Neuromorphic Computing and Engineering

PURPOSE-LED PUBLISHING™

Biologically inspired memristive neuron capable of on-chip learning

To cite this article before publication: Nadia Jimenez Olalla *et al* 2026 *Neuromorph. Comput. Eng.* in press https://doi.org/10.1088/2634-4386/ae9640

Biologically inspired memristive neuron capable of on-chip learning

Nadia Jimenez Olalla[1*], Ashira Long Ching Ip[2], Michael Baumann[1], Marco Dober[1], Juerg Leuthold[1] and Pau Vilimelis Aceituno[2*]

[1] Institute of Electromagnetic Fields (IEF), ETH Zürich, Switzerland

[2] Institute of Neuroinformatics, ETH Zürich & University of Zürich, Switzerland

*** Correspondence:**

Corresponding Author

nolalla@ethz.ch and pau@ini.uzh.ch

Keywords: bio-plausible learning rule, on-chip learning, neuromorphic, memristive neuron, non-linear operations, reconfigurable circuit.

Abstract

The brain embeds learning algorithms within its physical architecture, enabling on-site learning processes with remarkable advantages such as energy efficiency, learning autonomy and continual learning capabilities. Current memristive neuromorphic implementations, despite being inspired by the brain, are predominantly trained via offline, software-based methods. Consequently, this paradigm restricts their potential to realize on-chip learning. To address this limitation, we implement a simple memristive-based electronic circuit with an op-amp that emulates a neuron with biologically inspired learning rules, characterized by recent electrophysiological observations. To demonstrate its potential as a fundamental computational primitive, we integrate it into a network with one hidden

layer capable of learning all fundamental binary operations. As a proof-of-concept, we build a physical circuit that successfully learns and performs XOR, AND, and OR logic operations. To validate the scalability of our approach beyond simple logic, we also perform SPICE simulations on a compressed MNIST classification task. Therefore, the proposed circuit could be leveraged to build memristive neuromorphic systems capable of doing on-chip learning.

1 Introduction

The proliferation of Artificial Intelligence (AI) has marked a transformative era, with deep learning models achieving state-of-the-art performance across numerous domains. As Artificial Neural Networks (ANNs) grow to tackle arbitrary tasks of larger complexity, the demand for computational resources leads to an unsustainable surge in energy consumption, a major bottleneck for future progress [1].

In the quest for a more efficient computational paradigm, the scientific community is increasingly turning to the human brain for inspiration. The brain performs complex cognitive tasks with remarkable energy efficiency, operating orders of magnitude more efficiently than any existing computer [2]. This is primarily attributed to the co-location of memory and computation with algorithmic sparsity and decentralized, asynchronous and adaptive processing. Beyond energy efficiency, the integration of learning mechanisms within the hardware architecture enables significant functional advantages, including continual learning and intrinsic adaptation to changing environments.

This disparity between ANNs and the brain has catalyzed the field of neuromorphic computing, which aims to emulate the brain's structure and principles in hardware [3–5]. While traditional CMOS-based technologies have been used for initial neuromorphic implementations, their inherent limitations in density and power consumption have spurred research into novel nanodevices [6,7]. Among these, memristors have emerged as leading candidates for building artificial synapses [8–10]. This interest is driven by their ability to dynamically modulate resistance in response to electrical stimuli and their non-volatile nature, reducing their static power consumption to zero.

However, current research predominantly focuses on utilizing memristive neuromorphic systems to exclusively execute inference tasks. Thereby, the network is trained offline on energy-hungry,

external conventional computers, and the resulting synaptic weights are subsequently programmed onto the memristive elements. Consequently, the development of effective on-chip learning schemes with memristors remains a major unresolved challenge, due to the inability of analog hardware to meet the stringent requirements of current optimization algorithms [11].

In this work, we propose an on-chip learning scheme exploiting the nonvolatile and dynamic characteristics of memristors to emulate a brain-inspired learning rule. We first explain the current state of memristive neuromorphic hardware which provides the background and context for this research. Next, we introduce the recent theory of biological learning and how it influences synaptic adaptation. Thereafter, we compare memristors and biological synapses and exploit those similarities to build individual neurons that emulate biological learning. We demonstrate that our neurons can learn linear input-output associations and, by combining three of those neurons, we build a proof-of-concept of a reconfigurable network capable of dealing with nonlinear problems. To evaluate the scalability of this architecture on more complex datasets, we further present SPICE simulations of a compressed MNIST classification task. Finally, we discuss the implications of our work and our concluding remarks.

1.1 On-chip learning in memristive neuromorphic hardware

The realization of robust, on-chip learning in memristive neuromorphic hardware remains a central challenge in the field. While Backpropagation (BP) is the workhorse algorithm for training deep neural networks, its direct implementation is fundamentally incompatible with the physical constraints of analog hardware [11]. The key challenge resides in the dependency of the backward pass on the exact weights and activities of neurons during the feedforward pass. This requirement implicitly assumes conductance stability, ignoring any weight drift that might occur within the physical elements leading to wrong error correction calculations. Furthermore, the inherent non-idealities of memristors, like device-to-device variability and asymmetric, history-dependent weight updates, exacerbates the complexity of its integration [12,13].

A potential solution to these problems is the co-design of hardware-specific algorithms that directly confront memristive non-idealities. For example, algorithms like “AGAD” and "Tiki-Taka" [14,15] are BP-based algorithms designed to compensate for asymmetric device updates by first performing a calibration step to find a device's "symmetry point". However, this comes with the cost of increasing

peripheral circuitry, leading to higher complexity, area and power consumption. A second approach focuses on three-factor learning rules. In this supervised paradigm, the third factor is a non-local "error" signal broadcast to the synapses to approximate BP [16]. Similarly to the previous approach, it re-introduces non-locality, necessitating complex peripheral overhead for error calculation and routing. A broader perspective of current developments of memristive neuromorphic systems capable of doing on-chip learning is included in Supplementary Table 2.

In this work, we focus on investigating learning algorithms in the brain, which must account for asymmetries, non-idealities, and noise, where learning algorithms may differ from the principles of BP [17].

1.2 Learning algorithms in the brain

The question of which algorithms train biological deep networks has been investigated in depth for decades and recently resulted in two Nobel prizes for proponents of BP and the so-called energy models.

Although early studies argued for backpropagation of error [18], the biological implausibility of it was evident early on [19]. Subsequent groups have relaxed the assumptions of backpropagation to make it more compatible with the brain. Proposed solutions include incorporating random backward passes, which degrade performance [20], or adding dedicated circuitry, which decreases efficiency [21,22].

In contrast, energy models followed a distinct evolutionary path. While originally employed to explain simple associations [17], they were soon adapted to develop predictive coding circuits [18]. These circuits have since been widely utilized for more complex operations like predictions, control problems, image processing and classification problems. [23]. Originally, they relied on weight symmetry, but modern control-based methods have eliminated this constraint. This shift has yielded robust results in recurrent architectures like FORCE [24] and FOLLOW [25]. Notably, these models now match the performance of backpropagation in simple tasks while retaining the advantage of requiring no a priori knowledge of weights and activations [26].

Conceptually, the principle of the energy model is straightforward. Rather than measuring the activity during the feed-forward pass and computing the corresponding modification, energy models first ensure that the network solves the desired task through an external forcing signal. The system then changes the network parameters to minimize the forcing signal [27,28]. Importantly, the signal used for forcing does not need to have precise information about the network activations to enforce the activities and achieve backpropagation-like performances [26,29]. Finally, recent studies have found experimental evidence of such forced activations during learning [28]. In the field of computational neuroscience, the modern, generalized version of energy models is known as Target Learning (TL) or Prospective Configuration [26,29,30]. The mathematical background for Target Learning is provided in the Supplementary Section 1.4, Appendix D.

