "1", thereby allowing digital storage. By applying voltage pulses to the RRAM electrode resistance of the TiO_2 film can be changed. The change in resistance is dependent on the frequency as well as the amplitude of the pulses applied. The RRAM can be read by applying a small voltage pulse and reading the output currents without disturbing the resistance state.

The MIM layer format is used to create the structure of RRAM as shown in Figure 10. The resistive switching mechanism

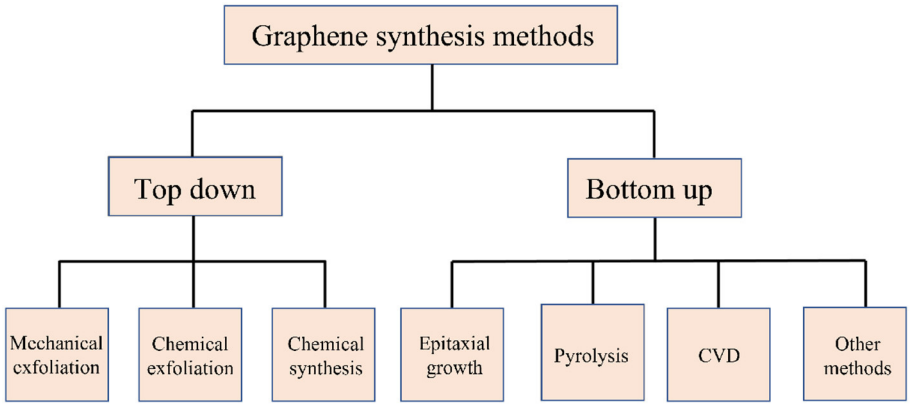


FIGURE 9
Schematic representation of different methods of graphene synthesis.

is enabled with applications of voltage across the two terminals of RRAM to define the resistive state. The HRS is considered the OFF state, and the LRS is regarded as the ON state. The switching mechanism from HRS to LRS happens through the application of external voltage. Some of the materials which exhibit this switching include the oxides of hafnium (Long et al., 2013; Zhao et al., 2014b; Feng et al., 2016), titanium (Yang et al., 2009; Bousoulas et al., 2016), tantalum (Chiu et al., 2012; Prakash et al., 2015; Huang et al., 2016), zinc, nickel (Lee et al., 2008b), manganese (Zhang et al., 2009), magnesium (Chiu et al., 2012), aluminum (Wu et al., 2010), and zirconium (Lin et al., 2007; Wang et al., 2009). In RRAM, the choice of electrode material is critical since it affects the switching property of the system. A small read voltage is applied to understand the system's current state (either ON or OFF) without disturbing the system's state. Since RRAM is a non-volatile memory, it will preserve the state even after removing the external voltage.

RRAM can be classified into two types depending on the voltage polarity to unipolar and bipolar resistive switching. The RRAM is unipolar when the used voltage polarity is the same, and it is called bipolar if reverse voltage polarity is used for switching between the different resistance states (LRS and HRS).

The insulating and conducting mechanisms in the RRAM occur from the breakdown and growth of the filament on the application of an external voltage. Depending on the resistive mechanism, RRAM can be classified into (i) metal ion-based RRAM and (ii) oxygen vacancies filament-based RRAM. In metal ion-based RRAM, the switching mechanism happens by the migration of metal ions in the filament and the oxidation and reduction mechanism. The steps followed in the process of transitioning of conducting state to the insulating state are depicted in Figure 11.

This type of mechanism happens in the case of metal electrodes such as Au, Ni, or Cu at the top-level electrode. The migration of metal ions occurs through the dielectric layer, and the subsequent reduction or oxidation happens at the bottom. This will create a metal filament between the two metal electrodes through the dielectric barrier. This metal filament formation possesses the LRS state, and the disappearance of the same enables the HRS state. In Figure 11, the Ag/a-ZnO/Pt RRAM cells demonstrate the resistive

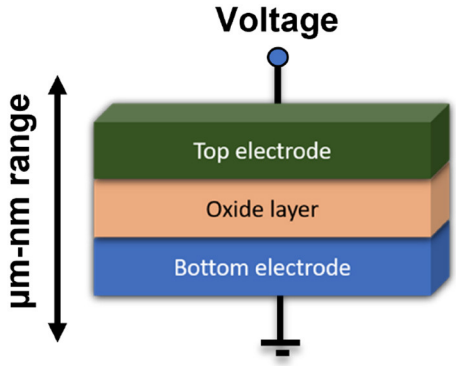


FIGURE 10
Schematic structure of RRAM with metal-insulator-metal layer structure.

switching mechanism. In this case, the Ag electrode is the active element that takes part in the filament formation mechanism, and the Pt electrode is inert. The state of the RRAM devise in the absence of an external electric field is shown in Figure 11A. On applying an external voltage, the oxidation of silver takes place, and it starts to get deposited on the dielectric layer. The bottom electrode, having a negative polarity, will attract these ions, and the ions get deposited on the bottom layer. The formation of metal filament through this process puts the device in the LRS state, as shown in Figures 11B–D. The device can be switched to the HRS state by applying the voltage in the reverse direction, as shown in Figure 11E. We can use graphene as a top/bottom electrode as well as an active insulating layer instead of other materials, as discussed in the following section.

In the case of oxygen vacancies-based RRAM, the resistance-switching mechanism occurs with the creation of oxygen vacancies. The reaction of oxygen ions with the anode material will create the conducting filament. The properties of RRAM will depend on the type of materials present in the top electrode, bottom electrode, and middle layer. Different substitutions of the top and bottom electrodes and middle layers with different materials can enhance

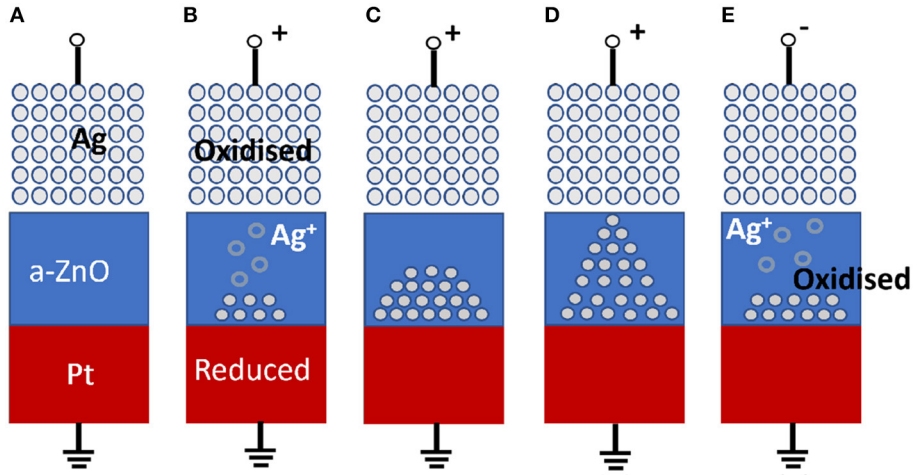


FIGURE 11
Schematic of the switching mechanism of conductive bridge RRAM. **(A)** The pristine state of the RRAM device. **(B, C)** Oxidation of Ag and migration of Ag+ cations toward the cathode and their reduction. **(D)** Accumulating Ag atoms and Pt electrodes leads to the growth of highly conductive filaments. **(E)** Filament dissolution takes place by applying a voltage of opposite polarity (Zahoor et al., 2020).

the properties of RRAM. The use of 2D materials has shown an enhancement in endurance, switching speed, threshold voltage, retention time, etc. The graphene-based RRAM shows promising results in the modification of RRAM toward better performance and for making the system a multilevel cell storage device for the application of MAC computing.

Different parameters will affect the performance of the RRAM device. This study mainly focuses on the variability-averse multi-level cell storage in the graphene-based RRAM system. The RRAM devices have shown a large variability due to the stochastic nature of the switching process.

4. Graphene-based RRAM

Improving the reliability, scalability, and cost-effectiveness of the RRAM device is an essential requirement for practically realizing in-memory and neural computing applications. Graphene-based RRAMs (GRRAM) have different characteristics: low power consumption, higher density, transparency, and homogeneity. GRRAM can be divided into two sub-parts: graphene RRAM and graphene oxide (GO) /reduced graphene oxide (rGO) RRAM. In graphene RRAM, graphene is used as an electrode, whereas in graphene oxide/reduced graphene oxide RRAM, GO or rGO can either be used as a dielectric layer or electrode to enhance the device's performance.

4.1. Graphene as the electrode in RRAM

The main property of RRAM is the resistive switching mechanism which has various difficulties related to the selection of electrodes and the dielectric layer. The high conductivity and high surface area-to-volume ratio of graphene makes it suitable for electrodes. The power consumption is significantly less in graphene-based electrodes in RRAM compared to conventional metal electrodes in RRAM memory devices. Graphene as an electrode offers various advantages over traditional metal electrodes. The greater mechanical scalability, higher conductivity, and ultrathin nature of graphene help to design non-volatile RRAM memory devices. The mechanical properties of graphene, including exceptional strength, flexibility, and elasticity, make it an ideal candidate for use in RRAM devices. These properties enable the fabrication of ultrathin memory cells and provide the potential for integrating RRAM into complex, multi-layered device architectures (Novoselov et al., 2005; Zhang, 2015). The mechanical scalability of graphene allows for the creation of densely packed memory arrays, contributing to higher storage capacities and improved device performance (Papageorgiou et al., 2017). Furthermore, graphene exhibits exceptional electrical conductivity due to its unique electronic band structure (Yung et al., 2013). The switching mechanism in RRAM involves the controlled migration of ions within the memory cell, leading to changes in resistive states. Graphene's high conductivity facilitates efficient charge transport during these switching processes, resulting in fast and reliable switching. The high conductivity of graphene also helps reduce power consumption and enables high-speed read and write operations in RRAM devices.

Lee et al. (2010) report a detailed study on resistive switching characteristics of non-volatile memory devices with nano-materials. 2D material and nanomaterial are the extreme candidates in the nano industry where organic channels and metal electrodes decrease the transmittance value (transmittance decrease of 25%) of the memory devices (Lee et al., 2010). Graphene is used as electrodes, and single-wall carbon nano-tube (SW CNT) is assumed as active layers between metals in non-volatile memory devices. They implemented this memory device with ozone treatment as graphene and oxygen atoms are bonded together. The fabricated memory device revealed that it provides an acceptable transmittance value. Graphene as an electrode provides

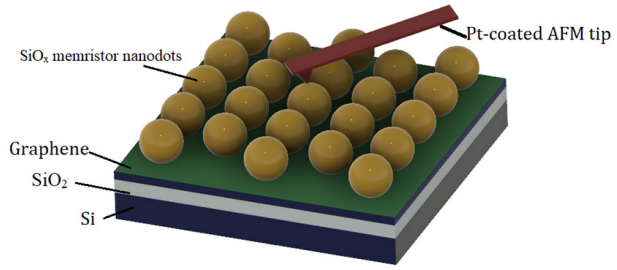


FIGURE 12
Schematic structure of memristor nanostructures on metal and graphene electrodes by a block copolymer self-assembly process.

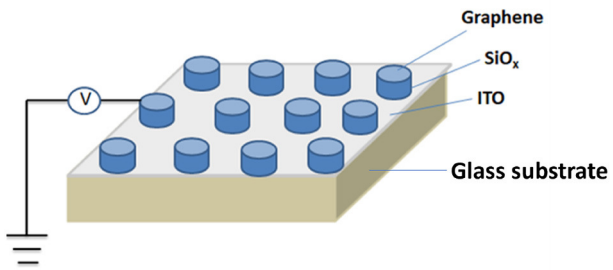


FIGURE 13
Schematic diagram of graphene-SiO_x -indium tin oxide (ITO) device.

a minimum decrease of transmittance of 3.6 %, which is 11.4 % and 25 % in Au and Al. They discovered that the non-volatile memory device with graphene electrodes exhibits better conduction with high mobility of $44 cm^2 V^{-1} s^{-1}$ and a switching speed of 100 ns. The graphene-based memory device performs better than metallic electrodes such as Au, Al, and Ag. The graphene SWCNT memory device improves switching characteristics enhanced by 2×10^2 (Yu et al., 2011b).

Ji et al. (2011) approached a design to integrate an 8×8 crossbar array of organic memory devices with graphene. This multi-layer graphene is an intermediate layer between insulating polyamide (IP) layers. A fabrication process integrates this device with the help of PET (polyethylene terephthalate) substrate. This device offers a high switching ratio current of 10^6 with write-once-read-many (WORM) characteristics. The bending cycle is 10 orders larger (Lee et al., 2010) and exhibits excellent cell-to-cell uniformity. The retention time of the memory device has been controlled in the order of 10^4. Their approach has maintained stable and reliable device characteristics without degrading the current performance. The WORM-type devices store the data permanently without losing any unintended data.

Park et al. (2012) demonstrated a detailed fabrication and characterization of high-density memristor nanodots with platinum and graphene electrodes by a block copolymer self-assembly process. Graphene is used as the bottom electrode, and Pt is a top electrode, where silicon dioxide (SiO_2) is considered an active layer for resistive switches where the memory device has been fabricated with a minimum process cost and less complexity. The fabricated device exhibits a switching ratio of 10^2, an endurance of 80 voltage sweeps, and a unipolar switching mechanism independent of the supply voltage. The formation of a memristor on a graphene electrode is shown in Figure 12.

As transparent electronics devices are in high demand for the electronics industry. Yao et al. (2012) have configured a transparent non-volatile memory device based on SiO_x active layer, indium tin oxide, and graphene as bottom and top electrodes with the glass substrate. Studies on the various device sizes are pursued to enhance the reliability of non-volatile memory. Their study revealed that the conduction filament generated in SiO_2 active layer maintains the constant current as the device size increases or decreases. The switching ratio (10^5) and electrical endurance (300 voltage sweeps) have improved compared to Park et al. (2012). They have also explored how the proposed device with graphene electrode offers better transparency characteristics and low retention time would be beneficial for device application. Figure 13 shows the graphene-SiO_x-indium tin oxide (ITO) device.

A glass platform is a suitable choice for constructing transparent memory devices. The RRAM is constructed with indium tin oxide as the top electrode, alumina as the functional oxide layer, and graphene as the bottom electrode. The non-volatile memory device of this composition has a high transmittance of 82% in the visible region. It is stable and has non-symmetrical bipolar switching properties with low set and reset voltages (less than 1 volt). With its vertical two-terminal configuration, the device has good resistive switching performance and a high on-off ratio (switching ratio) (5×10^3) (Dugu et al., 2018). The figure representing the device structure is shown in Figure 14. Furthermore, transparent materials can be integrated with other optical components to manipulate and direct light within the sensing system. This integration enhances the functionality and performance of optical sensors. Transparent RRAM devices could be integrated with optical sensors, enabling direct interaction between optical input data and neural network processing. This could find applications in fields such as image recognition or computer vision (Zhou et al., 2019; Kalaga et al., 2020).

A graphene-based memristive device (GMD) has been proposed by Qian et al. (2014) and presented a comparative analysis of output performance with a Pt-based memristive device (PtmD). The schematic structure for PtMD and GMDs is shown in Figure 15. The graphene electrode is integrated into TiO_x by the CVD fabrication method to obtain ultra-low switching power and non-linearity. Unlike Yao et al. (2012), they have used graphene as the bottom electrode, whereas Ti/Pt is used as the top electrode. The GMD is fabricated on polyethylene naphthalate (PEN) and offers excellent retention against mechanical bending. They discovered that GMDs have less switching power compared to PtMDs, which helps to protect the device from any thermal damage. Tunable, ultralow-power switching in memristive devices are enabled by a heterogeneous graphene oxide interface. The summary of RRAM devices graphene as top and bottom electrode along with typical characteristics are listed in Table 2.

Similar to Qian et al. (2014), Lee et al. (2015) fabricated a graphene SET electrode-RRAM (GS-RRAM) memory device and

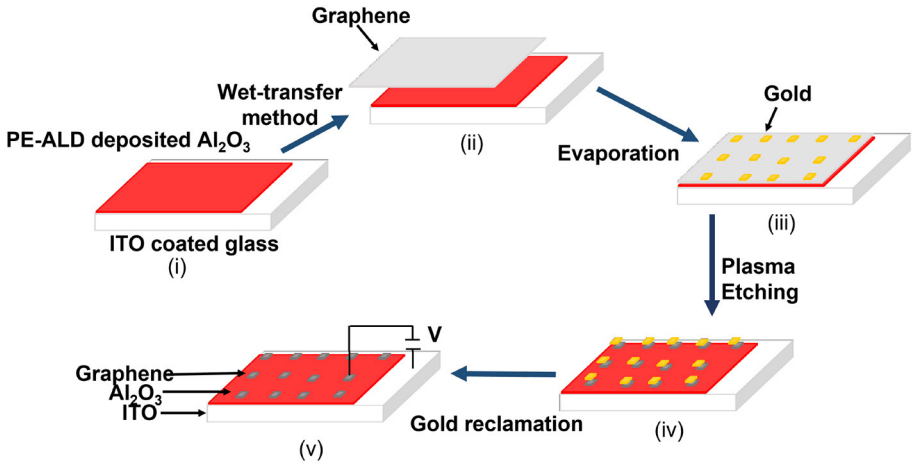


FIGURE 14
Schematic structure of the Ti/ZrO_2/Pt RRAM device (Dugu et al., 2018).

compared it with a Pt-RRAM memory device. In this study, a thin monolayer graphene that serves as a SET electrode is considered to make a thin memory cell structure. The graphene SET electrode helps to store (SET) and restore (RESET) oxygen ions during the programming process. They revealed that the proposed model with a graphene edge electrode has a lower SET compliance current, low RESET current, and low programming voltages, where the Pt-RRAM device cannot deal with low programming voltage or current due to degradation issues of the memory window. The efficient ion-storing capability of graphene helps reduce the power consumption 300 times more in Pt-RRAM. Metal oxide-resistive memory using graphene-edge electrodes (Chakrabarti et al., 2014) explored the performance of RRAM, where graphene is used as top and bottom electrodes. The $TiO_x/Al_2O_3/TiO_2$ dielectric layer is sandwiched between the top and bottom electrodes. The device exhibits forming-free switching characteristics where the device transitions between different resistance states (HRS/LRS) without requiring a separate "forming" process.The forming-free behavior reduces the device complexity and faster the switching process. The proposed device has increases the non-linearity of the current-voltage characteristic with a reduced value of current compliance. When the device exhibits increased non-linearity, the relationship between voltage and current is not linear and more complex and may involve various mechanisms, such as threshold effects, hysteresis, or other non-linear behaviors. This non-linearity can be influenced by factors such as the material properties of the dielectric layer and the electrodes as well as the specific design and operating conditions of the device. A stable retention time of 10^4s, a switching ratio of 10^4, and a greater endurance value ($>$ 200 cycles) have been obtained for the graphene-insulator-graphene (G-I-G) based RRAM configuration. Sohn *et al.* reported a graphene-based 3D RRAM structure where the oxygen ions originating from HfO_x migrate toward the graphene layer, where they aggregate to create a conductive filament (Sohn et al., 2015). This filamentary layer exhibits exceptional thinness, primarily attributed to the atomic-thick nature of graphene. This aligns with the switching mechanism observed in HfO_2 RRAM devices

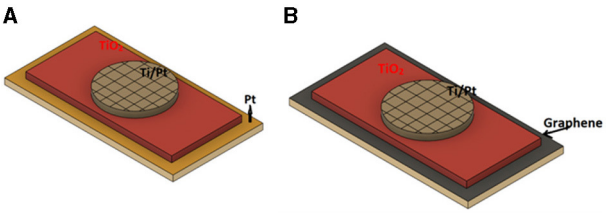


FIGURE 15
Schematic structure for **(A)** platinum-based memristors devices (PtMDs) and **(B)** graphene-based memristors devices (GMDs).

utilizing a top electrode composed of TiN in conjunction with a passive bottom electrode (Yu et al., 2012).

4.2. Graphene as the middle layer in RRAM

Other than electrodes, graphene can also be used as a middle layer in GRRAM for optimizing the switching properties. The incorporation of graphene in the middle layer helps the filament growth by generating a local internal field and acts as a trapping site in the RRAM. The graphene middle layer is usually used for multilevel switching. It is reported that graphene flakes when used as a middle layer help trap charge and act as a storage medium.

Doh and Yi (2010) proposed few-layer graphene (FLG) as an active layer in field-effect devices/ferroelectric devices. They studied the effect of the graphene thickness variation to observe the electrical performance. They discovered that the device has bistable resistance characteristics with long retention time. The resistance difference ratio has decreased with the increased value of graphene film thickness. They also demonstrated that power consumption is high due to the high value of operational voltage ($V_G > 30V$). He et al. (2012) proposed nanographene (NG) which acted as an active layer fabricated on a SiO_2 substrate. Various

TABLE 2 Graphene as the top and bottom electrode.

References	Bottom electrode	Top electrode	Active layer	Substrate	Switching ratio	Endurance	Retention ratio
Yu et al. (2011b)	Graphene	Graphene	SWCNT	PET	10^3	-	10^2
Ji et al. (2011)	Al	Graphene	polyimide and 6-phenyl-C61 butyric acid methyl ester (PI:PCBM)	PET	10^6	-	10^4
Park et al. (2012)	Graphene	Pt	SiO_2	Si	10^2	80	-
Yao et al. (2012)	ITO	Graphene	SiO_x	Glass	10^5	300	-
Dugu et al. (2018)	ITO	Graphene	Alumina	Glass	$> 10^3$	-	-
Chakrabarti et al. (2014)	Graphene	Graphene	TiO-x/Al_2O_3/TiO_2	-	10^4	> 200	10^4
Ji et al. (2011)	Graphene	Graphene	SWCNT	PET	10^3	10^2	10^3
Ji et al. (2011)	Graphene	Graphene	ZnO	Si	10^3	50	-
Chakrabarti et al. (2014)	Graphene	Graphene	TiO-x/Al_2O_3/TiO_2	-	10^4	10^2	10^4
Yao et al. (2012)	Graphene	Graphene	SiO_x	Plastic	10^6	10^2	
Park et al. (2012)	Graphene	SiO_x	Pt	Si	10^2	80	10^4
Ying-Chih Lai et al. (2013)	Graphene	Al	PMMA:P3BT	PET	10^5	10^7	10^4

10.3389/fnins.2023.1253075

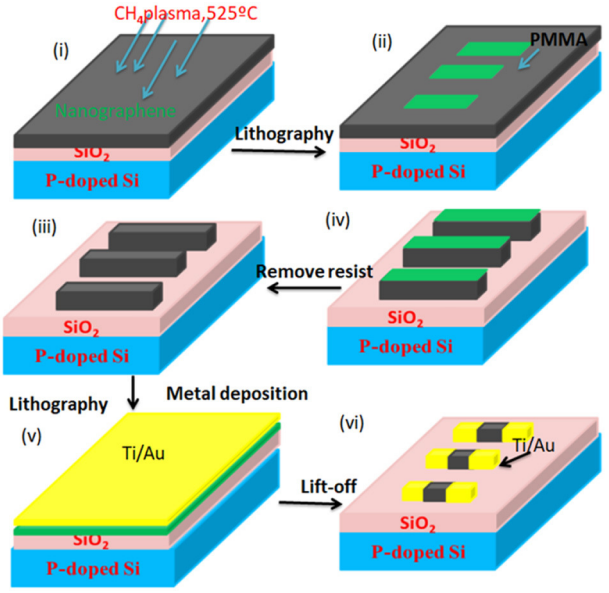


FIGURE 16
Fabrication flowchart for nanographene (NG).

multi-level switching mechanisms have been observed, such as unipolar, bipolar, and non-polar characteristics. Nanographene as an active layer in RRAM has several advantages, such as tunable conductivity and an easy fabrication process, unlike other materials. This research has shown a better endurance value of 10^4 cycles, a faster-switching speed of 500 ns, and a longer retention time of 10^5 cycles. The fabrication flow chart is shown in Figure 16.

Shindome et al. (2013) experimented with single and multi-layer graphene nanoribbon RRAM device characteristics. The drain current performance has been obtained for changing metal electrodes. They revealed that drain current is more for multi-layer graphene RRAM devices than single-layer graphene RRAM. The research also exhibits lower switching energy with a decreased value of channel width, which increases the packing density of the device. Graphene nanoribbon RRAM can possibly scale down to 30nm. Shin et al. (2010) proposed the charging and discharging effect (CDE) to study the bistable switching effects in graphene devices. They also demonstrate bandgap engineering to improve the switching ratio of the device. Two different charge carriers, p-type and n-type, have been considered for this study. The proposed study revealed that the current hysteresis of p-type graphene is inverted into n-type graphene, which increases the stability of the device. The summary of RRAM devices with graphene as an active layer along with typical characteristics are listed in Table 3.

5. Graphene oxide (GO)/reduced graphene oxide (rGO) RRAM

Graphene as a two-dimensional crystal has received more attention from researchers in the semiconductor industry due to its ultrahigh mobility, high thermal conductivity, and transparency characteristics. Graphene oxide is a layered structure consisting of a monolayer of graphene bound to oxygen in carboxyl, hydroxyl, or epoxy groups. Having a high energy band-gap of graphene oxide is possible to reduce the energy band-gap by removing the C-O bonds and offers high solubility. Graphene oxide and reduced graphene oxide are the two important carbon materials mainly used in bioelectrochemical systems (BESs). Graphene oxide offers a large hydrophilic surface area with oxygen-containing functional groups, facilitating microbial attachment and tailored electrochemical reactions on its electrode surface (Singh et al., 2018). On the other hand, rGO, obtained from GO through reduction processes, provides enhanced electrical conductivity, improved biocompatibility, and potential catalytic activity, making it an ideal candidate for efficient electron transfer and biofilm formation in BES systems (Wu et al., 2019). Graphene oxide can be deposited on any substrate due to its flexible nature. Nowadays, GO is a good insulating and semiconductor material compared to other materials and is highly used for RRAM devices. Graphene oxide-based RRAM devices have various pros compared to other materials. The RRAM device with GO can be scaled down in nano-regime and increases the packing density due to the easy fabrication process. Hu *et al.* studied graphene oxide (GO) based RRAM device flexible non-volatile memory. For the purpose of this study, aluminum (Al) has been chosen as the top and bottom electrodes, while GO functions as the active layer. When a negative voltage is applied, it induces an electric field that prompts the migration of oxygen ions within the GO layer. This migration leads to the formation of localized conductive filaments (CFs), consequently causing the device to switch to a LRS. Notably, at the LRS, ohmic conduction is not observed due to the transformation of the GO film into a sp3-bonded state in the absence of CFs (Jeong et al., 2010). During the forming process, a positive voltage bias applied to the Al layer initiates the creation of a highly resistive region in proximity to the tunneling electrode (TE). In the presence of an external electric field, oxygen ions present in the dielectric layer migrate toward the electrode. This migration fosters the continuous development of an sp^3 hybridization layer between the Al electrode and the GO layers that have undergone structural modifications, leading to the high-resistance state (HRS). Subsequently, when a negative voltage bias is applied to the TE Al layer, the reverse diffusion of oxygen ions occurs, resulting in the formation of CFs that lead to the low-resistance state (LRS) near the contact interface, driven by the influence of a negative electric field (Panin et al., 2011). In 2009, He et al. (2009) first explored the RRAM device with graphene oxide (GO) thin films, which are processed by the vacuum filtration method. They found that the device has a low switching voltage and offers a low switching ratio, which is improved later by many researchers (Kim et al., 2011; Yi et al., 2014). Jeong et al. (2010) fabricated a GO-based RRAM device prepared by the spin casting method at room temperature and found to be more reliable and flexible. This study has increased the retention and endurance of the device, which would be helpful for memory applications.

Graphene oxide (GO) can be used for non-volatile and bistable memory devices for its high optical transparency and flexibility. Vasu et al. (2011) studied the unipolar switching effect

TABLE 3 Graphene as an active layer.

Reference	Bottom electrode	Top electrode	Active layer	Substrate	Switch ing ratio	Endurance	Retention time
He et al. (2012)	Ti/Tu	SiO_2	NanoGraphene	p-doped Si	-	10^4	10^5
Shin et al. (2010)	Cr/Au	Al	Graphene	SiO_2	-	10^2	-
Shindome et al. (2013)	Ti/Au	Ti/Au	Graphene	SiO_2	10^3	10^4	10^5
Wu et al. (2012)	ITO	ITO	Graphene	Glass	10^6	-	10^4
He et al. (2013)	Ti/Au	Ti/Au	Graphene	Si/ SiO_2	10^5	-	-
Shindome et al. (2013)	Ti/Cr/Au	Ti/Cr/Au	Graphene nanoribbon	Si/ SiO_2	10^6	10^2	10^3

on reduced graphene oxide (rGO) with the glass substrate to obtain a high switching ratio and switching speed. The obtained results exhibit a switching ratio of 10^5 and switching speed of 10 μs.

Rani et al. (2012) implemented a cost-effective non-volatile memory behavior in rGO memory devices for extracting better endurance and retention time. It is found that the rGO memory device exhibited an endurance value of 10^2 and a retention time of 10^5. Ho et al. (2014) demonstrated a comparative analysis between rGO and GO RRAM devices for impedance spectroscopy and current-voltage analysis. The impedance spectroscopy and current-voltage analysis have been studied to determine the possible physical mechanism for resistive switching behavior. It is observed that switching behavior can be noticed in rGO-based RRAM devices due to its oxidation and reduction at the top electrode. The obtained results for rGO were better with the retention time of 10^6s. However, the rGO memory device provides a large value of the operating voltage of 4V, which increases the power consumption.

Pradhan et al. (2016) proposed a non-volatile rGO-based RRAM memory device to reduce the threshold voltage, which solves the power losses problem of the device more than Ho et al. (2014). Pradhan et al. (2016) proposed an rGO RRAM device which exhibits a threshold value of less than 1V where 4V was achieved by Ho et al. (2014). They also checked the variability of device size, film thickness, and scan votlage.

Kim et al. (2014) demonstrated a transparent memory cell, where reduced graphene is placed between two ITO electrodes to observe the multi-level resistive switching purpose. This memory device offers 80% optical transmittance where the amplitude of applied pulse voltage was varied from 2 to 7V.

Lin et al. (2015) developed a ZnO RRAM device with a capping rGO layer to study the resistive switching behavior. They concluded that introducing the rGO layer increases the stability of the ZnO memory device with a switching ratio of 10^5. The rGO layer acts as an oxygen reservoir in the ZnO memory device where ions are transit easily. On the other hand, oxygen vacancies of the rGO layer oppose reacting with Al electrodes. They also mentioned that ZnO RRAM device offers a great value of endurance of 10^8. The summary of RRAM devices with graphene oxide and reduced graphene oxide as an active layer is listed in Table 4.

6. Comparison of the properties of graphene-based materials with other 2D materials

In sections 4 and 5, the details of graphene, graphene oxide, and reduced graphene oxide base RRAM and its characteristics are discussed. There are other 2D materials such as transition metal dichalcogenides (TMDs) (molybdenum disulfide (MoS_2) and tungsten diselenide (WSe_2) etc.), which offer a diverse range of electronic properties as discussed in section 3. TMDS based RRAM is an emerging technology in the field of non-volatile memory and nanoelectronics (Zhu et al., 2019). In TMD-based RRAM, a thin layer of TMD material is used as the switching medium between two electrodes. The resistance of this TMD layer can be altered by applying an electric field, which changes the oxidation state or defects in the TMD material (Zhang et al., 2018; Jian et al., 2022). However, there are also challenges to overcome, such as ensuring stable and reliable switching behavior, understanding the underlying mechanisms that control resistance switching, and developing scalable manufacturing processes (Zhang et al., 2018). Table 5 presents the comparative study of different properties of graphene and TMDs based RRAM. TMDs can form stable heterostructures with graphene, combining the strengths of both materials for various functionalities.

Metal oxides such as hafnium oxide (HfO_2) and titanium dioxide (TiO_2) provide unique electronic properties suitable for different device applications (Meyer et al., 2012). Along with memory, they are used in optoelectronic devices, catalysis, and sensing applications. Metal oxides exhibit resistive switching behavior, which makes them suitable for RRAM applications, where the metal oxide layer acts as the switching medium (Sawa, 2008). When a voltage is applied across the electrodes, localized changes in the metal oxide's resistance state occur due to various mechanisms, such as the formation and dissolution of conductive filaments or changes in oxygen vacancy concentration (Kumar et al., 2017). One advantage of metal oxide-based RRAM is the potential for high memory cell density, HfO_2 based systems provide multilevel cell storage capabilities (Qi

TABLE 4 Graphene oxide and reduced graphene oxide as an active layer.

Reference	Bottom electrode	Top electrode	Active layer	Substrate	Switching ratio	Endurance	Retention time
Wu et al. (2014)	Pt	Cu	GO	$Ti/SiO_2/Si$	20	10^2	10^4
Hong et al. (2010)	ITO	Al	GO	Glass	10^3	10^2	10^9
Jeong et al. (2010)	Al	Al	GO	PET	10^2	10^2	10^5
Wang et al. (2012a)	ITO	Al	GO	Glass		10^3	10^2
Hu et al. (2012a)	Pt	Pt	GO	SiO_2 /Si	10^4	10^2	10^5
Liu et al. (2013)	GO	GO	GO	PET	10^2	10^3	10^3
Wang et al. (2012b)	Pt	Al	GO	Si	10^4	10^2	10^3
Venugopal and Kim (2012)	Ag	Ag	GO	SiO_2	10	-	10^3
Wang et al. (2012a)	ITO	Al	GO	PET	10^2	10^2	10^4
Pradhan et al. (2016)	Al	Al	GO	Glass	10^2	10^2	10^4
Banerjee et al. (2015)	ITO	Au	GO	Glass	10	10^2	-
Wu et al. (2015a)	ITO	ITO	GO	PES	10	-	10^5
Nagareddy et al. (2017)	Ti/Pt	Ti/Pt	GO	Si/ SiO_2	10^3	10^4	10^5
Kim et al. (2018)	Pt	Pt	rGO	Si/SiO_2	10^5	-	-
Saini et al. (2018)	ITO	Al/Au	GO	Glass	10^5	-	–
Han et al. (2014)	Ag	Au	rGO	PET	10^4	10^2	10^5
Kim et al. (2014)	ITO	ITO	rGO	Glass	10^3	10^5	10^7

et al., 2018; Milo et al., 2019). Achieving stable and repeatable resistive switching behavior is crucial for reliable memory operation. Uniformity of switching characteristics across large arrays of memory cells is also important for commercial viability (Guan et al., 2012). Table 6 presents the comparative study of different properties of graphene and metal oxide-based RRAM devices.

7. RRAM for multi-level cell storage

Multilevel cell storage in RRAM helps to increase the storage density of the memory cell without reducing its size of. In the normal method, the cell size needs to be reduced to increase the density, which requires complex patterning techniques. In the case of multilevel cell storage, the number of bits stored per cell can be increased to n (any integer above 2), increasing the density to n times with 2^n number of available states in the cell. Among the different memory devices such as Spin Transfer Torque RAM (STTRAM) and phase change memory. RRAM shows excellent scalability beyond the 10 nm technology node. The resistive switching mechanism in RRAM helps to attain different intermediate levels by varying the programming current. The size of the conducting filament in an RRAM device depends directly on the applied current. Thus, by adjusting the value of the current, different resistance states can be attained in the system.

The multilevel cell storage can be attained via different methods such as (i) varying compliance current, (ii) adjusting reset voltage, and (iii) changing the pulse width of program/erase operation (Prakash and Hwang, 2016). The most common method among these is the controlling of compliance current to obtain multi-level cell storage. The effect of compliance current on the switching mechanism of the $Ti/ZrO_2/Pt$ is studied by Lei et al. (2014), and the device structure is as shown in Figure 17. In the $Ti/ZrO_2/Pt$ device architecture, the multilevel cell storage is achieved by controlling the magnitude of the compliance current. The observed multilevel cell storage is explained using the voltage divider rule in a series circuit model. By varying the compliance current, the number of traps in the device is controlled; hence, the conductance is varied. A low voltage four-level cell storage is attained in Ta_2O_5/TiO_2 system by controlling the R_L and R_S state of the device (Terai et al., 2010). They found that multilevel cell storage can be achieved by varying the reset voltage as well. In this study, Ru et al. is used as the top and bottom electrode, and the combination of Ta_2O_5/TiO_2 is used as the middle layer. This device achieved a 2-bit/cell storage by multi R_H level operation. In another study of the HfO_2-based RRAM system, the multilevel cell storage is achieved by controlling either I_{set} or V_{stop} (Lee et al., 2008a).

In order to obtain the stable states in the multilevel cell storage system, it is important to distinguish the reference states from one another. The factors affecting the stability of resistance states are cycle-to-cycle variability, device-to-device variability,

TABLE 5 Comparative analysis of graphene-based RRAM and 2D TMDC materials-based RRAM devices.

Device name	Reference	Bottom electrode	Top electrode	Active layer material	Substrate	Switching speed	Endurance	Retention time
Graphene oxide based RRAM	Wu et al. (2014)	Pt	Cu	GO	Ti/SiO_2 /Si	20	10^2	10^4
	Liu et al. (2013)	ITO	GO	GO	PET	10^2	10^3	10^3
	Nagareddy et al. (2017)	Ti/Pt	Ti/Pt	GO	Si/SiO_2	10^3	10^4	10^5
	Wang et al. (2012b)	Pt	Al	GO	Si	10^2	10^2	10^3
	Sun et al. (2015)	FTO	Ag	MoS_2	Glass	10^3	-	10^2
2D TMDs based RRAM (MoS_2, WS_2, $MoSe_2$ based)	Zhou et al. (2017)	ITO	Ag	MoS_2	Glass	10^4	10^2	10^3
	Das et al. (2019)	ITO	Al	MoS_2	Glass	10^2	10^4	10^7
	Kumar et al. (2018)	Ni-Mn-In	Cu	MoS_2	Si	10^2	10^2	10^3
	Rehman et al. (2017)	Ag	Ag	Ws_2	PET	10^3	10^2	10^5
	Zhou et al. (2016)	Ag	Ag	MoS_2	SiO_2	10^2	10^2	10^3

TABLE 6 Comparative analysis of graphene-based RRAM and metal-based RRAM devices.

Device name	Reference	Bottom electrode	Top electrode	Active layer material	Switching speed	Endurance	Retention time
Graphene Oxide based RRAM	Wang et al. (2012b)	Pt	Al	GO	10^4	10^2	10^3
	Pradhan et al. (2016)	Al	Al	GO	10^2	10^2	10^3
	Wang et al. (2012a)	ITO	Al	GO	10^2	10^2	10^4
Metal oxide-based RRAM	Park et al. (2012)	Graphene	Pt	SiOx	10^2	80	10^3
	Yao et al. (2012)	ITO	Graphene	SiO_x	10^5	10^2	10^5
	Tsigkourakos et al. (2017)	TiN/Ti	Au	TiO_{2-x}	-	>50 cycles	10^5
	Wu et al. (2018)	Pd	TiN	HfO_x /Ag/NPs	-	-	10^4
	Chen et al. (2017)	Al	Al	HfO_x	10^4	-	-

operation temperature, random telegraph noise, and interstate switching variability. The study of the retention characteristics and endurance of the device will help to understand the reliability of the multiple resistance levels. It is observed that the retention time for the low resistance state highly depends on the operating current of the device (Ninomiya et al., 2013). With the incorporation of graphene, it is expected to obtain multiple states in the RRAM system. The property of this multi-level cell storage will enable the graphene-based systems to act as a synapse for neuromorphic computing and many other applications.

8. Commercially available RRAM models and its fabrication

For several years, researchers have demonstrated the potential of memristive devices in laboratory experiments. As a result, there

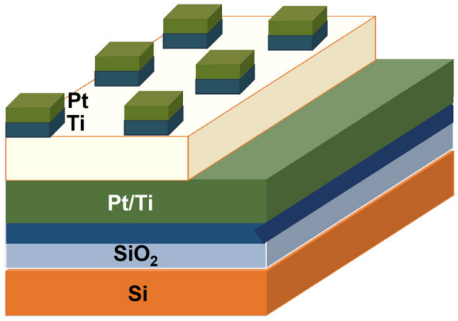


FIGURE 17
Schematic structure of the Ti/ZrO_2/Pt RRAM device (Lei et al., 2014).

have been successful demonstrations of these devices in commercial applications, with RRAM devices being particularly noteworthy in solid-state drives (SSDs) and Internet of Things (IoT) devices. Li et al. (2017) proposed a memory-centric computing approach based on RRAM that leverages on-chip non-volatile memories to perform local information processing in a highly energy-efficient manner. Three in-memory operation schemes using 3D RRAM has been developed and experimented to ensure their effectiveness and reliability, allowing for enhanced local information processing that is highly efficient and optimized for memory-centric computing systems. Wang et al. (2018) demonstrated the integration of 1-transistor/1-resistor (1T1R) memory cells using monolayer MoS_2 transistors and few-layer hBN RRAMs, creating a two-level stacked 3D monolithic structure. The fabrication process was conducted at temperatures below 150 °C. It is observed that this configuration exhibits forming-free (at $< 1V$) gradual set and reset, where the filament formation process in RRAM is not required for achieving the resistance states and which is particularly advantageous for linear weight updating in neuromorphic computing. However, some renowned company has developed various kind of RRAM devices. Adesto Technologies has recently launched a new chip family called Moneta, which utilizes CBRAM (Conductive Bridging Random Access Memory) technology. The Moneta family offers ultra-low power memory solutions that are designed to significantly reduce the overall energy consumption of connected devices. The chips demonstrate read and write operations at 50-100 times lower power compared to competitive solutions. The company has already begun shipping samples of the Moneta family in four different densities, including 32 Kbit, 64 Kbit, 128 Kbit, and 256 Kbit. Fujitsu recently developed RRAM product which offers 1.5 times higher memory density compared to the existing 8 Mbit RRAM. Other renowned foundries such as Intel, Panasonic, and Samsung have been developing RRAM technology. These companies have been investing heavily in RRAM research and development to improve the performance, reliability, and scalability of this promising memory technology.

9. Graphene-based RRAM applications

The researchers are investigating using graphene or graphene oxide (GO) as electrodes or switching material of RRAM targeting in-memory computing for neuromorphic behavior (Izam et al., 2016; Liu et al., 2018; Yan et al., 2018; Abunahla et al., 2020a). The control of resistance for multiple states by memorizing the previous state enables to mimic of biological synapses in the human brain neural network (Sparvoli and Marma, 2018; Xu et al., 2019b; Schranghamer et al., 2020; Kireev et al., 2022). With the large development in memristive materials, an excessive amount of work is being conducted in 2D materials-based memristors for neuromorphic computing (Abunahla et al., 2020a,b; Alimkhanuly et al., 2021). The graphene crossbar variability can be used to build a unique physical unclonable function (PUF), which can be used for various applications. Table 7 presents the review on graphene/GO RRAM for neuromorphic computing.

9.1. Memory

The characteristic features of RRAM such as simple structure, non-volatile, scalability, low power, and fast operation speed makes it a prominent place for future memory devices. In comparison with other materials, the 2D materials-based RRAM devices offer better transparency and flexibility. The incorporation of graphene will provide more feasible and effective methods to increase the capacity of storage devices. The SET current/voltage, I_{set}/ V_{set}, RESET current/voltage, I_{reset}/ V_{reset}, resistance ratio R_{OFF}/ R_{ON}, programming speed, power, and retention time are the parameters for the evaluation of memory devices. Table 8 shows the list of RRAM architecture in the literature with the evaluation parameters.

Zhao et al. (2014a) experimentally demonstrated that the graphene electrode layer provides high built-in series resistance to exhibit good device-to-device uniformity. This exhibits narrow resistance/voltage variations in both ON and OFF states. The switching characteristics of ITO/Al_2O_3/Graphene RRAM is compared with ITO/Al_2O_3/Pt RRAM devices in Dugu et al. (2018). The results in Dugu et al. (2018) show that graphene shows a low SET/RESET current/voltage in comparison with conventional RRAM electrodes such as Pt. A perceptron model is experimentally in Sparvoli and Marma (2018).

