


The neuromorphic Mosaic: re-configurable in-memory small-world graphs

Working Paper**Author(s):**

Dalgaty, Thomas; Moro, Filippo; De Pra, Alessio; [Indiveri, Giacomo](#) ; Vianello, Elisa; Payvand, Melika

Publication date:

2021-08-06

Permanent link:

<https://doi.org/10.3929/ethz-b-000529332>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Originally published in:

Research Square, <https://doi.org/10.21203/rs.3.rs-780916/v1>

The neuromorphic Mosaic: re-configurable in-memory small-world graphs

Thomas Dalgaty^{*1,2}, Filippo Moro², Alessio De Pra², Giacomo Indiveri^{*3}, Elisa Vianello^{*2}, and Melika Payvand^{*3}

¹CEA, LIST, Université Paris-Saclay, Palaiseau, France

²CEA, LETI, Université Grenoble Alpes, Grenoble, France

³Institute for neuroinformatics, University of Zurich, Zurich, Switzerland

*thomas.dalgaty@cea.fr, giacomo@ini.uzh.ch, elisa.vianello@cea.fr, melika@ini.uzh.ch

ABSTRACT

Thanks to their non-volatile and multi-bit properties, memristors have been extensively used as synaptic weight elements in neuromorphic architectures. However, their use to define and re-program the network connectivity has been overlooked. Here, we propose, implement and experimentally demonstrate Mosaic, a neuromorphic architecture based on a systolic array of memristor crossbars. For the first time, we use distributed non-volatile memristors not only for computation, but also for routing (i.e., to define the network connectivity). Mosaic is particularly well-suited for the implementation of re-configurable small-world graphical models, with dense local and sparse global connectivity - found extensively in the brain. We mathematically show that, as the networks scale up, the Mosaic requires less memory than in conventional memristor approaches. We map a spiking recurrent neural network on the Mosaic to solve an Electrocardiogram (ECG) anomaly detection task. While the performance is either equivalent or better than software models, the advantage of the Mosaic was clearly seen in respective one and two orders of magnitude reduction in energy requirements, compared to a micro-controller and address-event representation-based processor. Mosaic promises to open up a new approach to designing neuromorphic hardware based on graph-theoretic principles with less memory and energy.

Introduction

Graphs are omnipresent data structures which capture interactions (i.e., edges) between multiple units (i.e., nodes). They are the backbone of many computational systems that represent relational information between their interacting entities¹. Neural networks are an example of a graph. Graphs can be used to study and represent both biological and artificial neural networks, where neurons correspond to the nodes of a graph and the connections between them (i.e., weights or synapses) correspond to edges. Biological nervous systems, shaped over millions of years of evolution, have developed many computational principles that can be captured using graphical networks. Therefore, building computing architectures based on the same organizational principles is a promising path towards realizing powerful artificially intelligent systems.

One such important organizing principle is “small worldness” which is found extensively in empirical studies of structural and functional biological neural networks^{2,3} (Fig. 1a). In such a structure, short paths connecting neighboring nodes (neurons) are more common than long-range connections, which are sparse (Fig. 1b). The mix of dense local and sparse distal connectivity gives rise to efficient global coordination and information flow based on local interactions⁴. A connectivity matrix of an example small-world graph is plotted in Fig. 1c. It is characterized by the heavy connectivity along the matrix diagonal, with increasingly fewer connections between the further off-diagonal neuron pairs.

Crossbars of conductive memory elements have often been proposed as a means of realizing such models on hardware (Fig. 1d)⁵⁻¹¹. In these structures, a memory element connects a series of vertically running metal lines (i.e., columns) with orthogonal ones (rows). The conductance state of each memory corresponds to the synaptic weight parameter of a neuron, which is located at the end of each row. Such architectures perform matrix multiplication, the core operation of a neural network, in-memory and in an analog fashion. Relative to a von-Neumann architecture, this dramatically reduces the volume of data movement which in turn largely reduces the energy required to run neural network models¹²⁻¹⁷.

Resistive Random Access Memory (RRAM) devices, otherwise referred to as memristors, have emerged as a promising memory element for such in-memory crossbar architectures¹⁸⁻²³. They can be programmed with multiple discrete conductance levels²⁴ corresponding to different synaptic weight values in the connectivity matrix of Fig. 1c. Moreover, RRAMs retain information in a non-volatile fashion, which eliminates the static power consumption related to the storage of neural network weights²⁵. In particular, biologically inspired Spiking Neural Networks (SNNs) are well matched to RRAMs since the devices in the crossbar are read asynchronously and sparsely - thus dynamic power is also reduced.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

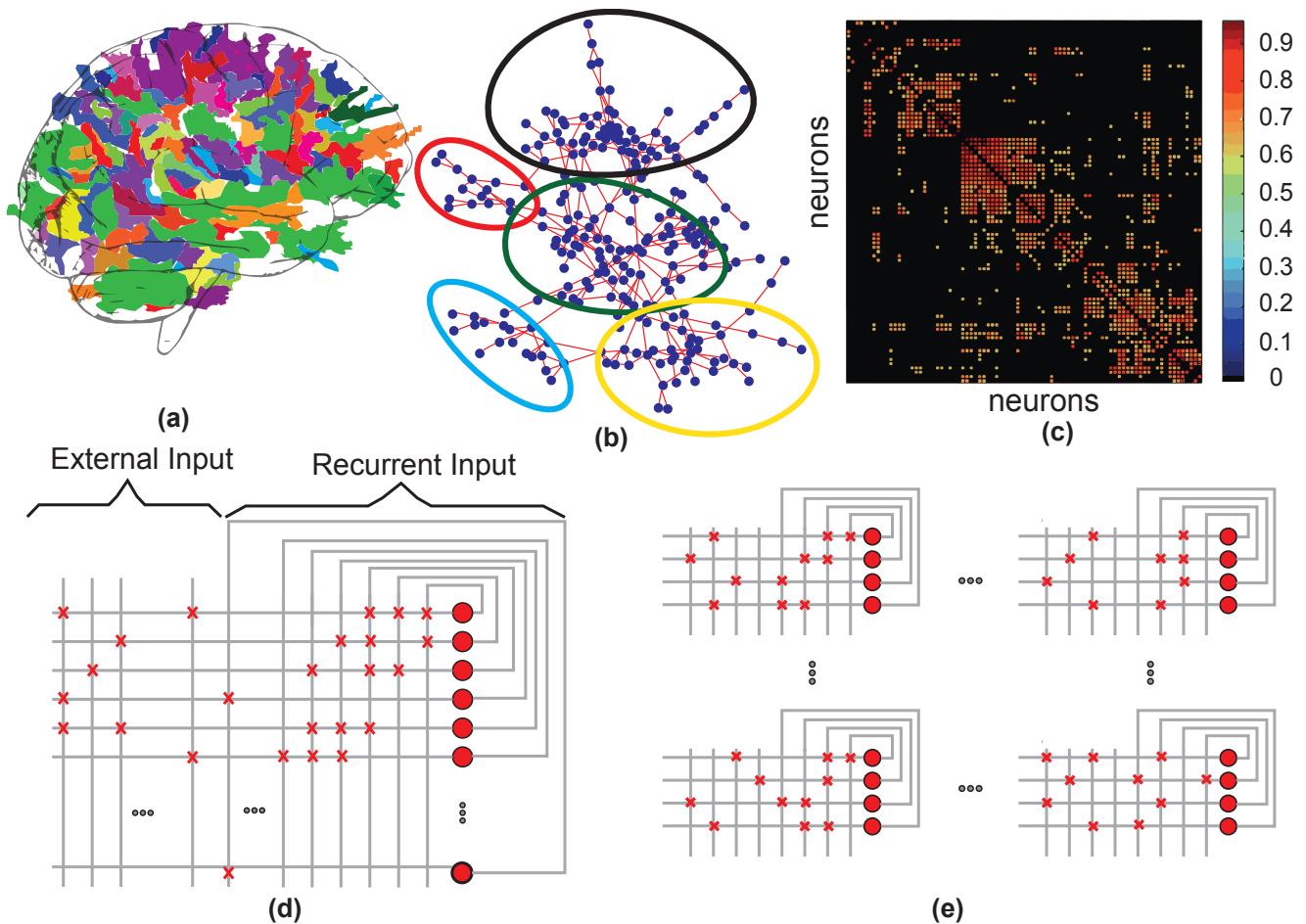


Figure 1. Small-world graphs in biological and graphical neural networks. (a) Depiction of small-worldness in the brain with highly clustered neighboring regions highlighted with the same color. (b) An example network model characteristic of a small-world graph. Five local clusters of nodes connect densely with each other and are interconnected with a sparse set of hub-like nodes. (c) (adapted from²⁸). The functional connectivity matrix based on data from human functional Magnetic Resonance Imaging showing the properties of a small-world graph. The rows and columns represent neuron indices. The diagonal region of the matrix contains the strongest connectivity which represent the connections between the neighboring neurons. The off-diagonal elements are not connected. (d) Hardware implementation of the connectivity matrix in c. Neurons and synapses are arranged in a crossbar architecture, where the inputs are in columns and the sum of the products of inputs and synaptic weights are calculated at the row. The column input could either be recurrent (coming from other neurons) or external (coming from real-world signals). (e) The Mosaic architecture with small "tiles" distributed in a two-dimensional mesh.