TL has previously shown promise in non-memristive neuromorphic hardware [33,34], suggesting that it can alleviate the problems of analog hardware. Here, we push this idea to novel materials that naturally capture intrinsic properties of biological synapses: memristors.

2 Filamentary Memristors: A Key Candidate as Electronic Synapse

To build a memristive circuit based on TL, we first determine the extent to which memristive responses align with that of biological synapses.

Biological synapses are connections between neurons that can naturally be understood as resistor devices: When a synapse presents a large weight, it transmits the signal from the sending neuron to the receiving neuron with small losses, inducing a strong effect on the latter. In contrast, when the synaptic weight is small, the influence on the receiving neurons is proportionally weaker [31]. In electronics, resistors conduct current in proportion to their conductance. Therefore, the internal strength of biological synapses can be understood as the conductance of electronic resistors.

Having established the equivalence between synapse and resistance, the next question is how to adapt them. In biology, synaptic plasticity refers to the modification of synapses in response to inputs. More specifically, an increase of synaptic strength can be induced by the combination of a strong, long-lasting voltage drop across the cell membrane together with an input signal [17,32,33]. Figure 1(a) shows this relationship using the data extracted from [17], which demonstrates that the synapse

strength increases if sufficient accumulated voltage is reached [34,35]. Similarly, memristors undergo conductance change when the accumulated voltage exceeds a threshold, after which it increases in an analog manner until saturation, see Figure 1(b). In section 5, we provide more information about the memristors used in this work.

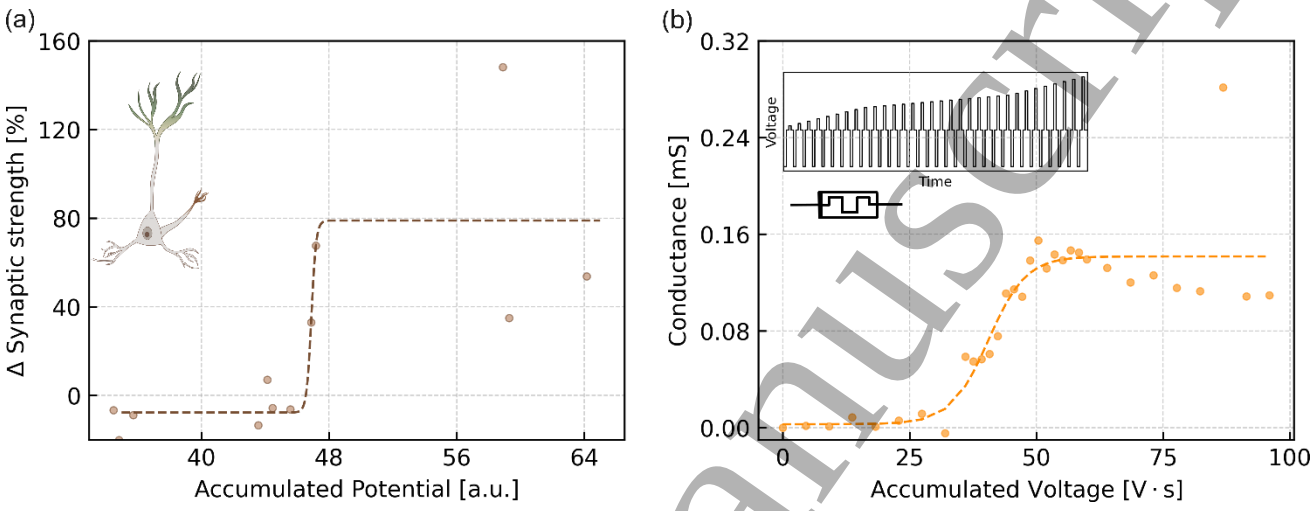

Figure 1 Comparison of synaptic response in a biological cell with memrsitive device characteristics. (a) Experimental evidence of an electrophysiological correlation between change of synaptic strength and accumulated voltage across the cell membrane over time (data extracted from [17]). (b) Memristive conductance evolution as a function of the accumulated voltage over time. The inset shows the pulse scheme applied to switch the device. Following each incremental potentiating pulse, the application of a negative pulse restores the device to its baseline state.

The results in Figure 1 go beyond the common analogy between biological synapses and memristors, which typically focuses on their shared ability to modulate their strength (synapses) or conductance (memristor). In fact, both memristors and synapses show a similar non-linear dependency on the integral of the voltage over time. Crucially, this indicates that the biophysical processes that build learning algorithms in the brain are naturally present in filamentary memristors. In Supplementary Section 1.5 (Appendix D), we mathematically demonstrate how our PN memristive model meets the Target Learning conditions. In the next section, we leverage those dynamical processes to build electronic neurons.

3 Results

3.1 Associative learning

To create a neuron circuit capable of inducing a controlled conductance change in the memristive synapse, we take inspiration from Pyramidal Neurons (PNs).

PNs are the fundamental computational unit and the center piece of learning algorithms in cortex [36,37]. We illustrate how PNs work in Figure 2(a). PNs receive inputs S_i through basal synapses with weight w_i and learning signals S_L arriving through a special receiving compartment referred to as apical synapses. The input terminals of the neuron are called dendrites and lead the signals to the core of the cell, the soma. The latter integrates the signal over time and sends out a response through the axon to neighboring neurons. Inducing plasticity at basal synapses w_i requires minimum signal strength achieved through the synchronous arrival of inputs S_i and an instructional control or learning signals S_L, thereby changing the synaptic weight (i.e. creating associations) between them [17]. The equivalent perceptron model is represented in Figure 2(b).

We build an electronic circuit mimicking a PN capable of implementing TL, see Figure 2(c). The circuit comprises Ta_2O_5-based memristors in parallel and an operational amplifier (op-amp) (see Section 5 for more in-depth details of the individual components). First, the memristors emulate parallel electronic synapses that receive the signals S_i. Subsequently, the resulting currents are summed and directed to the inverting terminal of the op-amp. Additionally, the learning signal S_L can be supplied through the non-inverting terminal of the op-amp, which serves as a common third terminal for all memristive synapses. The op-amp now plays a critical role in the learning process. It creates a virtual short between the non-inverting (+) and the inverting (-) inputs. I.e. a voltage S_L applied to the non-inverting input will be mapped to the inverting input. As a consequence, when an input signal S_i arrives at (-) simultaneously as the learning signal S_L at (+), the total voltage over each device corresponds to $S_i - S_L$. The combined effect of S_L and S_i builds up a voltage that is large enough to induce a synaptic conductance change of those memristors with a signal S_i. The non-volatile memristors then keep the new conductance (weight). Thereafter, even without a learning signal S_L, an input signal S_i applied to a memristor with a low conductance would be transmitted through the memristor and generate a similar output signal as seen when S_L was present. S_{OUT} is then fed to a non-linear activation function that suppresses noise and enhances salient signal components. This concludes the PN circuit.

The design of our electronic neuron circuit has two critical conditions to work: first, the learning signal S_L must not be affected by memristive changes. Second, changes in the output voltage S_{OUT} induced by subsequent circuitry must not propagate backwards. An op-amp satisfies both conditions because its near-infinite input impedance at both the inverting and non-inverting terminals, coupled

with its near-zero output impedance, prevents signal interaction.