Lu et al. (2022) have developed a two-terminal memristor synapse based on a silicon-argon composite film. In the case of the biological synapse, the weight is varied by the release of neurotransmitters from the preneuron induced by spikes. Thus, similar to that, this memristive synapse varies its conductance by the migration of the ions upon an external electrical signal or stimuli.

9.2. Neural networks

The RRAM crossbar in-memory computing is considered to be a potential solution for implementing power-efficient neural network architectures (Li et al., 2018; Mehonic et al., 2020). The analog/digital feature of RRAM, with the ability to memorize, can be used to build artificial neural networks for neuromorphic computation (Mehonic et al., 2020). Figure 18 shows crossbar architecture using RRAM devices for realizing the

TABLE 7 A review on graphene RRAM for neuromorphic computing.

Sl no.	Reference	Graphene application	No. of conductance states	Target application
1	Abunahla et al. (2020a,b)	Au/ partially redued graphene oxide (prGO)/Au	7	ANN of size 5 × 4 and 4 × 4
2	Alimkhanuly et al. (2021)	electrode of 3D vertical RRAM	64	XNOR
3	Sparvoli and Marma (2018)	RRAM fabrication with doped graphene oxide with silver	2	RRAM bridge synapse
4	Schranghamer et al. (2020)	Graphene field effect transistor	16	RRAM synapse
5	Xu et al. (2019b)	Al_2O_3/graphene quantum dots/Al_2O_3	2	Synapse
6	Kireev et al. (2022)	Bilayer Graphene-based Artificial Synaptic Transistors (BLAST)	100	Synapse transistor

TABLE 8 Different RRAM architectures.

RRAM structure	I_{set}/ V_{set}	I_{reset}/ V_{reset}	R_{OFF}/R_{ON} Ratio	SET/R ESET speed	Power
MLG/Dy_2O_3/ITO (Zhao et al., 2014a) Unipolar	1 μA/0.4V	20 μA /0.2V	$> 10^5$	60 ns	4.4 μW
ITO/Al_2O_3/ Graphene (Dugu et al., 2018) Bipolar	2.1 μA/0.8V	1.55 mA/-0.65V	$\sim 3.5 \times 10^3$	NA	$\sim 1mW$
Al_2O_3/GQD/Al_2O_3 (Xu et al., 2019b)	$< 5nA$/1.2V	$< 5nA$/-1.2V	NA	NA	NA
ITO/GO+0.1 % Ag/Al (Sparvoli and Marma, 2018) Unipolar	$< 4.78mA$/0.8V	2 pA/0.25V	7.5×10^8	$10\mu s$	NA
G/SiO_x/ITO (Yao et al., 2012) Unipolar	2 μA/4.26V	2 mA/10V	10^4	$50ns$	20 mW
Au/prGO/Au (Abunahla et al., 2020a,b) Unipolar	25 mA/3V	10 mA/-6.5V	10	$10s$	NA
TiN/HfO_x/ Graphene (Alimkhanuly et al., 2021) bipolar	$< 1\mu$ A/1.27V	$< 10\mu$A/ -1.37V	$> 10X$	500 ns	NA

MLG, multi-layer graphene; ITO, indium tinoOxide; GQD, graphene quantum dots; NA, not available.

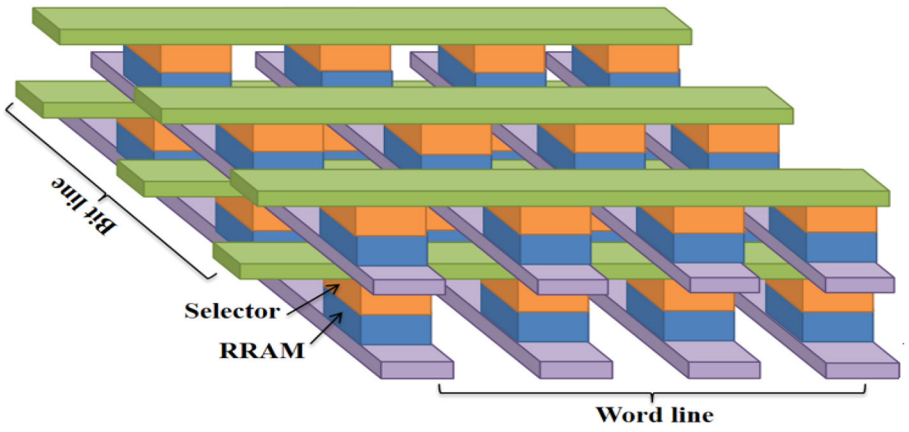


FIGURE 18
3D RRAM crossbar array.

neuromorphic computations. The weights of neural computations are programmed onto the RRAM devices during the write mode. Only a few studies have been reported in the literature using graphene/GO-based RRAM for neuromorphic computing (Abunahla et al., 2020a,b; Alimkhanuly et al., 2021). Both 2D and 3D crossbar architecture with RRAM have been discussed in the literature for neuromorphic computing. HebaAbunahla et al. presented a novel planar analog memristor crossbar with partially reduced graphene oxide (prGO) thin film (Abunahla et al., 2020a,b). In Abunahla et al. (2020a,b), the crossbar array has been fabricated and tested using the Iris dataset with an accuracy of 96.67%. 5 × 4 and 4 × 4 crossbar arrays have been fabricated, which is then used to classify the iris flower based on its petal and sepal length and width into different classes.

Alimkhanuly et al. (2021) demonstrated a 3D vertical RRAM (VRRAM) by replacing the metal-based interconnects with graphene due to the remarkable electronic and thermal conductivities. In Alimkhanuly et al. (2021), the authors fabricated a 416 × 224 × 8 size 3D array system. The recognition performance of the fabricated 3D graphene RRAM (Gr-RRAM) has been tested for the MNIST dataset. The network size is 400 input, 200 hidden, and 10 output neurons. The performance accuracy of Gr-RRAM is compared with platinum RRAM (Pt-RRAM), and the results show that the overall accuracy levels degrade for Pt-RRAM due to high read inaccuracy.

9.3. Logic gates

The logic computing is yet another application of memristor crossbar structure. The XOR operation-focused 3D VRRAM array architecture is demonstrated in Alimkhanuly et al. (2021). The XOR architecture using graphene-based VRRAM arrays have the potential of a highly stackable nature for parallel processing of multiple layers (Alimkhanuly et al., 2021). An XNOR logic-inspired architecture is designed to integrate 1-bit ternary precision synaptic weights into graphene-based VRRAM is presented in Alimkhanuly et al. (2021). However, robustness to device variability by using graphene-based RRAM in logic computing is not yet investigated in the literature and still remains an open problem.

9.4. Cryptography

The memristor crossbar arrays is also applied for cryptography applications (Cai et al., 2022; Yu et al., 2023). An in-memory hyperdimensional encryption using a memristor crossbar array is presented in Cai et al. (2022). The robustness of binary hypervectors against memristor crossbar non-ideality helps to control the impact of noise generated by the memristor crossbar for encryption. A 4D memristive hopfield neural network (MHNN) is proposed in Yu et al. (2023) for image encryption applications. The majority of memory-based cryptographic techniques for hardware security are based on physical unclonable functions (PUFs) (James, 2019). A large number of memory-crossbar-based PUFs have been proposed in the literature, for example, metal-oxide memristor-based or RRAM (Rose and Meade, 2015; Yansong et al., 2015; Uddin et al., 2017; Khan et al., 2021; Kim et al., 2021) etc. The PUF methods use variations in device parameters such as resistance state, switching time, and threshold voltages. These unpredictable probabilistic characteristics of memristor crossbars form the basis for PUF applications. The variations in device parameters and process variations affect the current flow through the device. Any temporal or spatial variations affect all aspects of resistive switching. The variation in PUF characteristics with the properties of graphene has not been explored yet in the literature.

The stochasticity in graphene-RRAM device response has not been extensively studied in the existing literature. The repeatability of fabricated Gr-RRAM devices are experimentally evaluated in the literature. The SET voltage varies for cycle-cycle variations for Gr-RRAM was found to be 6.4% in Alimkhanuly et al. (2021). As discussed in Kim et al. (2021), the SET voltage variations in Gr-RRAM crossbar array can also be used for PUF generation in cryptographic applications. The other device variations such as resistance state, switching time, and threshold voltages have not been considered for analysis with device-to-device and cycle-to-cycle variations. The stochasticity in graphene-RRAM variation for cryptography or PUF characteristics has not been explored yet in the literature and is an open problem.

10. CMOS compatibility

CMOS technology faces various unwanted problems due to the scaling of device attributes. The semiconductor industry is planning to replace the silicon material with graphene material. Since graphene is a conducting material and no energy band gap is present in it, it is very difficult to use graphene for digital device applications due to high-off state leakage and non-saturating drive currents. However, graphene-based devices are more acceptable for low-noise amplifiers and radio-frequency (RF) in analog device applications (Banerjee et al., 2010). Rodriguez et al. (2012) compared the RF behavior between graphene-based field effect transistor (GFET) and Si-based metal oxide field effect transistor (MOSFET). It is observed that the GFET device is more acceptable for the narrow range of drain voltage and drain current compared to Si-MOSFET. Cisneros-Fernández et al. (2019) proposed frequency domain multiplexing of liquid-gate GFET sensor for micro electrocorticogram (ECoG) recording purpose. The proposed work also allows hybrid integration.

Nowadays, graphene with Si CMOS circuits can also be constructed together for making heterogeneous devices. The demonstration of graphene and Si CMOS hybrid circuits has reduced barriers to entry of graphene in electronics. Huang et al. (2014) constructed a low-temperature hybrid integrated circuit where graphene devices and Si-CMOS circuits integrated together. Gilardi et al. (2019) designed relaxation oscillators using a GFET, Si CMOS D latch, and timing RC circuit. It is observed that the introduction of graphene material in the Si-CMOS logic circuit has improved the circuit complexity and also added other device functionality. One of the truly unique electronic properties of graphene not exhibited by conventional

semiconductors is ambipolarity. The ambipolarity of graphene helps to simplify the circuit and provide additional functionality. Graphene's ambipolarity eliminates the need for separate electron and hole transistors, reducing the overall transistor count and circuit complexity (Jabeur et al., 2010). The integration of graphene into Si CMOS logic circuits could offer a feasible approach for both simplification and enhanced functionality. Zhang et al. (2014b) proposed CMOS-compatible all-metal-nitride RRAM based on aluminum nitride (AlN). It is observed that the proposed device provides a lower operation current of 100 A, retention time $3x10^5$, and endurance value of 10^5 Hz. AlN has high thermal stability, allowing it to withstand the elevated temperatures used in CMOS processes. This makes it possible to integrate AlN-based RRAM fabrication steps into standard CMOS processes without causing significant damage to the underlying circuitry (Jackson et al., 2013). AlN can be deposited using various techniques that are already employed in CMOS manufacturing, such as PVD and CVD (Perez-Campos et al., 2015). PVD and CVD methods allow for conformal deposition of thin AlN films over complex three-dimensional structures, including the intricate features found in modern CMOS circuits (Cansizoglu et al., 2015). This conformal deposition capability is crucial for integrating RRAM cells within the existing CMOS architecture. The temperature requirements and chemical interactions during AlN deposition are generally more manageable compared to some graphene synthesis methods. Graphene-based RRAM, on the other hand, could face more integration challenges due to the specialized processes required for graphene synthesis and transfer. Graphene synthesis and transfer techniques involve high-temperature processes and chemical treatments that could affect the performance of the graphene itself (Choi et al., 2022). Achieving high-quality, defect-free graphene layers on a large scale while maintaining CMOS compatibility remains a significant hurdle (Moon and Gaskill, 2011). Yeh and Wong (2015) proposed a cost-competitive One-Transistor-N-RRAM (1TNR) array architecture for advanced CMOS technology where one committed transistor controls the access of one RRAM. It is observed that the 1TNR array architecture provides less leakage current than the cross-point array. Therefore, there is the possibility that graphene-based RRAM memory devices can be considered in CMOS technology soon.

11. Challenges and future scope

Due to its unique and interesting features, graphene has surpassed all other nanomaterials in terms of its use in electronic devices. Additionally, it was shown that graphene's greater mobility, less light absorption, and excellent mechanical qualities enhance the functionality of transparent flexible electronic devices. The difficulty is that the cost of manufacturing graphene will increase the overall price of the device. The transfer of graphene from one substrate to another without causing any damage is a tedious process, which requires the need of sophisticated instruments. Efficient methods need to be implemented to overcome these drawbacks.

The past several years have seen a substantial increase in research into new memory technologies, and numerous prototype RRAM products have been created to show the potential for high-speed and low-power applications. The CMOS compatibility and ability to fabricate in smaller dimensions make the RRAM a suitable candidate for device applications. A high endurance is reported in graphene-based RRAM devices. To date, in a single RRAM device, no technology has reported fast switching, low power, and stable operation simultaneously. In a graphene-based RRAM device, the properties need to be enhanced for better performance of the device.

12. Conclusion

This review article offers an insightful look into the topic of developing graphene-based RRAM devices in terms of neural computing by giving a concise overview of the development of memory architecture, the current trends, and the constraints. The importance of graphene based RRAM, as well as its structure, operation, and classification, have all been highlighted in a thorough discussion. The methodology and a detailed investigation on the MLC capabilities of RRAM have been presented. It is proposed that the graphene-based RRAM can be used for multilevel cell storage. This modified memory device, with 2D material can be used as a synapse. Along with this, the implementation of graphene based RRAM for various important applications such as hardware security and neuromorphic computing have been highlighted.

Author contributions

RTR, RD, and CR: Conducted literature review, prepared a part of the draft copy of the manuscript, and revision of the manuscript. AJ: Contributed to the theoretical framework development, provided critical insights during data analysis and interpretation, manuscript correction and writing, funding acquisition and supervision.

Conflict of interest

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

Publisher's note

References

Abunahla, H., Halawani, Y., Alazzam, A., and Mohammad, B. (2020a). Neuromem: Analog graphene-based resistive memory for artificial neural networks. *Sci. Rep*. 10, 1–11. doi: 10.1038/s41598-020-66413-y

Abunahla, H., Halawani, Y., Mohammad, B., and Alazzam, A. (2020b). "Tunable non-volatile analog resistive memory and its application in AI," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 464–467.

Adam, K., Smagulova, K., and James, A. P. (2018). "Memristive lstm network hardware architecture for time-series predictive modeling problems," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 459–462.

Alimkhanuly, B., Sohn, J., Chang, I. J., and Lee, S. (2021). Graphene-based 3d xnor-vrram with ternary precision for neuromorphic computing. *NPJ 2D Mater. Applicat*. 5, 55. doi: 10.1038/s41699-021-00236-x

Ambriz-Vargas, F., Kolhatkar, G., Broyer, M., Hadj-Youssef, A., Nouar, R., Sarkissian, A., et al. (2017). A complementary metal oxide semiconductor process-compatible ferroelectric tunnel junction. *ACS Appl. Mater. Interfaces*. 9, 13262–13268. doi: 10.1021/acsami.6b16173

Banerjee, I., Harris, P., Salimian, A., and Ray, A. K. (2015). Graphene oxide thin films for resistive memory switches. *IET Circuits Devices Syst*. 9, 428–433. doi: 10.1049/iet-cds.2015.0170

Banerjee, S. K., Register, L. F., Tutuc, E., Basu, D., Kim, S., Reddy, D., et al. (2010). Graphene for cmos and beyond cmos applications. *Proc. IEEE* 98, 2032–2046. doi: 10.1109/JPROC.2010.2064151

Bill, J., and Legenstein, R. (2014). A compound memristive synapse model for statistical learning through STDP in spiking neural networks. *Front. Neurosci*. 8, 120754. doi: 10.3389/fnins.2014.00412

Bousoulas, P., Stathopoulos, S., Tsialoukis, D., and Tsoukalas, D. (2016). Low-power and highly uniform 3-b multilevel switching in forming free TiO2-x-based RRAM with embedded Pt nanocrystals. *IEEE Electron Device Lett*. 37, 874–877. doi: 10.1109/LED.2016.2575065

Boybat, I., Le Gallo, M., Nandakumar, S. R., Moraitis, T., Parnell, T., Tuma, T., et al. (2018). Neuromorphic computing with multi-memristive synapses. *Nat. Commun*. 9, 1–12. doi: 10.1038/s41467-018-04933-y

Brivio, S., Ly, D. R., Vianello, E., and Spiga, S. (2021). Non-linear memristive synaptic dynamics for efficient unsupervised learning in spiking neural networks. *Front. Neurosci*. 15, 580909. doi: 10.3389/fnins.2021.580909

Cai, J., Amirsoleimani, A., and Genov, R. (2022). "Hyperlock: In-memory hyperdimensional encryption in memristor crossbar array," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 960–964.

Cansizoglu, H., Yurukcu, M., Cansizoglu, M. F., and Karabacak, T. (2015). Investigation of physical vapor deposition techniques of conformal shell coating for core/shell structures by Monte Carlo simulations. *Thin Solid Films* 583, 122–128. doi: 10.1016/j.tsf.2015.03.071

Cao, G., Meng, P., Chen, J., Liu, H., Bian, R., Zhu, C., et al. (2021). 2D material based synaptic devices for neuromorphic computing. *Adv. Funct. Mater*. 31, 2005443. doi: 10.1002/adfm.202005443

Chakrabarti, B., Roy, T., and Vogel, E. M. (2014). Nonlinear switching with ultralow reset power in graphene-insulator graphene forming-free resistive memories. *IEEE Electron Device Letters* 35, 750–752. doi: 10.1109/LED.2014.2321328

Chen, D., Wang, F., Li, Y., Wang, W.-W., Huang, T.-X., Li, J.-F., et al. (2019). Programmed electrochemical exfoliation of graphite to high quality graphene. *Chem. Commun*. 55, 3379–3382. doi: 10.1039/C9CC00393B

Chen, Y.-C., Chang, Y.-F., Wu, X., Zhou, F., Guo, M., Lin, C.-Y., et al. (2017). Dynamic conductance characteristics in hfo x-based resistive random access memory. *RSC Adv*. 7, 12984–12989. doi: 10.1039/C7RA00567A

Chithra, R., Aswani, A. R., and James, A. (2022). "Memristive cnn for wafer defect detection," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 3284–3288. Austin, TX: IEEE.

Chiu, F. C., Li, P. W., and Chang, W. Y. (2012). Reliability characteristics and conduction mechanisms in resistive switching memory devices using ZnO thin films. *Nanoscale Res. Lett*. 7:1–9. doi: 10.1186/1556-276X-7-178

Choi, S. H., Yun, S. J., Won, Y. S., Oh, C. S., Kim, S. M., Kim, K. K., et al. (2022). Large-scale synthesis of graphene and other 2D materials towards industrialization. *Nat. Commun*. 13, 1–5. doi: 10.1038/s41467-022-29182-y

Chua, L., and Yang, L. (1988a). Cellular neural networks: applications. *IEEE Trans. Circuits Syst*.35, 1273–1290. doi: 10.1109/31.7601

Chua, L., and Yang, L. (1988b). Cellular neural networks: theory. *IEEE Trans. Circuits Syst*. 35, 1257–1272. doi: 10.1109/31.7600

Cisneros-Fernández, J., Dei, M., Terés, L., and Serra-Graells, F. (2019). "Switch-less frequency-domain multiplexing of GFET sensors and low-power CMOS frontend for 1024-channel μECoG," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)* (Sapporo: IEEE), 1–5.

Das, U., Bhattacharjee, S., Sarkar, P. K., and Roy, A. (2019). A multi-level bipolar memristive device based on visible light sensing mos2 thin film. *Mater. Res. Express* 6, 075037. doi: 10.1088/2053-1591/ab154d

Doh, Y.-J., and Yi, G.-C. (2010). Nonvolatile memory devices based on few-layer graphene films. *Nanotechnology* 21, 105204. doi: 10.1088/0957-4484/21/10/105204

Duan, S., Hu, X., Dong, Z., Wang, L., and Mazumder, P. (2015). Memristor-based cellular nonlinear/neural network: design, analysis, and applications. *IEEE Trans. Neural Netw. Learn. Syst*. 26, 1202–1213. doi: 10.1109/TNNLS.2014.2334701

Dugu, S., Pavunny, S. P., Limbu, T. B., Weiner, B. R., Morell, G., and Katiyar, R. S. (2018). A graphene integrated highly transparent resistive switching memory device. *APL Mater*. 6, 058503. doi: 10.1063/1.5021099

Ejigu, A., Le Fevre, L. W., Fujisawa, K., Terrones, M., Forsyth, A. J., and Dryfe, R. A. W. (2019). Electrochemically exfoliated graphene electrode for high-performance rechargeable chloroaluminate and dual-ion batteries. *ACS Appl. Mater. Interf*. 11, 23261–23270. doi: 10.1021/acsami.9b06528

Farajian, A. A., Mortezaee, R., Osborn, T. H., Pupysheva, O. V., Wang, M., Zhamu, A., et al. (2019). Multiscale molecular thermodynamics of graphene-oxide liquid-phase exfoliation. *Phys. Chem. Chem. Phys*. 21, 1761–1772. doi: 10.1039/C8CP07115B

Feng, W., Shima, H., Ohmori, K., and Akinaga, H. (2016). Investigation of switching mechanism in HfOx-ReRAM under low power and conventional operation modes. *Scientific Rep*. 6, 1–8. doi: 10.1038/srep39510

Fujita, J. I., Hiyama, T., Hirukawa, A., Kondo, T., Nakamura, J., Ito, S. I., et al. (2017). Near room temperature chemical vapor deposition of graphene with diluted methane and molten gallium catalyst. *Scient. Rep*. 7, 1–10. doi: 10.1038/s41598-017-12380-w

Galashev, A. E., and Rakhmanova, O. R. (2014). Mechanical and thermal stability of graphene and graphene-based materials. *Physics-Uspekhi* 57, 970–989. doi: 10.3367/UFNe.0184.201410c.1045

Gilardi, C., Pedrinazzi, P., Patel, K. A., Anzi, L., Luo, B., Booth, T. J., et al. (2019). Graphene-si cmos oscillators. *Nanoscale* 11, 3619–3625. doi: 10.1039/C8NR07862A

Guan, X., Yu, S., and Wong, H. S. (2012). On the switching parameter variation of metal-oxide RRAM - Part I: Physical modeling and simulation methodology. *IEEE Trans. Electron Devices* 59, 1172–1182. doi: 10.1109/TED.2012.2184545

Han, S. T., Zhou, Y., Yang, Q. D., Zhou, L., Huang, L. B., Yan, Y., et al. (2014). Energy-band engineering for tunable memory characteristics through controlled doping of reduced graphene oxide. *ACS Nano* 8, 1923–1931. doi: 10.1021/nn406505t

He, C., Li, J., Wu, X., Chen, P., Zhao, J., Yin, K., et al. (2013). Tunable electroluminescence in planar graphene/SiO2 memristors. *Adv. Mater*. 25, 5593–5598. doi: 10.1002/adma.201302447

He, C., Shi, Z., Zhang, L., Yang, W., Yang, R., Shi, D., et al. (2012). Multilevel resistive switching in planar graphene/sio2 nanogap structures. *ACS Nano* 6, 4214–4221. doi: 10.1021/nn300735s

He, C., Zhuge, F., Zhou, X., Li, M., Zhou, G., Liu, Y., et al. (2009). Nonvolatile resistive switching in graphene oxide thin films. *Appl. Phys. Lett*. 95, 232101. doi: 10.1063/1.3271177

Ho, N. T., Senthilkumar, V., and Kim, Y. S. (2014). Impedance spectroscopy analysis of the switching mechanism of reduced graphene oxide resistive switching memory. *Solid State Electron*. 94, 61–65. doi: 10.1016/j.sse.2014.02.002

Hong, S. K., Kim, J. E., Kim, S. O., and Cho, B. J. (2010). "Non-volatile memory using graphene oxide for flexible electronics," in *2010 10th IEEE Conference on Nanotechnology, NANO 2010*, 604–606.

Hu, B., Quhe, R., Chen, C., Zhuge, F., Zhu, X., Peng, S., et al. (2012a). Electrically controlled electron transfer and resistance switching in reduced graphene oxide noncovalently functionalized with thionine. *J. Mater. Chem*. 22, 16422–16430. doi: 10.1039/c2jm32121a

Hu, X., Chen, G., and Duan, S. (2016). A novel memristive cellular neural network with time-variant templates. *Perspect. Sci*. 7, 126–132. doi: 10.1016/j.pisc.2015.11.021

Hu, X., Duan, S., Wang, L., and Liao, X. (2012b). Memristive crossbar array with applications in image processing. *Sci. China Informat. Sci*. 55, 461–472. doi: 10.1007/s11432-011-4410-9

Huang, H.-M., Wang, Z., Wang, T., Xiao, Y., and Guo, X. (2020). Artificial neural networks based on memristive devices: from device to system. *Adv. Intell. Syst*. 2, 2000149. doi: 10.1002/aisy.202000149

Huang, L., Xu, H., Zhang, Z., Chen, C., Jiang, J., Ma, X., et al. (2014). Graphene/si cmos hybrid hall integrated circuits. *Sci. Rep*. 4, 1–6. doi: 10.1038/srep05548

Huang, Y., Gu, Y., Wu, X., Ge, R., Chang, Y.-F., Wang, X., et al. (2021). Rese2-based rram and circuit-level model for neuromorphic computing. *Front. Nanotechnol*. 3, 782836. doi: 10.3389/fnano.2021.782836

Huang, Y., Shen, Z., Wu, Y., Wang, X., Zhang, S., Shi, X., et al. (2016). Amorphous ZnO based resistive random access memory. *RSC Adv*. 6, 17867–17872. doi: 10.1039/C5RA22728C

Im, I. H., Kim, S. J., and Jang, H. W. (2020). Memristive devices for new computing paradigms. *Adv. Intell. Syst*. 2, 2000105. doi: 10.1002/aisy.202000105

Izam, N. I. B., Aziz, T. N. T. A., Rahman, R. A., Malek, M. F., Herman, S. H., and Zulkifli, Z. (2016). *2016 IEEE Student Conference on Research and Development (SCOReD)*. Kuala Lumpur. 1–6.

Jabeur, K., Navarro, D., O'Connor, I., Gaillardon, P. E., Ben Jamaa, M. H., and Clermidy, F. (2010). *Proceedings of the 2010 IEEE/ACM International Symposium on Nanoscale Architectures*. Anaheim, CA. 47–52.

Jackson, N., Keeney, L., and Mathewson, A. (2013). Flexible-CMOS and biocompatible piezoelectric AlN material for MEMS applications. *Smart Mater. Struct*. 22, 115033. doi: 10.1088/0964-1726/22/11/115033

James, A. P. (2019). An overview of memristive cryptography. *Eur. Phys. J. Spec. Top*. 228, 2301–2312. doi: 10.1140/epjst/e2019-900044-x

Jeong, H. Y., Kim, J. Y., Kim, J. W., Hwang, J. O., Kim, J. E., Lee, J. Y., et al. (2010). Graphene oxide thin films for flexible nonvolatile memory applications. *Nano Lett*. 10, 4381–4386. doi: 10.1021/nl101902k

Ji, Y., Lee, S., Cho, B., Song, S., and Lee, T. (2011). Flexible organic memory devices with multilayer graphene electrodes. *ACS Nano* 5, 5995–6000. doi: 10.1021/nn201770s

Jian, J., Dong, P., Jian, Z., Zhao, T., Miao, C., Chang, H., et al. (2022). Ultralow-power rram with a high switching ratio based on the large van der waals interstice radius of TMDs. *ACS Nano* 16, 20445–20456. doi: 10.1021/acsnano.2c06728

Jung, J., Shin, D., Lee, Y., and Pak, J. J. (2021). Fabrication of solution-processed SnO2–Based flexible ReRAM using laser-induced graphene transferred onto PDMS. *Curr. Appl. Phys*. 25, 70–74. doi: 10.1016/j.cap.2021.02.009

Kalaga, P. S., Kumar, D., Ang, D. S., and Tsakadze, Z. (2020). Highly transparent ITO/HfO2/ITO device for visible-light sensing. *IEEE Access* 8, 91648–91652. doi: 10.1109/ACCESS.2020.2994383

Khan, M. I., Ali, S., Ikram, A. A., and Bermak, A. (2021). Optimization of memristive crossbar array for physical unclonable function applications. *IEEE Access* 9, 84480–84489. doi: 10.1109/ACCESS.2021.3087810

Kim, D., Kim, T.-H., Choi, Y., Lee, G. H., Lee, J., Sun, W., et al. (2021). Selected bit-line current puf: implementation of hardware security primitive based on a memristor crossbar array. *IEEE Access* 9, 120901–120910. doi: 10.1109/ACCESS.2021.3108534

Kim, H., Lee, S., and Kim, H. (2019). Electrical heating performance of electro-conductive para-aramid knit manufactured by dip-coating in a graphene/waterborne polyurethane composite. *Sci. Rep*. 9, 1511. doi: 10.1038/s41598-018-37455-0

Kim, H.-D., Yun, M. J., Lee, J. H., Kim, K. H., and Kim, T. G. (2014). Transparent multi-level resistive switching phenomena observed in ito/rgo/ito memory cells by the sol-gel dip-coating method. *Sci. Rep*. 4, 1–6. doi: 10.1038/srep04614

Kim, I., Siddik, M., Shin, J., Biju, K. P., Jung, S., and Hwang, H. (2011). Low temperature solution-processed graphene oxide/pr0. 7ca0. 3mno3 based resistive-memory device. *Appl. Phys. Lett*. 99, 042101. doi: 10.1063/1.3617426

Kim, K. H., Gaba, S., Wheeler, D., Cruz-Albrecht, J. M., Hussain, T., Srinivasa, N., et al. (2012). A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Lett*. 12, 389–395. doi: 10.1021/nl203687n

Kim, S., Jung, H. J., Kim, J. C., Lee, K. S., Park, S. S., Dravid, V. P., et al. (2018). *In situ* observation of resistive switching in an asymmetric graphene oxide bilayer structure. *ACS Nano* 12, 7335–7342. doi: 10.1021/acsnano.8b03806

Kireev, D., Liu, S., Jin, H., Patrick Xiao, T., Bennett, C. H., Akinwande, D., et al. (2022). Metaplastic and energy-efficient biocompatible graphene artificial synaptic transistors for enhanced accuracy neuromorphic computing. *Nat. Commun*. 13, 1–11. doi: 10.1038/s41467-022-32078-6

Ko, T. J., Li, H., Mofid, S. A., Yoo, C., Okogbue, E., Han, S. S., et al. (2020). Two-dimensional near-atom-thickness materials for emerging neuromorphic devices and applications. *iScience* 23, 101676. doi: 10.1016/j.isci.2020.101676

Krestinskaya, O., James, A. P., and Chua, L. O. (2020). Neuromemristive circuits for edge computing: a review. *IEEE Trans. Neural Netw. Learn. Syst*. 31, 4–23. doi: 10.1109/TNNLS.2019.2899262

Krishnaprasad, A., Choudhary, N., Das, S., Dev, D., Kalita, H., Chung, H. S., et al. (2019). Electronic synapses with near-linear weight update using MoS2/graphene memristors. *Appl. Phys. Lett*. 115, 10. doi: 10.1063/1.5108899

Kumar, A., Pawar, S., Sharma, S., and Kaur, D. (2018). Bipolar resistive switching behavior in mos2 nanosheets fabricated on ferromagnetic shape memory alloy. *Appl. Phys. Lett*. 112, 26. doi: 10.1063/1.5037139

Kumar, D., Aluguri, R., Chand, U., and Tseng, T. Y. (2017). Metal oxide resistive switching memory: Materials, properties and switching mechanisms. *Ceramics Int*. 43, S547–S556. doi: 10.1016/j.ceramint.2017.05.289

Kumar, M., Ban, D. K., Kim, S. M., Kim, J., and Wong, C. P. (2019). Vertically aligned WS2 layers for high-performing memristors and artificial synapses. *Adv. Electr. Mater*. 5, 1900467. doi: 10.1002/aelm.201900467

Kurian, M. (2021). Recent progress in the chemical reduction of graphene oxide by green reductants–a mini review. *Carbon Trends* 5, 100120. doi: 10.1016/j.cartre.2021.100120

Lee, H. Y., Chen, P. S., Wu, T. Y., Chen, Y. S., Wang, C. C., Tzeng, P. J., et al. (2008a). *Low Power and High Speed Bipolar Switching With a Thin Reactive ti Buffer Layer in Robust HfO2 Based RRAM*. Washington, DC: Technical Digest - International Electron Devices Meeting, IEDM.

Lee, S., Sohn, J., Jiang, Z., Chen, H.-Y., and Philip Wong, H.-S. (2015). Metal oxide-resistive memory using graphene-edge electrodes. *Nat. Commun*. 6, 1–7. doi: 10.1038/ncomms9407

Lee, S. B., Chae, S. C., Chang, S. H., Lee, J. S., Seo, S., Kahng, B., et al. (2008b). Scaling behaviors of reset voltages and currents in unipolar resistance switching. *Appl. Phys. Lett*. 93, 212105. doi: 10.1063/1.3036532

Lee, S. W., Lee, H. J., Choi, J. H., Koh, W. G., Myoung, J. M., Hur, J. H., et al. (2010). Periodic array of polyelectrolyte-gated organic transistors from electrospun poly (3-hexylthiophene) nanofibers. *Nano Lett*. 10, 347–351. doi: 10.1021/nl903722z

Lei, X.-Y., Liu, H.-X., Gao, H.-X., Yang, H.-N., Wang, G.-M., Long, S.-B., et al. (2014). Resistive switching characteristics of Ti/ZrO2/Pt RRAM device. *Chinese Phys. B*. 23, 117305. doi: 10.1088/1674-1056/23/11/117305

Li, H., Wang, S., Zhang, X., Wang, W., Yang, R., Sun, Z., et al. (2021). Memristive crossbar arrays for storage and computing applications. *Adv. Intell. Syst*. 3, 2100017. doi: 10.1002/aisy.202100017

Li, H., Wu, T. F., Mitra, S., and Wong, H.-S. P. (2017). Resistive ram-centric computing: Design and modeling methodology. *IEEE TCAS-I*. 64, 2263–2273. doi: 10.1109/TCSI.2017.2709812

Li, N., Wang, Z., and Shi, Z. (2011). "Synthesis of graphenes with arc-discharge method," in *Physics and Applications of Graphene - Experiments*.

Li, Y., Wang, Z., Midya, R., Xia, Q., and Yang, J. J. (2018). Review of memristor devices in neuromorphic computing: materials sciences and device challenges. *J. Phys. D Appl. Phys*. 51, 503002. doi: 10.1088/1361-6463/aade3f

Li, Y.-T., Zhao, H.-M., Tian, H., Wang, X.-F., Mi, W.-T., Yang, Y., et al. (2016). "Novel graphene-based resistive random access memory," in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)* (Hangzhou), 462–465.

Lin, C.-L., Chang, W.-Y., Huang, Y.-L., Juan, P.-C., Wang, T.-W., Hung, K.-Y., et al. (2015). Resistance switching behavior of zno resistive random access memory with a reduced graphene oxide capping layer. *Jpn. J. Appl. Phys*. 54, 04DJ08. doi: 10.7567/JJAP.54.04DJ08

Lin, C. Y., Wu, C. Y., Wu, C. Y. Y., Lee, T. C., Yang, F. L., Hu, C., et al. (2007). Effect of top electrode material on resistive switching properties of ZrO2 film memory devices. *IEEE Electron Device Letters* 28:366–368. doi: 10.1109/LED.2007.894652

Lin, Z., McCreary, A., Briggs, N., Subramanian, S., Zhang, K., Sun, Y., et al. (2016). 2D materials advances: from large scale synthesis and controlled heterostructures to improved characterization techniques, defects and applications. *2D Materials* 3, 042001. doi: 10.1088/2053-1583/3/4/042001

Liu, B., Liu, Z., Chiu, I.-S., Di, M., Wu, Y., Wang, J.-C., et al. (2018). Programmable synaptic metaplasticity and below femtojoule spiking energy realized in graphene-based neuromorphic memristor. *ACS Appl. Mater. Interf*. 10, 20237–20243. doi: 10.1021/acsami.8b04685

Liu, J., Yin, Z., Cao, X., Zhao, F., Wang, L., Huang, W., et al. (2013). Fabrication of flexible, all-reduced graphene oxide non-volatile memory devices. *Adv. Mater*. 25, 233–238. doi: 10.1002/adma.201203349

Liu, J., Zeng, Z., Cao, X., Lu, G., Wang, L. H., Fan, Q. L., et al. (2012). Preparation of MoS2-polyvinylpyrrolidone nanocomposites for flexible nonvolatile rewritable memory devices with reduced graphene oxide electrodes. *Small* 8, 3517–3522. doi: 10.1002/smll.201200999

Long, S., Lian, X., Ye, T., Cagli, C., Perniola, L., Miranda, E., et al. (2013). Cycle-to-cycle intrinsic RESET statistics in HfO2-based unipolar RRAM devices. *IEEE Electron Device Lett*. 34, 623–625. doi: 10.1109/LED.2013.2251314

Long, Y., Wu, L., Pan, F., Zhang, Z., Yang, M., Tang, A., et al. (2019). A graphene spin coatings for cost-effective corrosion protection for the magnesium alloy az31. *J. Nanosci. Nanotechnol*. 19, 105–111. doi: 10.1166/jnn.2019.16436

Lu, Q.., Zhao, Y.., Huang, L.., An, J.., Zheng, Y.., Yap, E. H., et al. (2023). Low-dimensional-materials-based flexible artificial synapse: materials, devices, and systems. *Nanomaterials* 13, 373. doi: 10.3390/nano13030373

Lu, W., Bao, N., Zheng, T., Zhang, X., and Song, Y. (2022). Memristor-based read/write circuit with stable continuous read operation. *Electronics* 11, 2018. doi: 10.3390/electronics11132018

Ma, W., Zidan, M. A., and Lu, W. D. (2018). Neuromorphic computing with memristive devices. *Sci. China Informat. Sci*. 61, 1–9. doi: 10.1007/s11432-017-9424-y

Meena, J. S., Sze, S. M., Chand, U., and Tseng, T. Y. (2014). Overview of emerging nonvolatile memory technologies. *Nanoscale Res. Lett*. 9, 1–33. doi: 10.1186/1556-276X-9-526

Mehonic, A., Sebastian, A., Rajendran, B., Simeone, O., Vasilaki, E., and Kenyon, A. J. (2020). Memristors from in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing. *Adv. Intellig. Syst.* 2:2000085. doi: 10.1002/aisy.202000085

Meyer, J., Hamwi, S., Kröger, M., Kowalsky, W., Riedl, T., and Kahn, A. (2012). Transition metal oxides for organic electronics: energetics, device physics and applications. *Adv. Mater.* 24, 5408–5427. doi: 10.1002/adma.201201630

Milo, V., Zambelli, C., Olivo, P., Pérez, E., K., Mahadevaiah, M., et al. (2019). Multilevel HfO2-based RRAM devices for low-power neuromorphic networks. *APL Mater.* 7, 81120. doi: 10.1063/1.5108650

Moon, J. S., and Gaskill, D. K. (2011). Graphene: its fundamentals to future applications. *IEEE Trans. Microw. Theory Tech.* 59, 2702–2708. doi: 10.1109/TMTT.2011.2164617

Mueller, N. S., Morfa, A. J., Abou-Ras, D., Oddone, V., Ciuk, T., and Giersig, M. (2014). Growing graphene on polycrystalline copper foils by ultra-high vacuum chemical vapor deposition. *Carbon N. Y.* 78, 347–355. doi: 10.1016/j.carbon.2014.07.011

Nagareddy, V. K., Barnes, M. D., Zipoli, F., Lai, K. T., Alexeev, A. M., Craciun, M. F., et al. (2017). Multilevel ultrafast flexible nanoscale nonvolatile hybrid graphene oxide-titanium oxide memories. *ACS Nano* 11, 3010–3021. doi: 10.1021/acsnano.6b08668

Nicolosi, V., Chhowalla, M., Kanatzidis, M. G., Strano, M. S., and Coleman, J. N. (2013). Liquid exfoliation of layered materials. *Science* 340, 1226419. doi: 10.1126/science.1226419

Ninomiya, T., Wei, Z., Muraoka, S., Yasuhara, R., Katayama, K., and Takagi, T. (2013). Conductive filament scaling of TaOx bipolar ReRAM for improving data retention under low operation current. *IEEE Trans. Electron Devices* 60, 1384–1389. doi: 10.1109/TED.2013.2248157

Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D., Zhang, Y., Dubonos, S. V., et al. (2004). Electric field in atomically thin carbon films. *Science* 306, 666–669. doi: 10.1126/science.1102896

Novoselov, K. S., Jiang, D., Schedin, F., Booth, T. J., Khotkevich, V. V., Morozov, S. V., et al. (2005). Two-dimensional atomic crystals. *Proc. Natl. Acad. Sci. USA*. 102, 10451–10453. doi: 10.1073/pnas.0502848102

Pan, C., Miranda, E., Villena, M. A., Xiao, N., Jing, X., Xie, X., et al. (2017). Model for multi-filamentary conduction in graphene/hexagonal-boron-nitride/graphene based resistive switching devices. *2D Materials* 4, 025099. doi: 10.1088/2053-1583/aa7129

Panin, G. N., Kapitanova, O. O., Lee, S. W., Baranov, A. N., and Kang, T. W. (2011). Resistive switching in al/graphene oxide/al structure. *Jpn. J. Appl. Phys.* 50, 070110. doi: 10.1143/JJAP.50.070110

Papageorgiou, D. G., Kinloch, I. A., and Young, R. J. (2017). Mechanical properties of graphene and graphene-based nanocomposites. *Prog. Mater. Sci.* 90, 75–127. doi: 10.1016/j.pmatsci.2017.07.004

Park, W. I., Yoon, J. M., Park, M., Lee, J., Kim, S. K., Jeong, J. W., et al. (2012). Self-assembly-induced formation of high-density silicon oxide memristor nanostructures on graphene and metal electrodes. *Nano Lett.* 12, 1235–1240. doi: 10.1021/nl203597d

Perez-Campos, A., Iriarte, F., Hernando-Garcia, J., and Calle, F. (2015). Post-CMOS compatible high-throughput fabrication of AlN-based piezoelectric microcantilevers. *J. Micromech. Microeng.* 25, 025003. doi: 10.1088/0960-1317/25/2/025003

Pradhan, S. K., Xiao, B., Mishra, S., Killam, A., and Pradhan, A. K. (2016). Resistive switching behavior of reduced graphene oxide memory cells for low power nonvolatile device application. *Sci. Rep.* 6, 1–9. doi: 10.1038/srep26763

Prakash, A., Deleruyelle, D., Song, J., Bocquet, M., and Hwang, H. (2015). Resistance controllability and variability improvement in a TaOx-based resistive memory for multilevel storage application. *Appl. Phys. Lett.* 106, 23. doi: 10.1063/1.4922446

Prakash, A., and Hwang, H. (2016). Multilevel cell storage and resistance variability in resistive random access memory. *Nano Devic. Sens.* 4, 49–72. doi: 10.1515/9781501501531-004

Prakash, A., and Hwang, H. (2019). Multilevel cell storage and resistance variability in resistive random access memory. *Phys. Sci. Rev.* 1, 6. doi: 10.1515/psr-2016-0010

Prodromakis, T., and Toumazou, C. (2010). "A review on memristive devices and applications," in *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, 934–937. Athens: IEEE.