27 Figure 1d shows an example of a Recurrent Spiking Neural Network (RSNN), realized by an RRAM crossbar that contains
 28 both synapses receiving spike signals from external inputs, and from recurrent connections of the neurons in the network.
 29 However, scaling this to large SNNs requires a very large crossbar of memristors. Problems such as current sneak-paths,
 30 parasitic resistance and capacitance of the metal lines, as well as excessively large read currents limit their maximum size
 31 in practice^{26,27}. Moreover, a single large crossbar would result in a wasteful utilization of the off-diagonal devices in the
 32 implementation of bio-inspired graphs with small-world properties (Fig. 1(c)).

33 To implement artificial spiking small-world graphs more efficiently, we propose and experimentally demonstrate a new
 34 re-configurable neuromorphic computing architecture called the "Mosaic" (Fig. 1(e)). The Mosaic is a two-dimensional systolic
 35 matrix of distributed "tiles", each based on a small crossbar of RRAM, that can serve either as analog spiking or spike routing
 36 elements. Effectively, the Mosaic dices up one large crossbar into numerous smaller tiles with different functions (Fig. 1(e)).
 37 Importantly, the Mosaic uses RRAM not only to store synaptic weights and carry out neural processing, but also to define the
 38 routing patterns linking up neighboring tiles.

39 The Mosaic lends itself to the implementation of small-world networks more efficiently, resulting in a better utilization of the

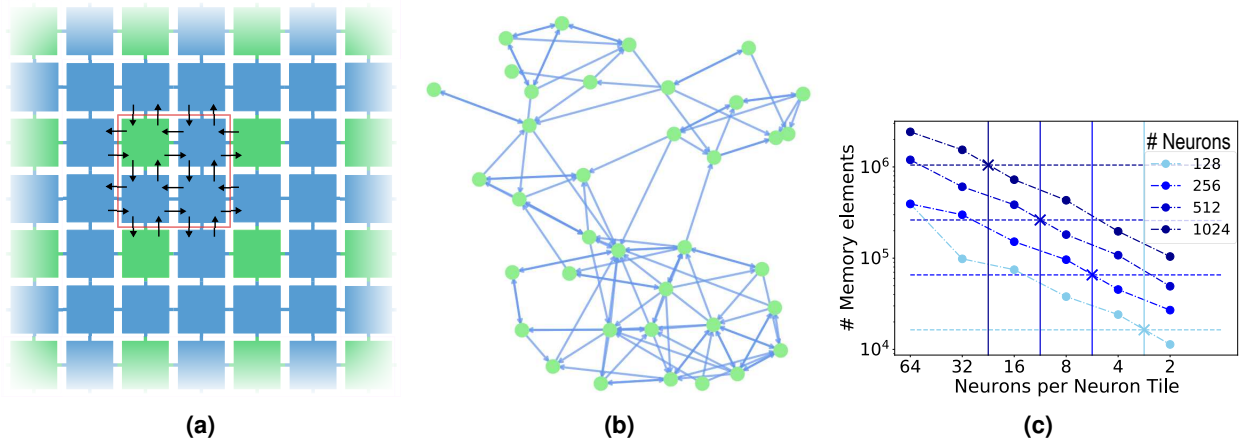


Figure 2. (a) The neuromorphic memory Mosaic. Green squares correspond to neuron tiles and blue squares to routing tiles. The bridges drawn between tiles correspond to the North, South, East, and West signal buses carrying the V_{in} and V_{out} voltage pulses. (b) An example graph resulting from the random programming of devices in each of the tiles in the Mosaic pictured in part (a). The green circles correspond to neurons which exist in the neuron tiles and the blue edges are defined by the resulting paths that are formed between neuron tiles through the routing tiles. (c) Plot showing the required bits of memory for different number of total neurons in a network model depending on the size of the neuron tile. The number of bits of memory is referred to as resistive memory devices programmed in a binary fashion. The horizontal dashed line indicates the number of required memory bits using a fully-connected RRAM crossbar array for different network sizes. The cross (X) illustrates the cross-over point beyond which Mosaic approach becomes favorable.

allocated memory resources. Moreover, it introduces a novel routing approach different from the conventional Address-Event Representation (AER) scheme in SNN hardware^{29,30} without the need for storing each neuron’s connectivity information in local memories that draw static power and can consume a large chip area (Supplementary Note 1).

In this Article, we first present the Mosaic architecture and mathematically quantify its memory footprint savings while implementing small-world neural networks compared to a single large memristor crossbar. We then report electrical circuit measurements from tiles that we designed and fabricated in 130 nm CMOS technology co-integrated with Hafnium dioxide-based RRAM devices. Calibrated on these measurements, we apply a simulation of the Mosaic to run a RSNN applied to the detection of arrhythmic heart beats from Electrocardiography (ECG) recordings. We compare our approach to equivalent implementations using a microprocessor and an AER-based neuromorphic processor. Per heartbeat we find that Mosaic achieves reductions in the total signal routing energy of two and one order of magnitudes respectively.

Results

The Mosaic architecture is illustrated in Fig. 2a as an array of tiles which are distributed in a two-dimensional systolic fashion. Each of the tiles consist of a small memristor crossbar which can receive and transmit spikes to and from their neighboring tiles to the North (N), South (S), East (E) and West (W) directions (Supplementary Fig. S1). The green squares represent “neuron tiles” and correspond to small crossbars (Fig. 1e) that store the synaptic weights of several Leaky Integrate and Fire (LIF) neurons. These neurons are implemented using analog circuits and are located at the termination of each row, emitting voltage spikes at their outputs³¹. These spikes are communicated between neuron tiles through a mesh of blue squares which represent “routing tiles”. Routing tiles encompass small crossbars that determine the connectivity patterns between neuron tiles. The state of each device in the crossbar determines the output direction (i.e., N, S, E, W) towards which its input spike propagates, i.e. steering it towards its intended target neuron elsewhere in the Mosaic. Together, the two tiles give rise to a continuous *mosaic* of neuromorphic computation and memory for realizing spiking small-world neural networks.

An example small-world neural network topology, obtained by randomly programming memristors in a computer model of the Mosaic (see Methods) is shown in Fig. 2b. The resulting graph exhibits an intriguing set of connection patterns that reflect those found in many of the small-world graphical motifs observed in animal nervous systems. For example, central ‘hub-like’ neurons with connections to numerous nodes, reciprocal connections between pairs of nodes reminiscent of winner-take-all mechanisms, and a number of heavily connected local neural clusters³. If desired, these graph properties could be adapted on-the-fly by the re-programming the RRAM states in the two tile types (Supplementary Fig. S2). For example, a set of desired small-world graph properties can be achieved by randomly programming the RRAM devices into their High-Conductive State (HCS) with a certain probability (Supplementary Fig. S3). Random programming can for example be achieved elegantly