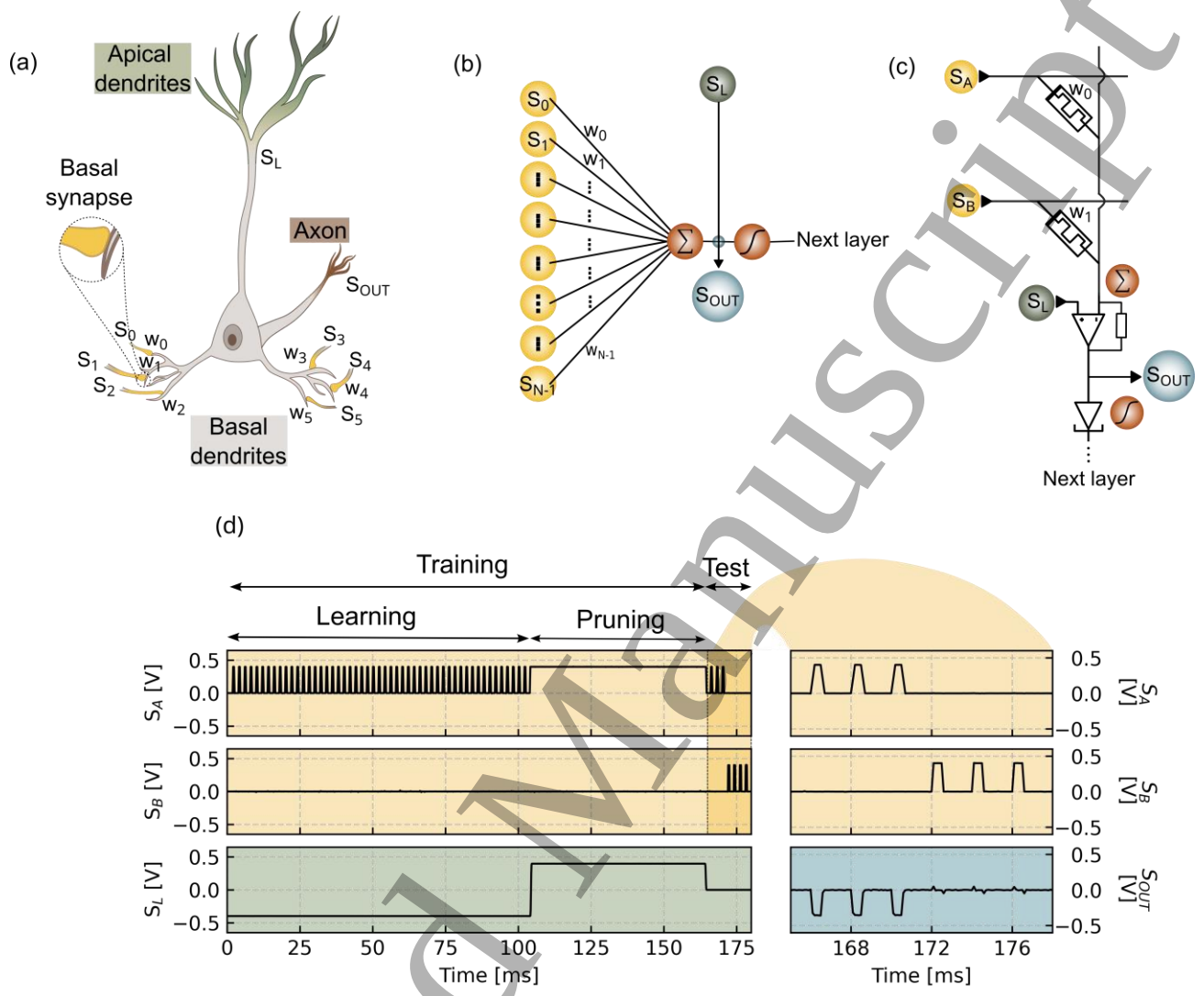

Figure 2 Associative Learning: (a) Pyramidal neuron (PN) model showing input signals S_i basal synapses w_i a learning signal coming from the apical dendrites S_L, and output signal travelling through the axon S_{OUT}. (b) Perceptron equivalent of the PN model. (c) Our electronic implementation of the PN model consisting of two Ta_2O_5-based memristors as synapses and an op-amp, together with few more passive elements. The input voltage S_i applied to the memristors in parallel generates current which is summed up and directed to the inverting input of the op-amp. S_L is applied in the non-inverting input, shifting the reference voltage at the inverting terminal of the op-amp. In consequence, S_L drops in every memristor changing the susceptibility of every memristive synapse to switch. The result of the multiplication and summation can be read in the output of the op-amp, S_{OUT}. To emulate the function of a full perceptron, an activation function follows S_{OUT}. (d) Experimental demonstration of associative learning. During the training phase, only a signal S_A consisting of a sequence of 0.4V pulses is applied to the upper memristor while S_B stays at 0V. Simultaneously, S_L, a plateau potential of -0.4V, shifts the reference voltage of both devices. Effectively, 0.8V drop in the first memristor with weight w_0, while 0.4V in the second one with weight w_1. In consequence, the first memristor undergoes switching, whereas the other one does not. This is observable in the test phase (see zoom-in on the right), where S_{OUT} responds to the pulsing of a signal S_A (i.e. the upper memristor is conductive) and there is no response to a signal S_B (i.e. the bottom memristor is non-conductive). These experimental results are in agreement with our simulations (see supplementary section 1.5 Appendix E).

We now experimentally verify that our circuit works as expected in Figure 2(d). The input signals S_i

is formed by a series of ms-long pulses representing the neuron inputs and S_L emulates the learning signal coming from apical dendrites, which consists of long sustained potential pulses.

When the learning signal S_L is applied to the non-inverting input, it shifts the potential at the inverting input by S_L. In consequence, the voltage across the memristors becomes $S_i - S_L$. The upper memristor with weight w_0 receives a non-zero signal S_A during the training phase, resulting in a total voltage drop of $S_A - S_L$. When the accumulation of $S_A - S_L$ exceeds the memristive threshold, the memristor transitions from the low conductive state (LCS) to the high conductive state (HCS). The memristor retains the state due to its non-volatile nature and, therefore, learns that correlation. Conversely, input S_B is a zero signal, and thus, the lower memristor with weight w_1 does not switch because $S_B - S_L \cong S_L$ is insufficient to induce a change. Therefore, future inputs in S_B will not propagate to the neuron. Notice that in this experiment, we reset unwanted switching events occurring during the learning phase through a pruning phase included in the training.

Finally, we verify the proper operation in the test (interference) phase. The output signal in the blue-shaded area demonstrates that the circuit learned successfully. When applying an equivalent input signal to S_A and S_B, only the upper memristor w_0 which switched to the LCS leads to a response at S_{OUT}.

In brief, this basic unit emulates an electronic perceptron, the central component of neural networks and can be trained by a learning signal enforcing the desired neuronal activity.

3.2 Showcase: Reconfigurable Circuit

The previously introduced reconfigurable electronic perceptron is a building block for a neural network capable of performing linear operations. However, [38] demonstrated the need of multiple-layer networks to solve nonlinear problems, XOR being their main benchmark. Following the same logic, in this section, we stack multiple associative units which can be trained to solve nonlinear and linear problems. For simplicity, we focus on the fundamental logical operations, XOR, AND and OR, which are the bases of any Turing computable function.

To this end, we build a small population of three neurons, see (a), arranged in a 2-2-1 network. It consists of a two-variable input layer, a hidden layer formed by two neurons, and a single-neuron

output layer, see (b). Each neuron consists of our basic associative unit from Figure 2(c). To form a layer, multiple nodes are placed in parallel, and the sequential arrangement of layers yields a deeper network, see (c).