Puah, P. Y., Yusoff, U. H., Lee, P. C., MOH, P. Y., and How, S. E. (2020). Surface characterization, biocompatibility and osteogenic differentiation of drop-casted multilayer graphene oxide film towards human wharton's jelly derived mesenchymal stem cells. *Mater. Technol.* 35, 238–247. doi: 10.1080/10667857.2019.1674506

Qi, M., Tao, Y., Wang, Z., Xu, H., Zhao, X., Liu, W., et al. (2018). Highly uniform switching of HfO2–x based RRAM achieved through Ar plasma treatment for low power and multilevel storage. *Appl. Surf. Sci.* 458, 216–221. doi: 10.1016/j.apsusc.2018.07.095

Qian, K., Tay, R. Y., Nguyen, V. C., Wang, J., Cai, G., Chen, T., et al. (2016). Hexagonal boron nitride thin film for flexible resistive memory applications. *Adv. Funct. Mater.* 26, 2176–2184. doi: 10.1002/adfm.201504771

Qian, M., Pan, Y., Liu, F., Wang, M., Shen, H., He, D., et al. (2014). Tunable, ultralow-power switching in memristive devices enabled by a heterogeneous graphene-oxide interface. *Adv. Mater.* 26, 3275–3281. doi: 10.1002/adma.201306028

Rani, A., Song, J.-M., Jung Lee, M., and Lee, J.-S. (2012). Reduced graphene oxide based flexible organic charge trap memory devices. *Appl. Phys. Lett.* 101, 233308. doi: 10.1063/1.4769990

Rehman, M. M., Rehman, H. M. M. U., Gul, J. Z., Kim, W. Y., Karimov, K. S., and Ahmed, N. (2020). Decade of 2d-materials-based rram devices: a review. *Sci. Technol. Adv. Mater.* 21, 147–186. doi: 10.1080/14686996.2020.1730236

Rehman, M. M., Siddiqui, G. U., Doh, Y. H., and Choi, K. H. (2017). Highly flexible and electroforming free resistive switching behavior of tungsten disulfide flakes fabricated through advanced printing technology. *Semicond. Sci. Technol.* 32, 095001. doi: 10.1088/1361-6641/aa77db

Rodriguez, S., Vaziri, S., Ostling, M., Rusu, A., Alarcon, E., and Lemme, M. C. (2012). Rf performance projections of graphene fets vs. silicon mosfets. *ECS Solid State Lett.* 1, Q39. doi: 10.1149/2.001205ssl

Rose, G. S., and Meade, C. A. (2015). "Performance analysis of a memristive crossbar puf design," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. San Francisco, CA: IEEE, 1–6. doi: 10.1145/2744769.2744892

Saini, P., Singh, M., Thakur, J., Patil, R., Ma, Y. R., Tandon, R. P., et al. (2018). Probing the mechanism for bipolar resistive switching in annealed graphene oxide thin films. *ACS Appl. Mater. Interfaces*. 10:6521–6530. doi: 10.1021/acsami.7b09447

Sano, K., Hatsuda, Y., and Yamamoto, S. (2013). Multi-fpga accelerator for scalable stencil computation with constant memory bandwidth. *IEEE Trans. Parallel Distrib. Syst.* 25, 695–705. doi: 10.1109/TPDS.2013.51

Sawa, A. (2008). Resistive switching in transition metal oxides. *Mater. Today* 11, 28–36. doi: 10.1016/S1369-7021(08)70119-6

Schranghamer, T. F., Oberoi, A., and Das, S. (2020). Graphene memristive synapses for high precision neuromorphic computing. *Nat. Commun.* 11, 1–11. doi: 10.1038/s41467-020-19203-z

Sharma, I., Papanai, G. S., Paul, S. J., and Gupta, B. K. (2020). Partial pressure assisted growth of single-layer graphene grown by low-pressure chemical vapor deposition: implications for high-performance graphene FET devices. *ACS Omega* 5, 22109–22118. doi: 10.1021/acsomega.0c02132

Shehab, M., Abualigah, L., Omari, M., Shambour, M. K. Y., Alshinwan, M., Abuaddous, H. Y., et al. (2022). "Artificial neural networks for engineering applications: a review," in *Artificial Neural Networks for Renewable Energy Systems and Real-World Applications*. Cambridge, MA: Academic Press, 189–206.

Shin, Y. J., Kwon, J. H., Kalon, G., Lam, K.-T., Bhatia, C. S., Liang, G., et al. (2010). Ambipolar bistable switching effect of graphene. *Appl. Phys. Lett.* 97, 262105. doi: 10.1063/1.3532849

Shindome, A., Doioka, Y., Beppu, N., Oda, S., and Uchida, K. (2013). Experimental study of two-terminal resistive random access memory realized in mono- and multilayer exfoliated graphene nanoribbons. *Jpn. J. Appl. Phys.* 52, :04CN05. doi: 10.7567/JJAP.52.04CN05

Singh, D. P., Herrera, C. E., Singh, B., Singh, S., Singh, R. K., and Kumar, R. (2018). Graphene oxide: an efficient material and recent approach for biotechnological and biomedical applications. *Mater. Sci. Eng. C* 86, 173–197. doi: 10.1016/j.msec.2018.01.004

Sohn, J., Lee, S., Jiang, Z., Chen, H. Y., and Wong, H. S. (2015). *Atomically Thin Graphene Plane Electrode for 3D RRAM*. Washington, DC: Technical Digest - International Electron Devices Meeting, IEDM, 1–5.

Sparvoli, M., and Marma, J. S. (2018). "Development of resistive memories based on silver doped graphene oxide for neuron simulation," in *2018 International Joint Conference on Neural Networks*, Rio de Janeiro, Brazil: IEEE, 1–6.

Starzyk, J. A., and Basawaraj. (2014). Memristor crossbar architecture for synchronous neural networks. *IEEE Trans. Circuits Syst. I: Regul. Pap.* 61, 2390–2401. doi: 10.1109/TCSI.2014.2304653

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Sun, B., Zhao, W., Liu, Y., and Chen, P. (2015). Resistive switching effect of ag/mos 2/fto device. *Function. Mater. Lett.* 8, 1550010. doi: 10.1142/S1793604715500101

Sung, S. H., Kim, T. J., Shin, H., Im, T. H., and Lee, K. J. (2022). Simultaneous emulation of synaptic and intrinsic plasticity using a memristive synapse. *Nat. Commun.* 13, 1. doi: 10.1038/s41467-022-30432-2

Terai, M., Sakotsubo, Y., Kotsuji, S., and Hada, H. (2010). Resistance controllability of Ta2O5/TiO2 stack ReRAM for low-voltage and multilevel operation. *IEEE Electron. Device Lett.* 31, 204–206. doi: 10.1109/LED.2009.2039021

Tsigkourakos, M., Bousoulas, P., Aslanidis, V., Skotadis, E., and Tsoukalas, D. (2017). Ultra-low power multilevel switching with enhanced uniformity in forming free tio2- x-based rram with embedded pt nanocrystals. *Physica Status Solidi* 214, 1700570. doi: 10.1002/pssa.201700570

Uddin, M., Majumder, M. B., and Rose, G. S. (2017). Robustness analysis of a memristive crossbar puf against modeling attacks. *IEEE Trans. Nanotechnol.* 16, 396–405. doi: 10.1109/TNANO.2017.2677882

Vasu, K., Sampath, S., and Sood, A. (2011). Nonvolatile unipolar resistive switching in ultrathin films of graphene and carbon nanotubes. *Solid State Commun.* 151, 1084–1087. doi: 10.1016/j.ssc.2011.05.018

Venugopal, G., and Kim, S. J. (2012). Observation of nonvolatile resistive memory switching characteristics in Ag/graphene-oxide/Ag devices. *J. Nanosci. Nanotechnol.* 12, 8522–8525. doi: 10.1166/jnn.2012.6675

Vourkas, I., Stathis, D., Sirakoulis, G. C., and Hamdioui, S. (2016). Alternative architectures toward reliable memristive crossbar memories. *IEEE Transact. Very Large Scale Integrat. (VLSI) Syst.* 24, 206–217. doi: 10.1109/TVLSI.2015.2388587

Wang, C.-H., McClellan, C., Shi, Y., Zheng, X., Chen, V., Lanza, M., et al. (2018). "3d monolithic stacked 1t1r cells using monolayer mos 2 fet and hbn rram fabricated at low (150c) temperature," in *2018 IEEE International Electron Devices Meeting (IEDM)*, 22–5. San Francisco, CA: IEEE.

Wang, L.-H., Yang, W., Sun, Q.-Q., Zhou, P., Lu, H.-L., Ding, S.-J., et al. (2012a). The mechanism of the asymmetric set and reset speed of graphene oxide based flexible resistive switching memories. *Appl. Phys. Lett.* 100, 063509. doi: 10.1063/1.3681366

Wang, S. Y., Lee, D. Y., Tseng, T. Y., and Lin, C. Y. (2009). Effects of Ti top electrode thickness on the resistive switching behaviors of rf-sputtered ZrO2 memory films. *Appl. Phys. Lett.* 95, 112904. doi: 10.1063/1.3231872

Wang, Z., Tjoa, V., Wu, L., Liu, W., Fang, Z., Tran, X. A., et al. (2012b). Mechanism of different switching directions in graphene oxide based rram. *J. Electrochem. Soc.* 159, K177. doi: 10.1149/2.068206jes

Wu, C., Li, F., Zhang, Y., and Guo, T. (2012). Recoverable electrical transition in a single graphene sheet for application in nonvolatile memories. *Appl. Phys. Lett.* 100, 042105. doi: 10.1063/1.3680093

Wu, C., Li, F., and Guo, T. (2014). Efficient tristable resistive memory based on single layer graphene/insulating polymer multi-stacking layer. *Appl. Phys. Lett.* 104, 183105. doi: 10.1063/1.4875596

Wu, H. Y., Lin, C. C., and Lin, C. H. (2015a). Characteristics of graphene-oxide-based flexible and transparent resistive switching memory. *Ceramics Int.* 41, S823–S828. doi: 10.1016/j.ceramint.2015.03.129

Wu, Q., Banerjee, W., Cao, J., Ji, Z., Li, L., and Liu, M. (2018). Improvement of durability and switching speed by incorporating nanocrystals in the hfox based resistive random access memory devices. *Appl. Phys. Lett.* 113, 5030780. doi: 10.1063/1.5030780

Wu, X., Saxena, V., and Zhu, K. (2015b). Homogeneous spiking neuromorphic system for real-world pattern recognition. *IEEE J. Emerg. Select. Top. Circ. Syst.* 5, 254–266. doi: 10.1109/JETCAS.2015.2433552

Wu, Y., Lee, B., and Wong, H. S. (2010). Al2O3-based RRAM using atomic layer deposition (ALD) with 1-μa RESET current. *IEEE Electron Devi. Lett.* 31, 1449–1451. doi: 10.1109/LED.2010.2074177

Wu, Y., Wang, L., Jin, M., Kong, F., Qi, H., and Nan, J. (2019). Reduced graphene oxide and biofilms as cathode catalysts to enhance energy and metal recovery in microbial fuel cell. *Bioresour. Technol.* 283, 129–137. doi: 10.1016/j.biortech.2019.03.080

Xia, Q., and Yang, J. J. (2019). Memristive crossbar arrays for brain-inspired computing. *Nature Mater.* 18, 309–323. doi: 10.1038/s41563-019-0291-x

Xie, J., Afshari, S., and Sanchez Esqueda, I. (2022). Hexagonal boron nitride (h-BN) memristor arrays for analog-based machine learning hardware. *NPJ 2D Mater. Appl.* 6, 1–7. doi: 10.1038/s41699-022-00328-2

Xu, R., Jang, H., Lee, M. H., Amanov, D., Cho, Y., Kim, H., et al. (2019a). Vertical MoS2 double-layer memristor with electrochemical metallization as an atomic-scale synapse with switching thresholds approaching 100 mV. *Nano Lett.* 19, 2411–2417. doi: 10.1021/acs.nanolett.8b05140

Xu, W., Wang, J., and Yan, X. (2021). Advances in memristor-based neural networks. *Front. Nanotechnol.* 3, 20. doi: 10.3389/fnano.2021.645995

Xu, Z., Li, F., Wu, C., Ma, F., Zheng, Y., Yang, K., et al. (2019b). Ultrathin electronic synapse having high temporal/spatial uniformity and an al2o3/graphene quantum dots/al2o3 sandwich structure for neuromorphic computing. *NPG Asia Mater.* 11, 1–11. doi: 10.1038/s41427-019-0118-x

Xue, C. J., Zhang, Y., Chen, Y., Sun, G., Yang, J. J., and Li, H. (2011). "*Emerging non-volatile memories: Opportunities and challenges*," in *Proceedings of the Seventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis* (New York, NY), 325–334.

Yakopcic, C., Fernando, B. R., and Taha, T. M. (2019). "Design space evaluation of a memristor crossbar based multilayer perceptron for image processing," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. Budapest: IEEE. doi: 10.1109/IJCNN.2019.8852005

Yan, X., Zhang, L., Chen, H., Li, X., Wang, J., Liu, Q., et al. (2018). Graphene oxide quantum dots based memristors with progressive conduction tuning for artificial synaptic learning. *Adv. Funct. Mater.* 28, 1803728. doi: 10.1002/adfm.201803728

Yang, J. J., and Williams, R. S. (2013). Memristive devices in computing system: promises and challenges. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* 9, 1–20. doi: 10.1145/2463585.2463587

Yang, L., Kuegeler, C., Szot, K., Ruediger, A., and Waser, R. (2009). The influence of copper top electrodes on the resistive switching effect in TiO2 thin films studied by conductive atomic force microscopy. *Appl. Phys. Lett.* 95, 013109. doi: 10.1063/1.3167810

Yansong, G., C., R. D., Said, A.-S., Kavehei, O., and Abbott, D. (2015). Memristive crypto primitive for building highly secure physical unclonable functions. *Sci. Rep.* 5, 12785. doi: 10.1038/srep12785

Yao, J., Lin, J., Dai, Y., Ruan, G., Yan, Z., Li, L., et al. (2012). Highly transparent nonvolatile resistive memory devices from silicon oxide and graphene. *Nat. Commun.* 3, 1101. doi: 10.1038/ncomms2110

Yeh, C.-W. S., and Wong, S. S. (2015). Compact one-transistor-n-rram array architecture for advanced cmos technology. *IEEE J. Solid-State Circ.* 50, 1299–1309. doi: 10.1109/JSSC.2015.2402217

Yi, M., Cao, Y., Ling, H., Du, Z., Wang, L., Yang, T., et al. (2014). Temperature dependence of resistive switching behaviors in resistive random access memory based on graphene oxide film. *Nanotechnology* 25, 185202. doi: 10.1088/0957-4484/25/18/185202

Ying-Chih Lai, C., Hsu, F.-C., Chen, J.-Y., He, J.-H., Chang, T.-C., Hsieh, Y.-P., et al. (2013). Transferable and flexible label-like macromolecular memory on arbitrary substrates with high performance and a facile methodology. *Adv. Mater.* 25, 2733–2739. doi: 10.1002/adma.201205280

Yu, F., Kong, X., Mokbel, A. A. M., Yao, W., and Cai, S. (2023). Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive hopfield neural network with a novel local active memeristor. *IEEE Trans. Circuits Syst II: Express Briefs* 70, 326–330. doi: 10.1109/TCSII.2022.3218468

Yu, J., Du Nguyen, H. A., Xie, L., Taouil, M., and Hamdioui, S. (2018). "Memristive devices for computation-in-memory," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Dresden: IEEE, 1646–1651.

Yu, J., Sun, W., Lai, J., Zheng, X., Dong, D., Luo, Q., et al. (2022). Performance improvement of memristor-based echo state networks by optimized programming scheme. *IEEE Electron Devi. Lett.* 43, 866–869. doi: 10.1109/LED.2022.3165831

Yu, S., Guan, X., and Wong, H. S. (2012). On the switching parameter variation of metal oxide RRAM - part II: model corroboration and device design strategy. *IEEE Trans. Electron Devices* 59, 1183–1188. doi: 10.1109/TED.2012.2184544

Yu, S., Jeyasingh, R., Wu, Y., and Philip Wong, H. S. (2011a). *Understanding the Conduction and Switching Mechanism of Metal Oxide RRAM Through low Frequency Noise and AC Conductance Measurement and Analysis*. Washington, DC: Technical Digest - International Electron Devices Meeting, IEDM.

Yu, W. J., Chae, S. H., Lee, S. Y., Duong, D. L., and Lee, Y. H. (2011b). Ultra-transparent, flexible single-walled carbon nanotube non-volatile memory device with an oxygen-decorated graphene electrode. *Adv. Mater.* 23, 1889–1893. doi: 10.1002/adma.201004444

Yung, K. C., Wu, W. M., Pierpoint, M. P., and Kusmartsev, F. V. (2013). Introduction to graphene electronics a new era of digital transistors and devices. *Contemporary Physics*. 54, 233–251. doi: 10.1080/00107514.2013.833701

Zahoor, F., Azni Zulkifli, T. Z., and Khanday, F. A. (2020). Resistive random access memory (RRAM): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications. *Nanoscale Res. Lett.* 15, 1–26. doi: 10.1186/s11671-020-03299-9

Zhang, F., Krylyuk, S., Zhang, H., Milligan, C. A., Zemlyanov, D. Y., Bendersky, L. A., et al. (2018). Electric field induced phase transition in vertical MoTe2 and Mo1-xWxTe2 based RRAM devices. *Nat. Mater.* 18, 55–61. doi: 10.1038/s41563-018-0234-y

Zhang, F., Li, C., Li, Z., Dong, L., and Zhao, J. (2023). Recent progress in three-terminal artificial synapses based on 2D materials: from mechanisms to applications. *Microsyst. Nanoeng.* 9, 1–17. doi: 10.1038/s41378-023-00487-2

Zhang, H. (2015). Ultrathin two-dimensional nanomaterials. *ACS Nano* 9, 9451–9469. doi: 10.1021/acsnano.5b05040

Zhang, S., Long, S., Guan, W., Liu, Q., Wang, Q., and Liu, M. (2009). Resistive switching characteristics of MnOx-based ReRAM. *J. Phys. D Appl. Phys.* 42, 055112. doi: 10.1088/0022-3727/42/5/055112

Zhang, Y., Ren, W., Jiang, Z., Yang, S., Jing, W., Shi, P., et al. (2014a). Low-temperature remote plasma-enhanced atomic layer deposition of graphene and characterization of its atomic-level structure. *J. Mater. Chem. C* 2, 7570–7574. doi: 10.1039/C4TC00849A

Zhang, Z., Gao, B., Fang, Z., Wang, X., Tang, Y., Sohn, J., et al. (2014b). All-metal-nitride rram devices. *IEEE Electron Dev. Lett.* 36, 29–31. doi: 10.1109/LED.2014.2367542

Zhang, Z., Yang, D., Li, H., Li, C., Wang, Z., Sun, L., et al. (2022). 2d materials and van der waals heterojunctions for neuromorphic computing. *Neuromorphic Comput. Eng.* 2, 032004. doi: 10.1088/2634-4386/ac8a6a

Zhao, H., Tu, H., Wei, F., and Du, J. (2014a). Highly transparent dysprosium oxide-based rram with multilayer graphene electrode for low-power nonvolatile memory application. *IEEE Trans. Electron Dev.* 61, 1388–1393. doi: 10.1109/TED.2014.2312611

Zhao, L., Chen, H. Y., Wu, S. C., Jiang, Z., Yu, S., Hou, T. H., et al. (2014b). Multi-level control of conductive nano-filament evolution in HfO2 ReRAM by pulse-train operations. *Nanoscale* 6, 5698–5702. doi: 10.1039/C4NR00500G

Zhong, Y. L., Tian, Z., Simon, G. P., and Li, D. (2015). Scalable production of graphene via wet chemistry: progress and challenges. *Mater. Today* 18 73–78. doi: 10.1016/j.mattod.2014.08.019

Zhou, F., Chen, J., Tao, X., Wang, X., and Chai, Y. (2019). 2D materials based optoelectronic memory: convergence of electronic memory and optical sensor. *Research*. 2019, :9490413. doi: 10.34133/2019/9490413

Zhou, G., Lu, Z., Yao, Y., Wang, G., Yang, X., Zhou, A., et al. (2017). Mechanism for bipolar resistive switching memory behaviors of a self-assembled three-dimensional mos2 microsphere composed active layer. *J. Appl. Phys*. 121, 15. doi: 10.1063/1.4980173

Zhou, G., Sun, B., Yao, Y., Zhang, H., Zhou, A., Alameh, K., et al. (2016). Investigation of the behaviour of electronic resistive switching memory based on mose2-doped ultralong se microwires. *Appl. Phys. Lett*. 109, 14. doi: 10.1063/1.4962655

Zhu, L., Zhou, J., Guo, Z., and Sun, Z. (2015). An overview of materials issues in resistive random access memory. *J. Materiom*. 1, 285–295. doi: 10.1016/j.jmat.2015.07.009

Zhu, W., Low, T., Wang, H., Ye, P., and Duan, X. (2019). Nanoscale electronic devices based on transition metal dichalcogenides. *2D Materials* 6, 032004. doi: 10.1088/2053-1583/ab1ed9

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 04 December 2023
DOI 10.3389/fnins.2023.1294954

OPEN ACCESS

EDITED BY
Pier Luigi Gentili,
Università degli Studi di Perugia, Italy

REVIEWED BY
Gyorgy Csaba,
Pázmány Péter Catholic University, Hungary
Konrad Szaciłowski,
AGH University of Science and Technology, Poland

*CORRESPONDENCE
Manuel Jiménez
✉ jimenez@imse-cnm.csic.es

RECEIVED 15 September 2023
ACCEPTED 10 November 2023
PUBLISHED 04 December 2023

CITATION
Jiménez M, Núñez J, Shamsi J, Linares-Barranco B and Avedillo MJ (2023) Experimental demonstration of coupled differential oscillator networks for versatile applications.
Front. Neurosci. 17:1294954.
doi: 10.3389/fnins.2023.1294954

Experimental demonstration of coupled differential oscillator networks for versatile applications

Manuel Jiménez[1]*, Juan Núñez[1], Jafar Shamsi[2], Bernabé Linares-Barranco[1] and María J. Avedillo[1]

[1]Instituto de Microelectrónica de Sevilla, IMSE-CNM (CSIC/Universidad de Sevilla), Seville, Spain, [2]Department of Cell Biology and Anatomy, University of Calgary, Calgary, AB, Canada

Oscillatory neural networks (ONNs) exhibit a high potential for energy-efficient computing. In ONNs, neurons are implemented with oscillators and synapses with resistive and/or capacitive coupling between pairs of oscillators. Computing is carried out on the basis of the rich, complex, non-linear synchronization dynamics of a system of coupled oscillators. The exploited synchronization phenomena in ONNs are an example of fully parallel collective computing. A fast system's convergence to stable states, which correspond to the desired processed information, enables an energy-efficient solution if small area and low-power oscillators are used, specifically when they are built on the basis of the hysteresis exhibited by phase-transition materials such as VO_2. In recent years, there have been numerous studies on ONNs using VO_2. Most of them report simulation results. Although in some cases experimental results are also shown, they do not implement the design techniques that other works on electrical simulations report that allow to improve the behavior of the ONNs. Experimental validation of these approaches is necessary. Therefore, in this study, we describe an ONN realized in a commercial CMOS technology in which the oscillators are built using a circuit that we have developed to emulate the VO_2 device. The purpose is to be able to study in-depth the synchronization dynamics of relaxation oscillators similar to those that can be performed with VO_2 devices. The fabricated circuit is very flexible. It allows programming the synapses to implement different ONNs, calibrating the frequency of the oscillators, or controlling their initialization. It uses differential oscillators and resistive synapses, equivalent to the use of memristors. In this article, the designed and fabricated circuits are described in detail, and experimental results are shown. Specifically, its satisfactory operation as an associative memory is demonstrated. The experiments carried out allow us to conclude that the ONN must be operated according to the type of computational task to be solved, and guidelines are extracted in this regard.

KEYWORDS

oscillatory neural networks, nano-oscillators, ASIC, phase-change material, neuromorphics, integrated circuits

1 Introduction

Current society demands more and more applications that require applying computationally hard and data-intensive algorithms, for example, neural networks. These are generally run on devices such as CPUs or GPUs, which offer great computing power but also require high energy consumption for their operation, which limits their use in edge computing. An alternative to

the use of CPUs or GPUs is their implementation in hardware. Currently, the development of these custom-specific hardware platforms is an area of high interest. It comprises many approaches, including digital and analog implementations. In the latter, the use of unconventional computing devices and paradigms is very promising. In this line of oscillatory neural networks (ONNs), the connection of a multitude of oscillator circuits by means of electrical coupling elements creates an intelligent collective system called oscillatory neural networks (ONNs) (Hoppensteadt and Izhikevich, 1999, 2000; Follmann et al., 2015; Sharma et al., 2015; Raychowdhury et al., 2019), with a high potential for energy-efficient computing. In ONNs, neurons are implemented with oscillators and synapses with resistive and/or capacitive coupling between pairs of oscillators. Computing is carried out on the basis of the rich, complex, non-linear synchronization dynamics of a system of coupled oscillators. When the oscillators synchronize in frequency, they tend to adopt a phase relationship that minimizes energy. The most commonly used ONN encodes information about the relationship between oscillator phases. Depending on the type of coupling, the phases of two interconnected oscillators tend to get closer (to be in phase) or to separate (to be out of phase or anti-phase). The energy landscape of the system is determined by the coupling configuration. Thus, the idea behind computing with ONNs is to map the solutions of the target task into their minimal energy states. The exploited synchronization phenomena are an example of what is called collective computing and are fully parallel. Convergence to the stable system state is fast, which paves the way for energy efficiency associated with low computation times. It has been proposed to be used as associative memory (AM) by configuring the couplings such that the patterns to be stored (training patterns) are minimal energy states of the system (Nikonov et al., 2015). ONNs are also useful for solving optimization problems by formulating them as an Ising model (Lucas, 2014) and mapping them to an ONN (Dutta et al., 2021). The Ising model problem is solved by the natural evolution of the ONN state to states associated with minimum values in its energy function (Hamiltonian). The relationship between ONNs and Hopfield neural networks (HNNs) (Hopfield, 1982) is evident at this point.

ONN implementations with different types of oscillators have been reported (phase-locked loops and voltage-controlled oscillators (Hoppensteadt and Izhikevich, 2000), non-volatile logic based on magnetic tunnel junctions (Calayir and Pileggi, 2013), micro-electro-mechanical systems and a feedback loop with transconductance amplifiers (Kumar and Mohanty, 2017), comparator and a digital circuit in Jackson et al. (2018), CMOS ring oscillators (Csaba et al., 2016; Ahmed et al., 2021; Moy et al., 2022), STOs (Popescu et al., 2018), or VO_2 (Corti et al., 2018), (Dutta et al., 2019, 2021; Corti et al., 2020; Núñez et al., 2021).

Structurally, the ONN resembles an artificial network based on the Hopfield model, HNN, which has been studied in-depth with regard to AM and pattern recognition tasks. The HNN has a simple conceptual model comprising a single, recurrent, fully connected layer of neurons with synaptic weights. Typically, the HNN model considers bipolar-state neurons. The state of each neuron is represented by S_i and it takes values in $\{-1, +1\}$. State updates as:

$$S_i = sign\left(\sum_j W_{ij} S_j\right) \quad (1)$$

with W_{ij} the weight of the synapse connecting neuron i and neuron j, $W_{ij} = W_{ji}$, and $W_{ii} = 0$. The HNN gradually transitions from an initial input state until a fixed point is reached. Fixed (stable or attractors) states are determined by the synaptic weights. In AM applications, the weight values are assigned (trained) in such a way that the patterns to be stored are fixed as attractor states. When an input pattern is applied, it evolves toward the closest stored pattern. In other words, when a distorted version of a training pattern is applied to the HNN, the original one is retrieved (inference).

Of course, energy-efficient oscillators are also necessary to achieve the target goal of energy-efficient computation. In this sense, oscillatory-based computing is not new. There were early contributions from pioneers such as Von Neumann (1957) and Goto (1959) in the 1950s. However, recent advances in technology have made it a popular and active research area. This is due to the emergence of phase-transition devices that can implement highly efficient and compact oscillators with minimal energy consumption based on various physical phenomena. VO_2 devices, in particular, stand out for their hysteresis in the characteristic I–V curve, which enables compact low-power relaxation oscillators (Csaba and Porod, 2020).

VO_2 material undergoes metal–insulator transitions under given electrical stimuli. That is, abrupt switching occurs from/to a high resistivity state (insulating phase) to/from a low resistivity state (metallic phase). Without electrical stimuli, it tends to stabilize in the insulating phase. When the applied voltage increases and the current density flowing through it reaches a given amount, an insulator-to-metal transition (IMT) occurs. Once in the metallic state, when the voltage decreases and the current density drops below a second given value, a metal-to-insulator transition (MIT) takes place. Figure 1A shows the I–V characteristic of a generic VO_2. A compact oscillator has been proposed on its basis (Figure 1B; Maffezzoni et al., 2015; Parihar et al., 2015). Figure 1C depicts waveforms for the oscillator output. The state of the VO_2 is also shown to better illustrate the circuit

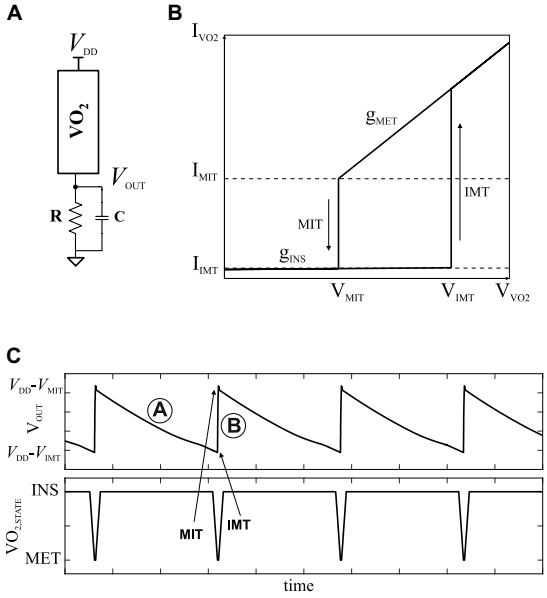


FIGURE 1
(A) I–V characteristic of the VO_2 device. **(B)** VO_2-based oscillator. **(C)** Output waveform of the oscillator including the state of the VO_2 device.

behavior. $VO_{2,STATE}$ = 'INS' means the device is in the insulating state. $VO_{2,STATE}$ = 'MET' corresponds to the device in the metallic state. Assuming that the VO_2 is in an insulating state (marked with "A" in Figure 1C), the oscillator output is discharged through the resistor. This increases the voltage drop across the VO_2 ($V_{DD} - V_{OUT}$) and so does the current through it. Once enough current density circulates, it switches to the metallic state (marked with "B" in Figure 1C). Equivalently, using the electrical model, switching to the metallic state occurs once the VO_2 voltage reaches V_{IMT}. The capacitor is then charged through the VO_2. This charging is very fast because of the low R_{MET} value. The voltage seen by the VO_2 decreases until it reaches V_{MIT} and the transition from metal-to-insulator state occurs. These nano-oscillators are attractive for their area and potential energy efficiency.

Figure 2 shows an ONN design using VO_2-based nano-oscillators as neurons and resistive couplings as synapses (Corti et al., 2018, 2020). In this work it is shown that two resistively-coupled oscillators synchronize in phase when coupling strength is high enough (resistance value low enough) and in anti-phase for large enough resistance values. That is, they proposed to use resistive coupling for both positive and negative weights. However, it is not easy to select suitable resistance values in actual applications. In fact, capacitively coupling is the easiest way to achieve anti-phase synchronization. Both types of coupling can be implemented using only resistance or capacitances with differential oscillators. A differential VO_2 oscillator has been proposed (Shamsi et al., 2021). It resorts to coupling two oscillators capacitively to force both outputs to be out of phase (180° apart). It allows for implementing both types of interactions using only capacitive or resistive coupling. This is very attractive from the point of view of implementing the coupling elements with memristor or ferroelectronic devices in crossbar architectures. On the basis of this oscillator, an ONN working as AM was shown by simulation (Shamsi et al., 2021).

In recent years, there have been numerous studies on ONNs using VO_2. Most of them report simulation results. Although in some cases experimental results are also shown (Shukla et al., 2016; Corti et al., 2018, 2020; Dutta et al., 2019), they are not implementing the design techniques that other works on electrical simulations report that allow to improve the behavior of the ONNs (Shamsi et al., 2021). Therefore, experimental validation of these approximations is necessary. In this study, we describe an ONN realized in a commercial CMOS technology in which the oscillators are built using a circuit that we have developed to emulate the VO_2 device. The purpose is to be able to study in-depth the synchronization dynamics of relaxation oscillators similar to those that can be performed with VO_2 devices. The ONN has been designed to emulate not only VO_2 devices but also fundamental characteristics of ONNs with this type of device, such as the fact that the interconnections between neurons are bidirectional. The fabricated circuit is very flexible since it allows programming the synapses to implement different ONNs, calibrating the frequency of the oscillators, or controlling their initialization. It uses differential oscillators and resistive synapses, equivalent to the use of memristors.

There are other two additional topics that must be introduced before proceeding with the CMOS ONN description.

The first one is the technique used to discretize the phase of the oscillators such that only two values are possible, and so the binary neurons of the reference HNN previously explained are reproduced. This can be achieved by Second Harmonic Injection Locking (SHIL) (Neogy and Roychowdhury, 2012). When a suitable synchronization signal, V_{SYNC}, is injected into a non-linear oscillator, SHIL occurs, and the oscillator adopts a frequency half the frequency of V_{SYNC} (f_{SYNC}) and becomes phase-synchronized within one of the two possible phases that are 180° apart. For this to occur, the natural frequency of the oscillator must be close to $f_{SYNC}/2$. SHIL is also extremely useful to stabilize the oscillator frequency against variability effects, easing the oscillators to synchronize in frequency, which is required for proper operation of the ONN.

Finally, the AM operation requires the application of an input pattern to the ONN. This is equivalent to forcing a given ONN state (a given phase pattern). In Corti et al. (2018), a method for this is presented. The authors proposed forcing a given initial state by selectively delaying the supply voltage of each neuron. For example, assuming that only binary patterns are applied, such as black and white pixel images, the initial state of the network has only two different phases, 180° apart. Those oscillators corresponding to black pixels are in one phase, and those corresponding to white ones are in the other phase. To achieve this, the black oscillators are switched on

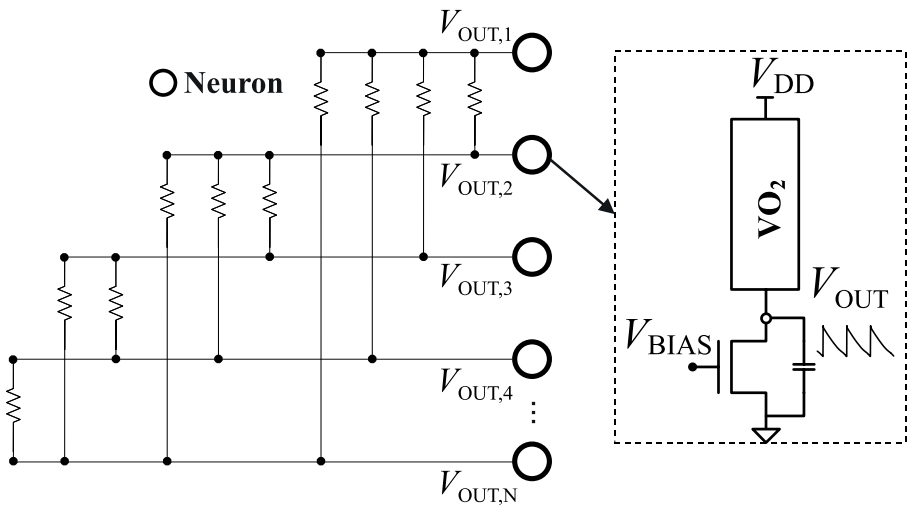


FIGURE 2
ONN design.

at T_0 and the white ones at $T_0 + T_{OSC}/2$, where T_{OSC} is the period of the oscillations.

The rest of the study is organized as follows. Section "Materials and Methods" describes in detail the designed and fabricated integrated circuit, along with the experimental setup prepared for testing it. Section 3 presents the experimental measurements and characterization of the circuit. Specifically, the ONN operation as an associative memory is demonstrated satisfactorily. The experiments carried out allow us to conclude that the ONN must be operated according to the type of computational task to be solved, and guidelines are extracted in this regard. Finally, section 4 summarizes the conclusions.

2 Materials and methods

2.1 Description of the fabricated CMOS differential ONN

An integrated circuit demonstrator of an analog 9-neuron ONN using a deep-submicron commercial CMOS technology (TSMC 65 nm – 1.2 V) has been designed, fabricated, and tested. The differential oscillators forming the neurons closely resemble those developed using VO_2 devices and previously introduced. Oscillator frequencies can be calibrated. Couplings or synapses are implemented with a four-terminal six-transistor circuit, which conductive characteristics are determined by two voltages, allowing the implementation of positive and negative weights. The ONN is fully connected and programmable. For flexibility, it can operate both with and without SHIL. Operation of the fully differential ONN described in this section was extensively validated at post-layout level simulation.

Figure 3 shows the layout of the fabricated circuit, showing the three types of circuits included (3×3 ONN, simple circuits, and differential oscillators with analog outputs) and their connections to the pad ring. The 3×3 ONN occupies a rectangle of 776 µm•747 µm with some empty area inside, and the complete chip area, including pad ring, is 1710 µm•1710 µm.

2.1.1 Differential oscillator

Each neuron consists of a differential relaxation oscillator that is formed by two single-ended oscillators whose outputs (V_{OUT1} and V_{OUT2}) are coupled by a capacitance (C_C). In turn, each of the single-ended oscillators consists of a couple of resistors (*R1* and *R2*), a P transistor, a capacitance (*C*), and a CMOS circuit that emulates the voltage–current characteristic of a VO_2, as shown in Figure 4A. The emulator (Figure 4B) has been designed using a Schmitt-Trigger inverter whose output is connected to a CMOS inverter that controls the gate voltage of an NMOS transistor (*N2*). Its drain and source terminals are the two terminals of the emulator. The input of the Schmitt-Trigger inverter is connected to the output of the oscillator. Unlike the conventional Schmitt-Trigger oscillator (Hodges and Jackson, 1983), in the proposed design, the output of the Schmitt-Trigger inverter is decoupled from the rest of the circuit, allowing its integration in complex oscillatory neural networks without penalty in energy efficiency by not having to increase its sizing. It also avoids using the floating resistor that appears in the conventional Schmitt-Trigger, whose implementation usually includes a switched capacitance and a switch. Additionally, the circuit includes the control functionality of the switching voltages V_{IMT} and V_{MIT}, with the voltage V_N on the gate of transistor *N3*, providing the proposed solution with greater flexibility as it allows to make programmable both the frequency and the amplitude of the resulting oscillator.

A step supply voltage is included for each single-ended oscillator ($V_{DD,OSC1}$ and $V_{DD,OSC2}$), through which the initial phase shift of the oscillator is controlled and, therefore, serves to establish the initial state of each neuron. Additionally, there are two inputs (V_{CTRL1} and V_{CTRL2}), whose aim is to help with the frequency synchronization of the different oscillators in case it is compromised by the inherent

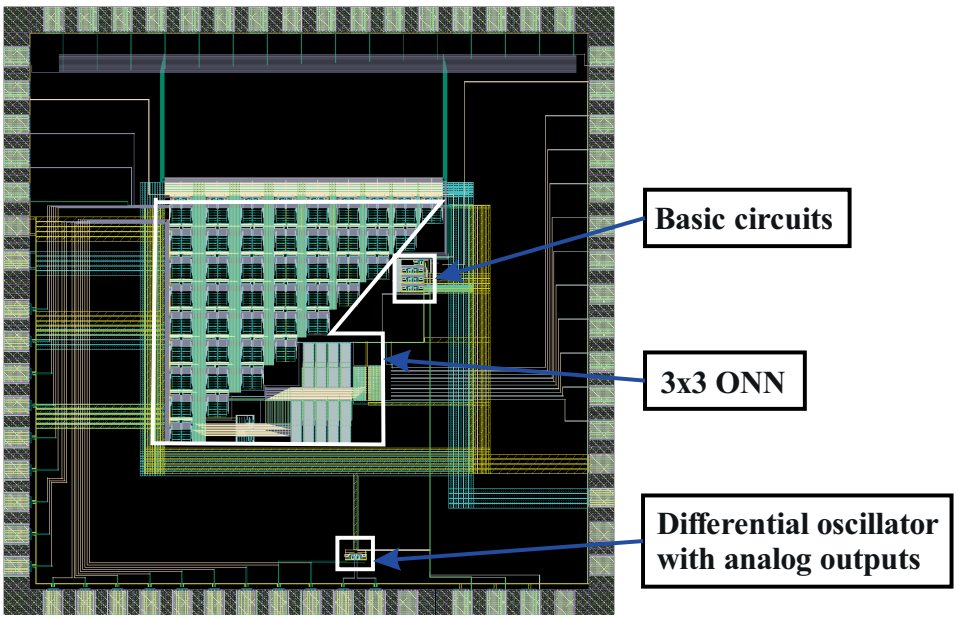


FIGURE 3
Layout of the fabricated circuit.

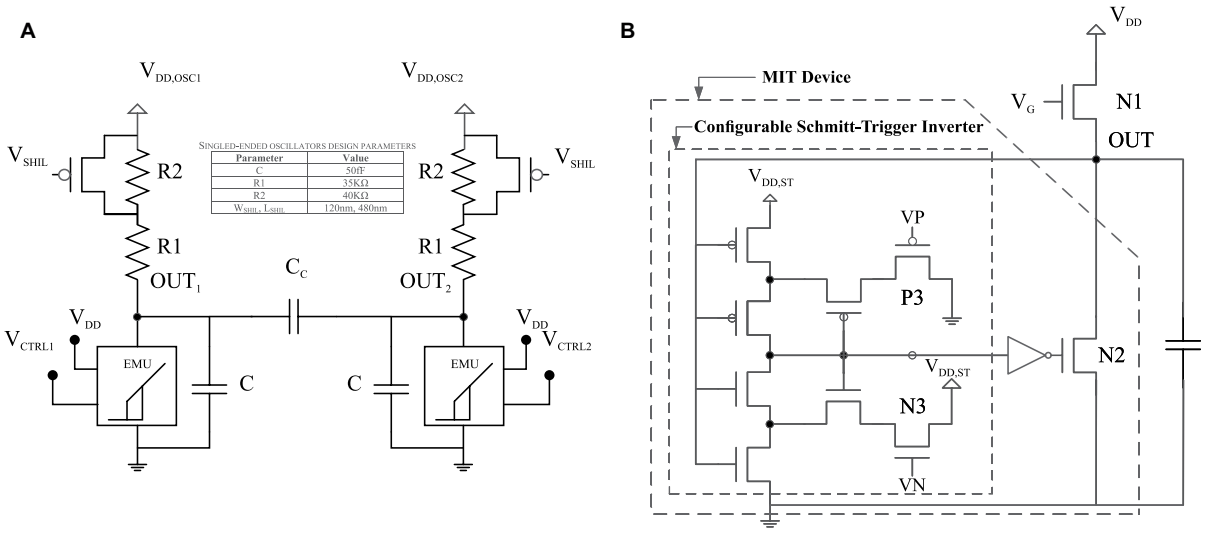


FIGURE 4
(A) Schematic of a differential oscillator. **(B)** Schematic of the circuit that emulates the behavior of the VO_2 device.

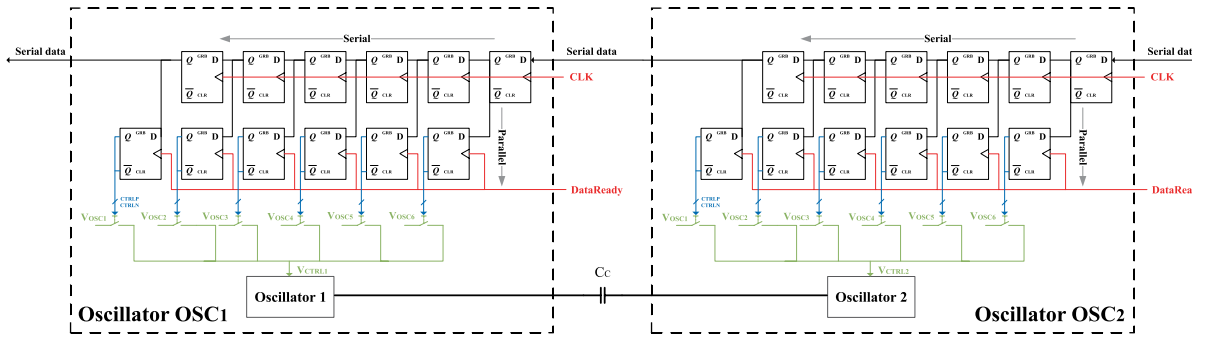


FIGURE 5
Schematic of oscillator calibration voltage selection based on series/parallel loading of the control word.

variability of the process and mismatch. Each oscillator can be connected to any of the six available calibration voltages by means of programmable switches controlled by programming registers. This scheme is illustrated in Figure 5.