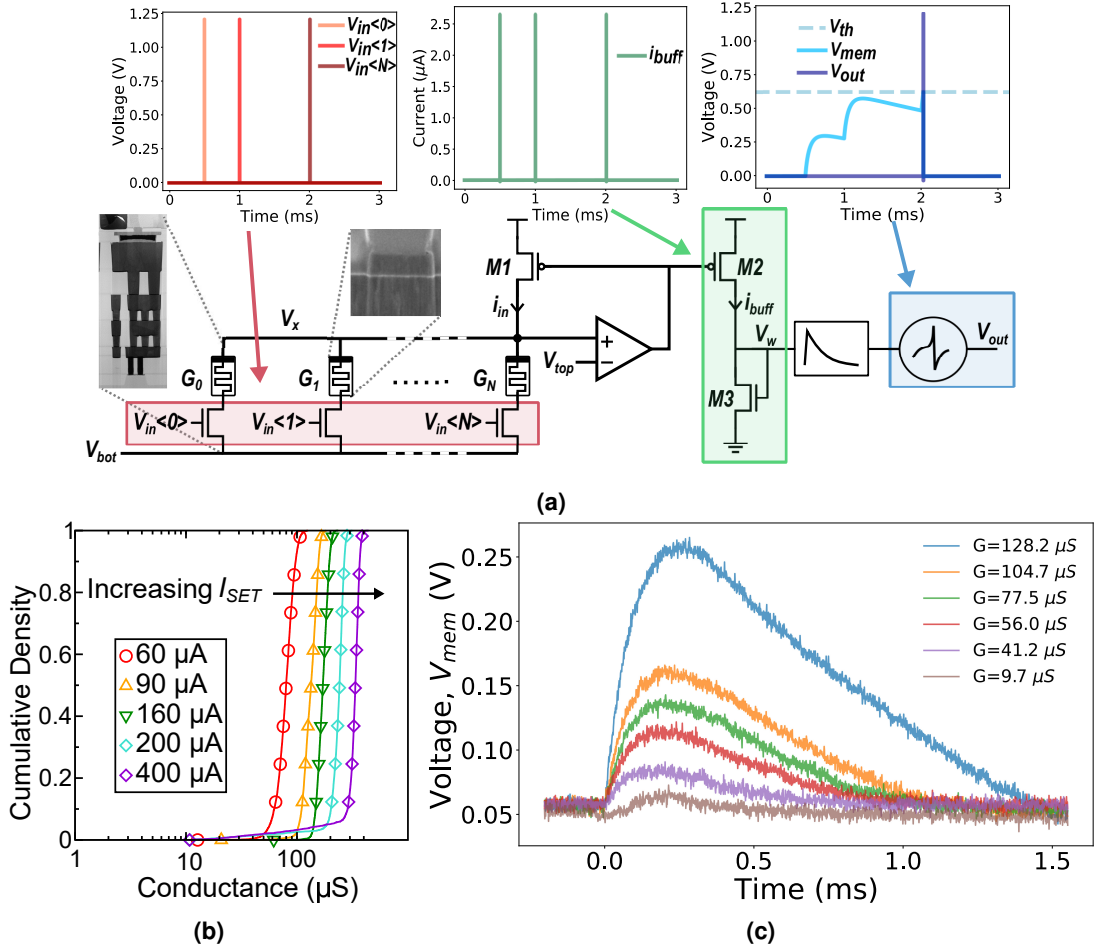


Figure 3. The neuron column circuit with example waveforms. (a) Input (red, left) voltage pulses (spikes), V_{in} , draw a current i_{in} proportional to the conductance state, G_i , of the 1T1R structures. This current is buffered (green, centre), i_{buff} , into a synapse circuit implementing a low pass filter, and in turn injects it into a neuron circuit. The neuron circuit integrates this current into a membrane voltage (blue, right), V_{mem} which causes the neuron to fire at the output after exceeding a threshold V_{th} . Insets of (left) scanning electron and (right) transmission electron microscopy images respectively show cross-sections of the 1T1R stack and the hafnium-dioxide layer sandwiched between top and bottom memristor electrodes. (b) Five cumulative distributions resulting from the application of a single SET programming pulse on each device in an array of 4096 RRAM devices over a range of SET programming currents, I_{SET} . (c) From an initial resting membrane voltage of 0.05V, the membrane voltage waveform recorded by an oscilloscope is plotted in time due to the arrival of a single input pulse. The conductance of the device being read is swept from $10\mu\text{S}$ to $125\mu\text{S}$ and the resulting waveforms are measured for each conductance value.

69 by simply modulating the RRAM SET voltage²⁵.

70 For Mosaic-based small-world graphs, we estimate the required number of memory devices (synaptic weight and routing
 71 weight) as a function of the total number of neurons in a network, through a mathematical derivation (see Methods). Fig. 2c
 72 plots the memory footprint as a function of the number of neurons in each tiles for different network sizes. Horizontal dashed
 73 lines show the number of memory elements using one large crossbar for each network size. The cross-over points, at which the
 74 Mosaic memory footprint becomes favorable, are denoted with a cross. While for smaller network sizes (here 128 neurons) no
 75 memory reduction is observed compared to a single large array, the memory saving becomes increasingly important as network
 76 is scaled. For example, given a network of 1024 neurons and 4 neurons per neuron tile, the Mosaic requires almost one order of
 77 magnitude fewer memory devices than a single crossbar.

78 Neuron tile circuits: small worlds

79 Each neuron tile in the Mosaic is composed of multiple “neuron columns”; a circuit that models a LIF neuron and its synapses.
 80 A neuron column circuit is shown in Fig. 3a. It has N parallel one-transistor-one-resistor (1T1R) RRAM structures at its input.
 81 The synaptic weights of each neuron are stored in the conductance level of the RRAM devices in one column.

The functionality of the neuron column is summarized in the three insets of Fig. 3a. Three input pulses of $V_{in} < 0 >, < 1 >, < N >$ are applied in sequence to the gate of the three 1T1R structures. This results in three current pulses, i_{buff} (green), proportional to the device conductance state. The currents are then injected to a circuit that models biological synaptic dynamics (see Supplementary S7a). This in turn injects an exponentially decaying current into a circuit modelling a biological neuron³². The injected current integrates as a voltage, V_{mem} , on the neuron's membrane capacitor (Supplementary Fig S7b). After the neuron circuit has integrated three input spikes, V_{mem} exceeds its firing threshold (V_{th}) and the circuit emits an output voltage spike.

We fabricated the neuron column of Fig. 3a in a 130 nm CMOS technology integrated with RRAM devices³³. In the fabricated circuit, the memristor corresponding to G_0 was programmed using a sweep of SET currents - resulting in a range of conductance values (Fig. 3b). After programming each device, we applied an input pulse to $V_{in} < 0 >$ and measured the signal V_{mem} which is plotted in Fig. 3c. This experimental result illustrates that the increase in RRAM conductance increases the peak voltage value resulting from a single input pulse, and thus serves well as a programmable synaptic weight element. A layout of this column circuit can be found in Supplementary Fig. S4.

To realize a network using such circuits, these neuron columns are agglomerated into a 'tile'. This is done through stacking consecutive columns side-by-side and connecting their gates row-wise to common input lines (i.e., a crossbar architecture). A simple neuron tile, composed of only two neuron columns receiving two inputs, is shown in Fig. 4a. The top two rows of the crossbar represent the neurons' synaptic weights corresponding to external inputs, while the bottom two represent those of the recurrent connections between neurons within the tile. Following a systolic organization³⁴, each input or output spike can enter from, and exit towards, the neighboring N, S, E, W tiles (Supplementary Fig. S1).

We mapped a simple network topology onto a fabricated neuron tile circuit depicted in Fig. 4a. Two devices highlighted in bold black were programmed to be in their HCS while the gray shaded ones were programmed in their Low-Conductive State (LCS). We then applied a train of input voltage spikes to $V_{in} < 0 >$. The experimental measurements are plotted in Fig. 4b whereby the membrane potential of neuron 0 is observed to periodically increase upon the arrival of each pulse. After the 6th input pulse, V_{mem} exceeds the threshold V_{th} , and the circuit generates an output spike. Because of the recurrent connection between the two neurons defined in the neuron tile, the membrane of neuron 1 integrates an excitatory post-synaptic potential at the same instant (shown in orange). Neuron 0 then enters a refractory period, during which it does not integrate incoming spikes.

Routing tile circuits: connecting small-worlds

A routing tile circuit is shown in Fig. 4c. It acts as a flexible means of configuring how spikes emitted from neuron tiles propagate locally between small-worlds. The functional principles of the routing tile circuits are similar to the neuron tiles. The only difference is the replacement of the biological synapse and neuron circuit models (shown in blue in Fig. 3a) with a simple current comparator circuit. On the arrival of a spike on the column, it compares the device read current (i_{buff} in Fig. 3a) to a reference. If it is greater than this reference, it generates an output spike. Otherwise the output remains at zero. Therefore, the state of the device serves to either pass or block input spikes: in Fig. 4c, each device determines whether input spikes arriving from different input ports (N, S, W, E) are propagated, or not, to each output port.