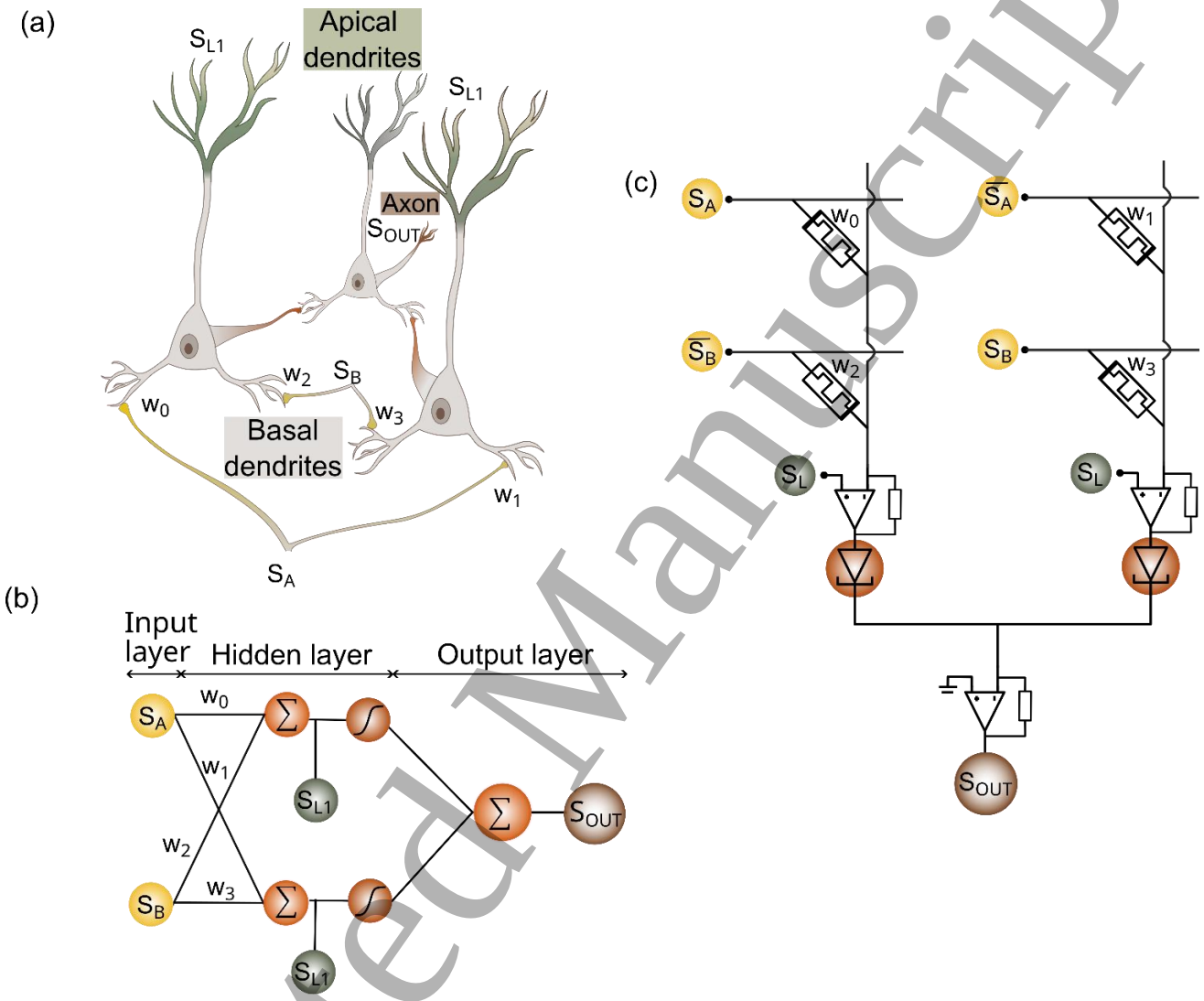

Figure 3 Reconfigurable circuit capable of computing both nonlinear and linear operations: (a) Small population of PNs. Two inputs S_A and S_B are applied to two basal dendrites of two different PNs and a learning signal S_L is applied through the apical dendrites. Their outputs are routed towards a third PN, which outputs S_{OUT} at the axon. This creates a neural network with a 2-2-1 configuration formed by 2 inputs, 1 hidden layer with two neurons and 1 output layer with 1 neuron. (b) Sketch of the respective network. (c) Reconfigurable circuit capable of performing XOR, AND and OR operations. Our reconfigurable electronic circuit is developed from perceptrons or associative units, see Figure 2(c). The electronic synapses are Ta_2O_5-based non-voltatile memristors, whereas the activation functions are Ta_2O_5-based MIM tunnel junctions. In this circuit S_A and S_B are fed to the positive electronic synapses, while their inverting counterparts $\overline{S_A}$ and $\overline{S_B}$ are applied to the negative ones. All those devices are subject to a common voltage shift generated by S_L. The output signal is filtered by an activation function and routed into the output layer.

The design of our electronic neural network has three critical considerations: first, synaptic weights must be able to adopt negative values but memristive conductance is inherently positive. Second, to

realize non-linear operations, we require elements with a non-linear response. Third, the dissipation of the signal when being transmitted through resistive elements requires an active gain to avoid signal loss.

To realize positive and negative synaptic weights, each synapse is implemented as a pair of non-negative conductance memristors of opposite polarity, one driven by the input and one by its inverted copy. Because both currents are summed at the inverting terminal, the effective weight seen by the neuron is the difference of the two conductances, which can take either sign while each individual conductance remains strictly positive. This results in two perceptrons with inputs S_A, $\overline{S_B}$ and $\overline{S_A}$, S_B, respectively. Second, to implement non-linearities, we include a MIM tunnel junction at the output of the op-amp, which filters the signal. Third, to compensate for the signal dissipation, we take advantage of the gain determined by the feedback resistor of the op-amp.

A schematic of the reconfigurable circuit is displayed in (c). In a similar manner as realized in Figure 2, the input signals S_A and S_B consists of a train of pulses while S_L consists of long sustained potential pulses. As before, currents through each memristor of an associative unit are summed up and fed to the inverting terminal of the op-amps. Subsequently, after propagating through the MIM activation functions, the two signals are added and transmitted through an op-amp to reach the output signal S_{OUT}. During training, S_L is applied to the non-inverting terminal. For more details about the electronic components employed for the experimental realization of the circuit in (c), see Section 5.

Figure 4 illustrates the experimental results of training the circuit to implement XOR, AND and OR operations. Each experiment is divided in training and test phase. We define a high pulse as a binary '1' and a low or absent pulse as a binary '0'. Holding the input S_A and S_B (yellow) identical, S_{OUT} (brown) changes during the test phase in response to a different S_L present during the training phase. More specifically, a weak S_L leads to AND operation and a strong S_L to OR. For the non-linear example XOR, we require a positive and a negative S_L to induce learning in memristors with both polarities.

To further clean up the signals and eliminate the small spikes at the output one could add a comparator downstream of S_{OUT}, which digitizes the signal by mapping values above and below a specific threshold voltage to high (1) and low (0) logic states, respectively (see Supplementary Figure 1).