As previously mentioned, a general method to improve the stability and synchronization of oscillatory neural networks is using SHIL. In this circuit, SHIL is injected through PMOS transistors, which are turned on and off through the input signal V_{SHIL}.

2.1.2 Synapse

As an analogy of the Wheatstone bridge, Figure 6A shows the schematic of the synaptic circuit, which is capable of providing positive, negative, and zero weights. Being a four-terminal circuit makes it appropriate for differential structures. Depending on the gate voltages, it is possible to have positive ($V_P > V_N$), negative ($V_N > V_P$), or zero weight ($V_P = V_N$). The two PMOS transistors are used for controlling the current between the neurons. Transistors between the positive branches can transfer current when $(V_P\text{-}V_1^+) < V_{TH}$ because V_P is applied to their gate. In addition, transistors between the negative branches can transfer current when $(V_N\text{-}V_2^+) < V_{TH}$ because V_N is applied to their gate. Therefore, the current between the neurons is controlled using these PMOS transistors. Figure 6B depicts the topology of the fabricated differential ONN.

A training rule is used to store patterns in neural networks, adjusting the synaptic weights accordingly. Once the weights are known, we propose a mapping rule to obtain the physical resistances for the memristor-bridge synapses. To store patterns in the ONN, we use the Hebbian rule to calculate the weights:

$$W_{ij} = \frac{1}{L}\sum_{k=1}^{P} b_i^k b_j^k \; i,j \in \{1,2,3,\ldots,L\} \tag{2}$$

where P is the number of stored patterns and L is the number of pixels in each pattern (which is equal to the number of neurons in the ONN). Elements b_i and b_j of all stored patterns are used to calculate the weight W_{ij}.

We propose here the following rules to map the sign and value of the above weights to the controlling voltages V_P and V_P. Weights W_{ij} are mapped to the V_P and V_P values using the following relation:

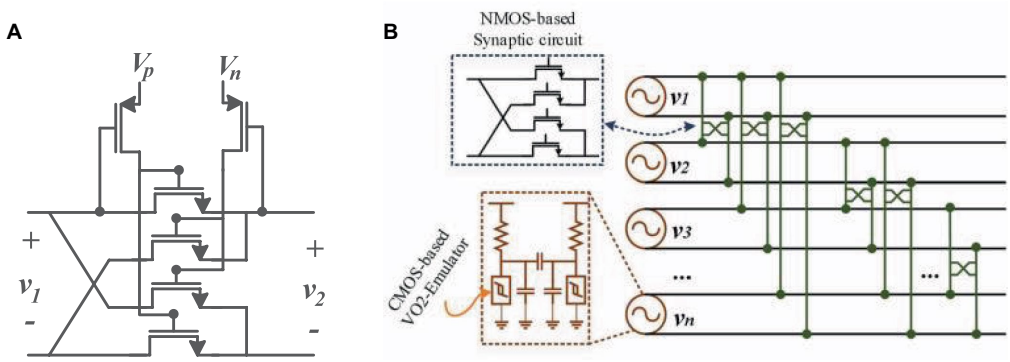


FIGURE 6
(A) Schematic of the synapsis. **(B)** Differential ONN implementation from Shamsi et al. (2021).

$$Map: \begin{cases} \alpha V_P = V_N = V_0 \; w_{ij} < 0 \\ V_P = \alpha V_N = V_0 \; w_{ij} > 0 \\ V_N = V_N = V_0 \; w_{ij} = 0 \end{cases} \quad (3)$$

where $\alpha > 1$ is a constant value. The design parameters of the synaptic circuit are W_{ij}, α, and V_0. Parameters α and V_0 will be obtained based on correctly functioning hardware.

Each synapse control voltage can be connected selectively to different voltages using programmable switches controlled by programming registers, similar to the oscillator calibration shown in Figure 5.

2.2 ASIC description

The ASIC consists of the following blocks:

- ONN of nine differential oscillators is fully interconnected with each other through 36 synapses. The ONN includes a control system from which the voltages defining the synapse weights can be selected, as well as an oscillator calibration mechanism to improve network synchronization. The oscillator outputs are connected to digital pads through Schmitt-Trigger buffers with configurable hysteresis to regenerate rail-to-rail signal swings and to cope with the waveform shape of relaxation oscillators.
- Basic circuits. Specifically, a single-ended oscillator, a differential oscillator, and two differential oscillators connected through a synapse similar to the one used in the ONN have been included. In all of them, the output is digital, as in the ONN. In addition, a differential oscillator has been included whose outputs are connected directly to analog pads in order to be able to observe the waveforms without digitizing.

2.2.1 Description of key signals/pads

The signals involved in the circuit are divided into the following categories:

- Oscillator supply voltages: Since differential oscillators are being used and there are nine oscillators in the ONN, 18 signals are required. These signals are generated externally and applied to digital pads that generate step signals between 0 V and 1.2 V. Controlling the relative timing on the initial phase selection step is essential: a delay between both signals corresponding to half a period involves applying input stimuli with opposite phases.
- Control system signals: Includes the clock signal, the signal that codifies the information to be loaded into the calibration/programming voltage selection registers, and the signal that indicates that the information has been loaded into the serial registers and serial-to-parallel conversion can be done.
- Oscillator calibration signals: These six signals can take values between 0 V and 1.2 V and allow the oscillator frequency to be tuned.
- Synapse programming signals: These 12 signals are used to set the weights for the synapses. They take values between 0 V and 1.2 V.
- Output stage configuration signals: These signals are used to set the thresholds of the ONN output Schmitt-Trigger buffer.
- Synchronization signal: It is a digital signal that ranges between 0 V and 1.2 V, with a frequency double that of the ONN's oscillators, used to enable the SHIL mechanism.

2.2.2 Control logic for calibration and programming

Each oscillator and each synapse have twice as many flip-flops as switches to be controlled. That is, 12 for each oscillator (see Figure 5) and 12 for each synapse. Half of them are configured in a single shift register, generating, therefore, a connection of $12 \bullet 9 + 12 \bullet 36 = 540$ memory elements. These registers are controlled by the clock signal. A control word is serially loaded in the shift register. It contains the calibrating and programming bits indicating which switches are closed and which are not. Obviously, for each oscillator or synapse, only one of its switches should be closed. Once the control word is fully loaded, a signal that indicates that the data are ready to be loaded is activated, and the information contained in the shift registers is loaded in parallel to the flip-flops directly controlling the switches.

2.3 Test board and experimental setup

The experimental verification of the ASIC has been performed using a custom-designed test PCB for this purpose. The block diagram of the setup and test equipment is shown in Figure 7A, together with

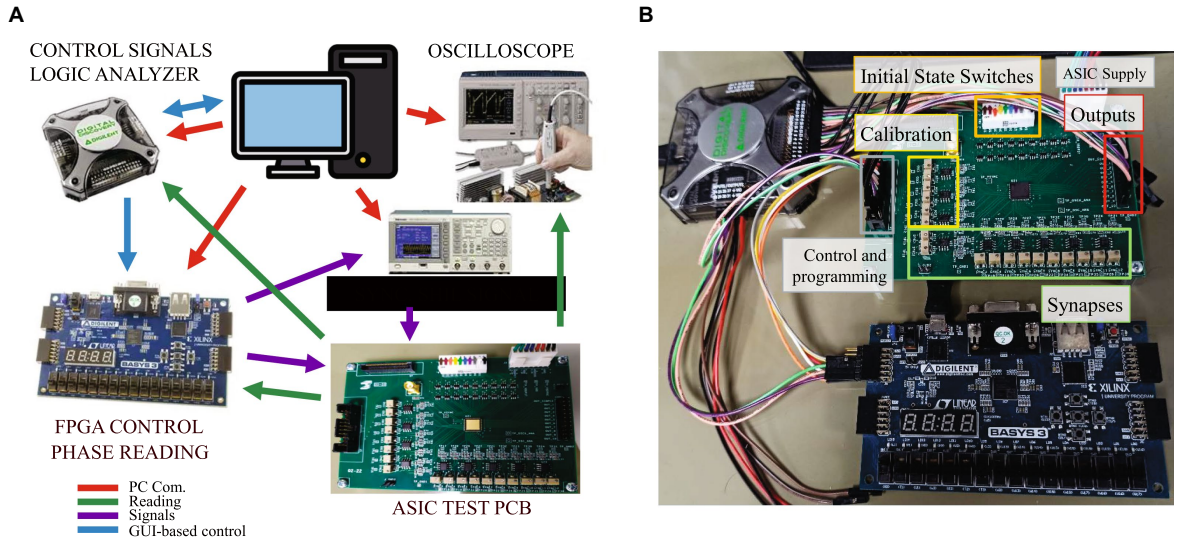


FIGURE 7
(A) Block diagram and **(B)** photograph of the test setup.

a photograph of the test PCB and its wiring in Figure 7B. The FPGA is used for programming the ONN (providing the bitstream that configures the assignment of synapses and calibration voltages) and for controlling the time-delayed power-on of the differential oscillators. A discrete micro-switch on-board allows to configure the initial state of the system, on which basis input patterns are applied to the ONN.

The main features of the PCB are summarized hereafter. First, the initialization of the differential oscillators is performed using two digital signals (common for each of them), which abruptly commute from low to high level with a delay equivalent to half an oscillator period. These signals are applied as the supply voltages of the differential oscillators. The order in which they are applied is given by the position of the micro-switch associated with each oscillator (labeled in Figure 7B as 'Initial State Switches'). These signals are generated and switched off by the custom digital design in the FPGA, commanded by START and RESET signals sent by the Digital Discovery instrument. A commercial software application linked to the instrument allows the PC user to trigger these signals.

In addition, the synchronization signal for SHIL is provided using the Tektronix AFG3102 function generator, whose output is connected to the SMA connector on the PCB, which has access to an ASIC's digital pad. Typically, the waveform used for the synchronization signal is a square signal with a voltage range between 0 V and 3.3 V. It is controlled by a trigger signal generated from the FPGA, allowing for the control of the time scheduling of both the start of the synchronization signal and the initialization of the oscillators.

The generation of the calibration and programming voltages is carried out on-board with a simple circuit consisting of an operational amplifier, a potentiometer, resistors, and capacitors. Each of the 12 programming and the six calibration voltages has one instance of this circuit dedicated, with an individual potentiometer, as can be seen in the 'Calibration' and 'Synapses' boxes in Figure 7B.

Finally, regarding output observation, digital ones are monitored using oscilloscope probes or the logic analyzer included in the Digilent Digital Discovery instrument. Furthermore, the FPGA I/O pins are compatible with reading it directly. Analog outputs can be observed using oscilloscope probes. A Keysight DSOX4104A oscilloscope has been used.

3 Results

3.1 Exploring the dynamics

The first aim of the ASIC was to be able to explore the dynamics of coupled oscillator systems. Thus, before describing its application to solve computation tasks, we report on the results of a set of experiments carried out to analyze the behavior of neurons, synapses, and the SHIL mechanism.

3.1.1 Oscillator performance

Although the ONN system has only digital outputs, simple analog oscillators were also included in the chip and connected to analog pads in order to be able to observe their behavior. Figure 8 depicts the experimental waveforms for an analog differential oscillator identical to the ones in the ONN. As expected, both outputs are 180° apart. The output average voltage ranges from 379 mV to 763 mV. Note that the small oscillation amplitude justifies the carefully designed Schmitt-Trigger-based output stage included for digitalization.

Figure 9 depicts the two outputs of one of the oscillators after digitalization and applying two calibration voltages. Figure 9A is for 1.2 V, where it can be observed that the outputs are out of phase, showing correct operation. The waveforms in Figure 9B correspond to the same experiment for a calibration voltage of 0.9 V. Note the frequency differences: by reducing the voltage applied to the calibration input, the frequency increases. The measured frequencies are 5.9 MHz and 7.18 MHz, respectively. The nine ONN oscillators have been characterized with a calibration voltage of 1.2 V. The average frequency obtained ranges between 5.5 MHz and 5.9 MHz, with a relative standard deviation between 25 m and 8 m. By varying the calibration voltage, it is possible to individually tune the frequencies of each oscillator between 6.02 MHz and 6.12 MHz, leading to a frequency difference reduction of a factor of 4.

Figure 10 depicts the obtained waveforms for the positive output of oscillator 1 with 1.2 V in the calibration voltages for different configurations of the output buffer stage. It can be observed that the

duty cycle of the digital oscillator changes. In Figure 10C, control voltages have been selected so that an undesired glitch is observed.

All these experiments have been carried out with deactivated SHIL. This is achieved by applying a constant of 3.3 V to the SHIL signal PAD.

3.1.2 Second harmonic injection locking performance

A common SHIL signal, externally provided, enables the synchrony between the oscillators when the SHIL frequency is found in a determined range related to the natural oscillator frequency. To illustrate the impact of applying SHIL, the average frequency and the deviation with and without SHIL were measured. With SHIL at 13.9 MHz, the average frequency increases to 6.95 MHz, and the relative deviation reduces to 4 m. Note that the oscillator synchronizes to half the SHIL signal as expected, and the SHIL signal helps to reduce the impact of oscillator jitter on frequency variation.

It is well-known that SHIL discretizes the oscillator phase so that only two phases can occur. These two phases are ideally 180° apart. This is shown in Figure 11. The output of two uncoupled oscillators is depicted with and without SHIL. Figures 11A,B corresponds to the case with SHIL. Since each oscillator can be in one of the two phases, they can both be either in-phase or anti-phase. We have been able to capture the two behaviors by slightly modifying the SHIL frequency. Note that the frequencies of the signals in Figures 11A,B are very close (6.769 MHz and 6.778 MHz), indicating that the modification of the SHIL signal has been minimal and yet able to introduce noise that leads to an oscillator being able to jump from one phase to the other. In no case has a situation been observed where the two oscillators have a phase difference other than 0° or 180°. By deactivating SHIL, any phase difference is possible (Figure 11C). Note that in these experiments, the two oscillators were individually calibrated to equalize their frequencies.

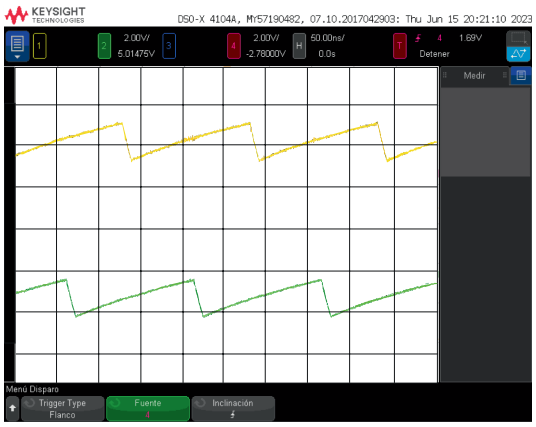


FIGURE 8
Experimental waveforms for a differential oscillator with analog outputs.

3.1.3 Synapse performance

As it was described, the implemented synapse can be programmed to enable both a positive and a negative coupling between a pair of oscillators. Positive (negative) coupling forces the two oscillators to be in phase (out of phase). Figure 12 shows the two cases. In Figure 12A, the synapse connecting the two oscillators was programmed with $V_P = 0$ V and $V_N = 0.95$ V. Figure 12B corresponds to $V_P = 0.95$ V and $V_N = 0$ V. The depicted waveforms have been obtained with SHIL activated. In this condition, the range of synapse voltages for which coupling is achieved is wide (from 0.25 V to 1.2 V). However, when there was no SHIL, this range was significantly reduced. The minimum required voltage increases to 0.95 V. Additionally, the range of valid SHIL frequencies is reduced with the synapse voltage. That is, both SHIL and coupling strength contribute to the operation of coupled oscillator systems.

To finish this first section on experimental results, we describe the behavior of three coupled oscillators with all-to-all connectivity. That is,

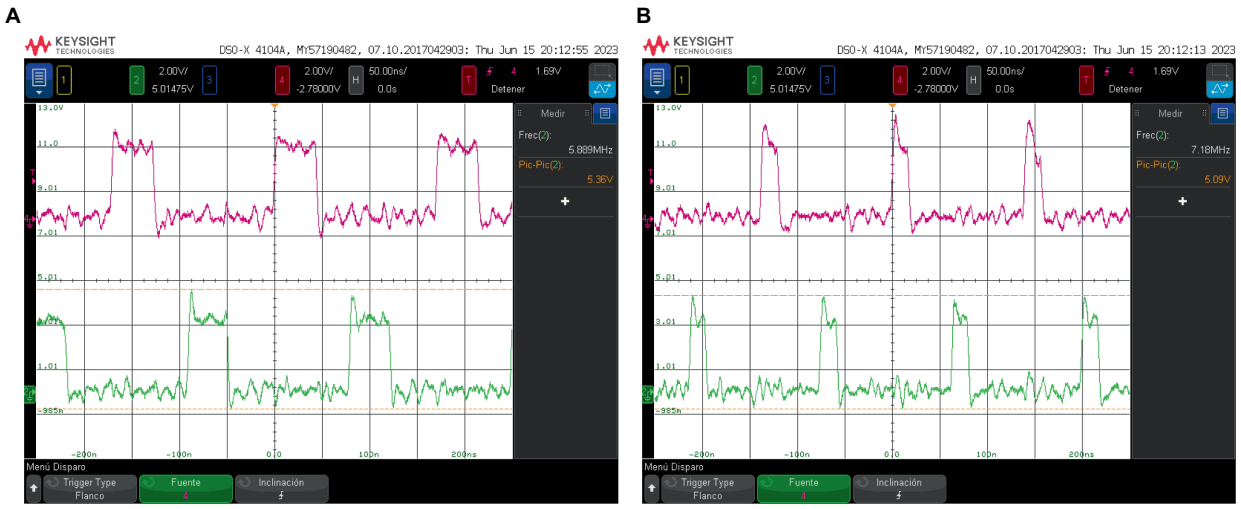


FIGURE 9
Experimental waveforms corresponding to the outputs of one of the oscillators after digitalization applying two calibration voltages: **(A)** 1.2 V and **(B)** 0.9 V.

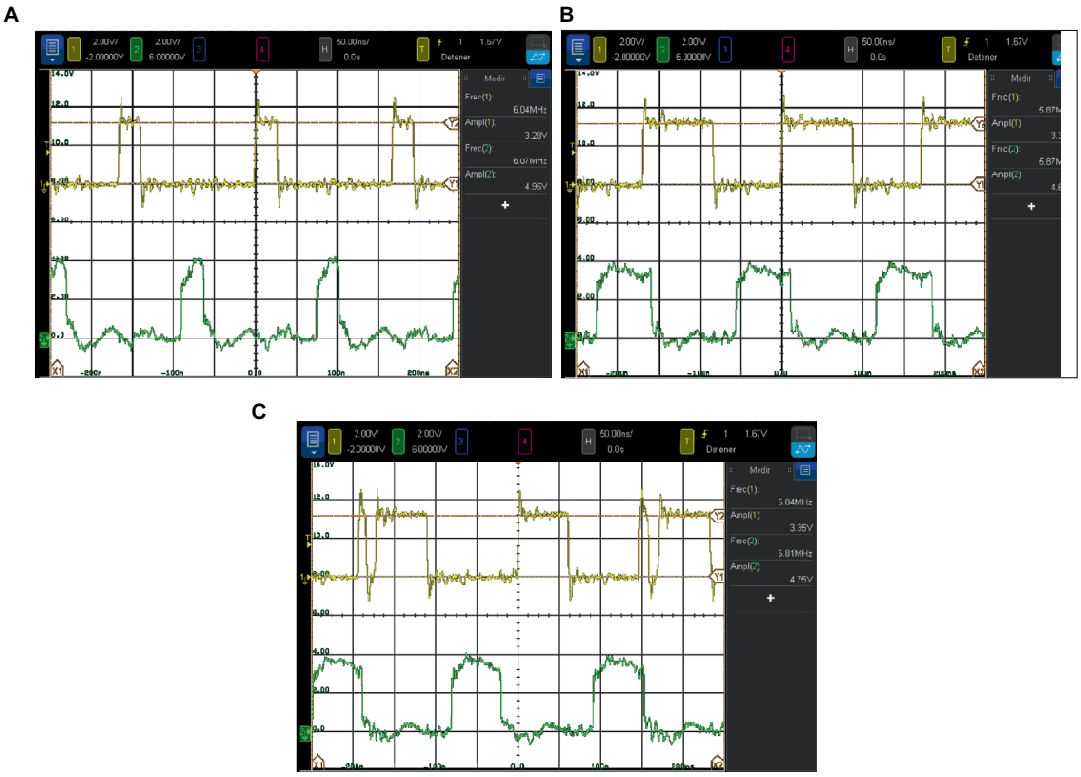


FIGURE 10
Impact of the configuration of the output buffer stage on the duty cycle of the output voltage. **(A, B)** show how the duty cycle of the output voltage varies when the output buffer configuration is modified. **(C)** shows that an undesired glitch may appear if the setting is not correct.

each of the three differential oscillators is coupled, as depicted in Figure 13A. The type of coupling is negative, and so the phases of each pair of oscillators are forced to separate from each other. We have just shown that a pair of negative-coupled oscillators evolved toward the anti-phase relationship. Clearly, when there are three connected oscillators, as in Figure 13A, it is not possible to satisfy that relationship for every pair of oscillators. It is not possible that O1 is in anti-phase with O2 and with O3, and, at the same time, O2 and O3 are also in anti-phase. It is interesting to check that our system behaves as expected. This expected behavior is completely different whether SHIL is applied or not. Assuming identical coupling, when no SHIL is applied, the three phases tend to be equally distributed (ideally to be 120° apart from each other). This is the state of the network that minimizes energy. With SHIL, since the phases are binarized, such a phase pattern is not allowed. The system tends to satisfy as many anti-phase relationships as possible. In this case, two out of the three can be satisfied.

The waveforms we experimentally obtained are depicted in Figure 13B when no SHIL was applied and in Figure 13C when SHIL was applied. The three synapses were programmed identically with $V_P = 0\,V$ and $V_N = 0.8\,V$. It can be observed that the expected behavior is obtained. Note that without SHIL, the three oscillators are not exactly 120° apart in phase. This can be due to variability in synapses, so that although the applied voltages are identical, the coupling strength can be slightly different.

It is interesting to point out that in these examples, we are in fact using physics to solve well-known computation problems. Without SHIL, the system solves the graph coloring problem (Wu et al., 2011; Parihar et al., 2017) associated with the triangle in Figure 13A. Different phases mean different colors for the nodes associated with the oscillators. With SHIL, the system obtains the Max-Cut of the corresponding graph. Nodes are split into two sets such that the number of edges between both groups is the maximum. A cut value of 2 was obtained in this case. Max-Cut is just one example of a problem that can be solved by coupled oscillator systems. A great interest has recently aroused in implementing oscillator-based Ising Machines (OIMs). OIMs efficiently solve Ising models, and there are procedures to map many hard-combinational problems into Ising models (Lucas, 2014).

3.2 ONN as associative memory

As it was described in the introductory section, an ONN can be used as an AM useful for pattern recognition applications. Unlike the graph coloring or Ising solver functionalities of the ONN described in the previous sub-section, the AM operation required applying an input pattern to the ONN. Thus, the initial phase pattern in the oscillators needs to be controlled in order to represent the input information. Corti et al. (2018) proposed to do it by controlling the timing of the power-on of each oscillator. Different works have shown that AM functionality can be achieved with this method (Núñez et al., 2021; Shamsi et al., 2021). In order to be able to use this initialization mechanism, the supply voltage of each oscillator can be independently controlled in our design, as previously described. Thus, we can test the operation of the fabricated ONN as an AM.

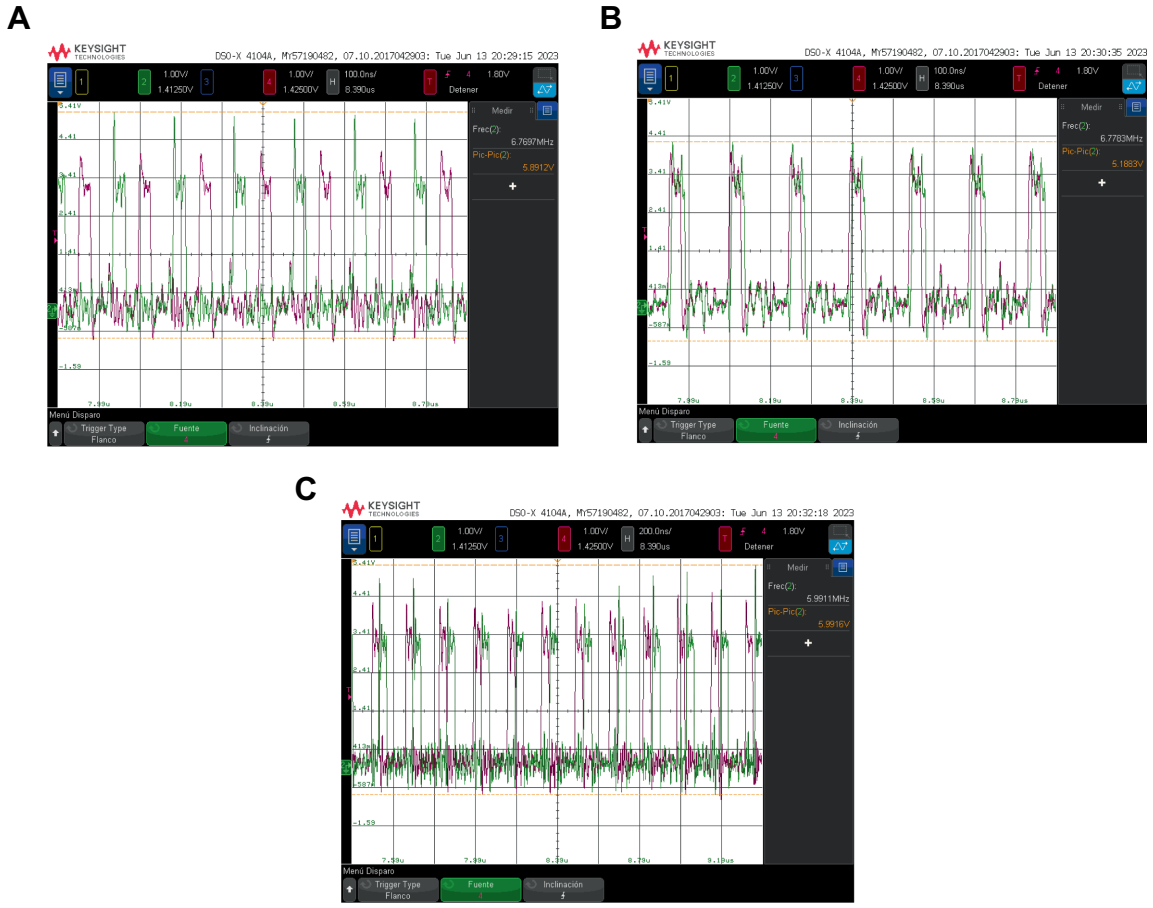


FIGURE 11
Output of two uncoupled oscillators is depicted with and without SHIL. **(A)** and **(B)** with SHIL and **(C)** without SHIL.

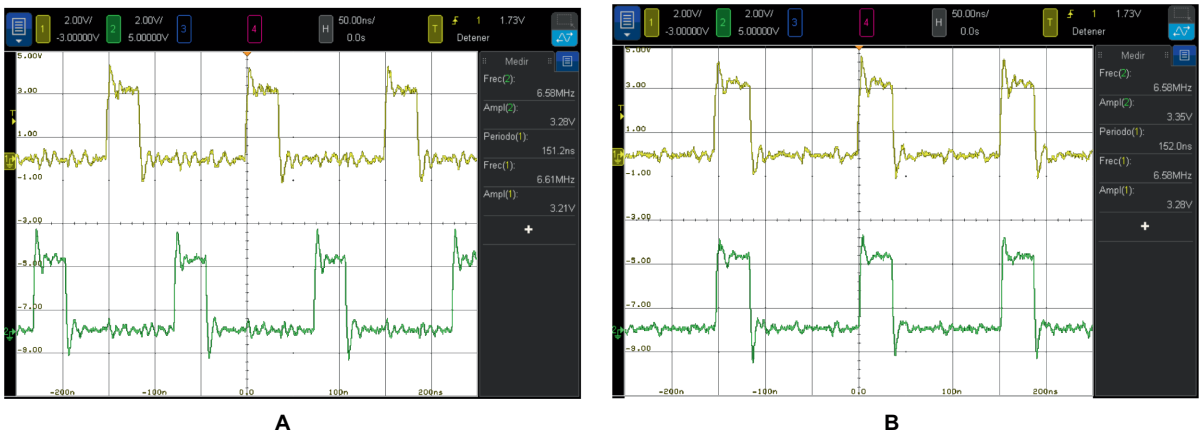


FIGURE 12
Two coupled oscillators in which the synapse is programmed with **(A)** negative coupling: $V_P = 0$ V and $V_N = 0.95$ V and **(B)** positive coupling: $V_P = 0.95$ V and $V_N = 0$ V.

Figure 14A depicts the two selected patterns to be stored, and Figure 14B shows the distorted test patterns applied to validate the AM operation. Note: We used binary patterns representing a black and white 3×3 image to approximate the pattern recognition application.

The required coupling type (positive or negative) and strength were determined for each synapse from the weight matrix obtained with Hebb's rule, and so the ONN was programmed accordingly. There were three different coupling values (one positive, one negative, and one null for uncoupling). Therefore, only two different voltages

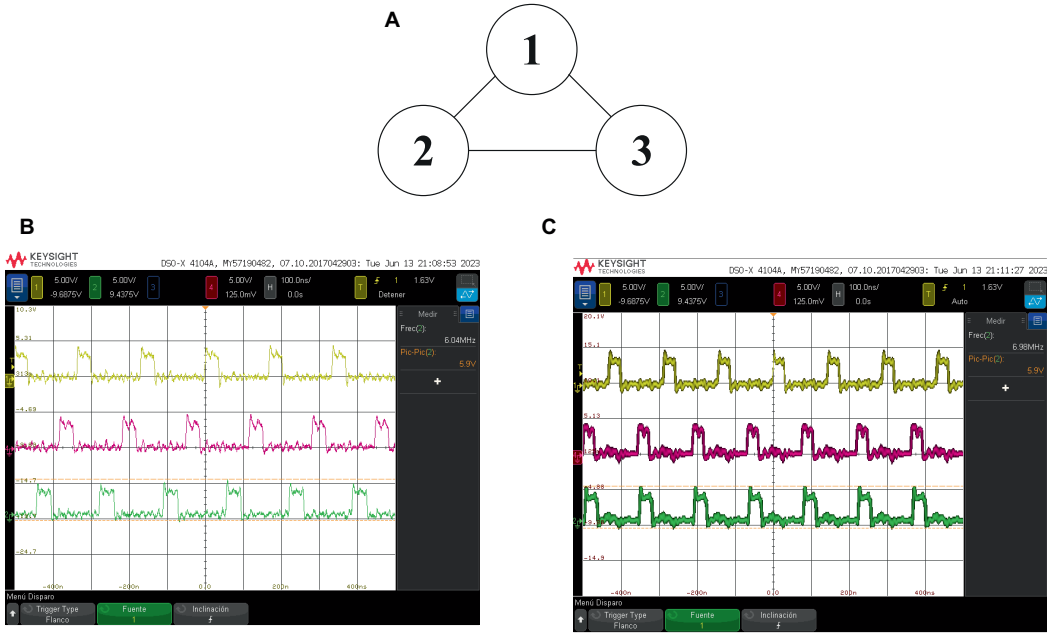


FIGURE 13
(A) Three coupled oscillators with an all-to-all connectivity. Output waveforms: **(B)** when SHIL is not applied and **(C)** when SHIL is applied.

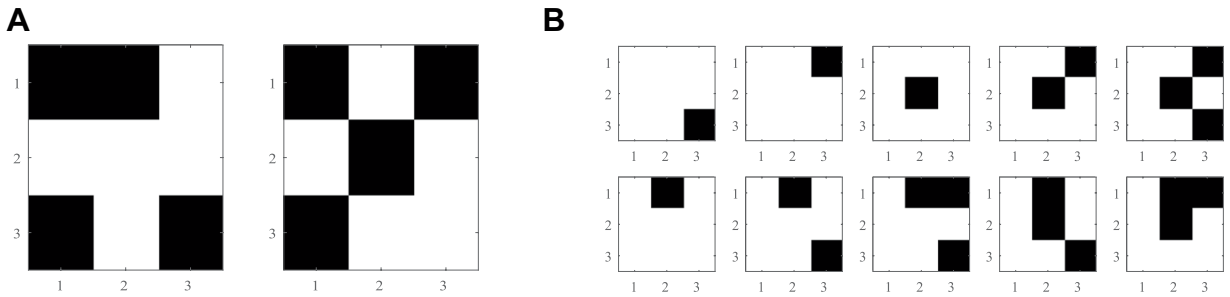


FIGURE 14
(A) Two stored patterns and **(B)** 10 test patterns selected for the measured 3×3 ONN AM.

were required for biasing the synaptic circuit. Positive (negative) couplings were coded with $V_P = 0.85\,V$ and $V_N = 0\,V$ ($V_P = 0\,V$ and $V_N = 0.85\,V$) and uncoupled with $V_P = V_N = 0\,V$. A SHIL signal of 14.3 MHz was injected.

Table 1 reports the obtained results for the two stored patterns (P1 and P2) and the 10 test patterns (T1–T10) that had been used during design for post-layout validation. For each, we indicate:

- Hamming distance to the closest store pattern (in parenthesis).
- The expected pattern is to be retrieved according to the associative memory functionality. For each test pattern, we expect to retrieve the closest stored one in terms of Hamming distance. This distance metric is the number of elements that are different between two patterns.
- Experimentally retrieved (read) pattern. A total of 100 trials were carried out for each input pattern. The number of times P1 is retrieved, followed by the number of times P2 is retrieved, is depicted. The reading operation is carried out at different time instants after the application of the test patterns, concretely at 3, 10, and 720 oscillation cycles from the beginning. Considering that the oscillation period is approximately 140 ns, these measurements correspond to 42 ns, 1.4 µs, and 100.5 µs.

The test patterns were evaluated with the mathematical HNN model obtaining the expected pattern for all of them, as well as the ONN does. Particularly, T4 is the only test pattern that did not converge the expected pattern in 90 out of 100 at the first reading at 3 cycles, but it quickly inferred and stabilized in the correct pattern from the second reading at 10 cycles, 1 µs later. Additionally, it can be observed from the third read at 720 cycles that the retrieved pattern is kept. So, it is concluded that the ONN successfully stores the two patterns. In fact, the only two stable states that were observed in all our experiments are those patterns. It has also been demonstrated that the ONN exhibits associative memory functionality. That is, it is able to retrieve a stored pattern from an applied pattern that is not a stored one.

TABLE 1 Summary of experimental results corresponding to the associative memory.

	Expected	Read (@ 3 cycles)	Read (@ 10 cycles)	Read (@ 720 cycles)
P1	P1	100/0	100/0	100/0
P2	P2	0/100	0/100	0/100
T1 (3)	P1	100/0	100/0	100/0
T2 (3)	P2	0/100	0/100	0/100
T3 (3)	P2	0/100	0/100	0/100
T4 (2)	P2	0/10	0/100	0/100
T5 (3)	P2	0/100	0/100	0/100
T6 (3)	P1	100/0	100/0	100/0
T7 (2)	P1	100/0	100/0	100/0
T8 (3)	P1	100/0	100/0	100/0
T9 (3)	P1	100/0	100/0	100/0
T10 (3)	P2	0/100	0/100	0/100

4 Discussion

A 9-neuron CMOS ONN resembling a VO_2-based ONN has been designed, fabricated, and tested. It uses a CMOS sub-circuit emulating the I–V characteristic of VO_2 devices to build differential oscillators. The synapse is implemented with a 6-transistor bridge topology, enabling resistive coupling among oscillators. Both positive and negative weights can be realized. The fabricated ASIC is programmable, with a large degree of controllability and observability to be able to dive into the dynamics of coupled non-linear oscillators, on which basis the computation is carried out.

Experiments carried out with two coupled oscillators (sub-section 3.1.3) have allowed us to experimentally sustain that both SHIL and coupling strength contribute to the synchronization of the oscillators.

The AM functionality has been demonstrated. It is important to point out that when we started to test the AM functionality, the results were not completely deterministic. That is, repeating an experiment several times led to different results. For some of the test patterns, sometimes P1 was retrieved, while for others it was P2. After carefully analyzing this behavior, we noted that conditions were not actually identical across the 100 trials since the SHIL signal was continuously running. So, the timing of oscillators power-on with respect to the SHIL phase was not fixed. We solved it by synchronizing the SHIL signal triggering and the initialization of the oscillators. Even after this modification of the experimental setup, there was still some indeterministic behavior in the system associated with input patterns equidistant (in terms of Hamming distance) to the two stored patterns. For those input patterns, sometimes P1 was retrieved while others were P2. It was due to the impact of noise.

Furthermore, it was observed that the system could evolve from one stable state to another. For example, as described in sub-section 3.1.2, a slight frequency shift of the SHIL induces noise that triggers a phase shift of an oscillator. Moreover, without any intended action on the experimental setup, and due to internal noise and other non-controllable noise sources, the phase flip can occur. It is extremely important to point out that the rate of this event is very different whether SHIL is applied or not. Under SHIL, this rate is much lower. In fact, we were not able to observe jumps from P1 to P2 in the AM with SHIL, although they occurred if SHIL was deactivated.

This observed behavior is very interesting from the point of view of the application of ONNs as Ising machines. The OIM application requires being able to escape from local energy minima. Our findings illustrate that scaping is easier in the absence of SHIL and that it can be enhanced by noise. This agrees with different works stating the importance of the SHIL signal schedule to improve the probability of exactly solving the associated Ising model. That is, obtaining a phase distribution corresponding to the minimum configuration of the Ising Hamiltonian. The next step in the exploitation of this integrated circuit is linked to the validation of the results reported in Avedillo et al. (2023), related to the resolution of combinatorial optimization problems.

Data availability statement

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

Author contributions

MJ: Writing – original draft, Writing – review & editing. JN: Writing – original draft, Writing – review & editing. JS: Writing – original draft, Writing – review & editing. BL-B: Writing – review & editing. MA: Writing – original draft, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work has been funded by Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía (BioVEO - Junta de Andalucía PAIDI 2020 - ProyExcel_00060). It has been also supported by the projects NeurONN (Horizon 2020 - Grant ID: 871501), PEROSPIKER (Horizon Europe ERC-2022-ADG - Grant ID: 101097688), and by Consejería de Transformación Económica, Industria, Conocimiento y Universidades de la Junta de Andalucía, dentro del Programa Operativo FEDER 2014-2020 (Project US-1380876).

Conflict of interest

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

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

References

Ahmed, I., Chiu, P. W., Moy, W., and Kim, C. H. (2021). A probabilistic compute fabric based on coupled ring oscillators for solving combinatorial optimization problems., IEEE J. *Solid State Circuits* 56, 2870–2880. doi: 10.1109/JSSC.2021.3062821

Avedillo, M. J., Jiménez, M., Delacour, C., Todri-Sanial, A., Linares-Barranco, B., and Núñez, J. (2023). Operating coupled VO2-based oscillators for solving Ising models. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 1:8887. doi: 10.1109/JETCAS.2023.3328887

Calayir, V., and Pileggi, L. (2013). *Fully-digital oscillatory associative memories enabled by non-volatile logic*. The 2013 international joint conference on neural networks.

Corti, E., Gotsmann, B., Moselund, K., Stolichnov, I., Ionescu, A., and Karg, S. (2018). *Resistive coupled VO_2 oscillators for image recognition, 2018*. IEEE International Conference on REBOOTING computing.

Corti, E., Khanna, A., Corti, E., Khanna, A., Niang, K., Robertson, J., et al. (2020). Time-delay encoded image recognition in a network of resistively coupled VO_2 on Si oscillators. *IEEE Electron. Device Lett.* 41, 629–632. doi: 10.1109/LED.2020.2972006

Csaba, G., and Porod, W. (2020). Coupled oscillators for computing: a review and perspective. *Appl. Phys. Rev.* 7:011302. doi: 10.1063/1.5120412

Csaba, G., Ytterdal, T., and Porod, W. (2016). Neural network based on parametrically-pumped oscillators. *IEEE Int. Conf. Electron. Circuits Syst.* 2016, 45–48. doi: 10.1109/ICECS.2016.7841128

Dutta, S., Khanna, A., Assoa, A. S., Paik, H., Schlom, D. G., Toroczkai, Z., et al. (2021). An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators. *Nat. Electron.* 4, 502–512. doi: 10.1038/s41928-021-00616-7

Dutta, S., Khanna, A., Gomez, J., Ni, K., Toroczkai, Z., and Datta, S. (2019). Experimental demonstration of phase transition Nano-oscillator based Ising machine. *IEEE Int. Electron Devices Meet.* 2019:3460. doi: 10.1109/IEDM19573.2019.8993460

Follmann, R., Macau, E. E., Rosa, E., and Piqueira, J. R. (2015). Phase oscillatory network and visual pattern recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 1539–1544. doi: 10.1109/TNNLS.2014.2345572

Goto, E. (1959). The parametron, a digital computing element which utilizes parametric oscillation. *Proc. IRE.* 47, 1304–1316. doi: 10.1109/JRPROC.1959.287195

Hodges, D. A., and Jackson, H. G. (1983). *Analysis and Design of Digital Integrated Circuits*. New York: McGraw-Hill.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational capabilities. *Proc. Natl. Acad. Sci.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hoppensteadt, F. C., and Izhikevich, E. M. (1999). Oscillatory neurocomputers with dynamic connectivity. *Phys. Rev. Lett.* 82, 2983–2986. doi: 10.1103/PhysRevLett.82.2983

Hoppensteadt, F. C., and Izhikevich, E. M. (2000). Pattern recognition via synchronization in phase-locked loop neural networks. *IEEE Trans. Neural Netw.* 11, 734–738. doi: 10.1109/72.846744

Jackson, T., Pagliarini, S., and Pileggi, L. (2018). An oscillatory neural network with programmable resistive synapses in 28 nm CMOS 2018. *IEEE Int. Conf. Rebooting Comput.* 2018, 1–7. doi: 10.1109/ICRC.2018.8638600

Kumar, A., and Mohanty, P. (2017). Autoassociative memory and pattern recognition in micromechanical oscillator network. *Sci. Rep.* 7:411. doi: 10.1038/s41598-017-00442-y

Lucas, A. (2014). Ising formulations of many NP problems. *Front. Phys.* 2:2014. doi: 10.3389/fphy.2014.00005

Maffezzoni, P., Daniel, L., Shukla, N., Datta, S., and Raychowdhury, A. (2015). Modeling and simulation of vanadium dioxide relaxation oscillators. *IEEE Trans. Circuits I* 62, 2207–2215. doi: 10.1109/TCSI.2015.2452332

Moy, W., Ahmed, I., Chiu, P. W., Moy, J., Sapatnekar, S. S., and Kim, C. H. (2022). A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nat. Electron.* 5, 310–317. doi: 10.1038/s41928-022-00749-3

Neogy, A., and Roychowdhury, J. (2012). Analysis and Design of sub-harmonically Injection Locked Oscillators. *Proc. IEEE DATE* 2018:677. doi: 10.1109/DATE.2012.6176677

Nikonov, D. E., Csaba, G., Porod, W., Shibata, T., Voils, D., Hammerstrom, D., et al. (2015). Coupled-oscillator associative memory Array operation for pattern recognition. *IEEE J. Explor. Solid State Comput. Dev. Circuits* 1, 85–93. doi: 10.1109/JXCDC.2015.2504049

Núñez, J., Avedillo, M. J., Jiménez, M., Quintana, J. M., Todri-Sanial, A., Corti, E., et al. (2021). Oscillatory neural networks using VO_2 based phase encoded logic. *Front. Neurosci.* 15:655823. doi: 10.3389/fnins.2021.655823

Parihar, A., Shukla, N., Datta, S., and Raychowdhury, A. (2015). Synchronization of pairwise-coupled, identical, relaxation oscillators based on metal-insulator phase transition devices: a model study. *J. Appl. Phys.* 117:783. doi: 10.1063/1.4906783

Parihar, A., Shukla, N., Jerry, M., Datta, S., and Raychowdhury, A. (2017). Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Sci. Rep.* 7:911. doi: 10.1038/s41598-017-00825-1

Popescu, B., Csaba, G., Popescu, D., Fallahpour, A. H., Lugli, P., Porod, W., et al. (2018). Simulation of coupled spin torque oscillators for pattern recognition. *J. Appl. Phys.* 124:152128. doi: 10.1063/1.5042423

Raychowdhury, A., Parihar, A., Smith, G. H., Narayanan, V., Csaba, G., Jerry, M., et al. (2019). Computing with networks of oscillatory dynamical systems. *Proc. IEEE* 107, 73–89. doi: 10.1109/JPROC.2018.2878854

Shamsi, J., Avedillo, M. J., Serrano-Gotarredona, T., and Linares, B. (2021). Hardware implementation of differential oscillatory neural networks using VO_2-based oscillators and Memristor-bridge circuits. *Front. Neurosci.* 15:674567. doi: 10.3389/fnins.2021.674567

Sharma, A., Bain, J. A., and Weldon, J. A. (2015). Phase coupling and control of oxide-based oscillators for neuromorphic computing. *IEEE J. Explor. Solid State Comput. Dev. Circuits* 1, 58–66. doi: 10.1109/JXCDC.2015.2448417

Shukla, N., Datta, S., Parihar, A., and Raychowdhury, A. (2016). "Computing with coupled relaxation oscillators" in *Future trends in microelectronics: Journey into the unknown*. eds. S. Luryi, J. Xu and A. Zaslavsky (Hoboken, NJ: Wiley-IEEE Press), 147–156.