Using a fabricated routing tile circuit, we demonstrate its functionality experimentally. Two devices (colored in green and red in Fig. 4c) were programmed in respective HCS and LCS. The other devices were left in the pristine state. This has the effect of allowing incoming pulses from N to propagate out to E , while blocking pulses coming from S direction. A pair of pulses were applied to N and S input ports of the fabricated circuit, plotted respectively in solid and dashed blue lines in Fig. 4d. While the E output port remains at zero due to the incoming pulses from the S input port, it switches to a high voltage as a result of incoming pulses from the N input port. This output pulse propagates on-wards to the next tile. Note that in Fig. 4d the output spike does not appear as rectangular due to the large capacitive load of the probe station (see Methods). To allow for greater configurability, more channels per direction can be used in the routing tiles (see Supplementary Fig. S5).

Application to ECG anomaly detection

RSNNs³⁵⁻³⁷ are networks of recurrently connected spiking neurons, whose internal dynamics are a function of the history of their input. They have been demonstrated to be able to process temporally changing sensory information as a result of their internal dynamics³⁸⁻⁴⁰.

Here, we apply a small-world RSNN implemented on the Mosaic to the detection of arrhythmic heartbeats from ECG signals⁴¹ (see Methods). First, we encode the continuous ECG time-series into trains of spikes using a delta-modulation technique, which describes the relative changes in signal magnitude^{42,43}. These spikes are then fed as input into the Mosaic small-world RSNN. As outputs, we designated two sub-populations of neurons within two pairs of the Mosaic's neuron tiles. Elevated spiking activity in either sub-population denotes a normal heart beat (black), or an anomalous one (red) (Fig. 5a).

We train the RSNN in an ex-situ fashion¹¹, using Backpropagation Through Time (BPTT)⁴⁴ with surrogate gradient approximations of the derivative of a LIF neuron activation function⁴⁵ (see Methods). We then transferred the resulting

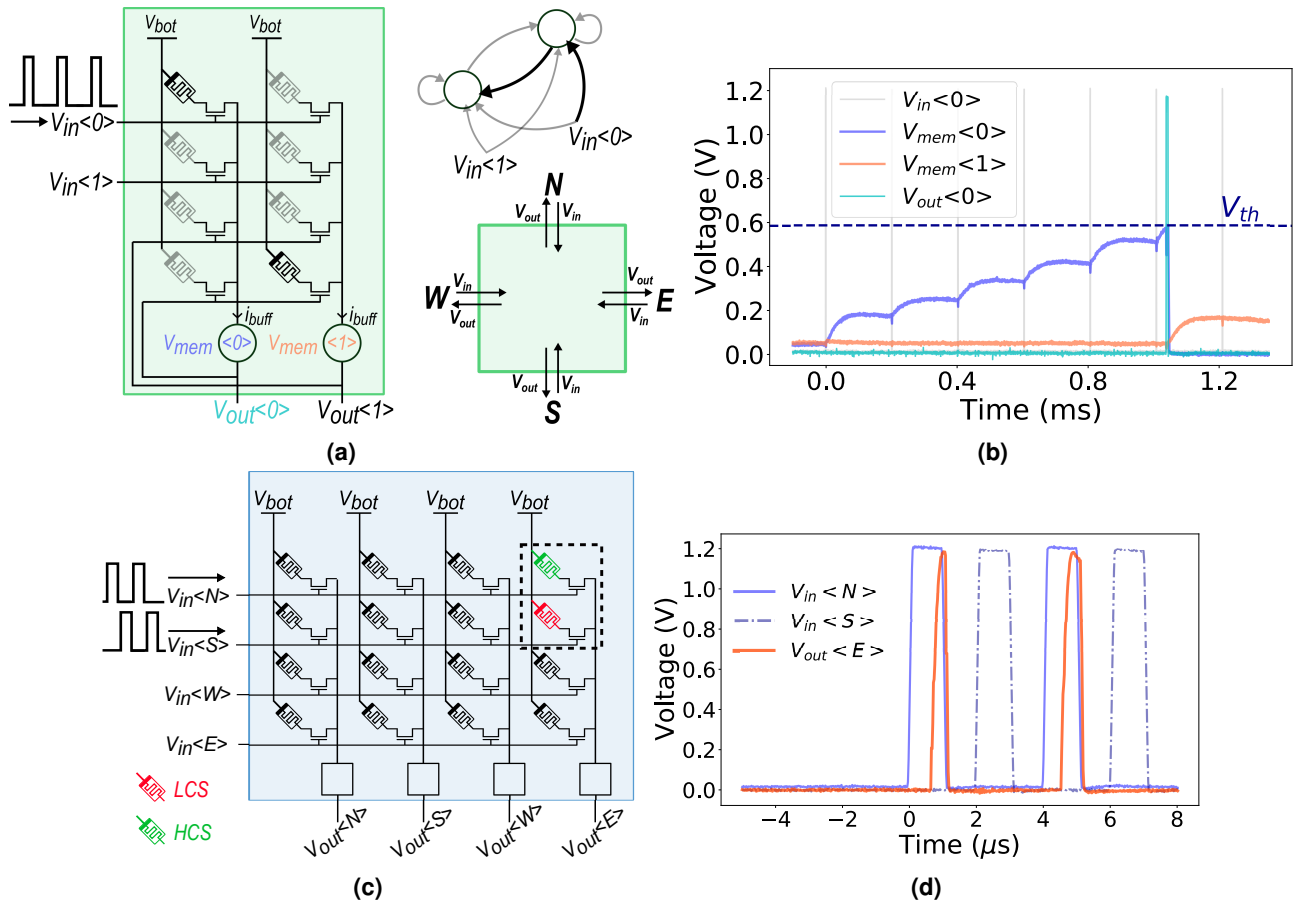


Figure 4. Experimental measurements of the fabricated tile circuits. (a) A depiction of a neuron tile containing two neuron columns. (left) Two stacked neuron columns realize a neuron tile circuit. Four of the devices (top of the array) define the synaptic connections from inputs $V_{in} < 0 >$ and $V_{in} < 1 >$ to the two neurons and an additional four devices (bottom of the array) define the recurrent connections between neurons. Devices are colored in black or gray to indicate respectively whether they were in HCS or LCS during our experimental results plotted in part b. Colored voltage labels and input voltage pulses are also in reference to plots in part b. (top right) A simple two-neuron network resulting from the tile circuit on the left. Circles are representing neurons and directed arrows indicate the synaptic connections between them. Arrows in bold correspond to the devices in the HCS in the tile shown on the left. (bottom right) The input and output voltage pulses can come from or be propagated to the neighboring tiles to the North (N), South (S), East (E) and West (W). (b) Voltage traces measured from a fabricated implementation of the neuron tile circuit in part a. Due to an input pulse train (gray pulses) at $V_{in} < 0 >$ the membrane of the zeroth neuron column in the tile integrates an increasing amount of voltage (purple trace) until, after six pulses, the neuron fires (light blue trace). As a result of the feedback connection to the other neuron column, neuron 1 also exhibits an increase in its membrane voltage. (c) Circuit schematic of a routing tile. Two devices colored green and red denote respectively the devices programmed in the HCS and LCS in the experiment of part (b). Rectangular pulse waveforms depicted on the left-hand side indicate where the input voltage pulses were applied during this experiment. (d) Experimental results from a fabricated version of the routing tile shown in part (c). Continuous and dashed blue traces show the waveforms applied to the N and S inputs while the orange trace shows the response of the output towards the E port. The E output port follows the N input resulting from the device programmed into the HCS in part (a).

136 floating-point precision weights to the low-precision conductance states of memristors in an experimental crossbar using a
 137 closed-loop iterative programming algorithm²⁴ (see Methods).

138 An example of the resulting spike trains produced in the Mosaic, due to an ECG time-series of the arrhythmic heartbeat
 139 plotted in Fig. 5a, is shown in Fig. 5b. The activity of the neurons in each predictive sub-population are bounded within red and
 140 black horizontal dashed lines. The neurons in the red sub-population fire more frequently than those in the black sub-population,
 141 here correctly identifying the heartbeat as arrhythmic.