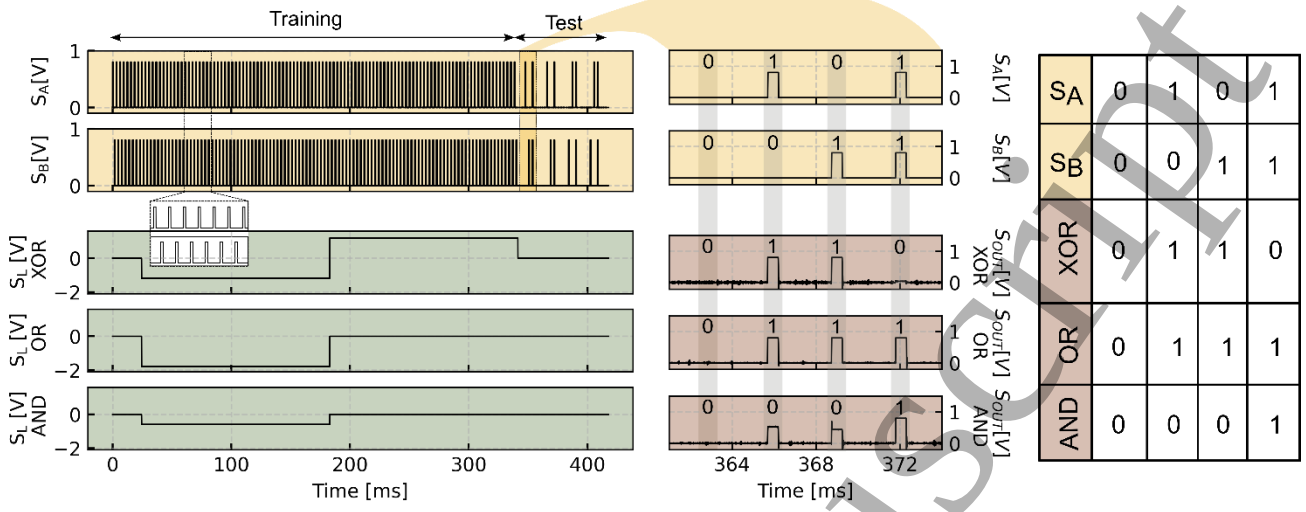

Figure 4 Experimental Result of Reconfigurable Circuit: During the training phase, two inputs consisting of alternating pulsing sequences of approximately 100 pulses 0.4ms long are applied, see S_A and S_B sequences (yellow). Training is initiated by applying any of the three possible learning sequences $S_{L,AND}, S_{L,OR}, S_{L,XOR}$ (green). In the inference or test phase (see the brown section in the zoom-in), the result of the learning process is revealed. While keeping the inputs S_A and S_B identical, the output signal S_{OUT} (brown) during the test phase depends upon the learning sequence. Particularly, applying alternative bias, in turn, causes the network to output XOR, a non-linear operation. Furthermore, mild negative bias induces an AND operation response, while strong negative bias leads to an OR operation. The table in the right side shows the truth table to facilitate the mapping to the operations during the test, indicated with indexes over the pulses.

3.3 Showcase: Compressed MNIST Classification

Based on the mathematical derivations of Target Learning, our memristive PNs can learn any function (see Supplementary Section 1.4, Appendix D). To demonstrate scalability beyond fundamental computational primitives, we elevate the task complexity by evaluating our PN circuit architecture on a compressed version of the MNIST dataset. This benchmark is executed using a physics-based SPICE hardware simulation to capture realistic circuit dynamics. Unlike conventional software-based machine learning frameworks that rely on idealized mathematical abstractions, our SPICE simulation models the analog transient behavior and device-level dynamics of the circuit.

We start by downsampling the standard images to a low-resolution 8x8 pixel format. This helps to reduce the high computational demands inherent to component-level transient simulation, see Figure 5(a). To encode the signal, we map pixel intensity to electrical pulses: white and gray pixels receive a voltage amplitude proportional to their grayscale value (scaling continuously between 0 V and a maximum pulse voltage), while black pixels yield no electrical input, see Figure 5(b). In a next step, see Fig. 5(c), we flatten the two-dimensional pixel array into a one-dimensional vector, yielding an

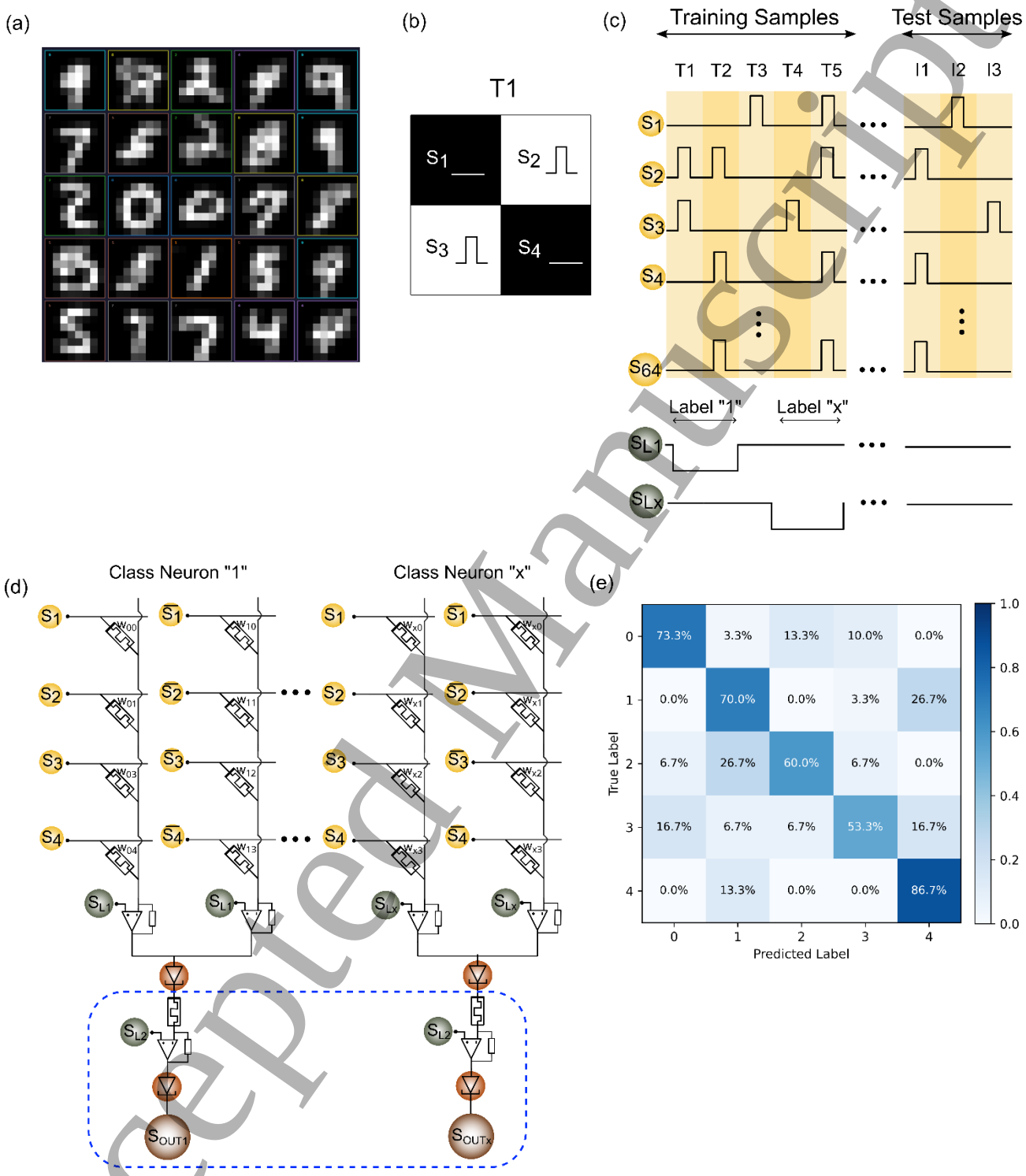

Figure 5 MNIST classification via target learning. (a) Representative downsampled 8×8 pixel MNIST images. (b) Illustration of a training example (T1), where black pixels translate to zero voltages and grayscale pixels translate to the applied pulses. (c) The 64 pixels of an image are first mapped to a training vector (e.g. T1). Subsequently, up to 100 training samples from within a digit class are lined up and used for training e.g. Lable "1". The testing is then performed with 30 testing samples per digit. (d) Circuit implementation, featuring a complete pyramidal neuron (PN) circuit for each digit class with dedicated potentiating and inhibiting units driven by positive and negative pulses, respectively. The output could be implemented by appending another PN neuron as indicated inside the blue dashed line. Here, it is not

included in the simulation. (e) Confusion matrix for a 5-digit classification task (digits "0"–"4"). Showing an average of 68.7% accuracy.

input layer dimensionality of 64. We apply a localized learning signal, S_L, concurrently with the corresponding training class, targeting it exclusively to the neuron representing the correct class (digits). For the training protocol, we concatenate the input image vectors of the same classes sequentially. The practical implementation of this learning process is depicted in Figure 5 (d). A dedicated electronic neuron, configured with a complementary potentiating and inhibiting column, detects each distinct digit class. The potentiating column processes the non-inverted input vector, whereas the inhibiting column receives the inverted signal. Both pathways adhere to the circuit topology of the associative pyramidal neuron detailed in (c) of the main text, where we route the learning signal to the non-inverting input. Upon spatial summation of the weighted inputs, a MIM tunnel junction applies a non-linear transformation to resolve the resultant output signal, S_{OUT}, see **Figure 6**(b). While the output calibration was done in software, it could be realized with another small PN neuron at the output, indicated by blue dashed lines. This will be realized in future work.