Von Neumann, J. (1957). *Non-linear capacitance or inductance switching, amplifying and memory devices*. U.S. patent no 2,815,488. Washington, DC: U.S. patent and trademark office.

Wu, J., Jiao, L., Li, R., and Chen, W. (2011). Clustering dynamics of nonlinear oscillator network: application to graph coloring problem. *Physica D* 240, 1972–1978. doi: 10.1016/j.physd.2011.09.010

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 04 March 2024
DOI 10.3389/fnins.2024.1307525

OPEN ACCESS

EDITED BY
Pablo Varona,
Autonomous University of Madrid, Spain

REVIEWED BY
Yan Fang,
Kennesaw State University, United States
Nicole Sandra-Yaffa Dumont,
University of Waterloo, Canada

*CORRESPONDENCE
Gyorgy Csaba
gcsaba@gmail.com

RECEIVED 04 October 2023
ACCEPTED 12 February 2024
PUBLISHED 04 March 2024

CITATION
Rudner T, Porod W and Csaba G (2024) Design of oscillatory neural networks by machine learning. *Front. Neurosci.* 18:1307525. doi: 10.3389/fnins.2024.1307525

Design of oscillatory neural networks by machine learning

Tamás Rudner[1], Wolfgang Porod[2] and Gyorgy Csaba[1]*

[1]Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary, [2]Department of Electrical Engineering, University of Notre Dame (NDnano), Notre Dame, IN, United States

We demonstrate the utility of machine learning algorithms for the design of oscillatory neural networks (ONNs). After constructing a circuit model of the oscillators in a machine-learning-enabled simulator and performing *Backpropagation through time* (BPTT) for determining the coupling resistances between the ring oscillators, we demonstrate the design of associative memories and multi-layered ONN classifiers. The machine-learning-designed ONNs show superior performance compared to other design methods (such as Hebbian learning), and they also enable significant simplifications in the circuit topology. We also demonstrate the design of multi-layered ONNs that show superior performance compared to single-layer ones. We argue that machine learning can be a valuable tool to unlock the true computing potential of ONNs hardware.

KEYWORDS

neuromorphic computing, oscillatory neural networks, machine learning design, ring oscillators, low-power computing

1 Introduction

The computing power of neuromorphic and artificial intelligence (AI) algorithms is greatly limited by the lack of low-power, energy-efficient hardware to run AI computing tasks. Outsourcing even the simplest AI processing primitives (such as pattern classification) to energy-efficient, specific-purpose hardware would greatly increase the prevalence and computational power of AI algorithms.

Neuromorphic analog computing elements are currently being intensely researched, as they promise significant energy savings in artificial intelligence (AI) computing tasks compared to their digital counterparts (Schuman et al., 2017). Among the many flavors of analog computing, oscillatory neural networks (ONNs) received special attention (Csaba and Porod, 2020a). This is due to the facts that (1) ONNs are realizable by very simple circuits, either by emerging devices or conventional transistor-based devices, (2) phases and frequencies enable a rich and robust (Csaba and Porod, 2020a) representation of information, and (3) biological systems seem to use oscillators to process information (Furber and Temple, 2007), likely for a reason.

Despite the significant current research efforts and the large literature, most ONNs seem to rely on some version of a Hebbian rule to define attractor states for the oscillator phases (Delacour and Todri-Sanial, 2021). The Hebbian rule is used to calculate the value of physical couplings between oscillators—such as resistances or capacitances—that define the circuit function. The reliance on the Hebbian rule turns most current ONNs into a sub-class of classical Hopfield networks, which are not very powerful by today's standards. While there are a few ONN implementations not relying on basic Hebbian rules (notably Vassilieva et al., 2011), it is likely that current ONNs do not fully exploit the potential of the hardware—due to the lack of a more powerful method to design the interconnections.

In this study, we show, using computer simulations, that a state-of-art machine learning method, namely Backpropagation Through Time (BPTT), when applied to a circuit-level model of the ONN, significantly enhances the computational power of ONNs. Our studied system is an ONN made of resistively coupled ring oscillators (Csaba et al., 2016; Moy et al., 2022), and its circuit topology is described in Section 2.1. Next, in Section 2.2 we develop the differential equations describing the circuit and show how a machine learning algorithm can be applied to design the circuit parameters. In Section 3.1, we apply the machine-learning framework for the design of an auto-associative memory and compare it to a standard Hebbian rule-based device. Section 3.1.3 furthers this concept by the design of a multi-layered network, which is a two-layer classifier and achieves superb performance compared to a single-layer device.

An AI processing pipeline typically has to process a large amount of input sensory data (such as audio, video, or text streams). These operations consume significant power, due to the sheer amount of sensory data. The ring oscillator-based ONN present here can do classification tasks in an energy-efficient way, and this way significantly increase the net power efficiency of the computing pipeline.

Overall, our study presents a design methodology that unlocks the true potential of oscillatory neural networks, overcoming the limitations imposed by simple learning rules. Additionally, the presented method allows for designing physically realizable structures: our networks rely on nearest-neighbor interactions, which is amenable to scaling, chip-scale realizations and uses significantly fewer neurons than fully connected networks.

2 Materials and methods

2.1 Resistively coupled ring oscillators for phase-based neuromorphic computation

It is well-established that the synchronization patterns of coupled oscillators may be used for computation (Csaba and Porod, 2020a). The idea of using phase for Boolean computation goes back to the early days of computer science (Wigington, 1959) and is being rediscovered these days (Roychowdhury, 2015). For neuromorphic computing, the original scheme of Izhikevich (Hoppensteadt and Izhikevich, 1999, 2000) was studied using various oscillator types and coupling schemes. A number of computing models were explored, ranging from basic convolvers (Nikonov et al., 2015) and pattern generators (Dutta et al., 2019) to hardware for handling NP-hard problems (Chai Wah Wu, 1998; Parihar et al., 2017; Moy et al., 2022).

Ring oscillators are among the simplest of oscillators. These devices consists only of (odd number of) inverters, capacitances, and resistances, see in Figure 1.

To give a simple example of how ring oscillators compute in phase space, Figure 1 shows a two-oscillator system. Nodes that are interconnected by a resistor will synchronize in phase. If identical nodes (say V_3, the 3rd voltage node of the ring oscillators) are interconnected, the oscillators will run in phase. However, in a 7-inverter ring oscillators, each node is phase-shifted by an angle of $2\pi/7$ with respect to their neighbor. If, say, V_3 of one oscillator is connected to say V_6 of the oscillators, the oscillators will pull toward an anti-phase configuration. The waveforms of these two cases are illustrated in the top part of Figure 2.

A larger network of oscillators with in-phase or out-of-phase pulling resistors will converge toward an oscillatory ground state configuration, which in fact maps to the solution of the Ising problem (Moy et al., 2022). Simply put, the phase of each oscillator will converge toward a value that optimally agrees to most of the constraints imposed on the oscillator by other oscillators it is coupled to. The dynamics of the coupled oscillator network will approximate the solution of a computationally hard optimization problem. For an Ising problem, the oscillator oscillator couplings are part of the problem description, and there is no need to calculate them.

While the Ising problem is important and shows the computational power of ONNs, an Ising solver alone is not very useful for solving most real-life, neuromorphic computing tasks. A neuromorphic computing primitive (such as a classification task) does not straightforwardly map to an Ising problem. So, if the oscillator network is to be used as a neuromorphic hardware, then the oscillator weights must be designed or trained to perform certain computational functions.

Most ONNs are used as auto-associative memories, making them applicable for simple pattern recognition/classification tasks. The weights are designed based on the Hebbian learning rule (Csaba et al., 2016; Delacour and Todri-Sanial, 2021), and this is one of the cases when the Ising model easily maps to a neuromorphic computing model. In fact, the connection between Ising and Hopfield's associative models (Hopfield, 1982; Michel et al., 1989; Smith, 1999) was designed by Hopfield early on Hopfield and Tank (1985). ONNs simply use oscillator phases as the state variable of Hopfield neurons.

The Hebbian rule (and even its improved variants Righetti et al., 2006; Tolmachev and Manton, 2020) has severe limitations: the rule works best on all-to-all oscillator (neural) connections and it does not trivially support learning on a set of training examples. In addition, simple Hopfield models are not very powerful neural networks by today's standards—for example, a Hebbian-trained Hopfield network achieves mediocre results in the standard MNIST classification tasks (Belyaev and Velichko, 2020). This is why our goal in this study is to go beyond these limitations and apply state-of-the-art-learning techniques to train ONN weights. This allows us to overcome the limitations of associative (Hopfield) type models and design ONN versions of many other neural network models.

2.2 Machine learning framework for circuit dynamics

Our methodology is to apply Backpropagation Through Time (BPTT) (Werbos, 1990) to an in-silico model of the oscillators. We constructed a circuit model of the coupled oscillator system; the resulting ODEs are solved and the value of the loss function is calculated at the end of the procedure. By backpropagating the error, we can optimize the circuit parameters in such a way that the ONN solves the computational task defined by the loss function. Once the circuit parameters are determined via this algorithm, they

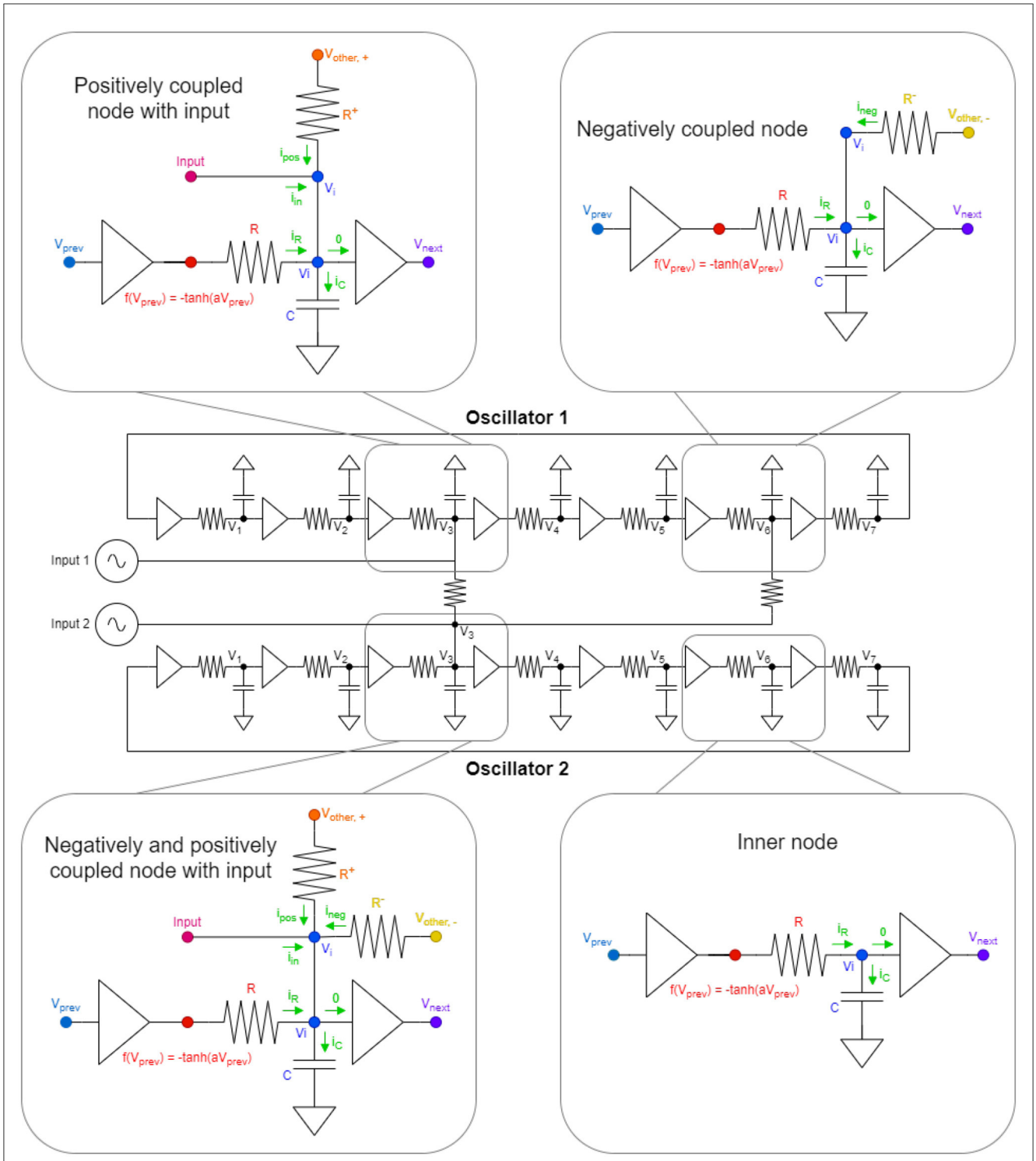


FIGURE 1
Here, in the middle, we can see a two oscillator system, coupled together in every possible way with numbered voltage nodes. The oscillators are built using seven inverters. The zoomed-in parts at the four corners are the different node types we can have in a given system. In those four schematics, the same colored nodes are at the same voltage levels and v_{prev} and v_{next} are used as the node before v_i and node after v_i, respectively, because of the circular design. The green arrows symbolize the currents flowing into the given node v_i. The 0-labeled current is a symbol for 0 current, as by definition, the input current to inverters is 0. In addition, the $v_{other,-}$ and $v_{other,+}$ is a symbol for the voltage node of another oscillator, which is coupled to the particular oscillator negatively and positively, respectively. The *Input*-labeled waveform generators can be anything feeding information into the system as external input currents.

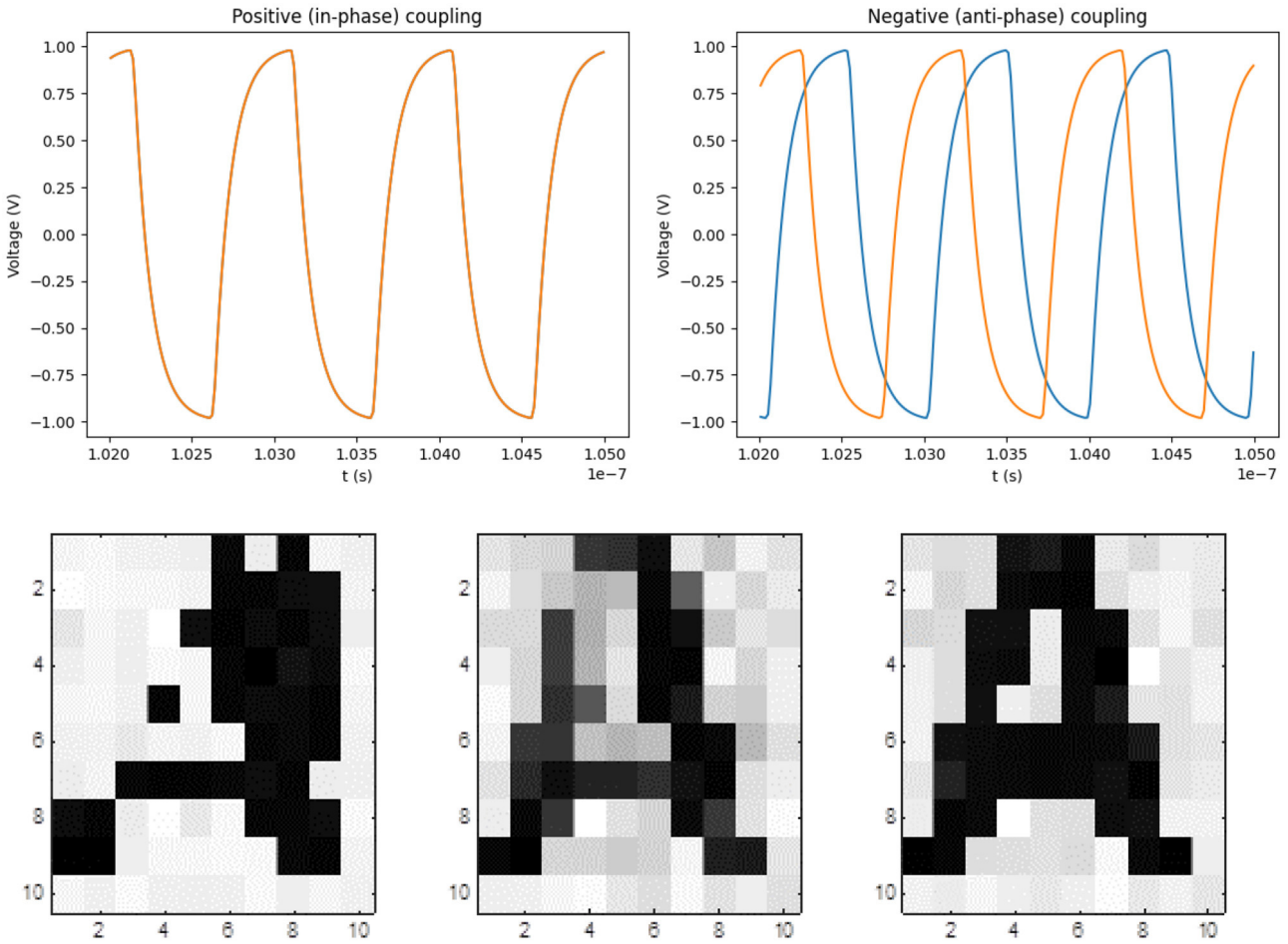


FIGURE 2
Phase-based computing by two ring oscillators: (**top left**) If R^- (or the resistance between different voltage nodes of two oscillators, say V_3 and V_6) dominates in the coupling, the oscillators run in phase, while (**top right**) if R^+ (or the resistance between same voltage nodes of two oscillators, say V_3) dominates, then anti-phase coupling is realised. The latter two are based on the criteria that coupling is realised by flowing currents and the larger the current is, the more influence the oscillators will have on one another. (**Bottom**) If phases correspond to pixels of a grayscale image, the phase dynamics may be used to converge to predefined patterns (Csaba et al., 2016). The illustrations of convergence to 'A' are taken from Csaba and Porod (2020b).

can be 'hard-wired' into a circuit (ONN hardware) for an effective hardware accelerator tool.

2.2.1 Computational model of resistively coupled ring oscillators

For the sake of concreteness, we assume that our circuit consists of n oscillators and each oscillator is composed of seven inverters. The circuit has k input nodes. We construct a simple ordinary differential equation (ODE)-based circuit model based on the equations derived by Lai and Roychowdhury (2005).

Each inverter is described on a behavioral level by a $f(x) = -\tanh(ax)$ non-linearity connected to an RC delay element. This way, a seven-inverter ring oscillator is modeled by seven first-order non-linear ODEs.

The mathematical formulation consists of three parts: internal dynamics of the oscillators (due to the inverters), dynamics due to external signals (inputs), and the coupling's dynamics.

In Figure 1, there can be seen the basis of the derivation of the ODE of the circuit model. There are four types of nodes in the system and for each of them, an ordinary, first-order differential equation can be derived using Kirchoff's current law as follows:

- Most nodes are inner-nodes (bottom right part in Figure 1) in the oscillators (5 in each) and their equation is rather easy to calculate:

$$C\frac{dv_i}{dt} = \frac{f(v_{prev}) - v_i}{R}$$

- There can also be negatively coupled nodes (top right part on Figure 1), which are a little bit more complex than the simple inner nodes. It also has another current component flowing to v_i, which is coming from the difference of the voltage of a different node of another oscillator and the voltage of the particular oscillator divided by the resistance between the nodes. Here, the requirement for negative coupling is that the two coupling nodes should be an odd even pair in

terms of numbering of voltage nodes. Here, the equation is the following:

$$C\frac{dv_i}{dt} = \frac{f(v_{prev}) - v_i}{R} + \frac{v_{other,-} - v_i}{R^-}.$$

- There can be positively coupled nodes with inputs (top left part on Figure 1). Positively coupled nodes are more complex than the negative coupled nodes previously described, as it not only has an incoming current from a different oscillator but also has an external input indicated by the waveform generators on Figure 1. Note that the requirement for positive coupling between the two oscillators is to have an even even or odd odd pairing of oscillators. The particular ODE for this kind of arrangement is as follows:

$$C\frac{dv_i}{dt} = \frac{f(v_{prev}) - v_i}{R} + \frac{v_{other,+} - v_i}{R^+} + Bu_{in},$$

  where B effectively controls the amplitude of the input waveform. It is worth mentioning that input is not necessarily present for this node, so it is possible that a node only has extra current coming from positive couplings without any kind of external input.
- The most complicated node is the one having positive and negative couplings and also some input (bottom left part on Figure 1). It is basically the merger of the previous two items which is manifested in the equations as well:

$$C\frac{dv_i}{dt} = \frac{f(v_{prev}) - v_i}{R} + \frac{v_{other,-} - v_i}{R^-} + \frac{v_{other,+} - v_i}{R^+} + Bu_{in}.$$

Combining the previously presented knowledge, for the whole system, we can arrive at the following ODE for the collection of voltages at all the nodes, which describes all parts if we write a differential equation for every node in the system using Kirchhoff's current law and assuming only resistors as couplings:

$$\frac{dV}{dt} = \frac{1}{RC}\Big(f\big(\mathbf{P}_\pi V\big) - V\Big) + \frac{1}{C}\mathbf{B}'u + \frac{1}{R_cC}\mathbf{C}'V,$$

where

$$f(x) = -\tanh(ax),$$

is the simplified characteristic of an inverter with some $a \in \mathbb{R}$. Furthermore, π is a permutation, such that

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

and $\mathbf{P} \in \mathbb{R}^{(7n)\times(7n)}$ is a block matrix in which for every $7x7$ matrix block in the main diagonal there is a permutation matrix corresponding to π. This orders the voltage nodes in the ring oscillator to calculate the voltage differences arising between the two endpoints of the resistors placed in between the two inverters. $\mathbf{B}' \in \mathbb{R}^{(7n)\times k}$ is the connector matrix for the inputs. The inputs are collected in $u \in \mathbb{R}^k$. $\mathbf{C}' \in \mathbb{R}^{(7n)\times(7n)}$ is the modified couplings matrix which is to be constructed from the real, humanly readable couplings matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$. The parameters R, $C \in \mathbb{R}^+$ are fixed for the oscillators; meanwhile, the $R_c \in \mathbb{R}^+$ coupling parameters are one of the two real, to-be-learnt parameters of the system that govern the whole coupling dynamics. The other ones are the parameters gathered in $\mathbf{B}'$, which directly relates to the amplitude of the input signal (typically a sinusoidal current generator).

The $\mathbf{C}_{i,j}$ is related to the couplings between oscillators i and j and the matrix is built the following way:

- All main diagonal entries are 0, as no oscillator is coupled to itself.
- All entries in the upper triangle of the matrix are corresponding to the positive (in-phase-pulling) couplings.
- All entries in the lower triangle of the matrix are corresponding to the negative (anti-phase-pulling) couplings.

The construction of $\mathbf{C}'$ can be done easily from C algorithmically. As every positive coupling is between 3-3 nodes of the oscillators and every negative connection is between 3-6 nodes of oscillators, the $\mathbf{C}'$ matrix is quite sparse. Similarly, because inputs are only fed into the 3rd node of every oscillator, the $\mathbf{B}'$ matrix is sparse.

The ODEs are constructed for the circuit of Figure 3, in case of a fully connected ONN. The oscillators are driven by sinusoidal current generators, and the phase of these signals carries the input. They define the initial states of the oscillators that is later changed by the couplings between the oscillators.

Each oscillator is connected by two resistors, the value of which has to be learned. The values of the coupling resistors are inversely related to the coupling parameters, which are stored in the $\mathbf{C}$ coupling matrix and the elements of this matrix are to be learned. In the equations above, R_c is a predefined, constant value which is the resistance scaling factor between two coupled nodes, usually around 10 $k\Omega$. The system learns the values in $\mathbf{C}$. From this matrix, the $\mathbf{C}'$ modified coupling matrix is built. The real physical coupling resistances' values between nodes i and j is given by $\frac{R_c}{\mathbf{C}_{i,j}}$.

Similarly, the values in $\mathbf{B}'$ are related to the input current generator's amplitude, but they are directly proportional to the real amplitude of input generators.

In the examples of the later sections, the grayscale pixel colors will typically correspond to the input phases of the current generators, and a pixel intensity from 0 to 1 is mapped to phases $\phi \in [0, \pi]$. Similarly, the output pattern is the stable, stationary phase pattern of the oscillators.

The circuit model we use (Lai and Roychowdhury, 2005) is simpler than a SPICE-level (Simulation Program with Integrated Circuit Emphasis) circuit model, as it takes into account the transistor characteristics by a behavioral curve. The internals of the MOS transistors are neglected. This simplification is done to facilitate learning as we will explain below.

2.2.2 Backpropagation for ONN circuit design

Backpropagation is the de facto standard algorithm used for the training of neural networks (LeCun et al., 2015). After each run of the neural network, the gradient of a properly defined loss function is computed with respect to the trainable parameters of the system, in an efficient manner.

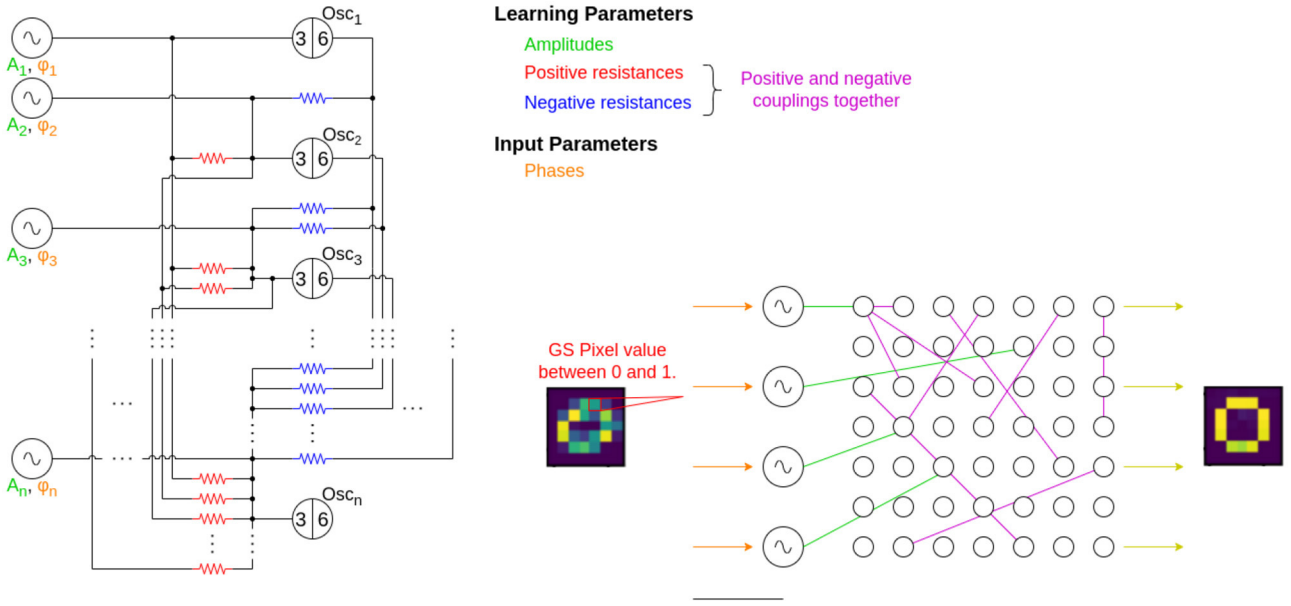


FIGURE 3
The circuit diagram of the entire computational layer. Input signal generators provide the sinusoidal signals with a phase that corresponds to an input pattern, such as pixels of an image. These generators are connected to the computing oscillators, whose phase pattern provides the solution to the problem. The 3–6 marks on the oscillators indicate the 3rd and 6th nodes in the ring oscillators' circuit. The green, red, and blue colored circuit elements' values are learned during the learning process and the phases, indicated with orange are the inputs. On the schematic figure, the purple connections indicate both the positive (red) and negative (purple) couplings. The grayscale pixel value is read from the image, converted to phase information, then the sinusoidal current generators are connected to the oscillators one-by-one. The yellow arrows show that the output is read from the oscillators and an image is formed.

BPTT (Backpropagation Through Time) is backpropagation applied to a dynamic system (i.e., an ODE-based description). The ODE is solved by a standard time-stepping technique, using discrete time. This dicretized solution may be viewed as a many-layer neural network such that one neural layer corresponds to a temporal snapshot of the system dynamics. The BPTT algorithm calculates and stores these layers (snapshots) in the forward pass of the calculation, then calculates the derivatives of an objective function with respect to trainable parameters in the backward pass.

To apply backpropagation or BPTT, a loss function (objective) function has to be defined, and this assumes a minimum value when the system is in the desired, computational state. The loss function is typically defined on the end state of the ODEs; in our case, this is the the stationary phase of oscillators at the end of the computation.

In this study, we apply BPTT to find out the circuit parameters that enable the ONN to perform useful computation. After the calculation of the gradient, a gradient descent method is used for learning, in order to minimize the loss function. Each gradient descent steps should bring the circuit parameters closer to their optimal value.

We have written our simulation code in Pytorch (Paszke et al., 2017)—the autograd feature of Pytorch makes the implementation of backpropagation and BPTT straightforward. We also used the *torchdiffeq* (Chen, 2018) package for implementing backward-differentiable ODE solvers. This external, third-party library is built upon PyTorch and provides various differentiable ODE solvers implemented for PyTorch. A particularly useful feature of *torchdiffeq* is that it can apply the adjoint method for the backward step (Chen et al., 2018) and calculate the gradients with a constant memory cost.

It must be noted that BPTT is computationally demanding for a complex dynamic system such as our ONN. The time-domain solution of a circuit model typically consists of thousands of time steps. As BPTT works by unwrapping the time-domain solution of an ODE to a many-layer neural network, the BPTT algorithm must handle a many-thousand layer network and this may yield to memory bottlenecks during the training.

Backpropagation through many layers inevitably suffers from the vanishing gradient problem (Lillicrap and Santoro, 2019). We found that our algorithm produces useful gradients up to a few thousand time steps (layers). The ONN is constructed to safely converge within this time frame.

The high computational demand of BPTT is the primary reason we have chosen a simplified circuit model for the simulation of ring oscillators. A typical Level 3 MOS model contains hundreds of parameters; while a SPICE-level simulation is straightforwardly possible even for larger circuits, learning (backpropagation) becomes computationally demanding for such models.

It is also important that BPTT supports only "*in silico*" training. The design of the ONN (i.e., the learning) takes place on a different hardware than the inference. The learning is done on a digital computer model. Once the learning is finished, the inference is done on a dedicated hardware that uses the computer-learned circuit parameters. Online learning is not possible this way, but our goal is to realize efficient, "hard wired" hardware for inference.

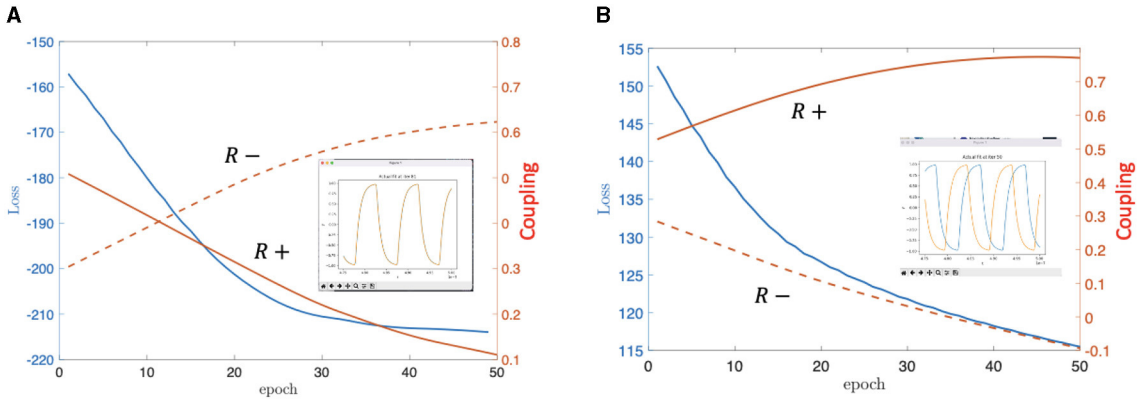


FIGURE 4
On **(A)** we can see the simulation's result for the positively coupled oscillators, meanwhile on **(B)**, there is the same for the negatively coupled 2-oscillator system. The loss changed in both cases from high value to low value. In addition, the orange curves are indicating the learning parameters' values contained in **C** and not the real values of the resistors. Note that in **(B)**, the parameter value corresponding to "R-" is going below 0, which would mean a negative resistance because of the connection of the parameters in **C** to the physical parameters, but this is only the mathematical solution, for a given simulation, the parameters were clamped to be non-negative and if they hit zero, the connection removed. **(A)** In-phase coupling learned. **(B)** Anti-phase coupling learned.

Figure 4 exemplifies the learning procedure for the two-oscillator system of Figure 2. We selected the loss function of the system as the dot product of the oscillator waveforms, which is a standard choice for this type of problems. The loss should be maximized (minimized) for in-phase (anti-phase) coupling. Such mean-square machine learning algorithm adjusts the value of the $C_{i,j}$ parameters (and the coupling resistors) until this desired phase configuration is reached.

This method can be straightforwardly generalized to achieve convergence toward more complex patterns. If the loss function aims to maximize the dot product of waveforms between same-colored pixels and minimize them between different-colored ones, then the phase pattern can converge toward any prescribed image. If the phase pattern made to converge toward different patterns for different inputs, then the ONN will act as an associative memory.

Since the Machine Learning (ML) technique is designing a physical circuit, safeguards were taken not to arrive to unrealizable circuit parameters such as negative resistances or exceedingly strong couplings that would quench oscillations. This was done by clipping the values after each learning step to a given interval.

3 Results

3.1 ONN-based pattern association on the MNIST dataset

We have chosen the standard MNIST database for testing the associative capabilities of our system. Since the BPTT algorithm is computationally demanding, we made a few simplifications. We downsampled the initially 28 × 28 pixel-sized picture from MNIST to have either 14 × 14 or 7 × 7 size using average pooling. This allowed us to have a reduced dimension for the input images, and also keep the necessary information because of the average pooling. In addition, 14x14 MNIST images are still recognizable as a human, so it allowed us to easily recognize if some patterns are easier for the algorithm to distinguish from the others.

3.1.1 Baseline: ONN-based associative memory with Hebbian learning

The simplest, well-studied ONN-based associative memory can be designed by the Hebbian rule. If we want the phase pattern to converge toward ξ or η for inputs resembling to ξ or η, then the weights that realize this associative memory are:

$$C_{ij}^{cpl} = \frac{1}{2}(\xi_i\xi_j + \eta_i\eta_j),$$

where ξ_i and ξ_j is the i-th and j-th element of the pattern ξ, and η_i and η_j is the i-th and j-th element of the pattern η, respectively.

The rule assumes all-to-all couplings, making a larger-scale network hard to physically realize.

In our Hebbian learning scheme, the weights were determined initially in a single-shot formula, and in our test case, we applied the learning to optimize the value of base coupling resistances, R_c, and the parameters in $\mathbf{B}'$, which are the amplitudes of input current generators.

The inner RC time constant of the ring oscillators was $2.0 \cdot 10^{-10}$ s, which translates into a 500 MHz oscillation frequency (time period $T = 2$ ns). The total simulation time for the network is 500 ns. The phase pattern is calculated from the last 300 ns window, so convergence is achieved after less than 100 oscillation cycles or 200 ns.

3.1.2 ONN-based associative memories with all-to-all and nearest-neighbor coupling

The same functionality that is realized by Hebbian learning can be achieved by the BPTT method. The loss function we selected was:

$$L = \frac{1}{n}\sum_{k=0}^{n}(O_k - T_k)^2,$$

where O_k is the pattern calculated from the output of the oscillators for the k-th input in the batch and T_k is the ground truth

for the same, which were ideal patterns of "0" and "1". In the above formula, n is the size of the batch used for learning.

Figure 5 compares results from the Hebbian- and BPTT-based designs. It is visually apparent that the BPTT-based design associates to the right pattern from very much distorted patterns. For the experiments seen in Figure 5, we downscaled the images from 28×28 to 7×7 which distorted many of the inputs. It helped speed up the computations, because an all-to-all coupled 728 oscillator system would result in almost 620000 resistors. This is hard to physically realize.

Most importantly, the BPTT-based design allows the design of sparsely interconnected circuit topologies. We used it to design the C_{ij} matrix of associative memory assuming only nearest neighbor interconnections. The nearest-neighbor interconnected, BPTT-designed network outperforms the fully interconnected Hebbian network, even if the number of trainable parameters in the system ($\approx 8n$ vs. $\frac{1}{2}n^2$) is significantly less. The qualitative results of this comparison can be seen in Figure 5.

The result that a nearest-neighbor (NN) interconnected (BPTT-designed) network outperforms the (Hebbian-designed) fully connected network is important. In a fully connected ONN, the number of connections grows quadratically with the number of oscillators, making large, fully connected circuits unrealizable. Only locally connected architectures yield to scalable, physically realizable ONN circuits.

Quantitatively, the results of the different approaches for the whole dataset $S = \{0, 1\}$ can be seen in Table 1.

3.2 Multi-layered ONNs for classification on the MNIST database

Single layer associative memories are not particularly efficient for classifying all the 10 MNIST classes, as there are strong

TABLE 1 The MSEs of all the elements from the set and their respective ground truths for the different methods in case of the associative learning.

Method	Hebbian	Proposed fully connected	Proposed NN connected
#Params	1,176	2,352	312
MSE	0.068	0.020	0.047

It is apparent that the fully connected network performed the best but even the nearest neighbor connected layer is good enough to beat the Hebbian learning in terms of quantitative association.

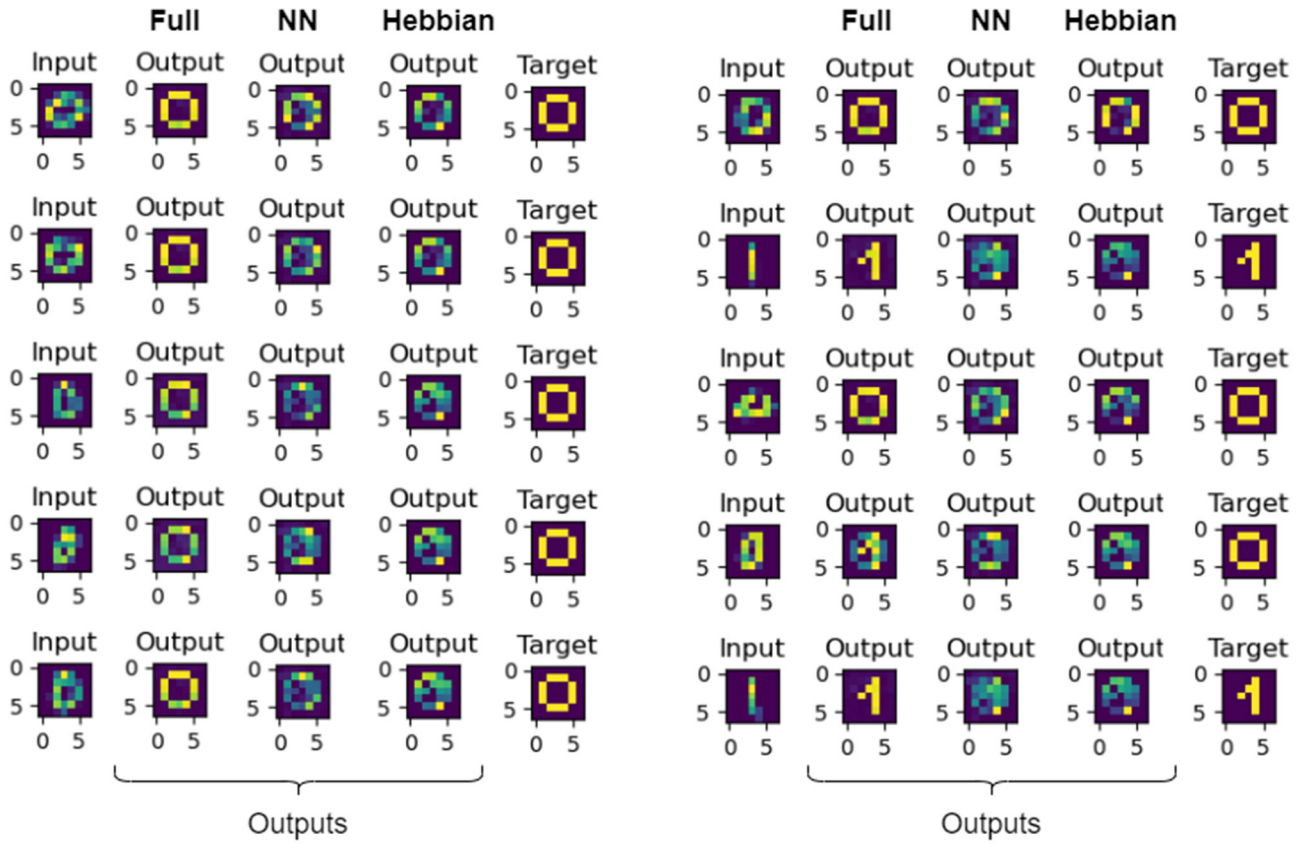


FIGURE 5
Here, we can see the comparison of the results of the fully connected, nearest neighbor connected, and the Hebbian-learned based networks. The two blocks of five inputs are shown side-by-side. The first column in each block corresponds to the input digit, the next three is the output of the systems (in order from left to right: fully coupled, nearest neighbor coupled and Hebbian-based). The last column in both blocks shows the target digit. It is apparent that the fully connected system worked best, but even our proposed, nearest neighbor connected topology was outperforming the Hebbian-based architecture.