142 The accuracy over the test set for 100 iterations of training, transfer and test is plotted in Fig. 5c using a boxplot. The median
 143 detection accuracy was 96.9%. This corresponds to a low drop in average accuracy compared to the original high-precision

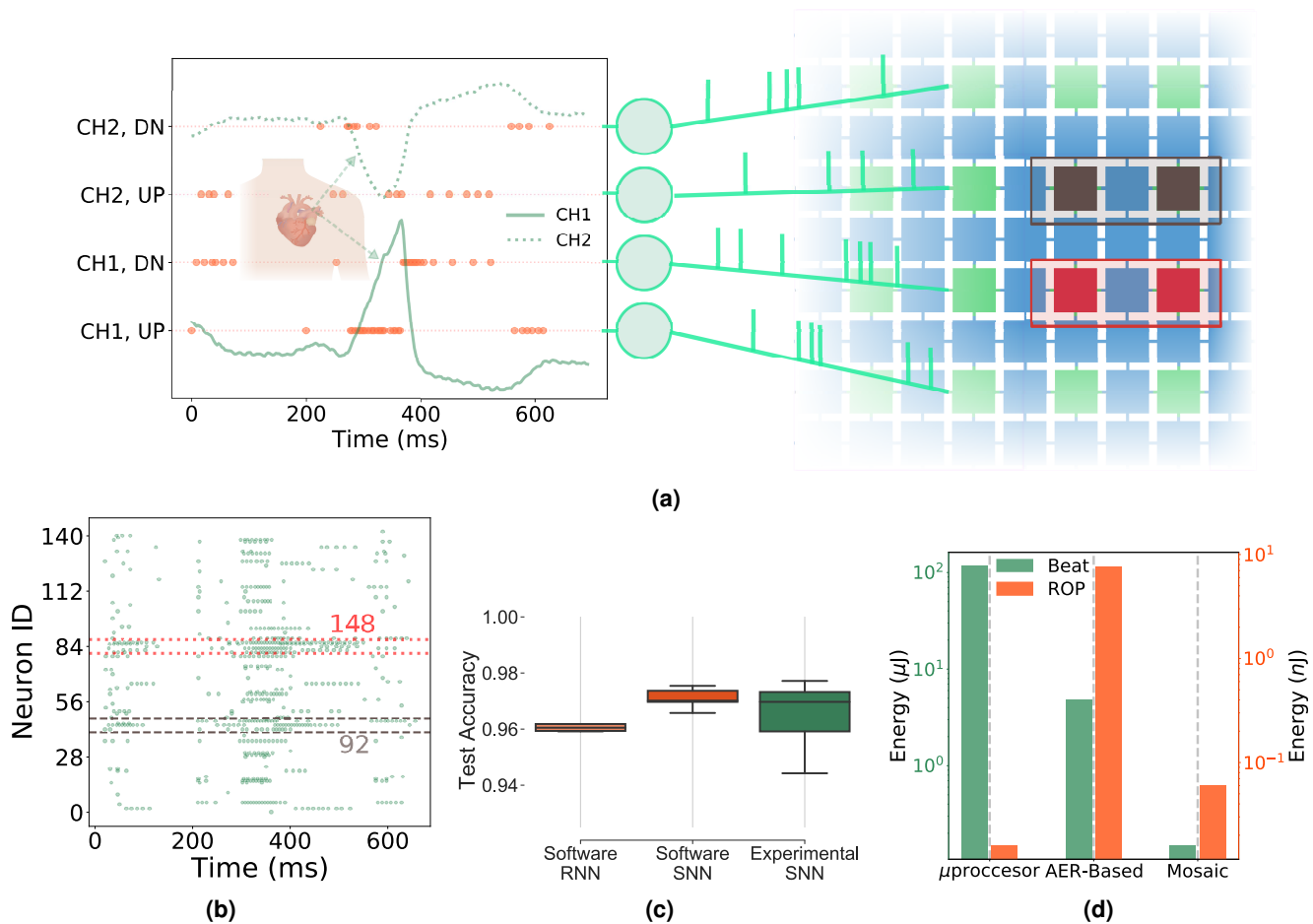


Figure 5. Results in applying the Mosaic to arrhythmia detection of two-channel ECG recordings. (a) A depiction of the ECG classification use case, addressed with the Mosaic architecture. Two-channel ECG waveforms are delta-modulated into four channels - describing upward and downward changes in their magnitude. These four channels correspond to four input neurons (green circles) that propagate events to all of the routing tiles across the Mosaic systolic array. Two groups of two neuron tiles (colored in black and red) are designated as the output neuron populations. Their total spikes counts are used as a means of classifying presented input waveforms. (b) An example raster plot, showing the activity of all of the neurons within the Mosaic, due to presentation of one input time series. Green points indicate the spike times of each neuron. Red and black dashed horizontal lines respectively indicate the anomalous and normal population activity used as the output neurons. (c) A comparison of the accuracy in this task. Boxplots show the accuracy distribution of a software based recurrent neural network (left, red), a software based spiking neural network (centre, orange) and the experimental Mosaic model with multi-level resistive memory devices acting as the synapses (right, green). (d) A comparison in terms of energy requirements between the Mosaic and two alternative event routing approaches. Green bar plots show the average total energy required to routing all spikes during presentation of single heartbeat. Orange bars show the energy required for a single routing operation (ROP).

software model (97.0%). Thus it illustrates that not only the imposed small-world structure of the Mosaic does not have a negative effect on the accuracy, but that the model is also robust to a severe degradation in the precision of the weights. Although, due to the variability in the transfer process, the gap between software and experimental models can sometimes as high as 1%. For further comparison, a non-spiking artificial Recurrent Neural Networks (RNN) was also applied to the same task, obtaining a median accuracy of 96.1%. This lower software accuracy compared with the experimental Mosaic's model, further confirms the Mosaic's computational power. This result is consistent with other observations whereby RSNNs have outperformed non-spiking equivalents⁴⁶.

Using estimates obtained from SPICE simulations of our fabricated test circuits and statistics from the Mosaic experiments (see Methods), the average energy per routing operation is estimated to be 60 pJ. Given the average number of spikes per heartbeat and the average number of routing tiles traversed between source and destination, the total energy required to process, and make a prediction regarding one heartbeat using the Mosaic is 150 nJ.

To gain a perspective on the energy efficiency of Mosaic relative to other hardware approaches, we compare this figure

156 to the energy required for running the same neural network model on a conventional microprocessor, and on an AER-based
157 Complementary Metal-Oxide-Semiconductor (CMOS) neuromorphic processor³⁰. While the energy required for a single
158 routing operation on a microprocessor, assumed to be equivalent to one Static Random Access Memory (SRAM) access, is
159 only 8 pJ, the total energy required per heartbeat is much greater - 116 μ J. This difference is in large part due to asynchronous
160 nature of memory access in the Mosaic approach relative to a microprocessor - where all variables are required to be updated
161 on each timestep of a numerical simulation. In an AER-based neuromorphic processor, 7.7 nJ is required to generate and route
162 an event between a source and a destination³⁰ (see Methods). Over the course of a full heartbeat, the total required energy is
163 therefore be 4.8 μ J.

164 Based on these estimations, the Mosaic achieves a reduction of two and one orders of magnitude in total routing energy
165 per heartbeat, relative to a microprocessor and an AER-based neuromorphic processor, respectively. This can be attributed
166 to the combination of Mosaic's low-energy routing memory access, along with its ability to access memory elements in an
167 asynchronous and event-based fashion (Supplementary Note 1).

168 Conclusion

169 We have proposed the Mosaic, a novel neuromorphic computing architecture based on a systolic array of small memristor
170 crossbars. The Mosaic is particularly well suited for the implementation of small-world graphical models, commonly found in
171 biological nervous systems. Crucially, the Mosaic uses distributed non-volatile resistive memory devices in an analog fashion,
172 not only for computation, but also to route spikes.

173 We showed mathematically that, particularly as network size increases, the Mosaic offers a means of implementing small-
174 world graphical models with less memory, and therefore energy, than previous approaches based on single large memristor
175 crossbars.

176 The two fundamental circuit blocks of the Mosaic, the neuron tile and the routing tile, were designed, fabricated and
177 experimentally demonstrated using a hybrid technology co-integrating 130 nm CMOS technology with resistive memory
178 devices. Based on the measurements of these circuits, a mixed hardware-software simulation of the Mosaic was developed and
179 the task of detecting arrhythmic heartbeats from ECG signals was addressed. The Mosaic was able to achieve an accuracy close
180 to that of an equivalent high-precision software model.