We evaluate this architecture on a 5-class subset (digits "0" to "4") (see Figure 5(d–e)), utilizing 100 training samples per digit and 30 testing samples per digit. Our physical PN model (based on experimental data) achieves an accuracy of 68.7%. This simple setup yields already 69% accuracy, despite being a hardware-constrained proof-of-concept: the classifier consists of a single-layer network, the training set is restricted to a small sample of compressed MNIST images, and the learning signal is a simple binary vector. Our primary objective was to demonstrate hardware viability rather than maximize classification benchmarks. Future work will focus on scaling the physical network to multi-layer architectures and refining the learning signals to improve classification accuracy on high-resolution datasets.

4 Conclusion

This work introduces a proof-of-concept circuit supporting on-chip training, opening new pathways towards fully independent neuromorphic systems. Inspired by a biologically validated learning process, learning signals enforce the desired neuronal activity and the optimization occurs while minimizing the enforcing signal. The proposed bio-inspired circuit aims to provide a device-agnostic solution, tolerant to device non-idealities (see Figure 3-5).

We have selected filamentary memristors to function as the electronic synapses, as their dynamic responses closely resemble those of biological synapses. Together with an operational amplifier and other passive elements, our simple circuit is capable of deploying associations. We utilize this building block to create a network with a hidden layer capable of realizing more complex nonlinear operations like XOR, and to reconfigure it to perform other logical operations like AND and OR. Furthermore, we demonstrate the scalability of our framework by applying it to a hand-written digit classification task using a compressed MNIST dataset.

Our circuit can be extended to contain multiple inputs per neuron and multiple hidden layers. Doing so would allow for more complex operations. Moreover, our current circuit is not restricted to digital operations, as memristive elements can exhibit analogue tuneability, thus enabling analogue computation. Both require finding the appropriate learning signals, which can be done by an analog circuitry [39]. Combining existing approaches to find target signal together with our memristive circuit will be considered in future work.

To our knowledge, this work represents the first demonstration that implements the recently discovered TL algorithm natively on-chip with memristors. Our neuron models rely on hardware-native local learning rules, which relax the stringent hardware requirements typically associated with BP, making the model suited for physical implementation on analog devices.

5 Materials and methods

5.1 Memristor characteristics

A memristor, or resistive switch, is an electronic device whose internal state changes under an external influence [40,41]. The signature of these devices is a “pinched” hysteresis loop in their voltage-current (V-I) characteristics, which is constrained to pass through the origin, see Figure 6(c). In this study we focus on a specific subclass of non-volatile memristors referred to as filamentary memristors or resistive random-access memories (RRAM). Those devices exhibit multi-level resistive switching and non-volatile retention, wherein the programmed resistive state is maintained without an applied external signal. The switching mechanism of this system is governed by the formation and dissolution of a conducting filament within an insulating layer induced by the application of a voltage to the metal/insulator/metal (MIM) structure [42].

Depending on the filament condition, the device can switch between High Conductance State (HCS) when the filament is formed, and Low Conductance State (LCS), when the filament is ruptured. Under a sufficiently high voltage input, the device undergoes a SET process, where it transitions from LCS to HCS. In contrast, a voltage in the opposite polarity switches the device back to LCS through a RESET process, see Figure 6(d). Notably, these events occur exclusively when the applied bias reaches the so-called SET and RESET voltage thresholds.

5.2 Device fabrication

In the case of the electronic synapses or non-volatile memristor we employ a well-known structure composed of Ta/Ta_2O_5/Pt with a nominal lateral size of 200 nm [43,44]. The 10 nm thick Ti adhesion layer and the 40 nm thick Pt bottom electrode of the Ta/Ta_2O_5/Pt stack were consecutively deposited on a 300 nm thick SiO_2 on Si substrate by electron beam evaporation at a base pressure of 10^{-6} mbar at a rate of 0.1 nm/s and 0.05 nm/s respectively. The following 10 nm thick Ta_2O_5 was deposited by reactive HiPIMS Magnetron Sputtering from a Ta target in a plasma formed by a ratio of 45 sccm Ar and 5 sccm O_2 at a base pressure of 5 mTorr using 150 W RF power. As an attempt to narrow the top electrode area, nm-sized holes were patterned and etched in the Ta_2O_5 layer using Electron Beam Lithography (EBL) and Reactive Ion Etching (RIE) below 100 nm, leading to area-dependent depths ranging from 2.5 to 7.5 nm deep. In the next step, the 40 nm Ta top electrode and 20 nm Pt capping layer were deposited using HiPIMS Magnetron Sputtering. The Ta layer was deposited using 130 W DC Sputtering at a base pressure of 6 mTorr and 45 sccm Ar flow. In contrast, the Pt layer was sputtered at 100 W RF sputtering at 6 mTorr and 45 sccm Ar flow. To finalize, the contact electrodes were composed by 10 nm thick Ti and 90 nm thick Au, both deposited consecutively using electron beam evaporation. This final layer was patterned using standard photolithography whereas the rest of the layers were patterned using EBL, see Figure 6 (a).