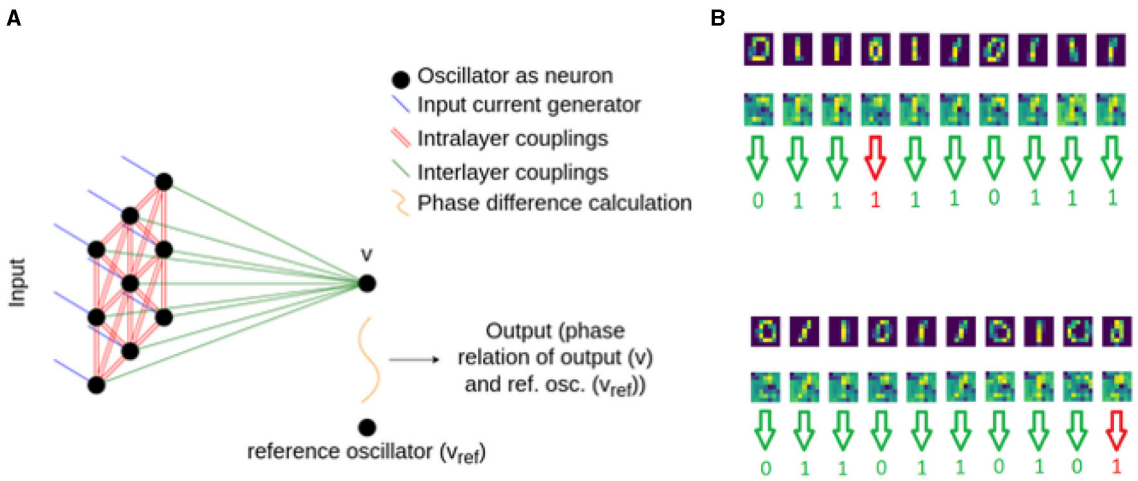


FIGURE 6
A simple two-layer classifier showing also the patterns forming in the hidden layer.

correlations between the different digits. The BPTT method does not require the oscillators of the network to converge to a prescribed phase pattern so there is no need to use associative memory for classification. For this reason, we investigate a simple multilayer ONN, where the second layer is a single oscillator connected to all oscillators of the input layer as illustrated in Figure 6.

We used architecture in Figure 6 in various ways: for binary classification, one hidden layer and a single output yielded decent results, as described in Section 3.1.3.1. For classifying all 10 digits, we trained 10 blocks (Figure 6), each responsible for recognizing one particular digit, and evaluated them with a winner takes all decision (see Section 3.1.3.2). Finally, we swapped the "winner takes it all" method for a small, MLP (multi-layered perceptron) model, composed of just a few neurons. This architecture is shown in Figure 7 and discussed in Section 3.1.3.3.

3.2.1 Binary classifiers with a single output

The two-layer classifier is shown in Figure 6. The phase of the output oscillator carries the classification result: we compare the output oscillator's phase with a reference oscillator's phase and maximize (minimize) their phase difference for one (or the other) pattern.

Since the optimal oscillator couplings are discovered by the BPTT algorithm, this device does not necessarily work as an associative memory. The phase patterns appearing in the hidden layer are non-intuitive, albeit occasionally they vaguely resemble the images to be recognized.

That having been said, without any apparent, clearly visible structure in the hidden layer, the network was predicting the two classes at a 98% success rate. The predictions made on some images are present in Figure 6.

3.2.2 10-digit classifier using a winner takes it all output

Classifying all 10 digits is a significantly more difficult task than the basic binary classifier and requires many more oscillators. Training a large number of oscillators simultaneously is prohibitively difficult with our method. Instead, training everything at once, we trained 10 separate blocks (subnetworks), each being responsible for recognizing one particular digit - as seen on Figure 7. The blocks themselves are nearest-neighbor connected. The individually trained networks are connected to a winner-takes-all circuit that decides the result of the 10-class classification.

The results of the distribution of average values of the predictions of each individual, competitive network can be seen in Figure 8. After further training, the output probabilities of the individual networks were improving, but still not aligned perfectly to the desired distributions as can be seen on Figure 9. Some digits predicts a high likelihood for the wrong classes. Using the winner takes it all algorithm (i.e., the decision is made by the ONNs using the 10 output likelihoods from the architectures and the highest one is the winner), we achieved an accuracy around 70%. To put this number in context, random guessing would be 10 %, but the state of art for MNIST digits is above 99%.

3.2.3 10-digit classifier using a trained second layer

Instead of the winner takes it all decision, we used a simple multilayered perceptron at the end to improve classification accuracy. It consists of 2 layers: one hidden layer and one output layer. The hidden layer has 15 and the output has 10 neurons. The structure of this new setup can be seen on Figure 10. This means that only 325 extra parameters are introduced, which is negligibly small compared to the roughly 16000 parameters of the ONN layers.

The reason we have chosen a traditional Feed Forward Neural Network (FFNN) layer to improve accuracy are entirely practical: such FFNN is easy to train and we could train it straightforwardly after all the ONN blocks were designed. We emphasize that this conventional NN layer does not alter our conclusions, and the vast majority of the computation is still done by the ONN network. It is worth to note that there are very few multi-layered ONNs in the literature [a few examples are Karg et al. (2021), Abernot and Aida (2023), or Velichko et al. (2019)].

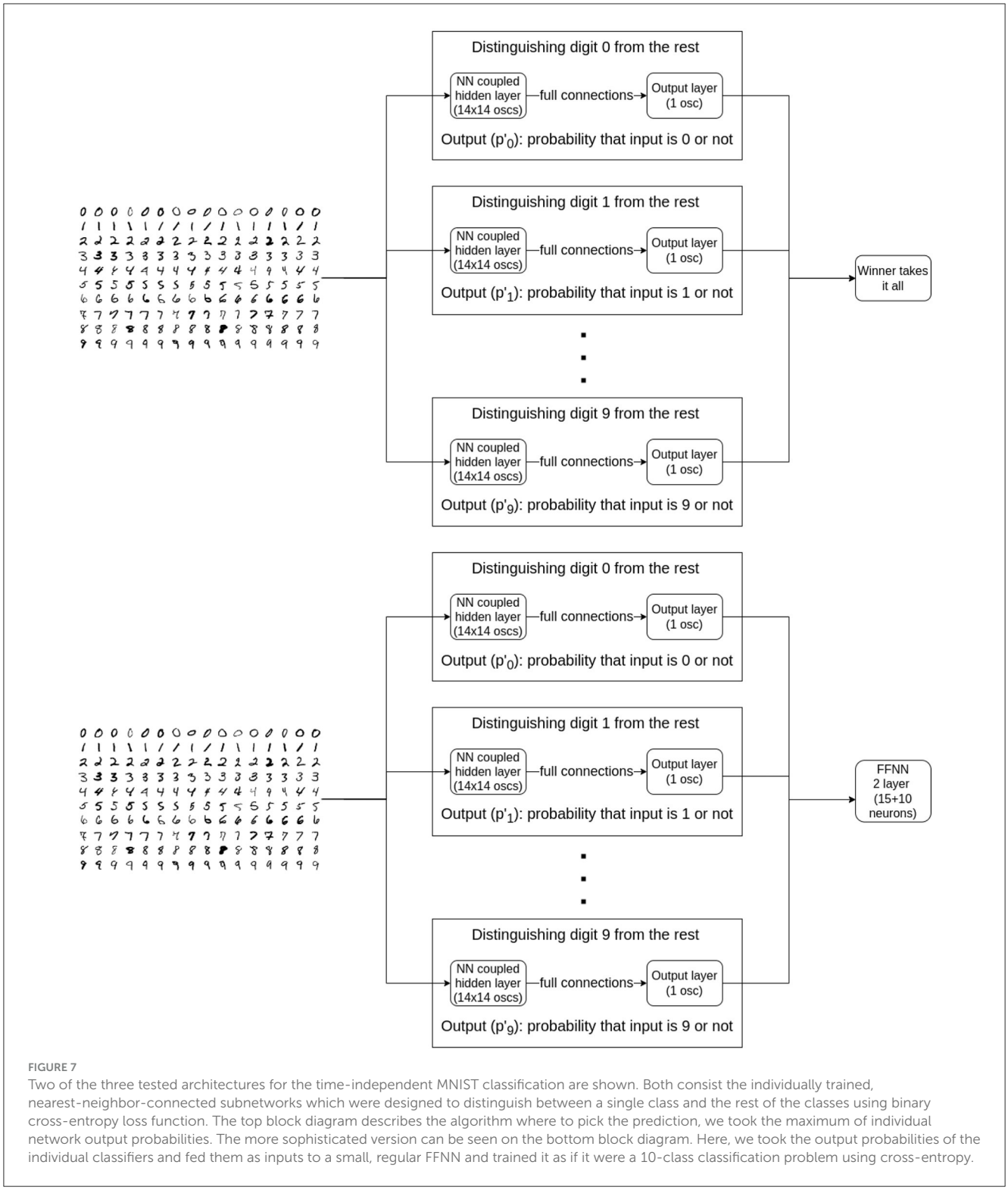


FIGURE 7
Two of the three tested architectures for the time-independent MNIST classification are shown. Both consist the individually trained, nearest-neighbor-connected subnetworks which were designed to distinguish between a single class and the rest of the classes using binary cross-entropy loss function. The top block diagram describes the algorithm where to pick the prediction, we took the maximum of individual network output probabilities. The more sophisticated version can be seen on the bottom block diagram. Here, we took the output probabilities of the individual classifiers and fed them as inputs to a small, regular FFNN and trained it as if it were a 10-class classification problem using cross-entropy.

Using the outputs of the competitive networks as inputs to this small neural network, we managed to reach 96.7% predictive accuracy. This is excellent accuracy for a network of this size. We implemented feedforward (perceptron) neural networks with identical number of parameters, and such networks typically reach 93–95% accuracy. While the MNIST problem is solved with fairly trivial networks with accuracy approaching 100%, these networks are using hundreds of thousands of parameters and we only had 20000 parameters in our training scheme.

We emphasize that in terms of computation workload, the heavy lifting in this architecture is done by the ONN-based preprocessing layer—the output layer contains a small number of parameters and it is a very small-scale neural network by any standard. The output layer is there since it is easily trainable so

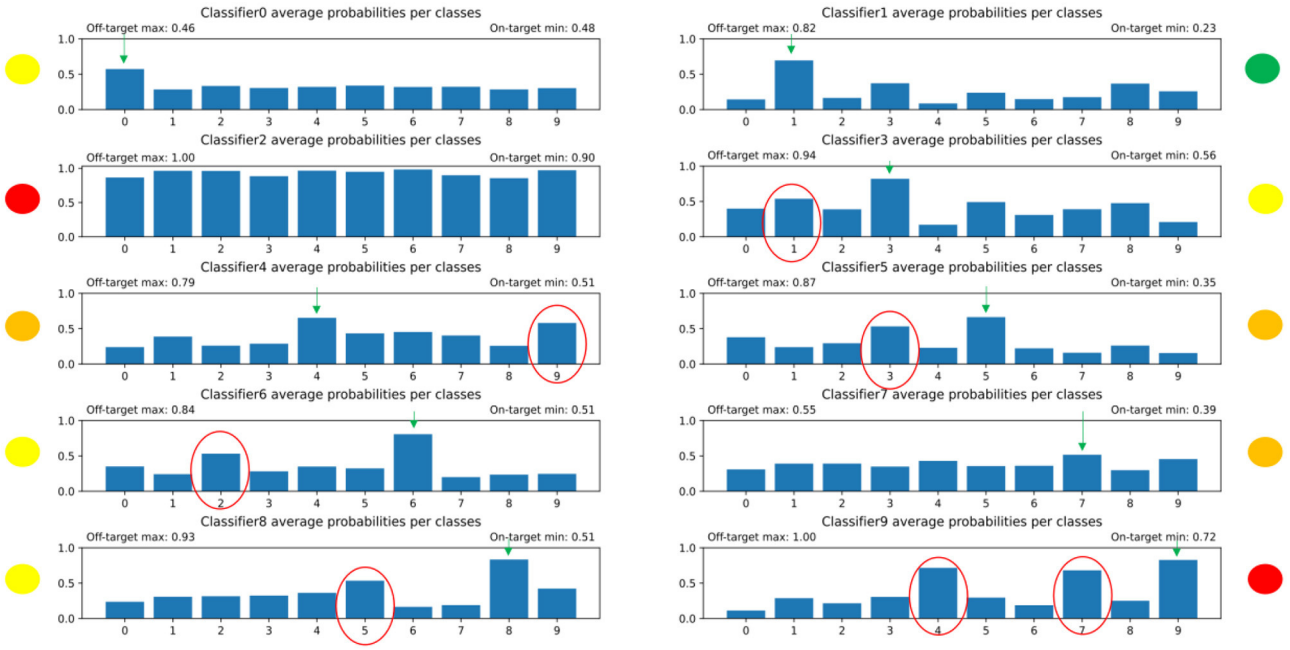


FIGURE 8
The distribution of predicted average probabilities for the individual, competitive networks in the winner takes it all model. The red-circled bars are those that on average were too high as probabilities because the given subnetwork should not have high values for that specific digit. The green arrows indicate which bar should be the highest. The yellow, orange, and red dots near the plots indicate how well the subnetwork managed to solve its task. It can be seen that this had to be improved.

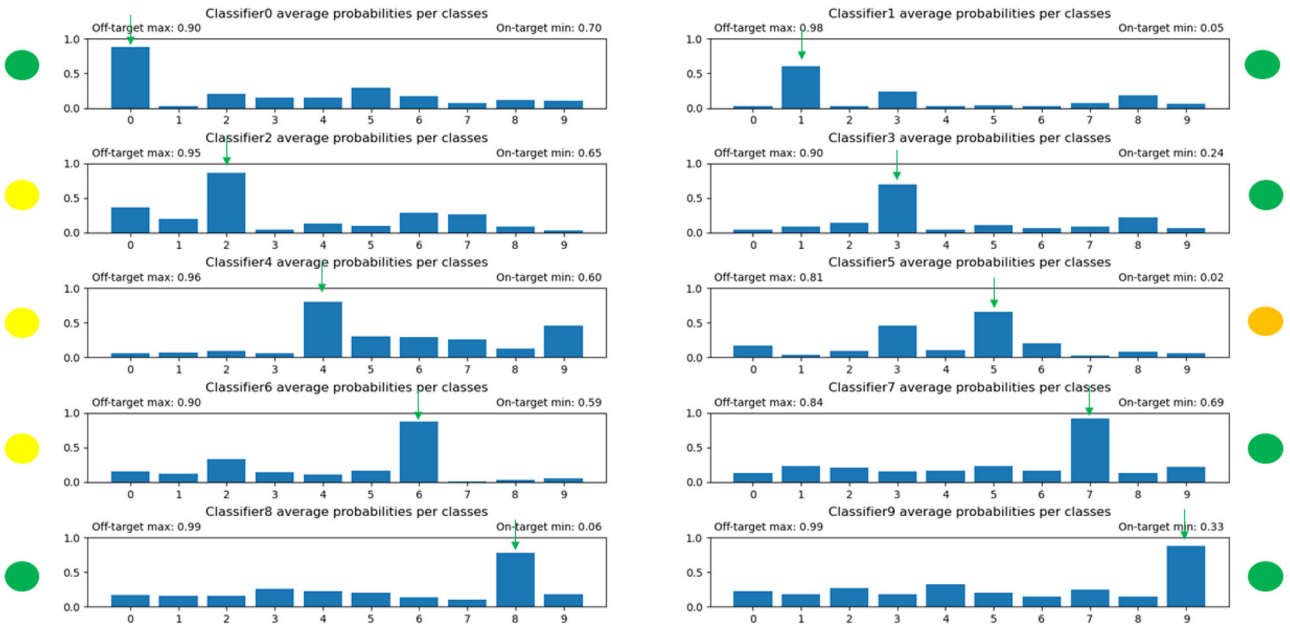


FIGURE 9
The distribution of the predicted average probabilities after extensive training. It is evident that the distributions improved, but there are still some outliers where the non-target digits are having too high probabilities.

it can maximize network performance at low training cost. The power consumption of the network is dominated by the ONN, and so the entire architecture benefits from the energy-efficient ONN operation. This result hints that ONNs excel as first layers (preprocessing layers) in an AI pipeline.

Integrating oscillatory neural networks (ONNs) with compact traditional neural networks, resembling perceptrons, presents a promising avenue to leverage their combined strengths. ONNs can perform complex, dynamic computation but they are difficult to train. Perceptrons (what we used here) can be easily trained to

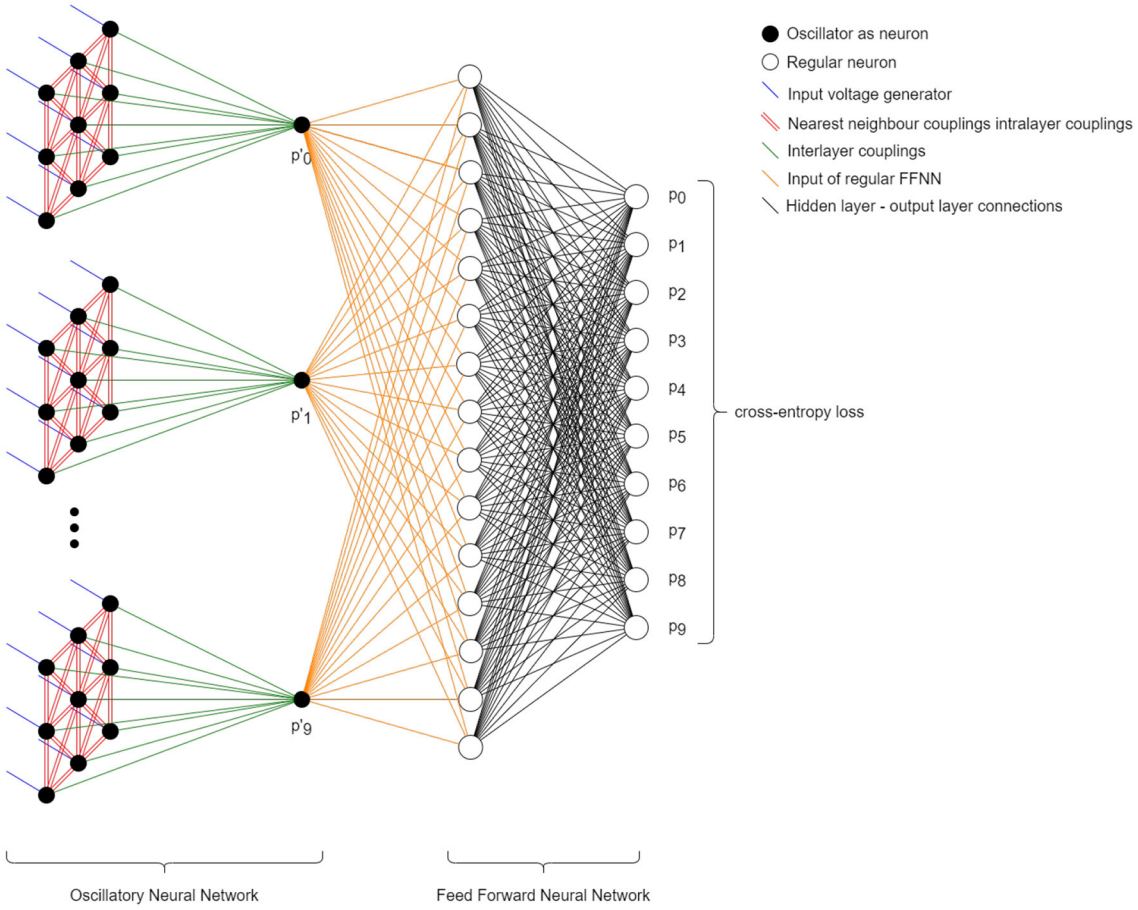


FIGURE 10
A network architecture with ONN layers as preprocessors and a traditional neural network postprocessing the results. The easy-to-train output layer significantly improves classification accuracy.

TABLE 2 **The quantitative comparisons of binary and multi-class classifiers with the parameter count indicated.**

	Binary classifiers		Multiclass classifiers with oscillators			Benchmark
Method	Fully	NN	FFNN-like	Winner takes all	Augmented	Perceptron FFNN
#Param	38,416	1,600	40,180	16,000	16,325	16,363
Perf. (%)	98	98	72.3 (70–75)	66.7 (65–70)	95.1 (93–97)	94.4 (93–95)

The "Augmented" network is the MLP-augmented network, using the two-layered MLP as the last function instead of the "Winner takes all". For the binary classifiers, the 98% performance is the worst case scenario. The multiclass classifiers and the benchmark model have ranges between the worst and best performances. The differences are the random initial values for the learning parameters.

specific tasks, but they have limited computational might. Putting ONNs close to sensory inputs, where most input data has to be handled (and where most power is consumed), and refining the computing function, a higher level with an easily trainable layer could harness the best of both worlds and yield the best overall power efficiency for the network.

3.3 Comparison of ONN classifier architectures

The Table 2 quantitatively summarizes some key findings of our study. Most importantly, the ONN-based network outperforms a standard FFNN with the same amount of parameters. This is not entirely surprising for two reasons: one is that ONNs are recurrent neural networks, exhibiting complex dynamics, unlike an FFNN. The other reason is that ONNs carry information in the phase, frequency, and amplitude of their signals, while a standard neuron outputs only one value (which is usually a static voltage in a hardware realization). So one may expect that an ONN, if properly trained, may be able to perform more complex functions with same number of neurons (processing units).

As a back-of-envelope calculation, if we assume a hardware similar to Moy et al. (2022), a single ring oscillator in our circuit would consume about a picojoule of power per inference, so the net power consumption of the competitive multi-layered device (with 20,000 ring oscillators) is estimated to be 4×10^{-8} joules/inference.

Highly optimized lightweight hardware neural networks achieve in the ballpark of 1μ J/inference for a similar problem (Dressen, 2023). GPU-based networks are usually designed to achieve higher accuracy at much higher power consumption, even if state-of-art GPU chips are manufactured using a much more advanced technology node than the work of Moy et al. (2022). Overall, these numbers suggest that building the ONN we studied here by simulations would give orders-of-magnitude improvements in power efficiency compared to state-of-art solutions.

In conclusion, the ONN is not only more economical in terms of parameters but does its job with a significantly higher power efficiency than the equivalent digital or software implementation.

4 Discussion

In this study, we introduced an in-silico method to design ONNs. We build a computational model of the ONN, apply BPTT techniques on this model and determine circuit parameters automatically using the BPTT training algorithm.

In the current literature of ONNs, Hebbian learning rules are used almost exclusively to realize associative memories or classifiers. The reader is referred to Núñez et al. (2021), Abernot and Aida (2023), Delacour and Todri-Sanial (2021), and Nikonov et al. (2015) and to the references therein. The performance and the capabilities of a simple Hebbian rule is quite limited when compared to modern ML algorithms. One may suspect that if an ONN is designed by Hebbian rules, the capabilities of the ONN will be more likely constrained by the learning rule, and not by the ONN hardware itself.

The BPTT-based design allowed us to use simulations for exploring the limits of ONN hardware without the limitations imposed by the simplicity of the training algorithm. We indeed found that the state-of-art learning method significantly increased the accuracy of the ONN classification, and this is one main result of this study.

Another key benefit of our method is that it allows the design of ONNs that is amenable to circuit realization. For example, we have shown that a nearest-neighbor-connected ONN that is designed by BPTT can outperform a fully connected Hebbian-trained device. Since only locally connected ONNs are scalable to meaningful problem sizes, this discovery opens the door to physically realizable ONNs, which perform complex processing functions without an unfeasibly high number of interconnections. In addition to that, the BPTT method may also be used to design higher-interconnected networks (such as all-to-all connected ones) that greatly outperform their Hebbian counterparts.

Another result of the study was the design of multi-layered ONN devices, of which very few exist in literature. The ONN first layer (preprocessing layer) is followed by a simple perceptron-based layer, and classification accuracy of 95 % is reached. In line with expectations, we find that multiple layers significantly enhance the capabilities of the network. We also find that the number of circuit parameters we had to train is smaller than the number of parameters of a similarly performing standard FFNN. This means that the ONN is more economical in terms of parameters. This benefit appears on top of the benefit in power efficiency: the analog ONN circuit dynamics does its job from the fraction of the power of a number-crunching digital solution.

Our design method is not without hindrances. One of its drawback is that it is not applicable to online training, the ONN must be trained on its computer model (*in silico*) and then the weights are hard-wired into a hardware circuitry. This is acceptable for an edge-AI accelerator, where energy-efficient operation is the main figure of merit. Further research is required to find training methods that would allow continuous, online learning.

Data availability statement

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

Author contributions

TR: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. WP: Conceptualization, Funding acquisition, Supervision, Writing – review & editing. GC: Conceptualization, Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This study was partially supported by a grant from Intel Corporation, titled HIMON: Hierarchically Interconnected Oscillator Networks. This project has received additional funding from the European Union's Horizon EU research and innovation programme under grant agreement No. 101092096 (PHASTRAC). The funder (Intel Corporation) had the following involvement with the study: helping to discover relevant literature and motivate the study. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

Acknowledgments

The authors are grateful for regular and fruitful discussions with the Intel team, in particular Narayan Srinivasa, Dmitri Nikonov, and Amir Khosrowshahi.

Conflict of interest

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

Publisher's note

References

Abernot, M., and Aida, T.-S. (2023). Simulation and implementation of two-layer oscillatory neural networks for image edge detection: bidirectional and feedforward architectures. *Neurom. Comput. Eng.* 3, 014006. doi: 10.1088/2634-4386/acb2ef

Belyaev, M., and Velichko, A. (2020). "Classification of handwritten digits using the hopfield network," in *IOP Conference Series: Materials Science and Engineering* (Bristol: IOP Publishing), 052048.

Chen, R. T. Q. (2018). *torchdiffeq*. Available online at: https://github.com/rtqichen/torchdiffeq

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). *Neural ordinary differential equations*. Advances in Neural Information Processing Systems.

Csaba, G., and Porod, W. (2020a). Coupled oscillators for computing: a review and perspective. *Appl. Phys. Rev.* 7, 011302. doi: 10.1063/1.5120412

Csaba, G., and Porod, W. (2020b). Noise immunity of oscillatory computing devices. *IEEE J. Explorat. Solid-State Comp. Dev. Circuits* 6, 164–169. doi: 10.1109/JXCDC.2020.3046558

Csaba, G., Ytterdal, T., and Porod, W. (2016). "Neural network based on parametrically-pumped oscillators," in *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Monte Carlo, Monaco: IEEE), 45–48.

Delacour, C., and Todri-Sanial, A. (2021). Mapping hebbian learning rules to coupling resistances for oscillatory neural networks. *Front. Neurosci.* 15, 694549. doi: 10.3389/fnins.2021.694549

Dressen, O. (2023). *Hardware Conversion of Convolutional Neural Networks: What Is Machine Learning*. Available online at: https://www.analog.com/en/resources/analog-dialogue/articles/hardware-conversion-of-cnns-what-is-machine-learning-part-3.html

Dutta, S., Parihar, A., Khanna, A., Gomez, J., Chakraborty, W., Jerry, M., et al. (2019). Programmable coupled oscillators for synchronized locomotion. *Nat. Commun.* 10, 3299. doi: 10.1038/s41467-019-11198-6

Furber, S., and Temple, S. (2007). Neural systems engineering. *J. Royal Soc. Interf.* 4, 193–206. doi: 10.1098/rsif.2006.0177

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hopfield, J. J., and Tank, D. W. (1985). "neural" computation of decisions in optimization problems. *Biol. Cybern.* 52:141–152. doi: 10.1007/BF00339943

Hoppensteadt, F. C., and Izhikevich, E. M. (1999). Oscillatory neurocomputers with dynamic connectivity. *Phys. Rev. Lett.* 82, 2983–2986. doi: 10.1103/PhysRevLett.82.2983

Hoppensteadt, F. C., and Izhikevich, E. M. (2000). Synchronization of laser oscillators, associative memory, and optical neurocomputing. *Phys. Rev. E* 62, 4010–4013. doi: 10.1103/PhysRevE.62.4010

Karg, S. F., Menges, F., and Gotsmann, B. (2021). *Multi-Layer Oscillating Network*. Patent 11,157,792, US. Washington, DC: Patent 11,157,792, US.

Lai, X., and Roychowdhury, J. (2005). "Analytical equations for predicting injection locking in lc and ring oscillators," in *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference, 2005* (San Jose, CA: IEEE).

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lillicrap, T. P., and Santoro, A. (2019). Backpropagation through time and the brain. *Curr. Opin. Neurobiol.* 55, 82–89. doi: 10.1016/j.conb.2019.01.011

Michel, A., Farrell, J., and Porod, W. (1989). Qualitative analysis of neural networks. *IEEE Trans. Circ. Syst.* 36, 229–243. doi: 10.1109/31.20200

Moy, W., Ahmed, I., Chiu, P.-,w., Moy, J., Sapatnekar, S. S., and Kim, C. H. (2022). A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nature Electron.* 5, 310–317. doi: 10.1038/s41928-022-00749-3

Nikonov, D. E., Csaba, G., Porod, W., Shibata, T., Voils, D., Hammerstrom, D., et al. (2015). Coupled-oscillator associative memory array operation for pattern recognition. *IEEE J. Explorat. Solid-State Comput. Dev. Circuits* 1, 85–93. doi: 10.1109/JXCDC.2015.2504049

Nú nez, J., Avedillo, M. J., Jiménez, M., Quintana, J. M., Todri-Sanial, A., Corti, E., et al. (2021). Oscillatory neural networks using vo2 based phase encoded logic. *Front. Neurosci.* 15, 655823. doi: 10.3389/fnins.2021.655823

Parihar, A., Shukla, N., Jerry, M., Datta, S., and Raychowdhury, A. (2017). Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Sci. Rep.* 7, 911. doi: 10.1038/s41598-017-00825-1

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). *Automatic Differenation in PyTorch*. Available online at: https://openreview.net/pdf?id=BJJsrmfCZ

Righetti, L., Buchli, J., and Ijspeert, A. J. (2006). Dynamic hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena* 216, 269–281. doi: 10.1016/j.physd.2006.02.009

Roychowdhury, J. (2015). Boolean computation using self-sustaining nonlinear oscillators. *Proceedings of the IEEE* 103:1958–1969. doi: 10.1109/JPROC.2015.2483061

Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., et al. (2017). A survey of neuromorphic computing and neural networks in hardware. *arXiv [Preprint]*. doi: 10.48550/arXiv.1705.06963

Smith, K. A. (1999). Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS J. Comput.* 11, 15–34. doi: 10.1287/ijoc.11.1.15

Tolmachev, P., and Manton, J. H. (2020). "New insights on learning rules for hopfield networks: Memory and objective function minimisation," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow: IEEE), 1–8.

Vassilieva, E., Pinto, G., Acacio De Barros, J., and Suppes, P. (2011). Learning pattern recognition through quasi-synchronization of phase oscillators. *IEEE Trans. Neural Netw.* 22, 84–95. doi: 10.1109/TNN.2010.2086476

Velichko, A., Belyaev, M., and Boriskov, P. (2019). A model of an oscillatory neural network with multilevel neurons for pattern recognition and computing. *Electronics* 8, 75. doi: 10.3390/electronics8010075

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 1550–1560. doi: 10.1109/5.58337

Wigington, R. (1959). A new concept in computing. *Proc. IRE* 47, 516–523. doi: 10.1109/JRPROC.1959.287311

Wu, C. W. (1998). Graph coloring via synchronization of coupled oscillators. *IEEE Trans. Circ. Syst. I: Fundam. Theory Appli.* 45, 974–978. doi: 10.1109/81.721263

frontiers | Frontiers in Neuroscience

TYPE Perspective
PUBLISHED 13 March 2024
DOI 10.3389/fnins.2024.1379205

OPEN ACCESS

EDITED BY
Pier Luigi Gentili,
Università degli Studi di Perugia, Italy

REVIEWED BY
Pasquale Stano,
University of Salento, Italy

*CORRESPONDENCE
Katja-Sophia Csizi
✉ kcs@zurich.ibm.com

RECEIVED 30 January 2024
ACCEPTED 19 February 2024
PUBLISHED 13 March 2024

CITATION
Csizi K-S and Lörtscher E (2024) Complex chemical reaction networks for future information processing.
Front. Neurosci. 18:1379205.
doi: 10.3389/fnins.2024.1379205

Complex chemical reaction networks for future information processing

Katja-Sophia Csizi* and Emanuel Lörtscher

Department of Science of Quantum and Information Technology, IBM Research Europe - Zurich, Rüschlikon, Switzerland

Tackling the increasing energy demand of our society is one of the key challenges today. With the rise of artificial intelligence, information and communication technologies started to substantially contribute to this alarming trend and therefore necessitate more sustainable approaches for the future. Brain-inspired computing paradigms represent a radically new and potentially more energy-efficient approach for computing that may complement or even replace CMOS in the long term. In this perspective, we elaborate on the concepts and properties of complex chemical reaction networks (CRNs) that may serve as information-processing units based on chemical reactions. The computational capabilities of simpler, oscillatory chemical reactions have already been demonstrated in scenarios ranging from the emulation of Boolean gates to image-processing tasks. CRNs offer higher complexity and larger non-linearity, potentially at lower energy consumption. Key challenges for the successful development of CRN-based computers are associated with their specific physical implementations, operability, and readout modalities. CRNs are sensible to various reaction triggers, and provide multiple and interlinked reaction pathways and a diverse compound space. This bears a high potential to build radically new hardware and software concepts for energy-efficient computing based on neuromorphic architectures—with computing capabilities in real-world applications yet to be demonstrated.

KEYWORDS

chemical computing, neuromorphic computing, chemical reaction networks, low-energy, brain-inspired

1 Introduction

Semiconductor technology constitutes one of the key-enabling technologies responsible for numerous innovations in modern times. Thanks to the continuous miniaturization of all integrated circuit components, the foundational technology has been successfully adapted to varying and diverse computing tasks over several decades. The validity and continuation of Moore's Law is currently being controversially discussed. At the same time, the increasing energy consumption of today's computing infrastructures—combined with other energy-expensive technologies—undoubtedly represents one of the largest challenges to our society. The projected energy demand might soon surpass the amount of energy being cumulatively generated. In the field of information and communication technologies, this is due to an ever increasing number of systems (Internet of Things (IoT), mobile systems, data centers, etc.) but also aggravated by emerging artificial intelligence (AI) applications (image and voice recognition, analog sensor signal processing, chatbots, etc.). The majority of AI applications entails workloads for which the classical von–Neumann architecture, with separated memory and processing units, was

originally not intended and now turns out to be costly in terms of energy consumption. Furthermore, maintaining and running AI systems creates similarly high costs as training thereof, currently already consuming >500 MWh per day. Overall, this trend will soon result in unaffordable energy demands beyond 100 TWh if the current systems cannot be substantially improved (de Vries, 2023). Consequently, there are tremendous efforts in semiconductor and related industries aiming at tuning existing semiconductor devices (e.g., phase-change, FPGA), architectures (e.g., specialized architectures), systems (e.g., GPUs, TPUs) or computing tasks (e.g., in-memory computing) for AI applications. Beyond those attempts, more disruptive and radically new ways of computing beyond the use of electrons and transistors are being evaluated. These initiatives include spintronics, quantum computing, optical computing, DNA-based computing, and neuromorphic computing. Generally, it is imperative for all new approaches to prioritize sustainability aspects over the entire life-cycle. This includes the use of abundant materials, green fabrication processes, complete recycling, etc. to contribute to a circular economy. To complement or replace existing technologies, it is essential not only to meet the prevailing standards of scalability and performance, but also to satisfy all aforementioned sustainability constraints. At the moment, there seems to be no obvious successor technology for CMOS. However, neuromorphic architectures appear to be a promising foundation as they conceptually mimic the human brain, which serves as an unparalleled role model in terms of energy efficiency. In addition, some neural networks are already designed from an implicit, simplified brain inspiration, but with orders of magnitude less complexity (Richards et al., 2019; Zador et al., 2023).

In this perspective, we present the novel class of bio-inspired, chemical information processing concepts that are based on complex chemical reaction networks (CRNs). CRNs are capable of processing information based on highly interconnected and interlinked chemical reactions. Due to their chemical self-organization and nonlinear characteristics, these systems provide potentially useful means for low-energy and massively parallel computing. To demonstrate the neuromorphic capabilities, scalable physical implementations and operational protocols must be developed.

2 Chemical reactions as information-processing units

The human brain with its interconnected neurons and the release of neurotransmitters in response to nerve impulses across localized information-processing centers is still unmatched in terms of energy efficiency. It consumes only around 20 watts of power while performing more than 200 trillion operations per second. New classes of HPC systems (e.g., ICNS Deep South) parallel such cross-linked brain-inspired architectures and are predicted to reach more than 100 trillion synaptic operations per second at a significant—yet to be measured—energy reduction. The cross-linking and collocation of memory and information-processing units will be at the heart of next-generation, semiconductor-based neuromorphic HPC to solve the von-Neumann bottleneck. Additionally, radically new approaches may take up the information-processing concept of our brain even closer by using chemical compounds and chemical reactions to encode and process information for computing purposes: Not only does information processing on the chemical level in living entities regulate and control fundamental processes like immune response, growth, or gene expression, the human brain runs entirely on chemical reactions for "logic" information processing. It is therefore conceptually appealing to draw direct analogies between the chemical compound and chemical reaction space to bio-inspired brain-type architectures with reactions emulating synapses and compounds representing neurons. By their very nature, molecules can carry out complex tasks such as molecular recognition and chemical reactions with the smallest possible footprint and energy requirements. Furthermore, chemical reactions can be cascaded, and are typically highly non-linear. It has been demonstrated that interconnected chemical systems are capable of mimicking Boolean logic gates (Tsompanas et al., 2021), carrying out pattern recognition (Gizynski and Gorecki, 2017; Parrilla-Gutierrez et al., 2020) or image processing tasks (Rambidi et al., 1998), finding shortest paths (Rambidi and Yakovenchuk, 2001), or solving optimization problems (Guo et al., 2021). Like other non-conventional computing architectures, these attempts predominantly exploited time-dependent event-driven paradigms, either in the form of spike-induced, or self-induced excitations (in analogy to oscillatory, or spiking neural networks).

2.1 Beyond Belousov–Zabotinsky reactions

Chemical computing was pioneered using the Belousov–Zabotinsky (BZ) reaction. The underlying chemical reactions result in nonlinear temporal oscillations and spatial self-organization. In the BZ oscillator, the time-evolution of excitations is determined by chemical reactions and diffusion, therefore referred to "reaction–diffusion" computing. In a very simplified representation, the BZ can be described by three main reactions that form a closed-loop catalytic cycle, as illustrated in Figure 1A. In BZ oscillations, the clock rate correlates with the intrinsic oscillation frequency and is somewhere between 1 and 100 Hz, not comparable to the GHz frequencies of semiconductor devices (GHz). Apart from that, the aforementioned complexity of information processing in living entities may require massive parallel operation in interlinked compartments. As a potential alternative, chemical reaction networks that have a higher complexity than the BZ reaction have been recently proposed as a chemical computing platform (Ivanov et al., 2023). In principle, any real-world chemical system can be encoded in the form of a chemical reaction network, although the network width and depth (i.e., the number of compounds formed and the number of reactions or reaction sequences that connect these compounds) varies significantly. Figure 1 conceptually illustrates, in a very simplified manner, the different degrees of complexity and interlinkage of chemical reactions suited for computing purposes.

Under the aspect of nonlinearity and complexity, CRNs more closely resemble bio-inspired systems than oscillatory reactions. The prebiotically relevant formose reaction is one of the archetype CRNs of that kind to show a temporal evolution over time once

the self-condensation of formaldehyde is energetically overcome. As recently demonstrated, the formose CRN provides a high-dimensional state space, nonlinear interactions, a fading memory effect, and discrete output signals—namely products derived after derivatization of the reaction mixture—that all depend susceptibly on input variations (Robinson et al., 2022). A fading-memory effect can moreover be realized by forcing the CRN into an out-of-equilibrium steady state, where the system can then receive inputs from and adjust its response to environmental conditions by dynamically changing its underlying reactions. With all these properties, the formose CRN can dictate some "design rules" and properties of an artificial CRN to be used for future computing:

1. *Complexity* and *nonlinearity*: The evolution of a reaction network constitutes a highly non-linear self-organization process, as demonstrated for instance by van Duppen et al. (2023) for the formose CRN;
2. *Dynamicity*: This evolution is highly time-dependent, generating complex temporal patterns as a function of different chemical inputs. These patterns can then be modulated in a dynamical way by steering a CRN's steady state through variation of the input parameters;
3. *Parallelizability*: In CRNs, chemical reactions occur simultaneously and independently in a massively parallel manner, realizing the processing of a large amount of information concurrently;
4. *Low-energy operability*: Due to the parallelization and autonomous self-organization capabilities of CRNs, these systems can be operated with extremely low external energy. Furthermore, chemical systems exhibit a propensity to favor pathways associated with the lowest overall system energy (if not steered externally) and therefore autonomously populate the kinetically least constrained reaction pathways;
5. *Determinism and reproducibility*: As chemical reactions are defined by the laws of quantum mechanics, a reaction's outcome and its corresponding rate under given conditions are unequivocally defined and should be precisely predictable and reproducible, following deterministic rules instead of stochastic (random) behavior. However, this does not directly translate to the macroscale operation in a real lab. This is subject to macroscopic effects, diffusion, local concentration effects, evaporation, etc., where the unique but convoluted CRN state must still be characterized by appropriate analytical techniques;
6. *Tunability*: A CRN must be tunable and its properties adaptable to different computing tasks.

The design of a chemical reaction system that can transmit signals, self-develop at corresponding non-equilibrium conditions and respond to external and internal triggers that affect the evolution to be used in the process of learning, are all crucial aspects when designing a CRN-based computer. Figure 2 depicts the basic components and a simple assembly of a chemical computer based on CRNs. The chemical processor is fed with an operational protocol derived from mapping real-world input data to reaction input parameters, which comprise the initial chemical composition and the reaction starting conditions. The reaction can then be dynamically controlled and steered by changing the chemical input flows and/or the reaction conditions. At different points in time, intermediates and products are formed, which need to be read out by some type of analytical instrument to collect output signals. For details and challenges associated with encoding and read-out, see Section 2.3.

2.2 Hardware requirements and challenges for physical implementation

Compared to CMOS architectures, characterized by its deterministic and scalable circuit design (rules), 3D assembly and hierarchies with device and redistribution layers, wiring schemes, thermal management and so forth, chemical reactions naturally occur in a liquid environment and are often not solid state in nature. Conceptually, computation of arbitrary complexity has been theoretically demonstrated to be Turing complete in principle, even by using only a small number of different molecular species (Soloveichik et al., 2008). This can be achieved by storing and processing information as integer counts of molecules in a well-mixed solution. If chemical systems are perceived as stochastic, the error probability is reduced at each computing step, and the total error probability can made arbitrarily small by adjusting the initial molecular species count. Then, a stochastic CRN can solve any computational problem—no matter how complex—given enough time and memory. However, a physical implementation of any chemical computing approach into real-world computing devices and systems will represent a disruptive change in design, fabrication and operation compared to existing semiconductor architectures with the following fundamental questions yet to be addressed:

1. Can automation and suitable hardware provide sufficient control over all chemical reactions to reproducibly create identical output states—both qualitative and quantitative—of a CRN?
2. Do CRNs behave chaotically or do their reaction pathways follow certain rules?
3. Can CRNs be cascaded to enable a scalable computing platform?
4. How fast can the system be encoded and what is the typical latency?
5. How can chemical reactions be fueled as reagents are being consumed?
6. What are means to clock a CRN?
7. Does self-organization and self-limitation within a CRN scale or is there any fundamental limitation when miniaturizing it?