181 Most importantly, relative to microprocessor-based and AER-based implementations of the same model, the Mosaic was
182 observed to permit respective two and one order of magnitude reductions in the total energy to process each heartbeat and
183 provide a prediction. Note that our evaluation is based on our design in 130 nm technology which imposes a minimum spike
184 pulse width which dominates the Mosaic's energy requirements (see Methods). It is expected that moving to more advanced
185 technologies would permit to substantially reduce the spike pulse-width and that the Mosaic would achieve even greater energy
186 reductions relative to these two approaches.

187 Graph-based computing is currently receiving attention as a promising means of leveraging the capabilities of SNNs⁴⁷. The
188 Mosaic is thus a timely dedicated hardware architecture optimized for a specific type of graph that is abundant in nature.

189 Methods

190 Design, fabrication of mosaic circuits

191 Neuron and routing column circuits

192 Both neuron and routing column share a common circuit in Fig. 3a which reads the conductances of the RRAM devices.
193 The RRAM bottom electrode has a constant DC voltage V_{bot} applied to it and the common top electrode is pinned to the
194 voltage V_x by a rail-to-rail operational amplifier (OPAMP) circuit. The OPAMP output is connected in negative feedback to
195 its non-inverting input (due to the 90 degrees phase-shift between the gate and drain of transistor M_1 in Fig. 3a) and has the
196 constant DC bias voltage V_{top} applied to its inverting input. As a result, the output of the OPAMP will modulate the gate voltage
197 of transistor M_1 such that the current it sources onto the node V_x will maintain its voltage as close as possible to the DC bias
198 V_{top} . Whenever an input pulse $V_{in} < n >$ arrives, a current i_{in} equal to $(V_x - V_{bot})G_n$ will flow out of the bottom electrode. The
199 negative feedback of the OPAMP will then act to ensure that $V_x = V_{top}$, by sourcing an equal current from transistor M_1 . By
200 connecting the OPAMP output to the gate of transistor M_2 , a current equal to i_{in} , will therefore also be buffered, as i_{buff} , into
201 the branch composed of transistors M_2 and M_3 in series. In the routing tile, this current is compared against a reference current,
202 and if higher, a pulse is generated and transferred onwards. The current comparator circuit is composed of two current mirrors
203 and an inverter (see Supplementary Fig. S6). In the neuron column, this current is injected into a CMOS differential-pair
204 integrator synapse circuit model³² which generates an exponentially decaying waveform from the onset of the pulse with an
205 amplitude proportional to the injected current. Finally, this exponential current is injected onto the membrane capacitor of a
206 CMOS leaky-integrate and fire neuron circuit model⁴⁸ where it integrates as a voltage (see Supplementary Fig. S7). Upon
207 exceeding a voltage threshold (the switching voltage of an inverter) a pulse is emitted at the output of the circuit. This pulse in

turn feeds back and shunts the capacitor to ground such that it is discharged. Further circuits were required in order to program the device conductance states. Notably, multiplexers were integrated on each end of the column in order to be able to apply voltages to the top and bottom electrodes the RRAM devices.

Fabrication/integration

The circuits described in Section have been taped-out in 130 nm technology at CEA-Leti, in a 200 mm production line. The Front End of the Line, below metal layer 4, has been realized by ST-Microelectronics, while from the fifth metal layer upwards, including the deposition of the composites for RRAM devices, the process has been completed by CEA-Leti. RRAM devices are composed of a 5 nm thick HfO_2 layer sandwiched by two 5 nm thick TiN electrodes, forming an $TiN/HfO_2/Ti/TiN$ stack. Each device is accessed by a transistor giving rise to the 1T1R unit cell. The size of the access transistor is 650 nm wide. 1T1R cells are integrated with CMOS-based circuits by stacking the RRAM cells on the higher metal layers. In the cases of the neuron and routing tiles, 1T1R cells are organized in a small - either 2x2 or 2x4 - matrix in which the bottom electrodes are shared between devices in the same column and the gates shared with devices in the same row. In this way, the devices can be accessed in a parallel manner. The circuits integrated into the wafer, were accessed by a probe card which connected to the pads of the dimension of $[50 \times 90] \mu m^2$.

Mosaic circuit measurement setups

The tests involved analyzing and recording the dynamical behavior of analog CMOS circuits as well as programming and reading RRAM devices. Both phases required dedicated instrumentation, all simultaneously connected to the probe card. For programming and reading the RRAM devices, Source Measure Units (SMU)s from a Keithley 4200 SCS machine were used. To maximize stability and precision of the programming operation, SET and RESET are performed in a quasi-static manner. This means that a slow rising and falling voltage input is applied to either the Top (SET) or Bottom (RESET) electrode, while the gate is kept at a fixed value. To the $V_{top}(t)$, $V_{bot}(t)$ voltages, we applied a triangular pulse with rising and falling times of 1 sec and peak V_{gate} . For a SET operation, the bottom of the 1T1R structure is conventionally left at ground level, while in the RESET case the V_{top} is equal to 0 V and a positive voltage is applied to V_{bot} . Typical values for the SET operation are V_{gate} in $[0.9 - 1.3] V$, while the V_{top} peak voltage is normally at 2.0 V. Such values allow to modulate the RRAM resistance in an interval of $[5 - 30] k\Omega$ corresponding to the Low-Resistive State (LRS) of the device. For the RESET operation, the gate voltage is instead in the $[2.75, 3.25] V$ range, while the bottom electrode is reaching a peak at 3.0 V. The High-Resistive State (HRS) is less controllable than the LRS due to the inherent stochasticity related to the rupture of the conductive filament, thus the HRS level is spread out in a wider $[80 - 1000] k\Omega$ interval. The reading operation is performed by limiting the V_{top} voltage to 0.3 V, a value that avoids read disturbances, while opening the gate voltage at 4.5 V.

Inputs and outputs are analog dynamical signals. In the case of the input, we have alternated two HP 8110 pulse generators with a Tektronix AFG3011 waveform generator. As a general rule, input pulses had a pulse width of $1 \mu s$ and rise/fall time of 50 ns. This type of pulse is assumed as the stereotypical spiking event of a Spiking Neural Network. Concerning the outputs, a 1 GHz Teledyne LeCroy oscilloscope was utilized to record the output signals.

Mosaic RSNN hardware-software simulation

The definition, training and test of the neural network was performed in a series of steps. First a recursive computer model of the Mosaic was used to generate a skeleton connectivity matrix that describes, for given Mosaic dimensions and device states, what neurons are connected to one another in a corresponding neural network. This model simulates the propagation of events, generated at the output of neuron tiles, through the mesh of routing tiles. The model propagates spikes from all neurons through all possible paths defined by the binary conductance states of the devices of the routing tiles. For each of the neuron columns that received this spike at their inputs, a flag was set in the appropriate index of a connectivity matrix describing the connectivity between all neurons.

The Mosaic model used in this Article was composed of routing tiles of 16×16 devices in a Mosaic of 11×11 tiles. Devices in the routing tiles were programmed to be in the HCS with a probability of 0.07. Neuron tiles were realized in a 20×4 array - this allows four signals from each of the four neighbouring tiles to be received independently, as well as four neurons to connect recurrently amongst themselves within a tile.

The resulting skeleton connectivity matrix was then exported to a PyTorch model of an RSNN model to be trained on the MIT-BIH heart arrhythmia dataset⁴¹. Specifically, all of the heartbeats of one patient (labelled as 201 in the dataset) were delta modulated into four spike train channels. These spike trains then served as an effective spiking input layer of the model.

Data points were presented to the model in mini-batches of sixteen. Two populations of neurons in two neuron tiles were used to denote whether the presented ECG signals corresponded to a healthy or an arrhythmic heartbeat. The softmax of the total number of spikes generated by the neurons in each population was used to obtain a classification probability. The negative log-likelihood was then minimized using the categorical cross-entropy with the labels of the signals. The derivative of the

260 Heaviside step function, that is used to rectify the membrane voltage of the LIF neurons into a zero or a one, was approximated
 261 using the function $1/abs(V_{mem} - V_{th})^2$ - inline with surrogate gradient training methods⁴⁵.

262 After training, the synaptic weights were transferred into a an array of 16 kb resistive memory devices co-integrated onto a
 263 130 nm CMOS technology. The synaptic weight of each synapse was defined by the subtraction of two conductance states of
 264 two devices. The process of transferring the high-precision software weights to the conductance states of the devices in the array
 265 was achieved using an iterative closed-loop multilevel programming algorithm. It is based on adapting the SET programming
 266 compliance current to obtain a conductance within a target range²⁴ and programming a device until its conductance falls within
 267 a pre-defined margin of tolerated error. Such an approach allows each device to be programmed with ten non-overlapping
 268 conductance levels.

269 The delta modulated test data was processed by a mixed hardware-software Moasic model. Whenever a pre-synaptic neuron
 270 emitted a spike in this model, the corresponding pair of devices (storing the synaptic weight connecting it to the post-synaptic
 271 neuron) are read in the RRAM array. This read value is then used to update the state of the post-synaptic neuron inline with
 272 the LIF neuron model implemented by the circuits. The data was classified based on which population of output neuron tiles
 273 produced the largest total count of spikes during the presentation of an input ECG time-series.

274 Preparation of the ECG dataset

275 The ECG dataset was downloaded from the MIT-BIH arrhythmia repository⁴¹. The database is composed of continuous
 276 30-minute recordings measured from multiple subjects. The QRS complex of each heartbeat has been annotated as either
 277 healthy or exhibiting one of many possible heart arrhythmias by a team of cardiologists. We selected one patient exhibiting
 278 approximately half healthy and half arrhythmic heartbeats. Each heartbeat was isolated from the others in a 700 ms time-series
 279 centered on the labelled QRS complex. Each of the two 700 ms channel signals were then converted to spikes using a delta
 280 modulation scheme⁴⁹. This consists of recording the initial value of the time-series and, going forward in time, recording
 281 the time-stamp when this signal changes by a pre-determined positive or negative amount. The value of the signal at this
 282 time-stamp is then recorded and used in the next comparison forward in time. This process is then repeated. For each of the
 283 two channels this results in four respective event streams - denoting upwards and downwards changes in the signals. During
 284 the simulation of the neural network, these four event streams corresponded to the four input neurons to the spiking recurrent
 285 neural network implemented by the mosaic.

286 Calculation of memory footprint

287 We calculate the Mosaic architecture's Memory Footprint (MF) in comparison to a large crossbar array, in building small world
 288 graphical models.

289 To evaluate the MF for one large crossbar array, the total number of devices required to implement any possible connections
 290 between neurons can be counted - allowing for any Spiking Recurrent Neural Networks (SRNN) to be mapped onto the system.
 291 Setting N to be the number of neurons in the system, the total possible number of connections in the graph is $MF_{ref} = N^2$.

292 For the Mosaic architecture, the number of RRAM cells (i.e., the MF) is equal to the number of devices in all the neuron
 293 tiles and routing tiles: $MF_{mosaic} = MF_{NeuronTiles} + MF_{RoutingTiles}$.

294 Considering each neuron tile with k neurons, each neuron tile contributes to $4 \times k^2$ devices (where the factor of 4 accounts
 295 for the four possible directions to which each tile can connect). Evenly dividing the N total number of neurons in each neuron
 296 tile gives rise to $T = ceil(N/k)$ required neuron tiles. This brings the total number of devices attributed to the neuron tile to
 297 $T \times 4 \times k^2$.

298 The number of routing tiles which connects all the neuron tiles depends on the geometry of the Mosaic systolic array. Here,
 299 we assume neuron tiles assembled in a square, each with a routing tile on each side. We consider R to be the number of routing
 300 tiles with $4k^2$ devices in each. This brings the total number of devices related to routing tiles up to $MF_{RoutingTiles} = R \times (4k)^2$.

The problem can then be re-written as a function of the geometry. Considering Fig.2a, let i be an integer and $(2i + 1)^2$
 the total number of tiles. The number of neuron tiles can be written as $T = (i + 1)^2$, as we consider the case where neuron
 tiles form the outer ring of tiles. As a consequence, the number of routing tiles is $R = (2i + 1)^2 - (i + 1)^2$. Substituting
 such values in the previous evaluations of $MF_{NeuronTiles} + MF_{RoutingTiles}$ and remembering that $k < N \times T$, we can impose that
 $MF_{Mosaic} = MF_{NeuronTiles} + MF_{RoutingTiles} < MF_{MF_{ref}}$. This results in the following expression:

$$MF_{Mosaic} = MF_{NeuronTiles} + MF_{RoutingTiles} < MF_{reference} \quad (1)$$

$$(i + 1)^2 4 \times k^2 + [(2i + 1)^2 - (i + 1)^2]((4k)^2) < (k(i + 1)^2)^2 \quad (2)$$

301 This expression can then be evaluated for i , given a network size, giving rise to the relationships as plotted in Fig.2c in the
 302 main text.

Calculation of routing energy

In state-of-the-art event-based neuromorphic chips, the information is communicated through the AER scheme²⁹. Whenever a spiking neuron in a chip (or module) generates a spike, its “address” (or any given ID) is written on a high speed digital bus and sent to the receiving neuron(s) in one (or more) receiver module(s). In our Mosaic structure, we have distributed the routing information in a two-dimensional matrix along with the computing units.

To compare the routing energy and latency of Mosaic with the AER systems, we have calculated the energy per spike routing in the best and worst case scenarios in both systems.

For AER-based systems, we are using the energy and latency numbers reported in Dynap-SE, as one of the most recent and optimized AER routing schemes³⁰. It is a multi-core neuromorphic comprising four cores; each core includes 256 neurons. It has a hierarchical asynchronous routing, combining a source-based routing mesh architecture with a destination-based hierarchical tree routing method. SRAM cells store the routing structure in the tree and the Content Addressable Memory (CAM) cells store the tag of the source address to which each neuron is connected.

Therefore, once a spike is generated, the least energy consumption happens in a scenario where the events should be routed locally, and thus 256 10-bit CAM cells are accessed. In the worst case, events have to travel from the first-level router to the higher levels and thus the energy of reading SRAM cells are added. Therefore, the energy of routing one spike in Dynap-SE can be calculated by the following equation:

$$E_{total} = E_{Spike} + E_{Pulse} + E_{En} + E_{BC} + RT \cdot E_{RT} \quad (3)$$

Where E_{Spike} is the energy to generate one spike, E_{Pulse} is the energy of the pulse extender circuit, E_{en} is the energy to encode one spike and append destination, E_{BC} is the energy to broadcast the event to the same core, RT is 1 if the spike has to be routed to other cores, otherwise zero, and E_{RT} is the energy to route the events to other cores. If $RT = 0$, total energy to route the event to the core sums up to 7.68 nJ. In case of the event routing to other cores, multiples of 360 pJ should be added to the energy consumption (energy required for reading SRAM at each hierarchical router level).

For the case of the microprocessor, the equivalent of routing an event would be to load into the arithmetic logic unit memory from an SRAM containing the synaptic weights and perform addition and multiplication operations to update neuron states and outputs before writing this back into SRAM. We assume that this is dominated by the SRAM access, and so take figures from the literature that give SRAM access energy figures⁵⁰.

Acknowledgements

We acknowledge funding support from the H2020 MeM-Scales project (871371) as well as the French ANR via Carnot funding.

Author contributions

T.D., G.I., E.V and M.P developed the mosaic concept. T.D. and M.P. designed and laid out the circuits for fabrication. F.M. and A.P. performed the measurements on the fabricated circuits. T.D., F.M and M.P. developed the Mosaic simulation and applied it to the arrhythmia detection task. All authors contributed to writing the paper.

References

1. Hamilton, W. L., Ying, R. & Leskovec, J. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).
2. Watts, D. J. & Strogatz, S. H. Collective dynamics of ‘small-world’ networks. *nature* **393**, 440–442 (1998).
3. Bullmore, E. & Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat. reviews neuroscience* **10**, 186–198 (2009).
4. Loeffler, A. *et al.* Topological properties of neuromorphic nanowire networks. *Front. Neurosci.* **14**, 184 (2020).
5. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
6. Ambrogio, S. *et al.* Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* **558**, 60–67, DOI: [10.1038/s41586-018-0180-5](https://doi.org/10.1038/s41586-018-0180-5) (2018).
7. Li, C. *et al.* Long short-term memory networks in memristor crossbar arrays. *Nat. Mach. Intell.* **1**, 49–57 (2019).
8. Wang, Z. *et al.* In-situ training of feed-forward and recurrent convolutional memristor networks. *Nat. Mach. Intell.* **1**, 434–442 (2019).

- 348 9. Woźniak, S., Pantazi, A., Bohnstingl, T. & Eleftheriou, E. Deep learning incorporating biologically inspired neural
349 dynamics and in-memory computing. *Nat. Mach. Intell.* **2**, 325–336 (2020).
- 350 10. Dalgaty, T. *et al.* In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling. *Nat.*
351 *Electron.* **4**, 151–161 (2021).
- 352 11. Dalgaty, T., Esmanhotto, E., Castellani, N., Querlioz, D. & Vianello, E. Ex-situ transfer of Bayesian neural networks to
353 resistive memory-based inference hardware. *Adv. Intell. Syst.* 2000103 (2021).
- 354 12. Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R. & Eleftheriou, E. Memory devices and applications for in-memory
355 computing. *Nat. nanotechnology* **15**, 529–544 (2020).
- 356 13. Chicca, E. & Indiveri, G. A recipe for creating ideal hybrid memristive-CMOS neuromorphic processing systems. *Appl.*
357 *Phys. Lett.* **116**, 120501, DOI: [10.1063/1.5142089](https://doi.org/10.1063/1.5142089) (2020).
- 358 14. Jouppi, N. P. *et al.* In-datacenter performance analysis of a Tensor Processing Unit. In *Proceedings of the 44th annual*
359 *international symposium on computer architecture*, 1–12 (2017).
- 360 15. Yu, S., Sun, X., Peng, X. & Huang, S. Compute-in-memory with emerging nonvolatile-memories: challenges and prospects.
361 In *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 1–4 (IEEE, 2020).
- 362 16. Joksas, D. *et al.* Committee machines—a universal method to deal with non-idealities in memristor-based neural networks.
363 *Nat. communications* **11**, 1–10 (2020).
- 364 17. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. electronics* **1**, 22–29
365 (2018).
- 366 18. Jo, S. H. *et al.* Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters* **10**, 1297–1301 (2010).
- 367 19. Ielmini, D. & Waser, R. *Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device*
368 *Applications* (John Wiley & Sons, 2015).
- 369 20. Serb, A. *et al.* Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses.
370 *Nat. Commun.* **7**, 12611 (2016).
- 371 21. Li, C. *et al.* Efficient and self-adaptive in-situ learning in multilayer memristor neural network. *Nat. Commun.* **9**, 1–8, DOI:
372 [10.1038/s41467-018-04484-2](https://doi.org/10.1038/s41467-018-04484-2) (2018).
- 373 22. Strukov, D., Indiveri, G., Grollier, J. & Fusi, S. Building brain-inspired computing. *Nat. Commun.* **10**, DOI: [10.1038/](https://doi.org/10.1038/s41467-019-12521-x)
374 [s41467-019-12521-x](https://doi.org/10.1038/s41467-019-12521-x) (2019).
- 375 23. Kingra, S. K. *et al.* SLIM: Simultaneous Logic-In-Memory computing exploiting bilayer analog OxRAM devices. *Sci.*
376 *reports* **10**, 1–14 (2020).
- 377 24. Esmanhotto, E. *et al.* High-density 3D monolithically integrated multiple 1T1R multi-level-cell for neural networks. In
378 *2020 IEEE International Electron Devices Meeting (IEDM)*, 36–5 (IEEE, 2020).
- 379 25. Dalgaty, T. *et al.* Hybrid neuromorphic circuits exploiting non-conventional properties of RRAM for massively parallel
380 local plasticity mechanisms. *APL Mater.* **7**, 081125 (2019).
- 381 26. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*
382 **521**, 61–64, DOI: [10.1038/nature14441](https://doi.org/10.1038/nature14441) (2015).
- 383 27. Yao, P. *et al.* Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–646 (2020).
- 384 28. Fornito, A., Zalesky, A. & Bullmore, E. T. Chapter 3 - connectivity matrices and brain graphs. In *Fundamentals of Brain*
385 *Network Analysis*, 89–113, DOI: <https://doi.org/10.1016/B978-0-12-407908-3.00003-0> (Academic Press, San Diego,
386 2016).
- 387 29. Boahen, K., Nomura, M., Vidal, E. R. & Rullen, R. V. Address-event senders and receivers: Implementing direction-
388 selectivity and orientation-tuning (1998).
- 389 30. Moradi, S., Qiao, N., Stefanini, F. & Indiveri, G. A scalable multicore architecture with heterogeneous memory structures
390 for dynamic neuromorphic asynchronous processors (DYNAPs). *Biomed. Circuits Syst. IEEE Transactions on* **12**, 106–122,
391 DOI: [10.1109/TBCAS.2017.2759700](https://doi.org/10.1109/TBCAS.2017.2759700) (2018).
- 392 31. Indiveri, G. *et al.* Neuromorphic silicon neuron circuits. *Front. Neurosci.* **5**, 1–23, DOI: [10.3389/fnins.2011.00073](https://doi.org/10.3389/fnins.2011.00073) (2011).
- 393 32. Chicca, E., Stefanini, F., Bartolozzi, C. & Indiveri, G. Neuromorphic electronic circuits for building autonomous cognitive
394 systems. *Proc. IEEE* **102**, 1367–1388, DOI: <https://doi.org/10.1109/JPROC.2014.2313954> (2014).

33. Grossi, A. *et al.* Fundamental variability limits of filament-based rram. In *2016 IEEE International Electron Devices Meeting (IEDM)*, 4.7.1–4.7.4, DOI: [10.1109/IEDM.2016.7838348](https://doi.org/10.1109/IEDM.2016.7838348) (2016). 395
396
34. Kung, H. T. & Leiserson, C. E. Systolic arrays for VLSI. Tech. Rep., Carnegie-Mellon Univ. Pittsburg PA (1978). 397
35. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002). 398
399
36. Lee, J. H., Delbruck, T. & Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Front. neuroscience* **10**, 508 (2016). 400
401
37. Zenke, F. & Vogels, T. P. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Comput.* **33**, 899–925 (2021). 402
403
38. Bellec, G. *et al.* A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* **11**, 1–15, DOI: [10.1038/s41467-020-17236-y](https://doi.org/10.1038/s41467-020-17236-y) (2020). 404
405
39. Bauer, F., Muir, D. & Indiveri, G. Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor. *Biomed. Circuits Syst. IEEE Transactions on* **13**, 1575–1582, DOI: [10.1109/TBCAS.2019.2953001](https://doi.org/10.1109/TBCAS.2019.2953001) (2019). 406
407
40. Yin, B., Corradi, F. & Bohté, S. M. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, 1–8 (2020). 408
409
41. Moody, G. B. & Mark, R. G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Medicine Biol. Mag.* **20**, 45–50 (2001). 410
411
42. Lee, H.-Y., Hsu, C.-M., Huang, S.-C., Shih, Y.-W. & Luo, C.-H. Designing low power of sigma delta modulator for biomedical application. *Biomed. Eng. Appl. Basis Commun.* **17**, 181–185 (2005). 412
413
43. Corradi, F. & Indiveri, G. A neuromorphic event-based neural recording system for smart brain-machine-interfaces. *IEEE transactions on biomedical circuits systems* **9**, 699–709 (2015). 414
415
44. Werbos, P. J. Backpropagation through time: What it does and how to do it. *Proc. IEEE* **78**, 1550–1560 (1990). 416
45. Neftci, E. O., Mostafa, H. & Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* **36**, 51–63 (2019). 417
418
46. Yin, B., Corradi, F. & Bohté, S. M. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, 1–8 (2020). 419
420
47. Davies, M. *et al.* Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proc. IEEE* **109**, 911–934 (2021). 421
422
48. Dalgaty, T., Payvand, M., De Salvo, B. *et al.* Hybrid CMOS-RRAM neurons with intrinsic plasticity. In *IEEE ISCAS*, 1–5 (IEEE, 2019). 423
424
49. Corradi, F., Bontrager, D. & Indiveri, G. Toward neuromorphic intelligent brain-machine interfaces: An event-based neural recording and processing system. In *Biomedical Circuits and Systems Conference (BioCAS)*, 584–587, DOI: [10.1109/BioCAS.2014.6981793](https://doi.org/10.1109/BioCAS.2014.6981793) (IEEE, 2014). 425
426
427
50. Pedram, A., Richardson, S., Horowitz, M., Galal, S. & Kvatinsky, S. Dark memory and accelerator-rich system optimization in the dark silicon era. *IEEE Des. & Test* **34**, 39–50 (2016). 428
429

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [suppinfo.pdf](#)