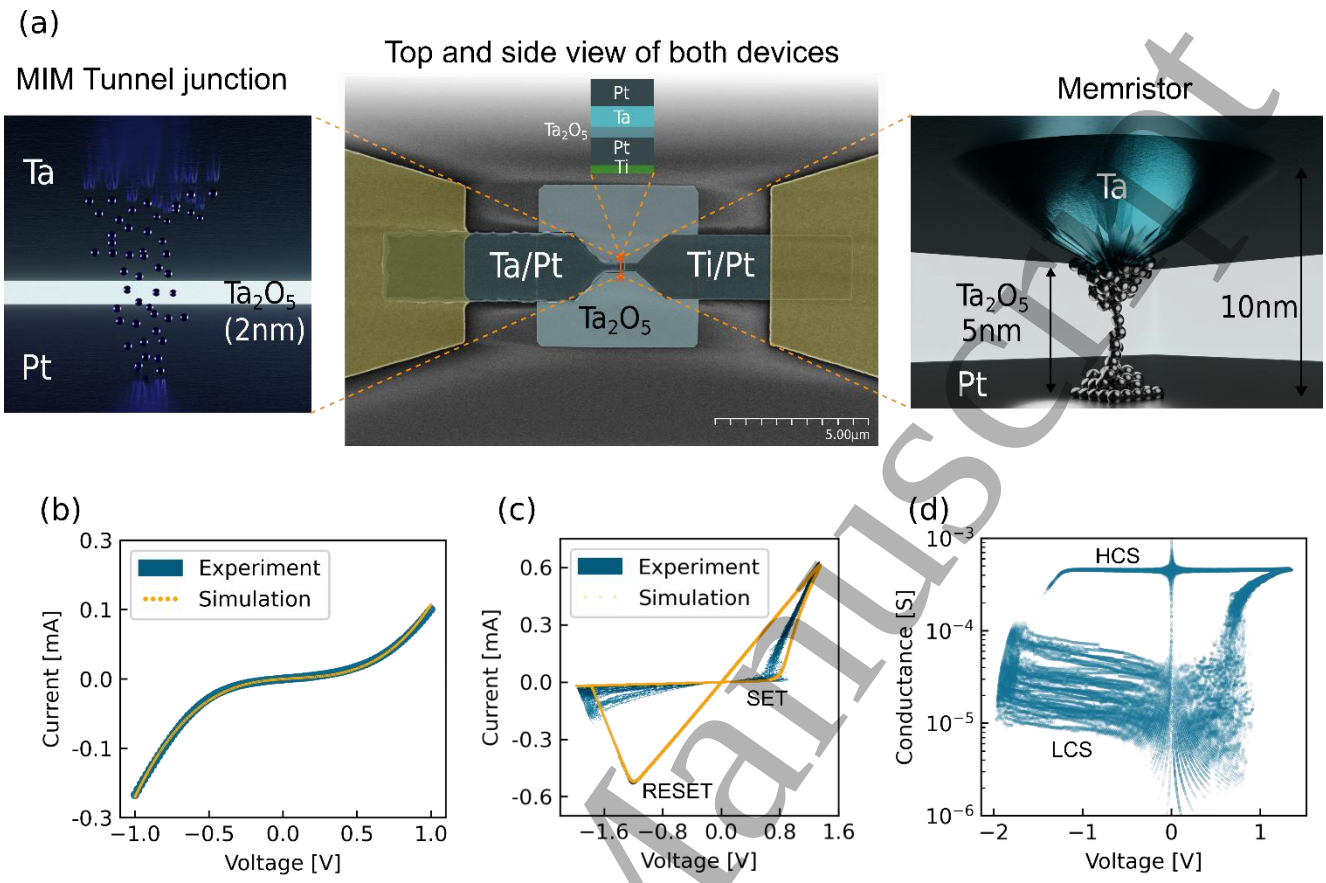

Figure 6 Device characteristics. (a) False-colored scanning electron microscopy image of a representative sample with their respective cross-sections (vertical cut along red line). Artistic illustration of MIM tunnel junction (left) and filamentary memristor (right). Both device types share the same layout (top and side view shown in the middle). Primarily, the difference is the insulating thickness, being ~2 nm for the MIM tunnel junction and ~5 nm for the Memristor. **(**b) I-V characteristics of MIM tunnel junction. The yellow trace corresponds to the I-V response from the tunnel junction simulation model. (c) I-V characteristics of a Ta_2O_5-based memristor. The yellow trace shows the I-V response of our memristor simulation model. (d) Equivalent conductance response.

The fabrication process of the MIM tunnel junction used as the activation function closely resembles the nonvolatile memristor's one. On top of the Ti/Pt bottom electrode, a 2 nm thick layer of Ta2O5 is sandwiched by the top 80 nm thick Ta electrode capping it finally with 30 nm thick Pt layer. In this case, no hole etching was performed. To finalize, the contact pads were evaporated using 10 nm thick Ti adhesion layer and 190 nm thick Au, to ensure good ohmic contacts, see Figure 6 (a).

5.3 Device custom models

To investigate the correspondence between the electrical behavior of filamentary memristors and basal synaptic dynamics, we developed a phenomenological model capable of predicting the device's

evolution under specified electrical stimuli, Figure 6(c). This approach abstracts the underlying switching mechanisms but accurately captures the essential dynamics of the memristive state, x, in response to external voltage inputs. To account for the influence of the system's history on the evolution of x, we employed a self-consistent dynamical simulation model, adapted by previous simulation models such as [45].

The maximum current I and the HCS, $G_{mem,HCS}$ are influenced by the value of a series resistor R_S and the input voltage V_{drive}, $G_{mem} = (\frac{V_{drive}}{I} - R_s)^{-1}$. They are determined at any given time stamp dt by the internal state x. x represents the internal evolution of the filament, being 1 when the filament is fully formed and 0 when the filament is broken. When comparing to the PN model in [17], the internal state x could be interpreted as the synaptic strength, where x=1 implies a strong connection, while x=0 indicates a weak synapse.

Regarding the modeling of the activation function device, the non-linear I-V characteristics are approximated by a superposition of two distinct transport regimes. At low bias, transport is dominated by Direct Tunneling (DT), described by a third-order Taylor expansion, $I = G_0V + \alpha V^2 + \gamma V^3$, which accounts for the junction's zero-bias conductance, G_0, inherent asymmetry of the MIM stack, α, and primary non-linearity, γ [46]. At higher bias, the electronic transport transitions to Fowler-Nordheim (F-N) Tunneling, modeled as $I_{FN} \propto V^2 \exp\left(-\frac{B}{V}\right)$, where B represents constant values. This composite model, combined with a parasitic series resistance, R_s, behaves like a threshold activation function, see Figure 6(b). While the discussed MIM tunnel junctions were selected for these experiments, other components represent a viable alternative with similar performance, for instance, two antiparallel diodes with different threshold voltages.

5.4 Transfer Function of PN Model

The transfer function of our PN model depicted in Figure 2(c) may be given by:

$$S_{OUT} = S_L + \sum_{i=1}^{N} (S_L - S_i)(\frac{w_i(x)}{w_F}),$$

where N is the total number of synapses. The term $w_i(x)$ denotes the conductance (weight) of the i-th non-volatile memristor which depends on its internal state x, while w_F represents the feedback

conductance of the system. The internal state variable *x* evolves dynamically over time based on the applied voltage drop across the device, governed by:

$$\frac{dx}{dt} = f(x, S_L - S_i),$$

Here, $S_L - S_i$ represents the programming voltage drop across the memristor terminals.

5.5 Simulation environment and experimental setup

We developed SPICE simulations of our electronic circuits using the open-source simulator LTSpice. We developed custom models of both the nonvolatile memristor used as electronic synapse and the MIM tunnel junction representing the activation function.

The circuit designs were first prototyped on a breadboard to serve as an initial proof of concept. Additional commercial devices were used, afterwards pinned and wired to the breadboard. To access the fabricated devices, 40 GHz RF Picoprobes were connected using SMA cables. A National Instruments Data Acquisition Card USB-6361 was used to send the analogue input signals S_A and S_B and collect the output signal S_{OUT}. The learning signal S_L was applied using a Keysight Arbitrary Wave Generator 33500.

The reconfigurable circuit was measured in three equivalent blocks as individual perceptrons and the resulting output was used as the input of the following block. To be more precise, each memristive column containing two devices was measured using the RF Picoprobes and lead into an op-amp. The two individual outputs were used as inputs to the output layer where the two MIM tunnel junctions were connected using the same RF picoprobes and directed into the final op-amp.

Acknowledgements

This research was financially supported by the Werner Siemens Stiftung (WSS) and the UZH postdoc grant (FK-24-122). We are thankful to the Cleanroom and Operation Team of the Binning and Rohrer Nanotechnology Center (BRNC) for their support in device fabrication.

This work also profited from discussions with Markus Fischer, Raphael Gisler, Kshipra Srikrishnaprabhu, Martin Stecher, Darius Urbonas, Miklós Csontos, Mathieu Luisier, Filippo Moro and Benjamin Grewe.

Author contribution

P.V.A., N.J.O. and A.L.C.I. developed the original idea. N.J.O. fabricated the devices. N.J.O. and A.L.C.I. performed the simulations. N.J.O., M.B. and M.D. executed the experiments. N.J.O. and P.V.A. wrote the original draft. P.V.A. and J.L. supervised the project. All the authors contributed to the discussions and the revision of the manuscript.

References

1. J. Kaplan, S. McCandlish, T. Henighan, et al., "Scaling Laws for Neural Language Models," arXiv:2001.08361 (2020).
2. G. Indiveri and S.-C. Liu, "Memory and Information Processing in Neuromorphic Systems," Proceedings of the IEEE **103**, 1379–1397 (2015).
3. D. V. Christensen, R. Dittmann, B. Linares-Barranco, et al., "2022 roadmap on neuromorphic computing and engineering," Neuromorph. Comput. Eng. **2**, 022501 (2022).
4. M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (2014), pp. 10–14.
5. P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," Science **345**, 668–673 (2014).
6. Y. Li, Z. Wang, R. Midya, et al., "Review of memristor devices in neuromorphic computing: materials sciences and device challenges," J. Phys. D: Appl. Phys. **51**, 503002 (2018).

7. W. Chen, L. Song, S. Wang, et al., "Essential Characteristics of Memristors for Neuromorphic Computing," Advanced Electronic Materials **9**, 2200833 (2023).
8. S. Jiang, S. Nie, Y. He, et al., "Emerging synaptic devices: from two-terminal memristors to multiterminal neuromorphic transistors," Materials Today Nano **8**, 100059 (2019).
9. E. Passerini, M. Lewerenz, A. Schneuwly, et al., "Reconfigurable artificial neuron and synapse enabled through a single alloyed memristor," Sci Rep **15**, 29745 (2025).
10. J. Bisquert, W. Shim, S.-Y. Kim, et al., "Synaptic Function in Memristor Devices for Neuromorphic Circuit Applications," Advanced Electronic Materials **11**, 2400903 (2025).
11. P. Baldi, P. Sadowski, and Z. Lu, "Learning in the machine: The symmetries of the deep learning channel," Neural Networks **95**, 110–133 (2017).
12. G. C. Adam, A. Khiat, and T. Prodromakis, "Challenges hindering memristive neuromorphic hardware from going mainstream," Nat Commun **9**, 5267 (2018).
13. Y. Zhang, Z. Wang, J. Zhu, et al., "Brain-inspired computing with memristors: Challenges in devices, circuits, and systems," Appl. Phys. Rev. **7**, 011308 (2020).
14. M. J. Rasch, F. Carta, O. Fagbohungbe, et al., "Fast and robust analog in-memory deep neural network training," Nat Commun **15**, 7133 (2024).
15. T. Gokmen and W. Haensch, "Algorithm for Training Neural Networks on Resistive Device Arrays," Front. Neurosci. **14**, (2020).
16. M. Payvand, M. E. Fouda, F. Kurdahi, et al., "On-Chip Error-Triggered Learning of Multi-Layer Memristive Spiking Neural Networks," IEEE Journal on Emerging and Selected Topics in Circuits and Systems **10**, 522–535 (2020).
17. P. V. Aceituno, S. de Haan, R. Loidl, et al., "Challenging Backpropagation: Evidence for Target Learning in the Neocortex," 2024.04.10.588837 (2025).
18. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1986).
19. F. Crick, "The recent excitement about neural networks," Nature **337**, 129–132 (1989).
20. T. P. Lillicrap, D. Cownden, D. B. Tweed, et al., "Random synaptic feedback weights support error backpropagation for deep learning," Nat Commun **7**, 13276 (2016).
21. J. Sacramento, R. P. Costa, Y. Bengio, et al., "Dendritic cortical microcircuits approximate the backpropagation algorithm," (2018).
22. A. Payeur, J. Guerguiev, F. Zenke, et al., "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," Nat Neurosci **24**, 1010–1019 (2021).

23. H. Song, M. Kim, D. Park, et al., "Learning from Noisy Labels with Deep Neural Networks: A Survey," (2022).
24. D. Sussillo and L. F. Abbott, "Generating Coherent Patterns of Activity from Chaotic Neural Networks," Neuron **63**, 544–557 (2009).
25. B. DePasquale, C. J. Cueva, K. Rajan, et al., "full-FORCE: A target-based method for training recurrent networks," PLOS ONE **13**, e0191527 (2018).
26. A. Meulemans, F. S. Carzaniga, J. A. K. Suykens, et al., "A Theoretical Framework for Target Propagation," (2020).
27. Y. Song, B. Millidge, T. Salvatori, et al., "Inferring neural activity before plasticity as a foundation for learning beyond backpropagation," Nat Neurosci **27**, 348–358 (2024).
28. P. V. Aceituno, S. de Haan, R. Loidl, et al., "Challenging Backpropagation: Evidence for Target Learning in the Neocortex," 2024.04.10.588837 (2025).
29. B. Podlaski and C. K. Machens, "Biological credit assignment through dynamic inversion of feedforward networks," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2020), Vol. 33, pp. 10065–10076.
30. C. Grienberger and J. C. Magee, "Entorhinal cortex directs learning-related changes in CA1 representations," Nature **611**, 554–562 (2022).
31. S. Holler, G. Köstinger, K. A. C. Martin, et al., "Structure and function of a neocortical synapse," Nature **591**, 111–116 (2021).
32. M. Graupner and N. Brunel, "Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location," Proc Natl Acad Sci U S A **109**, 3991–3996 (2012).
33. C. Clopath and W. Gerstner, "Voltage and Spike Timing Interact in STDP – A Unified Model," Front. Synaptic Neurosci. **2**, (2010).
34. B. Chih, H. Engelman, and P. Scheiffele, "Control of Excitatory and Inhibitory Synapse Formation by Neuroligins," Science **307**, 1324–1328 (2005).
35. J. T. R. Isaac, R. A. Nicoll, and R. C. Malenka, "Evidence for silent synapses: Implications for the expression of LTP," Neuron **15**, 427–434 (1995).
36. K. P. Körding and P. König, "Supervised and Unsupervised Learning with Two Sites of Synaptic Integration," J Comput Neurosci **11**, 207–215 (2001).
37. M. Larkum, "A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex," Trends Neurosci **36**, 141–151 (2013).

38. M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry* (The MIT Press, 2017).
39. A. M. E. Meulemans, B. F. Grewe, M. T. Farinha, et al., "Neural network circuit for deep artificial neuronal networks using deep feedback control," United States patent US20240281647A1 (August 22, 2024).
40. M. Lanza, S. Pazos, F. Aguirre, et al., "The growing memristor industry," Nature **640**, 613–622 (2025).
41. M.-K. Song, J.-H. Kang, X. Zhang, et al., "Recent Advances and Future Prospects for Memristive Materials, Devices, and Systems," ACS Nano **17**, 11994–12039 (2023).
42. A. Wedig, M. Luebben, D.-Y. Cho, et al., "Nanoscale cation motion in TaOx, HfOx and TiOx memristive systems," Nature Nanotech **11**, 67–74 (2016).
43. A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, et al., "Sub-nanosecond switching of a tantalum oxide memristor," Nanotechnology **22**, 485203 (2011).
44. M. Csontos, Y. Horst, N. J. Olalla, et al., "Picosecond Time-Scale Resistive Switching Monitored in Real-Time," Advanced Electronic Materials **9**, 2201104 (2023).
45. D. Molnár, T. N. Török, J. Volk, et al., "Neural Information Processing and Time-Series Prediction with Only Two Dynamical Memristors," Advanced Electronic Materials **12**, e00353 (2026).
46. W. F. Brinkman, R. C. Dynes, and J. M. Rowell, "Tunneling Conductance of Asymmetrical Barriers," Journal of Applied Physics **41**, 1915–1921 (1970).