An obvious approach to handle and govern control over wet-chemistry is to compartmentalize chemical reactions, e.g., by introducing physical reactor volumes, following similar strategies as found in biological systems and used when the required selectivity cannot be achieved (Ruiz-Mirazo et al., 2014). In that sense, semiconductor architectures and fabrication processes can be highly beneficial as they enable scalable reaction volumes down to *fL* with great flexibility regarding reactor volumes and types (static reactors vs. flow reactors), while offering a high chemical resistivity against corrosive solvents. In addition, the implementation of smallest channels for mass-flow, and ion- and proton-selective materials such as membranes is feasible. Furthermore, microfluidic

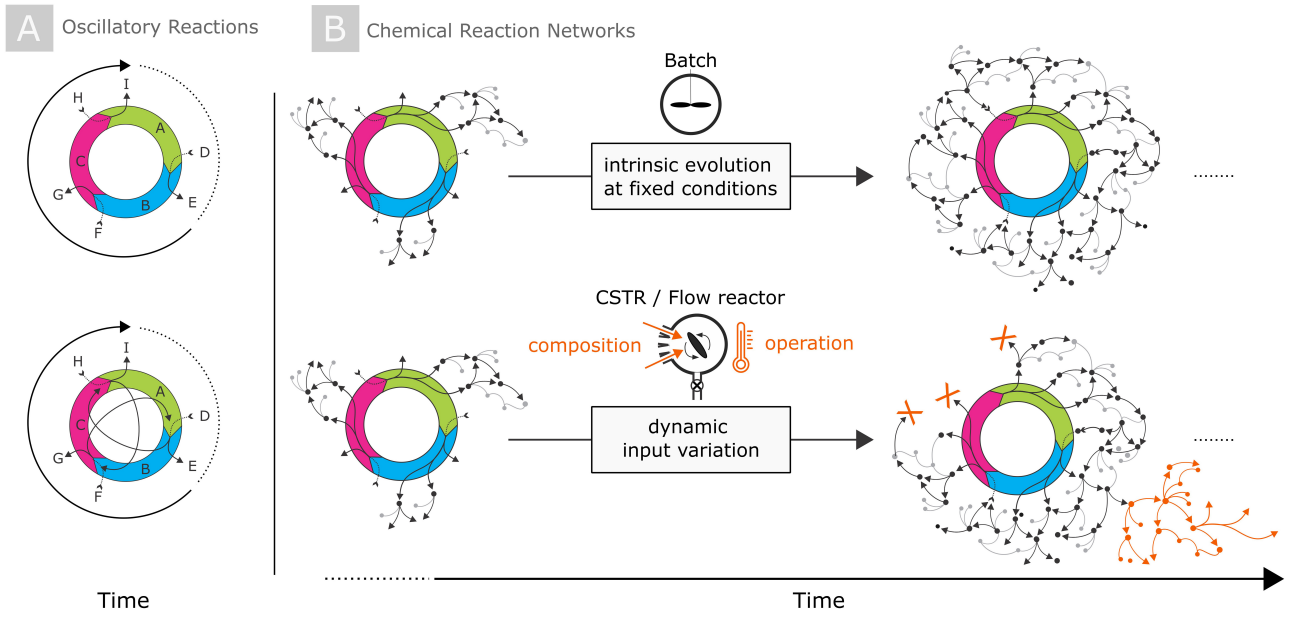


FIGURE 1
Schematic and highly simplified illustrations of different types of reaction networks: **(A)** The simplest form are closed-loop catalytic cycles as, for instance, occurring in oscillatory reactions including the BZ reaction. In these reaction sequences, either a few dominant intermediates are selectively formed **(top)** or, in more complex scenarios, cross-reactions between intermediates of the main reaction cycle occur **(bottom)**. **(B)** In contrast, CRNs comprise highly-interconnected, cascaded reaction sequences that populate individual branches of the reaction topology over time. The time-evolution of CRNs occurs intrinsically at given fixed environmental conditions **(upper row)**, or can be steered by dynamic input variation of the compositional and/or operational input parameters **(bottom row)**.

platforms provide means to control the reaction (e.g., dwell time, temperature, etc.) and to monitor and feedback-control it, for instance through electrode implementation. Highly complex 3D liquid networks can be envisioned that may enable site-selective supply of materials, e.g., to locally feed reactions or to steer the reaction by providing reagents. The precise supply of feedstock molecules is a crucial aspect in chemical computing, as chemical compounds are consumed over time and must be fed for long term operation. Currently, the lifetime of a chemical processor is limited to a few minutes to hours, depending on the CRN's kinetics. Furthermore, silicon-based microfluidics may enable a seamless integration into a CMOS stack or the direct use of CMOS components suitable for controlling and monitoring wet-chemical systems.

For a proof-of-concept, the chemical computer may still be operated manually, involving typical labor-intensive chemical procedures. For repeated use, unavoidable when processing larger data sets, efficient operation can only be achieved if the platform can be operated in full automation, ensuring reproducibility and scriptability of all components including in-line analytical readout, and *in silico* inference. In particular for CRNs where the composition must be very accurately controlled at any time, only a script-based orchestration of liquid handling hardware, reactor operation and analytics can provide a precision suitable for achieving reproducible chemical operations. In addition, handling and disposal of chemicals require compliance with various safety standards. Furthermore, the safe operation and risk assessment relies on prior knowledge and understanding of the intermediates and products formed. Another critical aspect is the realizable computational speed: The typical latency of the CRN, together with intrinsic kinetic properties that determine the reaction speed, can only be modulated to a certain degree, and therefore constitute a severe computing bottleneck. Furthermore, the determination of the CRN state might require time-intensive post-processing steps, e.g., derivatization, separation, etc. to characterize the products both quantitatively and qualitatively. All these parameters are by no means trivial to predict, and must be empirically addressed in time-consuming parameteric studies when designing a new CRN for computing.

2.3 Encoding and readout

Beyond the design and operation of the chemical processor itself, a real-world computing task must be encoded into the chemical world, and the corresponding solution decoded from the properties derived from the chemical system. For that purpose, specific problem-related data is mapped onto a typically rather sparse input subspace, which comprises the chemical composition and the operational input parameters at which the CRN can autonomously self-develop. The non-equilibrium conditions that span this subspace, however, are generally challenging to predict *a priori* and require time-intensive parametric studies of this multidimensional space. To obtain solutions to a given computing problem, suitable analytical methods are required to read out the

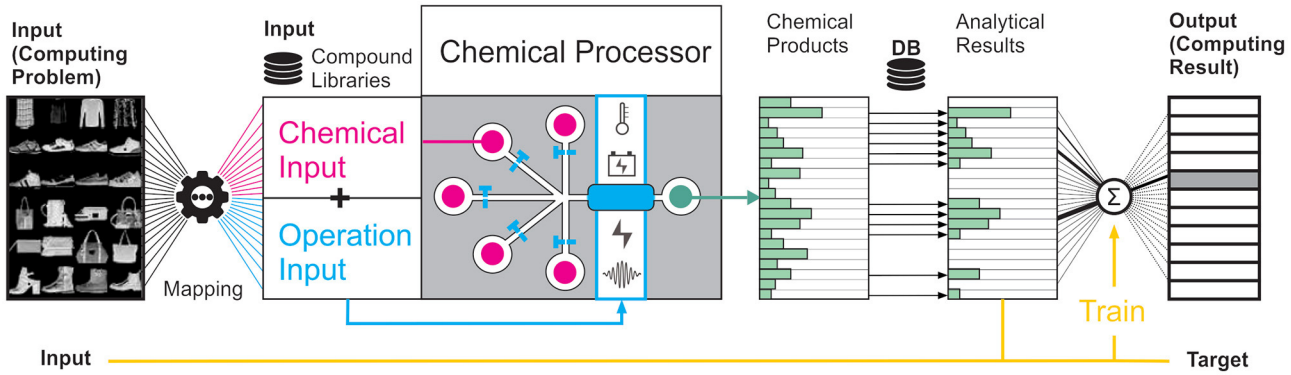


FIGURE 2
Methodological overview of the principle components and processing steps of a chemical computing platform: Problem-specific data is mapped to chemical (pink) and static as well as dynamic operational (blue) inputs for a given computing problem. These inputs and reaction conditions determine the evolution of the CRN, which acts as an information-processing unit based on the underlying CRN-embedded chemical reaction pathways. The product (or intermediate) space of the CRN (type of compounds and quantities thereof) must be analytically identified and quantified using libraries, and chemometric algorithms. All these characteristic features can be used to train a CRN's response to a desired target function *in-silico*, leading to classification, and forward prediction.

complex state of the CRN which is in principle given by the type of products and their concentrations. In case the response and temporal evolution of the CRN is not yet known for the entire parameter space (chemical input + reaction conditions + initial state), high-resolution analytics must be employed to identify and quantify all compounds. These methods are most often based on sample extraction, preparation, separation and physio-chemical sensing modalities, thereby creating a speed bottleneck for computing. Suitable instrumentation includes gas or liquid chromatography, mass spectrometry, trapped-ion mobility, differential ion mobility etc., many of them further need to be combined to achieve a complete picture. All these offline methods cannot directly be incorporated into the computing platform itself due to size limitations and an interruption of the chemical reaction process upon sample extraction. In return, compounds are identifiable through comparison with huge libraries, supported by chemometric algorithms. These quantifications, which can be traced down to parts-per-trillion levels, enable reasoning of the detailed behavior of a CRN, at the cost of speed and ease of operation.

At a later stage with known CRN behavior, such methods are not suitable any more. Instead, a lower analytical resolution is sufficient to determine essential features of the CRN state. Suitable inline/online methods enable direct sensing within the reactor or integration of the sensing component into the computing platform to enable automated and timely measurements without interrupting or terminating the platform operation. These methods comprise, for instance, pH and electrochemical potential measurements, ultraviolet—visible or Fourier-transformed infrared (FTIR) spectroscopy and benefit from real-time data acquisition. Mostly, however, quantitative information is presented in highly convoluted features. For instance, in FTIR spectroscopy, features may be assigned to characteristic functional groups, and the total absorption intensities can be integrated, but not dissected into individual compound's quantities. However, as long as enough of such features are present in the spectra – which is the case for CRNs with high chemical diversity in their output space – the signals represent accumulated concentrations, which are directly proportional to the individual compounds, thereby still representing the CRN state. Furthermore, the analytics may only identify a subset of all compounds generated or properties measured in an even more convoluted way, including for instance (spectrally broad) absorption or emission features, which are still cumulative properties of the analytical matrix. In principle, only state-representative, essential features are required to determine the state of the CRN. Hence fully untargeted fingerprinting may be another reasonable analytical modality for computing at much less experimental effort. Consequently and by its nature, any reasoning of the CRN's chemical composition and behavior will then not be possible anymore but the simpler readout may be better scalable, cheaper and faster while still providing enough features for computing. Hence, all these considerations constitute a trade-off between precision, resolution, speed and ease of use.

The exact timing and sequence of sampling of the CRN state by analytical means is not trivial to assess, and the nature of the data acquisition determines the scope of application. While measurements of the instant response of the CRN are only limited by the speed of instrumental data acquisition, monitoring of an equilibrium or steady state depends on the inherent chemical kinetics that govern reactions and that will lead to the formation of new chemical species. These kinetics dictate a specific evolution time for each type of CRN to unfold its complexity. Subsequently, a single CRN state can be sampled, yielding already enough data for classification tasks. In contrast, the CRN reservoir state can be modulated by static or dynamic changes in the input concentrations too, providing a timely response of the network to evolve and develop under these variations. This allows for monitoring a

time-resolved read-out of the CRN state. This data can be leveraged for various time-dependent computing tasks, such as forward prediction, modeling complex dynamics of biological systems, or solving voice recognition tasks, as the time-dependent CRN input and output can always be mapped to these types of problems.

2.4 Potential application areas

With the chemical computing platform depicted in Figure 2, various applications, both in chemical sciences but also general computing, can be envisioned. Conceptually, the CRN can be considered a material embodiment of a fixed and non-linear type of reservoir, whose properties are considered a black box as being unknown at the beginning. For small networks, the behavior of the reservoir may be mimicked by these AI/Machine Learning algorithms, whose structure and dynamic behavior are explainable. In reservoir computing, input variables are mapped to the dynamics of a fixed system called a reservoir, whose response is then read out by determining its state and mapped to the desired computing solutions. For CRNs, there are multiple features that may represent the state. A major advantage of the reservoir computing approach for CRNs is the comparatively low training effort as weights connecting reservoir nodes do not need to be assigned explicitly, but are chosen randomly such that only the readout-layer is trained. In a CRN, the intermediates and reaction paths connecting these intermediates must hence not be characterized explicitly, which would require the derivation of reaction rate constants. However, if the chemical behavior of the CRN is explainable, it can be encoded as a graph in which compound nodes are connected by weights derived from reaction kinetics, and must not be treated as a black box. However, the mapping of a CRN reservoir's input layer to product output data generates interpretable input–output correlations. These can be used to perform simple classification or optimization tasks, for instance the maximization of product outputs in the chemical discovery sector. In a more long-term vision, they could even be harnessed for the *in-situ* synthesis of drug molecules for personalized patient treatment in the health-care sector. This makes CRNs ideal candidates to forecast the spatiotemporal behavior of dynamic and even chaotic systems.

3 Discussion

The energy-related economic and societal boundary conditions imposed by an ever-increasing energy demand fuel the innovative pressure to design fundamentally new computing approaches. In this perspective, we discussed one emergent, brain-inspired computing paradigm that exploits complex chemical reaction networks as information processing units. Chemical reaction networks are highly nonlinear, energy-efficient, and parallelizable and therefore capable to mimic the information processing capabilities of living systems, whose computing efficiency is still unparalleled. However, an actual physical implementation of a chemical (reservoir) computer poses various challenges associated with operability, encoding of a real-world computing problem, and readout. These operational parameters must be addressed carefully, as they constitute a disruptive design change compared to the predominant semiconductor architectures. Achieving a profound understanding of the time-dependent behavior of complex chemical reaction networks will enable the comprehension of biological reaction networks and help improve automated and yield-optimized retrosynthesis with multiple applications not efficiently tackled by today's computing systems.

Author contributions

K-SC: Conceptualization, Methodology, Visualization, Writing – original draft, Writing – review & editing. EL: Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Visualization, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by the Swiss National Science Foundation – NCCR Molecular Systems Engineering under the grant number 51NF40-205608. Further funding is received from the European Union and the Swiss State Secretariat for Education, Research and Innovation (SERI) under the contract numbers 22.00017 and 22.00034 (Horizon Europe Research and Innovation Project CORENET).

Acknowledgments

The authors acknowledge scientific discussion with A. de la Escosura, W. T. S. Huck, D. Merckle, D. Widmer, R. Lovchik, H. Wolf, A. Valdés, and S. Vela Gallego.

Conflict of interest

K-SC and EL were employed by company IBM Research Europe - Zurich.

Publisher's note

References

de Vries, A. (2023). The growing energy foot print of artificial intelligence. *Joule* 7, 2191–2194. doi: 10.1016/j.joule.2023.09.004

Gizynski, K., and Gorecki, J. (2017). Cancer classification with a network of chemical oscillators. *Phys. Chem. Chem. Phys.* 19, 28808–28819. doi: 10.1039/C7CP05655A

Guo, S. Y., Friederich, P., Cao, Y., Wu, T. C., Forman, C. J., Mendoza, D., et al. (2021). A molecular computing approach to solving optimization problems via programmable microdroplet arrays. *Matter* 4, 1107–1124. doi: 10.1016/j.matt.2021.03.002

Ivanov, N. M., Baltussen, M. G., Regueiro, C. L. F., Derks, M. T. G. M., and Huck, W. T. S. (2023). Computing arithmetic functions using immobilised enzymatic reaction networks. *Angew. Chem. Int. Ed. Engl.* 135:e202215759. doi: 10.1002/ange.202215759

Parrilla-Gutierrez, J. M., Sharma, A., Tsuda, S., Cooper, G. J. T., Aragon-Camarasa, G., Donkers, K., et al. (2020). A programmable chemical computer with memory and pattern recognition. *Nat. Commun.* 11:1442. doi: 10.1038/s41467-020-15190-3

Rambidi, N. G., Kuular, T. O. O., and Makhaeva, E. E. (1998). Information-processing capabilities of chemical reaction–diffusion systems. 1. Belousov–Zhabotinsky media in hydrogel matrices and on solid supports. *Adv Mater Opt Electron.* 8, 163–171. doi: 10.1002/(SICI)1099-0712(1998070)8:4<163::AID-AMO347>3.0.CO;2-A

Rambidi, N. G., and Yakovenchuk, D. (2001). Chemical reaction-diffusion implementation of finding the shortest paths in a labyrinth. *Phys. Rev. E* 63:026607. doi: 10.1103/PhysRevE.63.026607

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., et al. (2019). A deep learning framework for neuroscience. *Nat. Neurosci.* 22, 1761–1770. doi: 10.1038/s41593-019-0520-2

Robinson, W. E., Daines, E., van Duppen, P., de Jong, T., and Huck, W. T. S. (2022). Environmental conditions drive self-organization of reaction pathways in a prebiotic reaction network. *Nat. Chem.* 14, 623–631. doi: 10.1038/s41557-022-00956-7

Ruiz-Mirazo, K., Briones, C., and de la Escosura, A. (2014). Prebiotic systems chemistry: new perspectives for the origins of life. *Chem. Rev.* 114, 285–366. doi: 10.1021/cr2004844

Soloveichik, D., Cook, M., Winfree, E., and Bruck, J. (2008). Computation with finite stochastic chemical reaction networks. *Nat. Comput.* 7, 615–633. doi: 10.1007/s11047-008-9067-y

Tsompanas, M. A., Fyrigos, I. A., Ntinas, V., Adamatzky, A., and Sirakoulis, G. C. (2021). Light sensitive Belousov–Zhabotinsky medium accommodates multiple logic gates. *Biosystems* 206:104447. doi: 10.1016/j.biosystems.2021.104447

van Duppen, P., Daines, E., Robinson, W. E., and Huck, W. T. S. (2023). Dynamic environmental conditions affect the composition of a model prebiotic reaction network. *J. Am. Chem. Soc.* 145, 7559–7568. doi: 10.1021/jacs.3c00908

Zador, A., Escola, S., Richards, B., Ölveczky, B., Bengio, Y., Boahen, K., et al. (2023). Catalyzing next-generation artificial intelligence through neuroai. *Nat. Commun.* 14:1597. doi: 10.1038/s41467-023-37180-x

frontiers | Frontiers in Neuroscience

TYPE Perspective
PUBLISHED 17 April 2024
DOI 10.3389/fnins.2024.1394271

OPEN ACCESS

EDITED BY
Pier Luigi Gentili,
Università degli Studi di Perugia, Italy

REVIEWED BY
Haitong Li,
Purdue University, United States
Max Talanov,
Institute for Artificial Intelligence R&D, Serbia

*CORRESPONDENCE
S. J. Ben Yoo
✉ sbyoo@ucdavis.edu

RECEIVED 01 March 2024
ACCEPTED 08 April 2024
PUBLISHED 17 April 2024

CITATION
El Srouji L, Abdelghany M, Ambethkar HR, Lee Y-J, Berkay On M and Yoo SJB (2024) Perspective: an optoelectronic future for heterogeneous, dendritic computing. *Front. Neurosci.* 18:1394271. doi: 10.3389/fnins.2024.1394271

Perspective: an optoelectronic future for heterogeneous, dendritic computing

Luis El Srouji, Mahmoud Abdelghany, Hari Rakul Ambethkar, Yun-Jhu Lee, Mehmet Berkay On and S. J. Ben Yoo*

Department of Electrical and Computer Engineering, University of California, Davis, Davis, CA, United States

With the increasing number of applications reliant on large neural network models, the pursuit of more suitable computing architectures is becoming increasingly relevant. Progress toward co-integrated silicon photonic and CMOS circuits provides new opportunities for computing architectures with high bandwidth optical networks and high-speed computing. In this paper, we discuss trends in neuromorphic computing architecture and outline an optoelectronic future for heterogeneous, dendritic neuromorphic computing.

KEYWORDS

neuromorphic computing, silicon photonic computing, dendritic computing, heterogeneous computing, analog computing

1 Introduction

Carver Mead introduced the term "neuromorphic" in 1990 in an invited article where he explained the inherent wastefulness of digital computation. In brief, he argued that each operation in a digital system requires the switching of about 10,000 transistors and that much of the power required to switch these transistors is actually due to the excess capacitance on each gate caused by wiring between each transistor. To reduce these problems, Mead (1990) argued that computing algorithms should be designed for less data movement and that engineers should use the natural properties of devices to perform various operations. Despite these arguments, the success of digital computers based on the von Neumann architecture continued to grow and dominate the market into the present day (Backus, 1978).

A human brain, on the other hand, is a highly parallelized computing system whose analog dynamics offer many advantages for high-performance computing. It is estimated that the human brain can process up to 10^{23} operations every second compared to the roughly 10^{9} operations per second possible with a traditional computer based on the von Neumann architecture (Thagard, 2002). Despite this fact, a desire for deterministic components has enforced a preference for digital circuits in computer architecture. Meanwhile, biological neural networks are surprisingly noise-tolerant despite synaptic efficacies as low as 20% (Stevens and Wang, 1994). Nonetheless, as the trend of Moore's Law (Theis and Wong, 2017) wanes, further advancements in computing can no longer rely on increasing transistor speeds and density. As a result, general-purpose computing systems are expected to be increasingly replaced by application-specific integrated circuits (ASICs) in various compute-intensive applications, including neural networks (Solli and Jalali, 2015; Ranganathan, 2020). While vectorized tensor processing units (TPUs) and graphical processing units (GPUs) have made traditional deep neural network (DNN) architectures more practical on digital systems, carefully designed analog and mixed-signal ASICs can often offer improvements to throughput and system latency while reducing power consumption.

Recent progress in the area of silicon photonics has pushed industry leaders such as GlobalFoundries to develop a co-integrated process design kit (PDK)—labeled GF 45SPCLO—that allows circuit designers to place photonic elements and CMOS circuits on the same physical substrate (Rakowski et al., 2020). This process opens new doors for optoelectronic ASICs that employ silicon photonic elements for high-bandwidth data communication networks (Beausoleil, 2011) alongside CMOS electronic circuits for high-speed computing structures (Hassan et al., 2023).

In the following manuscript, we will highlight two important features of biological neural networks from the perspective that co-integrated photonic and electronic technologies are key to the future of neuromorphic computing. Section 2 will begin by describing the advantages of heterogeneous neural dynamics and discuss the limited number of neuromorphic devices that incorporate this heterogeneity. Next, Section 3 will discuss the computational properties of biological dendrites and review existing approaches to implement dendritic computing. Finally, we discuss a proposed optoelectronic chiplet architecture that is capable of supporting these features in a scalable neuromorphic system.

2 Heterogeneous neural dynamics

Artificial neural networks employ a variety of nonlinear transformations (activation functions) to guide a model into choosing an efficient encoding for a given task (Rasamoelina et al., 2020; Mercioni and Holban, 2023). While artificial neural networks are considered to be universal function approximators (Hornik et al., 1989; Lu et al., 2017), it is well known empirically that the choice of nonlinearity can be pivotal to the success of the DNN. Similarly, it is known that biological neurons display a wide range of dynamic behaviors—see Figure 1C for examples of three typical behaviors of cortical neurons according to the Izhikevich model (Izhikevich, 2003). However, when designing hardware accelerators for neuromorphic computing, engineers must decide what level of specificity or generality is needed to support the various neural dynamics required for the most common DNNs. Following Mead's argument (Mead, 1990), it is far more efficient to use the natural physical properties of a device to provide nonlinear behavior, but these properties are often fixed after fabrication and impossible to program. Even in the case of digital systems, general-purpose computing elements necessarily consume more physical resources and power in order to serve the generic case.

For example, Intel's Loihi processor (Davies et al., 2018) is designed as a many-core digital system where each "neurocore" asynchronously updates a set of internal variables using a limited digital core. In the first iteration of the chip, the internal updates followed a fixed schedule and computed discrete updates to a current-based, leaky-integrate-and-fire (CUBA LIF) model that was not programmable. Loihi 2, however, included a micro-code programmable neuron that allows an arbitrary neuron model to be implemented as long as its instructions fit in the core's local memory (Orchard et al., 2021). Each core on Loihi 2 has the same memory size and must contain all the necessary parameters for neurons and synapses on that core. This means that more complicated neuron models limit the maximum neuron density of the core, and neurons which use different micro-code models must be implemented on separate neurocores. Nonetheless, Intel invested in this generality despite the additional implementation complexity because of the expectation that heterogeneous neural dynamics would hold several key computational benefits.

Biological neural networks are remarkably heterogeneous regarding individual neuron dynamics and morphological structure. Koch and Laurent (1999) argues that this heterogeneity is a direct consequence of the complexity of behaviors and sensory modalities that the brain must handle. To establish whether this heterogeneity is advantageous or purely an evolutionary epiphenomenon, several analyses have compared neural network structures with and without heterogeneity and shown that the variability of neural responses in heterogeneous populations increases the sensitivity of a population code and, therefore, improves the precision at which it can be read out (Shamir and Sompolinsky, 2006; Chelaru and Dragoi, 2008; Marsat and Maler, 2010). Population codes are often associated with sensory stimuli because of their ability to handle noisy input (Averbeck et al., 2006), and these results show that heterogeneity may be key to balancing sensitivity and signal-to-noise ratio. Perez-Nieves et al. (2021) also showed that the heterogeneity of synaptic time constants in a reservoir network improved generalization, robustness to hyperparameters, and overall learning performance. While these results were demonstrated by digital simulation, Mead's argument should remind us that an architecture that uses the natural dynamics of its computing elements would be more efficient than a digital emulator to implement a heterogeneous neural network.

2.1 Optoelectronic neurons

A number of efforts have been made to design analog photonic neurons by drawing a bijection between semiconductor laser and amplifier dynamics and the dynamics of a LIF neuron model (Tait et al., 2014; Prucnal et al., 2016). These efforts are motivated by the advantages in bandwidth and throughput of silicon photonic interconnects (Miller, 2000, 2009; Agarwal et al., 2019; Huang et al., 2022)—a crucial advantage considering that human synaptic fan-out is on the order of 10,000 synapses per neuron. However, optical dynamics are controlled by material parameters that are fixed after device fabrication and are mostly unprogrammable (i.e., carrier lifetimes in the gain medium of a laser). As a result, photonic neurons may not be sufficient to replicate the breadth of heterogeneity found in biological neural networks. An optoelectronic approach, however, may be more feasible.

Electronics are preferable for designing programmable circuits given the long history of well-developed design principles for CMOS circuits, and many such programmable circuit models have been demonstrated (Indiveri et al., 2011). An optoelectronic neuron would combine this programmability with the benefits of optical interconnects. Under this approach, photodetectors transduce optical signals into electrical currents and are analogous to the synaptic receptors in a biological neuron. These currents are collected by a capacitor in the circuit analogous to the membrane capacitance. A CMOS circuit behaves like the neuron soma and provides the feedback dynamics that generate the neuron's excitable

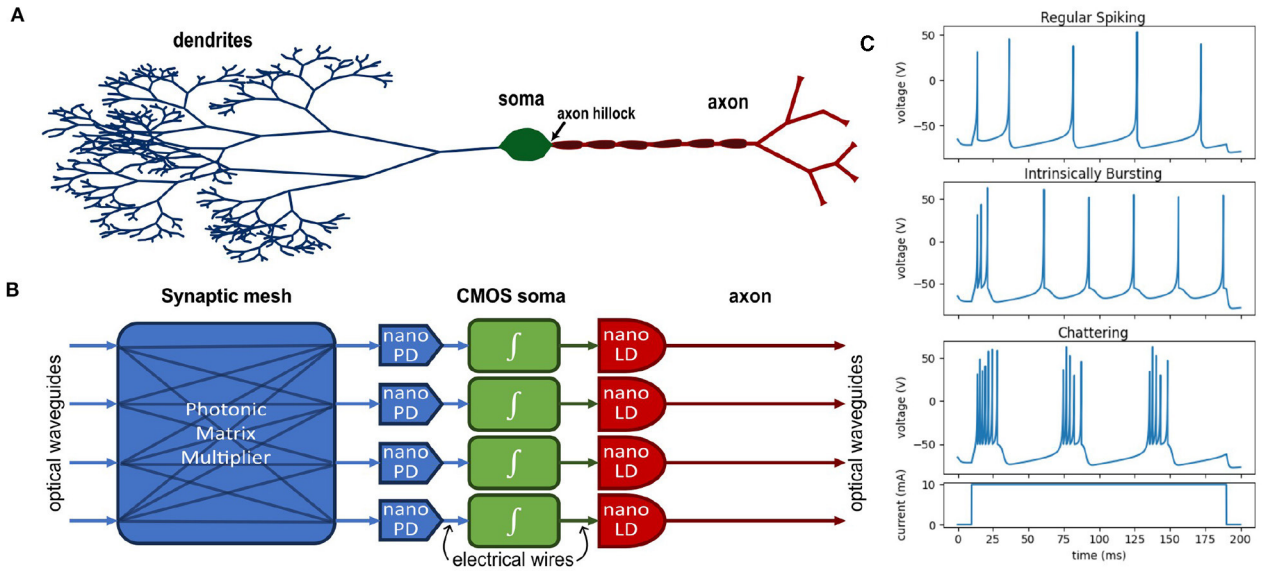


FIGURE 1
(A) Diagram of a biological neuron compared to **(B)** a four-neuron layer of an optoelectronic neural network—a nanophotodetector (nano PD) and nanolaser (nano LD) perform signal transduction. **(C)** Examples of various neural dynamics according to the Izhikevich neuron model (Izhikevich, 2003).

(spiking) behavior. In the biological neuron, the membrane potential is only propagated to the axon when the activity near the axon hillock reaches a threshold; similarly, in the optoelectronic neuron, a CMOS amplifier drives a laser only when the neuron is spiking. Figures 1A, B show a functional comparison between a biological neuron (A) and a block diagram of the optoelectronic neuron (B).

Few have demonstrated such optoelectronic neurons with spiking dynamics, and those existing implementations show limited or no programmable dynamics (Balle et al., 2013; Tait et al., 2015). More recently, Lee et al. (2024) demonstrated a programmable spiking optoelectronic neuron using the GF 45SPCLO PDK. However, because of the lack of on-chip lasers available in this process, an off-chip vertical cavity laser was externally connected to the neuron. The neuron efficiency was projected to improve on a more advanced CMOS node and with an on-chip micro-scale laser such as a low-threshold ring laser (Liang et al., 2016). Despite the strengths of this optoelectronic approach, a new process that can reliably integrate on-chip lasers alongside these CMOS and silicon photonic circuits is required to make packaging more feasible. Existing implementations of photonic matrix multipliers have large footprints, where each matrix element requires a roughly 900 μm^2 area (Ramey, 2020; Feldmann et al., 2021). This limits the number of synaptic connections and neurons that are available on a single chip. As such, an advanced packaging scheme is needed for photonic and optoelectronic neural networks to be practical at the scales of modern DNNs. A 3D photonic-electronic packaging scheme has been proposed for this purpose in which chiplets are stacked and tiled onto an interposer using a combination of 3D photonic and electronic interconnects (Zhang et al., 2020). Early results have been demonstrated in other application contexts (Chang et al., 2023); however, a complete photonic-electronic neuromorphic chiplet network has not yet been demonstrated.

3 Dendritic computing

Despite the variety of nonlinearities mentioned in Section 2, the vast majority of DNNs rely on a point-neuron model that lacks the temporal and spatial complexity of a biological neuron. Under this limited model, synaptic integration and nonlinearities across the network are considered instantaneous. Meanwhile, biological neurons vary so widely in morphology and dynamics that a standard taxonomy for neuron classification has yet to be established (Zeng and Sanes, 2017). For example, pyramidal neurons in Layer V are distinct from those in Layer II/III of the human cortex and carry distinct properties for synaptic integration even within the same cortical area (Spruston, 2008). This diversity in biological networks has led to the hypothesis that the increased spatio-temporal complexity could be related to several major advantages of biological neural networks over the modern implementation of DNN (Acharya et al., 2022).

Various ion channels line the cell membrane along these dendritic branches (see Figure 2A), leading to both passive and active effects that can be modeled by cable theory (Koch, 1984). In the passive case, propagation along the dendrite is often compared to a lossy transmission line that propagates a signal with both attenuation and dispersion (Dayan and Abbott, 2001). Neurons with dendrite models are also more expressive than point neurons; Acharya et al. (2022) summarize three major features of dendrite models: weight plasticity, delay plasticity, and structural plasticity. Weight plasticity is the synaptic strength as modeled in DNNs. In contrast, the delay and structural plasticities are unique features

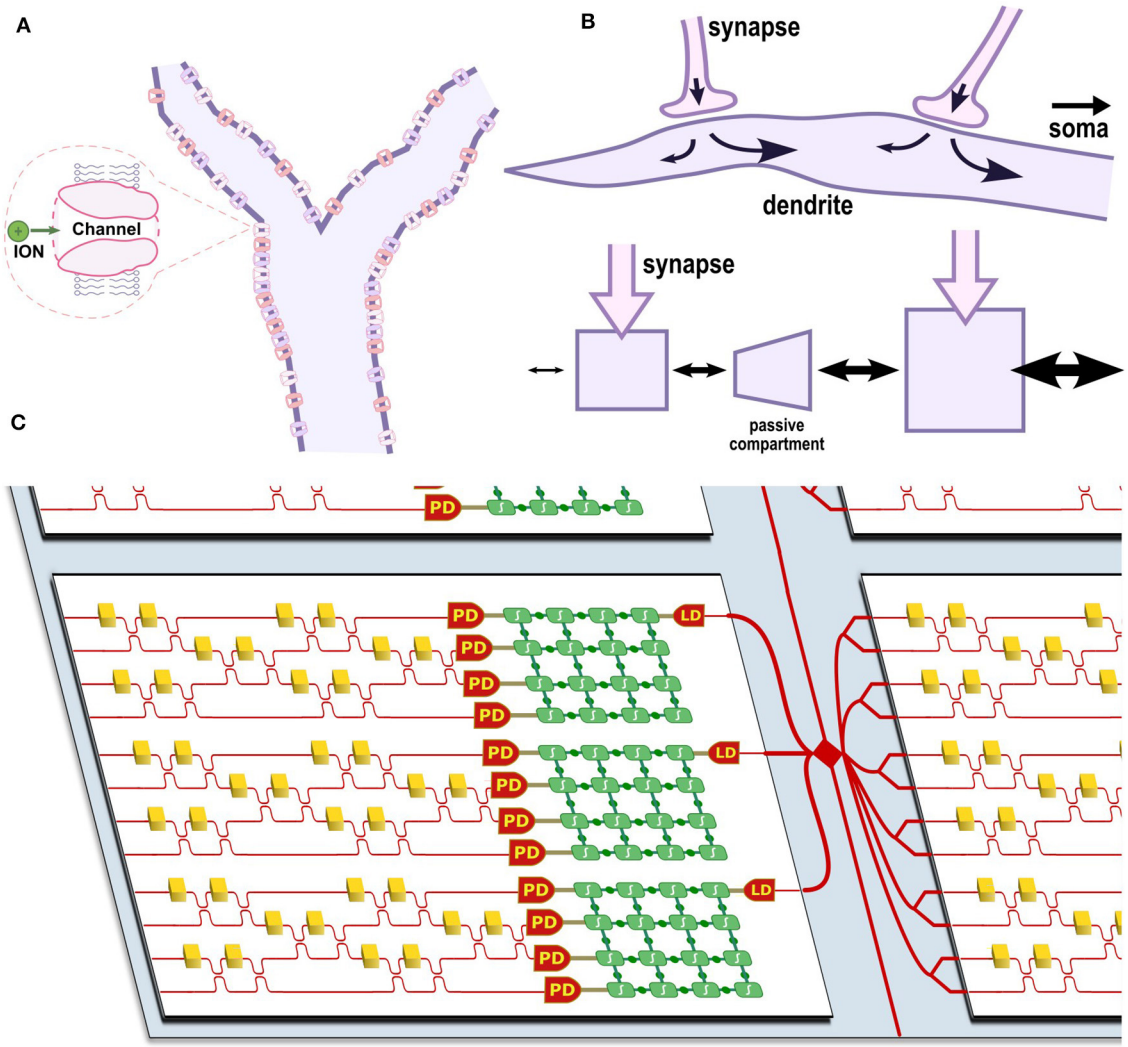


FIGURE 2
(A) Diagram of a dendritic branch with various ion channels. **(B)** Diagram of current flow between two synapses along a dendrite and an equivalent multi-compartment model where current flow is indicated by arrow size. **(C)** Diagram of a photonic-electronic chiplet architecture for optoelectronic dendritic computing. Each chiplet (white) shows several multi-compartment optoelectronic neurons and is coupled to other chiplets through an interposer (blue) containing an inter-chip routing network (red). The neurons contain multiple programmable CMOS blocks (green) interconnected via a switch matrix; leaf nodes receive optical input through a photodiode (PD), and root nodes emit optical output through a laser diode (LD). A Mach-Zehnder Interferometer mesh is shown as an example photonic synaptic mesh.

of the dendrite that allow the dendrite to process sequences of information (discussed in Section 3.2).

3.1 Multi-compartment models

A multi-compartment model can be used to discretize the dendritic tree into localized segments under which the membrane dynamics are considered uniform. Such a model is necessary for neuromorphic computing because the cable dynamics of a biological neuron membrane are too complex to model in a continuous manner. Multiple software libraries such as the NEURON (Migliore et al., 2006) and Brian 2 (Stimberg et al., 2019) simulators have been written to model networks with such neurons. Each local compartment model includes active dynamics and membrane leakage currents, while a conductive channel models the diffusive, axial current flow between compartments. This model can be summarized by the following equation where V_m represents the localized membrane potential, C_m represents the localized membrane capacitance, V_i indexes the connected compartments, $g_k(t)$ indexes synaptic conductances, E_k indexes the reversal potential associated with a given synaptic conductance, and $F(V_m, t)$ summarizes the local membrane dynamics:

$$C_m \frac{dV_m}{dt} = F(V_m, t) + \sum_k g_k(t)(E_k - V_m) + \sum_i g_i(V_i - V_m) \quad (1)$$

Equation (1) highlights the similarities and differences between synaptic currents and dendritic currents. When a signal is received on the synapse, its conductance increases, and a current is generated according to the voltage difference between membrane

potential and the equilibrium potential related to the given synaptic receptor. This equilibrium potential is often at the extrema of possible membrane potentials, and thus, a given synapse results in a polarizing or depolarizing current nearly exclusively.

In contrast, the conductance between dendritic compartments is time-independent, and the current flow direction can fluctuate depending on which compartment has the higher membrane potential at a given moment. The activity at each compartment is similar to the point model—a nonlinear response to a weighted sum—though the location of each compartment changes the efficacy and delay of its effect on the soma. By studying detailed models of hippocampal CA1 cells, Poirazi and Papoutsi (2020) showed that the complexity of these multi-compartment models could only be mapped onto a two-layer ANN, indicating that a single biological neuron is equivalent to a two-layer neural network. This result, however, only considers the rate-coded behavior of the neuron and does not fully capture the spike-timing sensitivity of dendritic models.

Biological dendrites are also tapered so that diffusive, axial currents preferentially flow in the feed-forward direction (toward the soma). This can be modeled as a decreasing resistance (increasing conductance) between compartments close to the soma. Figure 2B shows how this model captures the behavior of a dendrite with two synapses. The tapered end of the dendrite is shown on the left, and the increase in cross-sectional diameter toward the right of the dendrite allows more current flow toward the soma (not pictured) to the right. Note that the arrows shown in the figure indicate the directional preference for diffusive currents and do not represent static current flow. Similarly, the multi-compartment model has two compartments with synapses and a passive compartment in between that models the passive length of dendrite between the two synapses in the biological neuron. The following subsection discusses the consequences of this feature for sequence processing and highlights some early attempts at dendritic computing architectures.

3.2 Temporal complexity

Because the point neuron model lacks temporal dependence, machine learning tasks involving sequential data require carefully designed neural network models. Recurrent neural networks (RNNs) fake temporal complexity and memory by applying the network repeatedly for some number of simulated time steps (Elman, 1991; Lipton et al., 2015). Information is retained in memory based on feedback pathways, but each point neuron could instantaneously influence the output (Sutskever et al., 2014). Convolutional neural networks (CNNs) have also been used for processing sequences because a kernel could be used to detect a feature at any position in the input sequence. However, both RNNs and CNNs exhibit poor scaling properties, leading to difficulties in handling data with long sequences (Werbos, 1990; Kolen and Kremer, 2010). To combat this limitation, transformer models (Vaswani et al., 2017), use an attention mechanism that allows the network to process an entire sequence in a fixed number of operations while maintaining temporal dependence—future information is not available in the past. Because the network is designed for a fixed number of operations, its maximum sequence length—or context window—is inherently limited. Multiple solutions have been proposed to circumvent this limit (Ren et al., 2022; Hatamizadeh et al., 2023), but these solutions all aim to implement sequential processing on a model lacking temporal complexity.

In contrast, biological neuron dynamics show a wide range of temporal complexity. In addition to the heterogeneity of neural dynamics discussed in Section 2, the spatially distributed morphology of biological neurons gives rise to temporal delays that offer an additional dimension of encoding information: temporal ordering. A single-compartment neuron model is sensitive to the timing between incoming synaptic signals but not their order. In contrast, the aforementioned tapered geometry of biological dendrites allows a distinction between the stimulation of two synapses in the forward direction compared to the reverse direction, corresponding to a distinct temporal order.

Nease et al. (2012) first demonstrated a mapping of the cable model to reconfigurable analog CMOS blocks on a computing architecture known as the Field-Programmable Analog Array (FPAA). The device uses floating gates to set a switch matrix and control the flow of currents between computational analog blocks (CAB)—see Basu et al. (2010) for more details. Using these CABs, the architecture was able to accurately replicate the dynamics of a passive length of dendrite within its linear regime and also demonstrate favorable computational properties in the nonlinear regime. George et al. (2013) applied this architecture toward modeling a Hidden Markov Model (HMM) for word spotting. In this demo, the tapered effect of the dendrite was also modeled, allowing for the detection of syllables in a word only when presented with the correct sequential ordering. Because these temporal dependencies were computed using the passive transmission properties of the dendrite cable, the devices showed $> 1,000\times$ improvement in multiply-and-accumulate operations per Watt (MACs/W)—when compared to an equivalent HMM implemented on a digital system.

Boahen (2022) proposed a similar dendritic architecture in which several ferroelectric domains control the gate of a transistor. Under this architecture, the ferroelectric domains align only when a sequence arrives in the correct order; voltage is applied at the transistor source terminal while current is read out at the drain to form a temporal order detector, much like the dendrite. Boahen also argues that sequential encodings can sparsify communication because each pulse in a layer of N dendritic units represents a base-N digit and thus conveys $log_2(N)$ bits. As a result, Boahen argues that this architecture reduces the heat generated by an on-chip network and provides a more suitable architecture for 3D integrated electrical circuits.

These devices show how dendritic models provide a new dimension for encoding and decoding information that can reduce the power constraints of neural networks. In each of these examples, however, it is assumed that the dendrite is deliberately programmed to be selective to some sequence. Neither of the two architectures describes a method for learning or training the sequential encoding.

4 Discussion

An optoelectronic approach to neuromorphic computing is better suited to provide the interconnect bandwidths necessary to support the neuronal fan-in and fan-out required to model neural networks at biological scales while also allowing for flexible and programmable neural dynamics. The density of optoelectronic neurons may be limited due to the relatively larger scale of photonic devices compared to CMOS circuits. However, because neurons with dendritic trees are functionally similar to a two-layer neural network, the incorporation of a CMOS dendrite network would counteract these limitations by providing increased expressivity to each neuron. Additionally, a dendritic tree offers additional architectural flexibility to represent high-fan-in, low-fan-out functional units as dendritic compartments while low-fan-in, high-fan-out units are represented as neuron somas. As a result, an optoelectronic, dendritic-computing architecture is likely the key to advances in large-scale neuromorphic computing.

Figure 2 shows a diagram of an optoelectronic chiplet architecture that captures the advantages of neural heterogeneity and dendritic structures. A programmable electronic switch matrix connects analog blocks that model active and passive dendritic compartments. A photonic matrix multiplier—such as a Mach-Zehnder Interferometer mesh—forms the receiving synaptic mesh and serves as input to a number of dedicated leaf nodes containing photodetectors. Each dendritic tree would also contain a dedicated root node that models the soma and drives a laser output. Each chiplet would contain several multi-compartment optoelectronic neurons and be coupled through a shared photonic-electronic interposer, which provides a routing mesh between many chiplets. Using this architecture, neural networks could be emulated with much greater biological accuracy and at much lower power than existing neuromorphic solutions.

This optoelectronic approach to heterogeneous, dendritic neuromorphic computing would make the vision of brain-scale neuromorphic computing more feasible. However, this architecture relies on the development of packaging methods for 3D photonic and electronic integrated circuits, though an increasing number of challenges for scaling contemporary electronic systems is likely to provide a shared motivation toward the development of such integration methods. Alongside these packaging methods, more work is needed to determine an optimal number of dendritic compartments and an optimal analog model that concisely captures all of the relevant membrane dynamics of the neuron.

5 Conclusion

Biological neural networks benefit from heterogeneous neural dynamics and dendrite morphology that have been largely unexplored in hardware accelerators. An optoelectronic approach can implement high-bandwidth communication networks *and* programmable dynamical systems to provide a "best-of-both-worlds" solution for implementing biological complexity in neuromorphic computing architectures. More work is needed to optimize the architecture and computing model however, the stark contrast in the energy efficiency of human brains compared to modern computing systems offers substantial motivations to pursue novel computing methods.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

LE: Writing – original draft, Writing – review & editing. MA: Writing – review & editing. HA: Writing – review & editing. Y-JL: Writing – review & editing. MB: Writing – review & editing. SY: Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2021-21090200004]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government was authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Conflict of interest

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

Publisher's note

References

Acharya, J., Basu, A., Legenstein, R., Limbacher, T., Poirazi, P., Wu, X., et al. (2022). Dendritic computing: branching deeper into machine learning. *Neuroscience* 489, 275–289. doi: 10.1016/j.neuroscience.2021.10.001

Agarwal, S., Jacobs-Gedrim, R. B., Bennett, C., Hsia, A., Heukelom, M. S. V., Hughart, D., et al. (2019). "Designing and modeling analog neural network training accelerators," in *2019 International Symposium on VLSI Technology, Systems and Application, VLSI-TSA* (Hsinchu). doi: 10.1109/VLSI-TSA.2019.8804680

Averbeck, B. B., Latham, P. E., and Pouget, A. (2006). Neural correlations, population coding and computation. *Nat. Rev. Neurosci.* 7, 358–366. doi: 10.1038/nrn1888

Backus, J. (1978). Can programming be liberated from the von Neumann style? *Commun. ACM* 21, 613–641. doi: 10.1145/359576.359579

Balle, S., Figueiredo, J. M. L., Javaloyes, J., Romeira, B., Ironside, C. N., Piro, O., et al. (2013). Excitability and optical pulse generation in semiconductor lasers driven by resonant tunneling diode photo-detectors. *Opt. Express* 21, 20931–20940. doi: 10.1364/OE.21.020931

Basu, A., Brink, S., Schlottmann, C., Ramakrishnan, S., Petre, C., Koziol, S., et al. (2010). A floating-gate-based field-programmable analog array. *IEEE J. Solid-State Circuits* 45, 1781–1794. doi: 10.1109/JSSC.2010.2056832

Beausoleil, R. G. (2011). Large-scale integrated photonics for high-performance interconnects. *J. Emerg. Technol. Comput. Syst.* 7, 326–327. doi: 10.1109/PHO.2011.6110559

Boahen, K. (2022). Dendrocentric learning for synthetic intelligence. *Nature* 612, 43–50. doi: 10.1038/s41586-022-05340-6

Chang, P. H., Samanta, A., Yan, P., Fu, M., Zhang, Y., On, M. B., et al. (2023). A 3D integrated energy-efficient transceiver realized by direct bond interconnect of co-designed 12 nm finfet and silicon photonic integrated circuits. *J. Lightwave Technol.* 41, 6741–6755. doi: 10.1109/JLT.2023.3291704

Chelaru, M. I., and Dragoi, V. (2008). Efficient coding in heterogeneous neuronal populations. *Proc. Nat. Acad. Sci.* 105, 16344–16349. doi: 10.1073/pnas.0807744105

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Dayan, P., and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: The MIT Press.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* 7, 195–225. doi: 10.1007/BF00114844

Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., et al. (2021). Parallel convolutional processing using an integrated photonic tensor core. *Nature* 589, 52–58. doi: 10.1038/s41586-020-03070-1

George, S., Hasler, J., Koziol, S., Nease, S., and Ramakrishnan, S. (2013). Low power dendritic computation for wordspotting. *J. Low Power Electron. Appl.* 3, 73–98. doi: 10.3390/jlpea3020073

Hassan, A., Saha, S., and Carusone, A. C. (2023). "Fully Integrated photonic dot-product engine in 45-nm SOI CMOS for photonic computing," in *2023 IEEE Silicon Photonics Conference (SiPhotonics)* (Washington, DC: IEEE), 1–2. doi: 10.1109/SiPhotonics55903.2023.10141931

Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., and Molchanov, P. (2023). "Global context vision transformers," in *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, HI: PMLR), 12633–12646.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.* 2, 359–366. doi: 10.1016/0893-6080(89)90020-8

Huang, C., Sorger, V. J., Miscuglio, M., Al-Qadasi, M., Mukherjee, A., Lampe, L., et al. (2022). Prospects and applications of photonic neural networks. *Adv. Phys. X* 7:1981155. doi: 10.1080/23746149.2021.1981155

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:9202. doi: 10.3389/fnins.2011.00073

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Koch, C. (1984). Cable theory in neurons with active, linearized membranes. *Biol. Cybern.* 50, 15–33. doi: 10.1007/BF00317936

Koch, C., and Laurent, G. (1999). Complexity and the nervous system. *Science* 284, 96–8. doi: 10.1126/science.284.5411.96

Kolen, J. F., and Kremer, S. C. (2010). "Gradient flow in recurrent nets: the difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks*, eds S. C. Kremer, and J. F. Kolen (New York, NY: IEEE Press), 237–374. doi: 10.1109/9780470544037

Lee, Y.-J., On, M. B., Srouji, L. E., Zhang, L., Abdelghany, M., Yoo, S. B., et al. (2024). Demonstration of programmable brain-inspired optoelectronic neuron in photonic spiking neural network with neural heterogeneity. *J. Lightwave Technol.* 1–12. doi: 10.1109/JLT.2024.3368450. [Epub ahead of print].

Liang, D., Huang, X., Kurczveil, G., Fiorentino, M., and Beausoleil, R. G. (2016). Integrated finely tunable microring laser on silicon. *Nat. Photonics* 10, 719–722. doi: 10.1038/nphoton.2016.163

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv* [preprint]. arXiv:1506.00019. doi: 10.48550/arXiv.1506.00019

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). "The expressive power of neural networks: a view from the widt," in *Advances* in *Neural Information Processing Systems, Vol. 30* (Long Beach, CA), 1097–1105.

Marsat, G., and Maler, L. (2010). Neural heterogeneity and efficient population codes for communication signals. *J. Neurophysiol.* 104, 2543–55. doi: 10.1152/jn.00256.2010

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636. doi: 10.1109/5.58356

Mercioni, M. A., and Holban, S. (2023). "A brief review of the most recent activation functions for neural networks," in *2023 17th International Conference on Engineering of Modern Electric Systems, EMES* (Oradea). doi: 10.1109/EMES58375.2023.10171705

Migliore, M., Cannia, C., Lytton, W. W., Markram, H., and Hines, M. L. (2006). Parallel network simulations with NEURON. *J. Comput. Neurosci.* 21, 119–129. doi: 10.1007/s10827-006-7949-5

Miller, D. A. (2000). Rationale and challenges for optical interconnects to electronic chips. *Proc. IEEE* 88, 728–749. doi: 10.1109/5.867687

Miller, D. A. (2009). Device requirements for optical interconnects to silicon chips. *Proc. IEEE* 97, 1166–1185. doi: 10.1109/JPROC.2009.2014298

Nease, S., George, S., Hasler, P., Koziol, S., and Brink, S. (2012). Modeling and implementation of voltage-mode CMOS dendrites on a reconfigurable analog platform. *IEEE Trans. Biomed. Circuits Syst.* 6, 76–84. doi: 10.1109/TBCAS.2011.2163714

Orchard, G., Frady, E. P., Rubin, D. B. D., Sanborn, S., Shrestha, S. B., Sommer, F. T., et al. (2021). "Efficient neuromorphic signal processing with Loihi 2," in *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation* (Coimbra), 254–259. doi: 10.1109/SiPS52927.2021.00053

Perez-Nieves, N., Leung, V. C., Dragotti, P. L., and Goodman, D. F. (2021). Neural heterogeneity promotes robust learning. *Nat. Commun.* 12, 1–9. doi: 10.1038/s41467-021-26022-3

Poirazi, P., and Papoutsi, A. (2020). Illuminating dendritic function with computational models. *Nat. Rev. Neurosci.* 21, 303–321. doi: 10.1038/s41583-020-0301-7

Prucnal, P. R., Shastri, B. J., de Lima, T. F., Nahmias, M. A., and Tait, A. N. (2016). Recent progress in semiconductor excitable lasers for photonic spike processing. *Adv. Opt. Photonics* 8:228. doi: 10.1364/AOP.8.000228

Rakowski, M., Meagher, C., Nummy, K., Aboketaf, A., Ayala, J., Bian, Y., et al. (2020). 45nm "CMOS - silicon photonics monolithic technology (45CLO) for next-generation, low power and high speed optical interconnects," in *Optical Fiber Communication Conference (OFC) 2020*, (Washington, DC: Optica Publishing Group), T3H.3. doi: 10.1364/OFC.2020.T3H.3

Ramey, C. (2020). "Silicon photonics for artificial intelligence acceleration: Hotchips 32," in *2020 IEEE Hot Chips 32 Symposium, HCS 2020* (Palo Alto, CA). doi: 10.1109/HCS49909.2020.9220525

Ranganathan, P. (2020). Technical perspective: asic clouds: specializing the datacenter. *Commun. ACM* 63, 102–102. doi: 10.1145/3399738

Rasamoelina, A. D., Adjailia, F., and Sincak, P. (2020). "A review of activation function for artificial neural network," in *SAMI 2020 - IEEE 18th World Symposium on Applied Machine Intelligence and Informatics, Proceedings* (Herlany), 281–286. doi: 10.1109/SAMI48414.2020.9108717

Ren, P., Li, C., Wang, G., Xiao, Y., Du, Q., Liang, X., et al. (2022). "Beyond fixation: dynamic window visual transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, LA: IEEE), 11987–11997. doi: 10.1109/CVPR52688.2022.01168

Shamir, M., and Sompolinsky, H. (2006). Implications of neuronal diversity on population coding. *Neural Comput.* 18, 1951–1986. doi: 10.1162/neco.2006.18.8.1951

Solli, D. R., and Jalali, B. (2015). Analog optical computing. *Nat. Photonics* 9, 704–706. doi: 10.1038/nphoton.2015.208

Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nat. Rev. Neurosci.* 9, 206–221. doi: 10.1038/nrn2286

Stevens, C. F., and Wang, Y. (1994). Changes in reliability of synaptic function as a mechanism for plasticity. *Nature* 371, 704–707. doi: 10.1038/371704a0

Stimberg, M., Brette, R., and Goodman, D. F. (2019). Brian 2, an intuitive and efficient neural simulator. *Elife* 8:e47314. doi: 10.7554/eLife.47314.028

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems, Vol. 27* (Montréal, QC).

Tait, A. N., Nahmias, M. A., Tian, Y., Shastri, B. J., and Prucnal, P. R. (2014). *Photonic Neuromorphic Signal Processing and Computing*. Berlin, Heidelberg: Springer, 183–222. doi: 10.1007/978-3-642-40224-1_8

Tait, A. N., Prucnal, P. R., Shastri, B. J., de Lima, T. F. and Nahmias, M. A. (2015). Excitable laser processing network node in hybrid silicon: analysis and simulation. *Opt. Express* 23, 26800–26813. doi: 10.1364/OE.23.026800

Thagard, P. (2002). How molecules matter to mental computation. *Philos. Sci.* 69, 429–446. doi: 10.1086/342452

Theis, T. N., and Wong, H.-S. P. (2017). The end of Moore's law: a new beginning for information technology. *Comput. Sci. Eng.* 19, 41–50. doi: 10.1109/MCSE.2017.29

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. (2017). "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS 2017)*, *Vol. 30* (Long Beach, CA).

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 1550–1560. doi: 10.1109/5.58337

Zeng, H., and Sanes, J. R. (2017). Neuronal cell-type classification: challenges, opportunities and the path forward. *Nat. Rev. Neurosci.* 18, 530–546. doi: 10.1038/nrn.2017.85

Zhang, Y., Samanta, A., Shang, K., and Yoo, S. J. B. (2020). Scalable 3D silicon photonic electronic integrated circuits and their applications. *IEEE J. Sel. Top. Quantum Electron.* 26, 1–10. doi: 10.1109/JSTQE.2020.2975656

frontiers | Frontiers in Neuroscience

TYPE Opinion
PUBLISHED 10 September 2024
DOI 10.3389/fnins.2024.1443121

OPEN ACCESS

EDITED BY
André van Schaik,
Western Sydney University, Australia

REVIEWED BY
Roberto Marangoni,
University of Pisa, Italy
Konrad Szaciłowski,
AGH University of Science and Technology, Poland

*CORRESPONDENCE
Pier Luigi Gentili
✉ pierluigi.gentili@unipg.it

RECEIVED 03 June 2024
ACCEPTED 27 August 2024
PUBLISHED 10 September 2024

CITATION
Gentili PL, Zurlo MP and Stano P (2024) Neuromorphic engineering in wetware: the state of the art and its perspectives. *Front. Neurosci.* 18:1443121. doi: 10.3389/fnins.2024.1443121

Neuromorphic engineering in wetware: the state of the art and its perspectives

Pier Luigi Gentili[1]*, Maria Pia Zurlo[1] and Pasquale Stano[2]

[1]Department of Chemistry, Biology, and Biotechnology, Università degli Studi di Perugia, Perugia, Italy, [2]Department of Biological and Environmental Sciences and Technologies (DiSTeBA), University of Salento, Lecce, Italy

KEYWORDS

chemical artificial intelligence, chemical reaction networks, emergence, oscillatory chemical reactions, synthetic biology, DNA, proteins, fluidic memristors

1 Introduction

The UN General Assembly (2015) has compiled an Agenda, containing 17 goals to be pursued worldwide to promote a sustainable future by 2030. Accomplishing these goals requires designing and implementing more effective strategies to manage Complex Systems, including human beings and their societies, the world economy, urban areas, natural ecosystems, and the climate (Gentili, 2021a). A promising strategy, which is literally blooming, relies on the development of Artificial Intelligence (AI) and Robotics. AI helps humans collect, store, and process the Big Data required to monitor the constant evolution of Complex Systems (Corea, 2019). AI also assists us in making up our minds for controlling the behavior of Complex Systems. Hard and soft robotics allow humans to access environments otherwise precluded. For instance, they help us (1) investigate the geochemical characteristics of other planets and examine the abysses of our oceans to discover new mines of precious materials and energy resources, (2) access the interior organs of our bodies for less invasive surgery, (3) and work in dirty or dangerous places. Two are the principal and traditional approaches exploited to develop AI (Lehman et al., 2014; Mitchell, 2019). The first approach entails writing "intelligent" software that runs on electronic computers based on von Neumann's architecture, whose principal drawback is having processing and memory units physically separated. Some software mimics rigorous logical thinking, while others imitate the structural and functional features of neural networks to learn how to perform tasks from data. The second approach for developing AI entails implementing artificial neural networks in hardware for neuro-prosthesis or designing brain-like computing machines, with processors and memory confined in the same space (the so-called mem-computing; Sebastian et al., 2020). Artificial neural networks are rigid if they are made of silicon-based circuits or inorganic memristors; they are flexible if based on organic semiconductor films (Christensen et al., 2022; Lee and Lee, 2019; Wang et al., 2020; Zhu et al., 2020). They can be designed with three distinct architectures: (A1) feedforward (having trainable unidirectional connections), (A2) recurrent (with trainable feedback actions), or (A3) reservoir (consisting of an untrained non-linear dynamic system coupled to trainable input and output layers) network (Nakajima, 2020; Tanaka et al., 2019; Cucchi et al., 2022; see Figure 1A).

In the last decade or so, a novel promising strategy to develop AI has been put forward: it consists of mimicking human intelligence and the forms of intelligence exhibited by all the other living beings through molecular, supramolecular, and systems chemistry in wetware, i.e., liquid

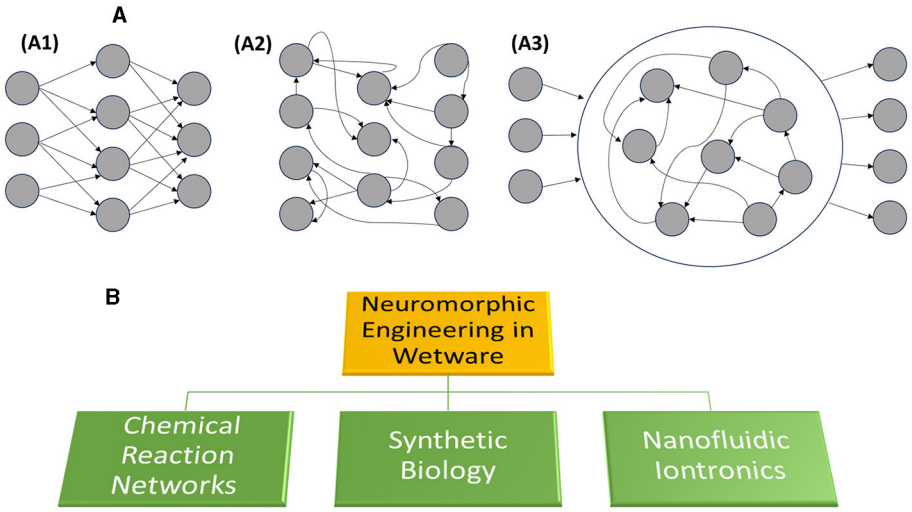


FIGURE 1
(A) Shows the three principal architectures of artificial neural networks: they are (A1) feedforward, (A2) recurrent, and (A3) reservoir networks. **(B)** Showcases the three principal methodologies for developing neuromorphic engineering in wetware.

solutions (Gentili and Stano, 2023a,b; Kuzuya et al., 2023; Murata et al., 2022), which is the peculiar phase supporting life. As we believe that this still not-well-explored field represents a huge opportunity to understand and exploit computation in the molecular realm—thus closely mimicking the natural (biological) cognitive abilities—here we would like to highlight the current methodologies. In particular, we focus on artificial neural networks in wetware and, hence, on the strategies to develop neuromorphic engineering in the fluid phase. The selection of topics presented in this short article is not meant to represent the whole diversity of this research area—it rather mirrors our specific interests. The variegate methodologies proposed so far can be grouped into three distinct approaches (see Figure 1B) presented succinctly in the next three paragraphs. Some future perspectives are shortly presented in the last paragraph.

2 Chemical reaction networks

Any liquid solution containing two or more reactive solutes may display some of the brain's dynamic features, especially if considered as a useful model or even a simplified version of it. Although any brain is a complex three-dimensional cellular architecture, chemical reaction networks can share some aspects of their organization. Indeed, it is still possible to draw direct analogies between the chemical compound and chemical reaction space to bio-inspired brain-type architectures with the reactive molecules of solutes representing the neurons and their mutual impacts being the synapses (Csizi and Lörtscher, 2024). The molecules of solvent, which do not react but assist the chemical transformations of solutes, are like the brain's glial cells. Some solute molecules' collisions trigger chemical reactions, whereas others are chemically ineffective. Specific steric and energetic conditions must be verified to render a molecular impact reactive. The Arrhenius law defines the transformation rate constant (k_r) of the reagents into the products and it formally corresponds to the activation function of the molecular nodes:

$$k_r = Ae^{\frac{-E_{act}}{RT}} \tag{1}$$

In Equation 1, the pre-exponential factor A is related to the steric requirements, whereas E_{act} is the minimum energy needed to make an impact reactive. Usually, it is the thermal energy, RT, available to all the molecules, which is exploited to overcome the barrier E_{act}, unless other energetic inputs are unleashed from outside. The kinetic constant k_r, defined in Equation 1, is related to the computational rate for the chemical reaction network: it increases by heating. If the concentration of i-th solute is $C_{0,i}$ (expressed in moles per volume of solution, i.e., in molarity M), the total number of molecular neurons (N) per unit of volume (expressed in liters) will be given by the Avogadro's number times the sum of the solutes' analytical concentrations:

$$N = \left(6.022 \times 10^{23}\right) \sum_i C_{0,i} \tag{2}$$

Molecular networks compute in a highly parallel manner, and their computational rate (C_R) might be remarkable: For a bimolecular reaction of the type $A + B \xrightarrow{k_r} P$, it will be:

$$C_R = \left(k_r C_{0,A} C_{0,B}\right)\left(6.022 \times 10^{23}\right) \tag{3}$$

When the rate-determining step is the encounter of the reactants (A and B) by diffusion, the apparent reactive constant $\left(k_r\right)_{app} \approx 10^9 M^{-1}s^{-1}$, and if $C_{0,A}C_{0,B} \approx 10^{-9}M^2$, then the computational rate is hundreds of zettaFLOPS (i.e., $\approx 10^{23}$) per unit of volume, i.e., five orders of magnitude faster than the best supercomputer in the world, according to the TOP500 project

(https://www.top500.org/). Of course, in a chemical reaction, even if carried out by billions and billions of molecules, it is generally impossible to address individual reaction events in order to distinguish them because they occur randomly distributed in space and time. The situation could be improved through micro-compartmentalization, but it remains far from the performances of the two-dimensional architecture of the processors inside an electronic computer and even further from the remarkable computational performances of the three-dimensional architecture of a biological brain.

In any fluid solution, the network's architecture is not fixed, but fluid, subjected to the constant movement of the molecular neurons, promoted by diffusion, stirring (if present), and advection (if induced). It is a reservoir network (Figure 1A3), whose overall shape and size are fixed by the solid device containing the solution (Adamatzky, 2019) and whose computational rate is directly proportional to the concentrations of the solutes. If the molecules constituting the network are prepared and maintained in a coherent quantum state, they can be employed to perform quantum neuromorphic computing (Ghosh et al., 2021). When molecular Brownian motion destroys the coherent quantum states, the chemical reservoir can be exploited to implement classical logic. If the input-output relationships are steep sigmoid functions, they are appropriate for implementing binary logic gates (De Silva, 2013). The molecular logic gates have been demonstrated to be reconfigurable because the input-output relationship can change depending on the technique used to monitor the read-out layer of Figure 1A3. When the input-output function is not sigmoid but hyperbolic or linear, the molecular network is appropriate for processing infinite-valued logic, like fuzzy logic (Gentili, 2018). Fuzzy logic is a model of human capability to make decisions using natural language. The words are fuzzy sets. It has been demonstrated that fuzzy sets can be chemically implemented through the context-dependent conformational distributions of compounds (Gentili, 2021b; Gentili and Perez-Mercader, 2022). The major challenges for neuromorphic engineering through chemical reaction networks are to connect (1) different chemical logic gates for the implementation of extended circuits analogous to those in electronics and (2) distinct chemical words to build molecular languages. One way is through optical signals (Andréasson and Pischel, 2015) and another through microfluidic platforms that allow controlling the encounter of molecular reagents (Kou et al., 2008).

Some chemical reactive systems produce intermediates that establish mutual strong non-linear relationships, typical of a recurrent network, and give rise to bottom-up emergent properties, such as spontaneous temporal and spatial self-organization phenomena (Epstein and Pojman, 1998; Ashkenasy et al., 2017). These chemical systems, whose iconic instance is the Belousov-Zhabotinsky reaction, have been proposed as dynamic surrogates of real neurons because they can reproduce their oscillatory, chaotic, and excitable regimes (Okamoto et al., 1995; Izhikevich, 2007; Gentili and Micheau, 2020). They can communicate through chemical, electrical, and optical signals, giving rise to spatio-temporal synchronization phenomena, analogous to those shown by real neural networks. The single neural surrogates can be confined to either macro- or micro-reactors. They have been arranged in all three archetypes of neural networks shown in Figure 1A: feed-forward, recurrent, and reservoir networks (Gentili et al., 2017; Litschel et al., 2018; Vanag, 2019; Gentili, 2022; Tomassoli et al., 2024).

When the molecules participating in the chemical reaction networks are biopolymers, such as DNA, RNA, and proteins, we enter the realm of synthetic biology, which constitutes the second strategy for developing neuromorphic engineering in wetware (Vasle and Moškon, 2024).

3 Synthetic biology

The non-linear reactivity of biopolymers, i.e., DNA, RNA, and proteins engaged in fundamental processes for cell life, is ideal for implementing reservoir and recurrent networks (Cameron et al., 2014; Tang et al., 2021) *in vivo* and *in vitro*. Since each biopolymer exists as a collection of conformers, whose features are context-dependent, the bio-chemical reaction networks are intrinsically fuzzy (Gentili, 2024). Fuzzy neural networks guarantee adaptability and the capability to make decisions in environments dominated by uncertainty and vagueness (Zadeh, 1997; Gentili and Stano, 2022). Within a cell, biopolymers participate in chemical reactions that occur in overcrowded micro- and nano-compartments (i.e., the organules), often at their interface, and involving tethered reactive species, limiting their random Brownian motions. This well-orchestrated and complex bio-chemical reaction network gives rise to an autonomous cellular computing system. A cell is capable of (1) collecting data about the external environment and its internal state through transmembrane sensory proteins; (2) processing the sensory data and making decisions, which (3) trigger the genetic module or (4) modify cellular metabolism (Roederer, 2005; Gentili and Stano, 2024). Living cells are too complex to be reproduced synthetically, from scratch, through a bottom-up approach. The synthetic cells (SCs) implemented so far are more similar to wetware machines that are programmed to compute and accomplish specific tasks, such as assaying chemical information and therapeutics (Chang, 1987; Guindani et al., 2022). However, attention has been recently paid to how to make them more organism-like, i.e., "minimally cognitive" (Damiano and Stano, 2018; Stano, 2023). For example, an explicitly declared goal is to implant a sort of minimal brain made of chemical reaction networks inside SCs (Braccini et al., 2023), aiming at a simple form of autonomy. A step further will be reachable when an SC could become a neural network node made of other SCs (with or without involving natural cells) to imitate the organizational and functional features of biological tissues. In these cellular networks, two- or three-dimensional cultures of human brain cells (the so-called brain organoids) will be employed, facilitating the reconstruction of the histoarchitecture and functionality of real neural networks (Smirnova, 2024).

4 Nanofluidic iontronics

Bioinspired nanofluidic iontronics represents the most recent approach for developing neuromorphic engineering in wetware (Hou et al., 2023). It consists of hybrid circuits made of solid nanochannels and electrically conductive ionic solutions to imitate real neurons that use ionic currents as information carriers. The

solid nanochannels are not simple containers: their shape and size affect the electrical properties of the devices. There are two groups of nanofluidic devices: (1) nanofluidic transistors that mimic structures and functionalities of biological ion channels, and (2) nanofluidic memristors that mimic synapses (Xiong et al., 2023a). Under nano-confinements, both water molecules and ions exhibit anomalous transport behaviors, such as ultrahigh ion/proton transport speed and selectivity (Robin et al., 2023). These nanofluidic devices not only reproduce brain-like neural electrical signals but also realize the logic operation or memory functionalities. The way to endow bioinspired nanofluidics with smart responsiveness is to modify the inner surface of the channels with various responsive molecules, such as aptamers and antibodies (Xiong et al., 2023b). A wide range of chemical species could coexist and move freely in electrolyte solutions contributing to abundant chemical information compared with solid memristors. The biological compatibility of fluidic memristors is convenient for the communication between real neurons and devices.

5 Discussion

Despite the recent impressive advancements in conventional (hardware/software) AI and Robotics, we expect a profound revolution in the sciences of the artificial (Cordeschi, 2002) will definitely come from exploring fluid chemical systems and their computational capabilities. The development of neuromorphic engineering in wetware requires an interdisciplinary effort, involving chemists, physicists, biologists, engineers, computer scientists, and neuroscientists. Differently from general-purpose electronic computers, neuromorphic devices in wetware will be specific-purpose. In computing, they will be particularly alluring for recognizing variable patterns, solving NP-hard problems, and processing vague information (Adleman, 1994; Adamatzky et al., 2005; Evans et al., 2024; Csaba and Porod, 2020; Gentili and Stano, 2024) because chemical reaction networks perform massive-parallel computations. Furthermore, neuromorphic devices in wetware will guarantee a seamless interface with living beings because they can interplay with living cells even at the molecular level. They will reciprocally communicate through both chemical and physical signals. Chemical communication can be carried out not only through diffusion, but also advection, chemical waves and motor proteins. Neuromorphic devices in wetware will monitor, and heal if required, biological functions through the implementation of multi-scale artificial and biological communication networks, called Internet of Nano/Bio-things (IoBNTs; Akyildiz et al., 2015; Stano et al., 2023). We think it is reasonable to expect that such IoBNTs will approach the power of biological intelligence to process information based on uncertain and context-dependent data without an excessive expenditure of energy.

Author contributions

PG: Conceptualization, Funding acquisition, Writing – original draft, Writing – review & editing. MZ: Writing – review & editing. PS: Conceptualization, Funding acquisition, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work has been funded by Università degli Studi di Perugia (Fondo Ricerca di Base 2022) and the European Union—NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041—VITALITY. We acknowledge Università degli Studi di Perugia and MUR for support within the project Vitality. The synthetic biology approaches to embodied AI and minimal cognition is a theme developed within the project: Org (SB-EAI)—An Organizational Approach to the Synthetic Modeling of Cognition based on Synthetic Biology and Embodied AI, funded by the Ministero dell'Università e della Ricerca (Italy), within the PRIN-2022 program, grant number: 20222HHXAX.

Acknowledgments

PG acknowledges the Università degli Studi di Perugia for the Ricerca di Base fund.

Conflict of interest

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

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

References

Adamatzky, A. (2019). A brief history of liquid computers. *Philos. Trans. R. Soc. B* 374:20180372. doi: 10.1098/rstb.2018.0372

Adamatzky, A., De Lacy Costello, B., and Asai, T. (2005). *Reaction-Diffusion Computers*. Amsterdam: Elsevier Science.

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021–1024. doi: 10.1126/science.7973651

Akyildiz, I. F., Pierobon, M., Balasubramaniam, S., and Koucheryavy, Y. (2015). The internet of bio-nano things. *IEEE Commun. Mag.* 53, 32–40. doi: 10.1109/MCOM.2015.7060516

Andréasson, J., and Pischel, U. (2015). Molecules with a sense of logic: a progress report. *Chem. Soc. Rev*. 44, 1053–1069. doi: 10.1039/C4CS00342J

Ashkenasy, G., Hermans, T. M., Otto, S., and Taylor, A. F. (2017). Systems chemistry. *Chem. Soc. Rev*. 46, 2543–2554. doi: 10.1039/C7CS00117G

Braccini, M., Collinson, E., Roli, A., Fellermann, H., and Stano, P. (2023). Recurrent neural networks in synthetic cells: a route to autonomous molecular agents? *Front. Bioeng. Biotechnol*. 11:1210334. doi: 10.3389/fbioe.2023.1210334

Cameron, D. E., Bashor, C. J., and Collins, J. J. (2014). A brief history of synthetic biology. *Nat. Rev. Microbiol*. 12, 381–390. doi: 10.1038/nrmicro3239

Chang, T. M. (1987). Applications of artificial cells in medicine and biotechnology. *Biomater. Artif. Cells Artif. Organs* 15, 1–20. doi: 10.3109/10731198709118504

Christensen, D. V., Dittmann, R., Linares-Barranco, B., Sebastian, A., Le Gallo, M., Redaelli, A., et al. (2022). 2022 roadmap on neuromorphic computing and engineering. *Neuromorph. Comput. Eng*. 2:e022501. doi: 10.1088/2634-4386/ac4a83

Cordeschi, R. (2002). *The Discovery of the Artificial. Behavior, Mind and Machines Before and Beyond Cybernetics*. Dordrecht: Springer.

Corea, F. (2019). *Introduction to Data. Everything You Need to Know About AI, Big Data, and Data Science*. Cham: Springer.

Csaba, G., and Porod, W. (2020). Coupled oscillators for computing: a review and perspective. *Appl. Phys. Rev*. 7:e011302. doi: 10.1063/1.5120412

Csizi, K.-S., and Lörtscher,. E. (2024). Complex chemical reaction networks for future information processing. *Front. Neurosci*. 18:1379205. doi: 10.3389/fnins.2024.1379205

Cucchi, M., Abreu, S., Ciccone, G., Brunner, D., and Kleemann, H. (2022). Hands-on reservoir computing: a tutorial for practical implementation. *Neuromor. Comput. Eng*. 2:e032002. doi: 10.1088/2634-4386/ac7db7

Damiano, L., and Stano, P. (2018). Synthetic biology and artificial intelligence. Grounding a cross-disciplinary approach to the synthetic exploration of (embodied) cognition. *Compl. Syst*. 27, 199–228. doi: 10.25088/ComplexSystems.27.3.199

De Silva, A. P. (2013). *Molecular Logic-Based Computation*. Cambridge: Royal Society of Chemistry.

Epstein, I. R., and Pojman, J. A. (1998). *Introduction to Nonlinear Chemical Dynamics*. New York, NY: Oxford University Press.

Evans, C. G., O'Brien, J., Winfree, E., and Murugan, A. (2024). Pattern recognition in the nucleation kinetics of non-equilibrium self-assembly. *Nature* 625, 500–507. doi: 10.1038/s41586-023-06890-z

Gentili, P. L. (2018). The fuzziness of the molecular world and its perspectives. *Molecules* 23:2074. doi: 10.3390/molecules23082074

Gentili, P. L. (2021a). Why is Complexity Science valuable for reaching the goals of the UN 2030 agenda? *Rend. Fis. Acc. Lincei*. 32, 117–134. doi: 10.1007/s12210-020-00972-0

Gentili, P. L. (2021b). Establishing a new link between fuzzy logic, neuroscience, and quantum mechanics through bayesian probability: perspectives in artificial intelligence and unconventional computing. *Molecules* 26:5987. doi: 10.3390/molecules26195987

Gentili, P. L. (2022). Photochromic and luminescent materials for the development of Chemical Artificial Intelligence. *Dyes Pigm*. 205:110547. doi: 10.1016/j.dyepig.2022.110547

Gentili, P. L. (2024). The conformational contribution to molecular complexity and its implications for information processing in living beings and chemical artificial intelligence. *Biomimetics* 9:121. doi: 10.3390/biomimetics9020121

Gentili, P. L., Giubila, M. S., Germani, R., Romani, A., Nicoziani, A., Spalletti, A., et al. (2017). Optical communication among oscillatory reactions and photo-excitable systems: UV and visible radiation can synchronize artificial neuron models. *Angew. Chem. Int. Ed*. 56, 7535–7540. doi: 10.1002/anie.201702289

Gentili, P. L., and Micheau, J. C. (2020). Light and chemical oscillations: review and perspectives. *J. Photochem. Photobiol. C Photochem. Rev*. 43:100321. doi: 10.1016/j.jphotochemrev.2019.100321

Gentili, P. L., and Perez-Mercader, J. (2022). Quantitative estimation of chemical microheterogeneity through the determination of fuzzy entropy. *Front. Chem*. 10:950769. doi: 10.3389/fchem.2022.950769

Gentili, P. L., and Stano, P. (2022). Chemical neural networks inside synthetic cells? A proposal for their realization and modeling. *Front. Bioeng. Biotechnol*. 10:927110. doi: 10.3389/fbioe.2022.927110

Gentili, P. L., and Stano, P. (2023a). Tracing a new path in the field of AI and robotics: mimicking human intelligence through chemistry. Part I: molecular and supramolecular chemistry. *Front. Robot. AI* 10:1238492. doi: 10.3389/frobt.2023.1238492

Gentili, P. L., and Stano, P. (2023b). Tracing a new path in the field of AI and robotics: mimicking human intelligence through chemistry. Part II: systems chemistry. *Front. Robot. AI* 10:1266011. doi: 10.3389/frobt.2023.1266011

Gentili, P. L., and Stano, P. (2024). Living cells and biological mechanisms as prototypes for developing chemical artificial intelligence. *Biochem. Bioph. Res. Co*. 720:150060. doi: 10.1016/j.bbrc.2024.150060

Ghosh, S., Nakajima, K., Krisnanda, T., Fujii, K., and Liew, T. C. H. (2021). Quantum neuromorphic computing with reservoir computing networks. *Adv. Quant. Technol*. 4:2100053. doi: 10.1002/qute.202100053

Guindani, C., da Silva, L. C., Cao, S., Ivanov, T., and Landfester, K. (2022). Synthetic cells: from simple bio-inspired modules to sophisticated integrated systems. *Angew. Chem. Int. Ed*. 61:e202110855. doi: 10.1002/anie.202110855

Hou, Y., Ling, Y., Wang, Y., Wang, M., Chen, Y., Li, X., et al. (2023). Learning from the brain: bioinspired nanofluidics. *J. Phys. Chem. Lett*. 14, 2891–2900. doi: 10.1021/acs.jpclett.2c03930

Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience*. Cambridge, MA: MIT Press.

Kou, S., Lee, H., van Noort, D., Swamy, K. M. K., Kim, S., Soh, J., et al. (2008). Fluorescent molecular logic gates using microfluidic devices. *Angew. Chem. Int. Ed*. 47, 872–876. doi: 10.1002/anie.200703813

Kuzuya, A., Nomura, S.-I. M., Toyota, T., Nakakuki, T., and Murata, S. (2023). From molecular robotics to molecular cybernetics: the first step toward chemical artificial intelligence. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* 9, 354–363. doi: 10.1109/TMBMC.2023.3304243

Lee, Y., and Lee, T. W. (2019). Organic synapses for neuromorphic electronics: from brain-inspired computing to sensorimotor nervetronics. *Acc. Chem. Res*. 52, 964–974, doi: 10.1021/acs.accounts.8b00553

Lehman, J., Clune, J., and Risi, S. (2014). An anarchy of methods: current trends in how intelligence is abstracted in AI. *IEEE Intell. Syst*. 29, 56–62. doi: 10.1109/MIS.2014.92

Litschel, T., Norton, M. M., Tserunyan, V., and Fraden, S. (2018). Engineering reaction-diffusion networks with properties of neural tissue. *Lab Chip* 18, 714–722. doi: 10.1039/C7LC01187C

Mitchell, M. (2019). *Artificial Intelligence. A Guide for Thinking Humans*. New York, NY: Farrar, Strauss and Giroux.

Murata, S., Toyota, T., Nomura, S. I. M., Nakakuki, T., and Kuzuya, A. (2022). Molecular cybernetics: challenges toward cellular chemical artificial intelligence. *Adv. Funct. Mat*. 32:2201866. doi: 10.1002/adfm.202201866

Nakajima, K. (2020). Physical reservoir computing-an introductory perspective. *Jpn. J. Appl. Phys*. 59:e060501. doi: 10.35848/1347-4065/ab8d4f

Okamoto, H., Tanaka, N., and Naito, M. (1995). Analogy between the stimulus-response characteristics of neuronal and electrochemical cells. *Chem. Phys. Lett*. 237, 432–436. doi: 10.1016/0009-2614(95)00336-3

Robin, P., Emmerich, T., Ismail, A., Niguès, A., You, Y., Nam, G. H., et al. (2023). Long-term memory and synapse-like dynamics in two-dimensional nanofluidic channels. *Science* 379, 161–167. doi: 10.1126/science.adc9931

Roederer, J. (2005). *Information and Its Role in Nature*. Berlin: Springer.

Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., and Eleftheriou, E. (2020). Memory devices and applications for in-memory computing. *Nat. Nanotechnol*. 15, 529–544. doi: 10.1038/s41565-020-0655-z

Smirnova, L. (2024). Biocomputing with organoid intelligence. *Nat. Rev. Bioeng*. 24:6. doi: 10.1038/s44222-024-00200-6

Stano, P. (2023). Chemical systems for wetware artificial life: selected perspectives in synthetic cell research. *Int. J. Mol. Sci*. 24:14138. doi: 10.3390/ijms241814138

Stano, P., Gentili, P. L., Damiano, L., and Magarini, M. (2023). A role for bottom-up synthetic cells in the internet of bio-nano things? *Molecules* 28:5564. doi: 10.3390/molecules28145564

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., et al. (2019). Recent advances in physical reservoir computing: a review. *Neural Netw*. 115, 100–123. doi: 10.1016/j.neunet.2019.03.005

Tang, T. C., An, B., Huang, Y., Vasikaran, S., Wang, Y., Jiang, X., et al. (2021). Materials design by synthetic biology. *Nat. Rev. Mater*. 6, 332–350. doi: 10.1038/s41578-020-00265-w

Tomassoli, L., Silva-Dias, L., Dolnik, M., Epstein, I. R., Germani, R., and Gentili, P. L. (2024). Neuromorphic engineering in wetware: discriminating acoustic frequencies through their effects on chemical waves. *J. Phys. Chem. B* 128, 1241–1255. doi: 10.1021/acs.jpcb.3c,08429

UN General Assembly (2015). *Transforming Our World: the 2030 Agenda for Sustainable Development. A/RES/70/1*. Available at: https://www.refworld.org/docid/57b6e3e44.html (accessed March 22, 2024).

Vanag, V. (2019). Hierarchical network of pulse coupled chemical oscillators with adaptive behavior: chemical neurocomputer. *Chaos* 29:e083104. doi: 10.1063/1.5099979

Vasle, A. H., and Moškon, M. (2024). Synthetic biological neural networks: from current implementations to future perspectives. *BioSystems* 237:105164. doi: 10.1016/j.biosystems.2024.105164

Wang, Z., Wu, H., Burr, G. W., Hwang, C. S., Wang, K. L., Xia, Q., et al. (2020). Resistive switching materials for information processing. *Nat. Rev. Mater.* 5, 173–195. doi: 10.1038/s41578-019-0159-3

Xiong, T., Li, C., He, X., Xie, B., Zong, J., Jiang, Y., et al. (2023a). Neuromorphic functions with a polyelectrolyte-confined fluidic memristor. *Science* 379,156–161. doi: 10.1126/science.adc9150

Xiong, T., Li, W., Yu, P., and Mao, L. (2023b). Fluidic memristor: bringing chemistry to neuromorphic devices. *Innovation* 4:100435. doi: 10.1016/j.xinn.2023.100435

Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90, 111–127. doi: 10.1016/S0165-0114(97)00077-8

Zhu, J., Zhang, T., Yang, Y., and Huang, R. (2020). A comprehensive review on emerging artificial neuromorphic devices. *Appl. Phys. Rev.* 7:e011312. doi: 10.1063/1.5118217

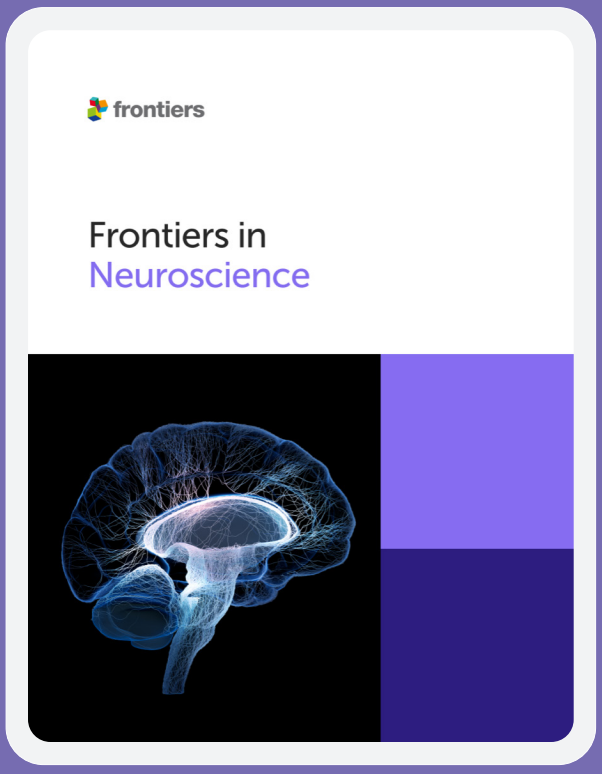
Frontiers in Neuroscience

Provides a holistic understanding of brain function from genes to behavior

Part of the most cited neuroscience journal series which explores the brain - from the new eras of causation and anatomical neurosciences to neuroeconomics and neuroenergetics.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact