

Memristor-based fully analog reservoir computing system for power-efficient real-time spatiotemporal signal processing

Yanan Zhong

Institute of Functional Nano & Soft Materials, Soochow University

Jianshi Tang

Tsinghua University <https://orcid.org/0000-0001-8369-0067>

Xinyi Li

Institute of Microelectronics, Tsinghua University

Xiangpeng Liang

University of Glasgow <https://orcid.org/0000-0001-5243-0457>

Zhengwu Liu

School of Integrated Circuits, Tsinghua University <https://orcid.org/0000-0001-7968-9469>

Yijun Li

School of Integrated Circuits, Tsinghua University

Peng Yao

Tsinghua University

Zhenqi Hao

School of Integrated Circuits, Tsinghua University

Bin Gao

Tsinghua University <https://orcid.org/0000-0002-2417-983X>

He Qian

Tsinghua University

Huaqiang Wu (✉ wuhq@tsinghua.edu.cn)

Tsinghua University <https://orcid.org/0000-0001-8359-7997>

Article

Keywords: dynamic memristor, reservoir computing, analog computing, arrhythm detection, dynamic gesture recognition

Posted Date: May 10th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1592701/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Nature Electronics on September 26th, 2022. See the published version at <https://doi.org/10.1038/s41928-022-00838-3>.

Abstract

Implementing power-efficient reservoir computing hardware systems is of great interest to the field of neuromorphic computing. More and more studies attempt to use analog devices or components, such as memristors and spintronic oscillators, to partially replace fully digital systems to boost the power efficiency. However, a reservoir computing system operating real-time in fully analog fashion has not been demonstrated yet. In this work, a fully analog reservoir computing system was implemented using two types of memristors, where dynamic memristors were used to construct the reservoir while non-volatile memristor arrays were used as the readout layer. The key features, such as threshold and window, extracted from the dynamic memristor-based physical nodes were found to have a significant impact on the system performance. By adjusting the features to the appropriate range, our system can efficiently process spatiotemporal signals in real time with extremely low power consumption, more than three orders of magnitudes lower than digital counterparts. Both temporal task of arrhythm detection and spatiotemporal task of dynamic gesture recognition were demonstrated, where high detection accuracy of 96.6% and recognition accuracy 97.9% were achieved respectively. Our work demonstrates that such memristor-based fully analog reservoir computing system could be attractive for spatial and temporal edge computing with extremely low power and hardware cost.

Introduction

With the rapid development of artificial intelligence, artificial neural networks (ANNs) are getting increasingly larger and deeper in scale, which imposes critical challenges to the computing power and efficiency of mainstream digital computers based on the von Neumann architecture^{1,2}. To break the von Neumann bottleneck, brain-inspired neuromorphic computing based on emerging devices, such as memristors^{3,4}, has attracted increasing attention in recent years⁵⁻⁷. For example, the computing-in-memory architecture with emerging resistive switching devices, including nonvolatile memristors (NVM), has exhibited orders of magnitudes higher energy efficiency than traditional computing platforms like central processing units (CPU) and graphics processing units (GPU)². This is mainly because parallel multiply-accumulate (MAC) computations can be implemented in a purely analog fashion via fundamental physics laws on a memristor array by making use of its analog resistive switching characteristics.

So far, most of the NVM-based ANNs implemented have been feed-forward networks, such as multilayer perceptron (MLP) and convolutional neural network (CNN)⁸⁻¹¹, which are difficult to handle temporal tasks. The other type of ANNs with recursive connections is usually called recurrent neural network (RNN), which is designed for dealing with temporal inputs¹²⁻¹⁴. However, the training of RNN is challenging as it usually faces the problem of exploding or vanishing gradients. Also, it is difficult to be implemented with NVM arrays especially in a fully analog fashion, because the inherent variation of NVM conductance and the noise in analog circuits could cause error accumulation during the recursive computations, which would eventually result in large output errors. Recently, an ingenious idea that replaces the recursive

network with a dynamic physical system has attracted wide attention¹⁵⁻¹⁷. Because this approach can not only circumvent the error accumulation problem in the recursive network, but also dramatically reduce hardware cost. The most representative example that embodies this idea is the reservoir computing (RC)¹⁸⁻²⁰, which was initially proposed as a special variant of RNN. Unlike traditional RNN, the recursive network in the reservoir is fixed and its weights do not need to be changed during the training process, which enables the reservoir layer to be implemented with a specific physical system. In addition, the training process of RC only involves the weights connecting to the final readout layer, where low-cost training algorithms, such as linear regression, can be used.

In principle, the hardware implementations of RC system can be roughly divided into the following three types. The first one is a fully digital RC system, where both the reservoir layer and the readout layer are implemented by digital components (such as field-programmable gate array and digital signal processor)²¹⁻²³ as shown in Fig. 1a. While such fully digital system can minimize noise interference, additional analog-to-digital converters (ADCs) are usually needed to convert the external analog signals into digital inputs, which would incur extra energy consumption and system latency. Furthermore, a large number of floating-point calculations in such digital system that adopts the von Neumann architecture with separated memory and computing units would also greatly increase the energy consumption and hardware resources. To address these problems, hybrid RC systems which have an analog reservoir layer and a digital readout layer are proposed as shown in Fig. 1b. In such hybrid systems, external analog signals can be directly input and processed in the reservoir which can be implemented with analog elements, such as spintronic oscillators²⁴⁻²⁶, photonic modules²⁷⁻²⁹, nanowire networks³⁰⁻³², or memristors³³⁻³⁵. These emerging devices can be treated as special physical systems³⁶ (or physical nodes) to equivalently realize the function of reservoir, which can reduce both energy consumption and area. However, such hybrid RC system still has shortcomings as it also requires a large number of ADCs to convert analog reservoir states into digital signals for the digital readout layer. In recent works, NVM arrays have been used to implement the analog readout layer^{32,35}. However, such systems reported so far still need to add certain digital units (e.g., ADCs and registers) between the reservoir module and the readout module for data conversion and buffering. A desired fully analog RC system, where both the reservoir and the readout layer are implemented with analog elements and analog signals can be directly transmitted and processed throughout the whole system without any conversion and buffering, is shown in Fig. 1c. Compared with the above two types of RC systems, such fully analog RC system, which can perform real-time spatiotemporal signal processing with extremely low power consumption and hardware cost, has not been demonstrated yet.

In this study, a memristors-based fully analog RC system was successfully demonstrated using two distinct types of memristors, where dynamic memristors (DMs) are used as the physical nodes to construct parallel reservoirs and NVM arrays are used as the readout layer. The relationship between the node features and the system performance was investigated through carefully adjusting hyperparameters. Two key node features, that are threshold and window, were found to have a large impact on the system performance. By engineering the DM-based reservoirs, the implemented RC system

successfully demonstrated real-time spatiotemporal signal processing with more than three orders of magnitudes lower power consumption than digital RC systems. Both arrhythm detection and dynamic gesture recognition tasks were demonstrated, achieving high accuracies of 96.6% and 97.9% respectively.

Results

System Architecture and Device Characteristics. The hardware implementation of a fully integrated DM-based RC (DM-RC) system is shown in Fig. 2a, which consists of power module, microcontroller unit (MCU), reservoir module and readout module. The power module provides stable positive and negative voltage sources, and the MCU generates control signals and handles data transfer. The reservoir module maps the features of low-dimensional temporal input signals to a high-dimensional space. Such high-dimensional features (that are reservoir states) are usually linearly separable, so that they can be classified using a simple fully connected readout layer. The core of the reservoir module is a set of DMs, which are connected to the printed circuit board (PCB) through a probe card. The DM used in this work has a cross-point structure with a material stack of Ti/TiO_x/Pd (110 nm/80 nm/50 nm), and the cross-sectional transmission electron microscope (TEM) image of the device is shown in the inset of the left panel of Fig. 2a. Here, 24 DMs are used in our system to form parallel reservoirs, and the memristive I-V curves along with the optical image of these devices are shown in the left panel of Fig. 2a. All the 24 DMs show similar I-V curves with a strong rectification feature, which offers the desired nonlinearity for designing the reservoir. To build a complete reservoir module, peripheral circuits, such as amplifiers and multiplexers (MUXs), are employed to realize the mask process³³ and signal transmission. One DM along with its peripheral circuits is treated as one nonlinear physical node, which we called a DM node in this work, so the entire reservoir module is composed of 24 DM nodes. The details of signal processing in a DM node are shown in Fig. 2b. First, the input signal is amplified by a factor of A_i through an input amplifier. Then, the amplified signal and its inverted counterpart are connected to a MUX which is controlled by a special mask sequence with a pulse width of δ . The masked signal is added by an input voltage bias B_i and then applied to the DM. After that, the output current signal of DM is first converted to a voltage signal through a trans-impedance amplifier (TIA) and then amplified by a factor of A_o through an output amplifier. The final output signal and its inverted counterpart are fed to the readout module.

The function of the readout module is to perform a weighted summation of all reservoir states to obtain the final system output. The readout module includes 4 NVM chips, each one consisting of an array of 2048 one-transistor-one-resistor (1T1R) cells. The NVM used in this work has a memristor material stack of TiN/TaO_x/HfAlO_y/TiN, where the HfAlO_y acts as the resistive switching layer while the TaO_x layer serves as the thermal enhanced layer to improve the analog switching characteristics^{8,37}. The cross-sectional TEM image of the NVM device is shown in the inset of the right panel of Fig. 2a. The I-V curves of NVM at different resistance states are shown in right panel of Fig. 2a, which exhibits good linearity. Such linear I-V characteristics ensure that the device has the same resistance under different input voltage levels, which allows us to directly use analog voltages as inputs. Circuit diagram of the complete DM-RC system is shown in Fig. 2c. The analog input signals are directly fed into the reservoir module,

which has multiple parallel DM nodes with different masks. The output signals of the reservoir module are applied to the bit lines (BLs) of the NVM array in the form of differential pairs in order to realize both positive and negative weights^{38,39}. The current signal outputs from the source lines (SLs) are collected by the integrators, whose output signals are sampled to yield the final result. In this way, the DM-RC system completes the signal processing in a full analog fashion as illustrated in Fig. 1c.

Hyperparameter Analysis. In order to optimize the performance of the DM-RC system, four hyperparameters, A_i , B_i , A_o and δ , in the reservoir module are set to be externally adjustable. Figure 3a illustrates the experimental approach to explore the relationship between the DM node behavior and the system performance. The DM node behavior is characterized by the Input-Output (I-O) curve measured from voltage sweep, and the DM-RC system performance is measured by the normalized root mean square error (NRMSE) on a waveform classification task (see **Supplementary Fig. 1** for more details). Different sets of hyperparameters are used when optimizing both the DM node behaviors and the system performance. Under each set of hyperparameters, the waveform classification results of the DM-RC system and the corresponding I-O curves of the DM nodes are recorded for subsequent analysis. Figure 3b shows the normalized I-O curves of DM nodes, where the input and output voltages are normalized to $[-1, 1]$ and $[0, 1]$, respectively. Typical experimental results are obtained when the hyperparameters are set to be $A_i = 1.0$, $B_i = 1.2$ V, $A_o = 2.2$ and $\delta = 0.6$ ms. In order to facilitate the analysis of the impact of DM node's behavior on the RC system performance, three key features, including threshold (T), slope (S) and window (W), are extracted from the normalized I-O curve. A simplified DM node model is established using these three features (i.e., T , S and W), and the details of this model are described in the **Methods** section. Figure 3c shows the simulated I-O curves of the established DM node model under a set of simulation parameters, which are consistent with the experimental results. This result confirms that the established model can retain the key characteristics of the DM node. In the following, we then use it to further build the reservoir model and explore the relationship between the three node features and the system performance. Through analysis of both experimental and simulation results, we can then clarify the relationship between the DM node features and the DM-RC system performance.

Figures 3d and 3e show the experimental and simulation results of system performance as a function of threshold and slope, respectively, where the value of the window remains constant ($W = 0.15$). The results in Figs. 3d and 3e show the effect of node nonlinearity on the performance of the reservoir. From both the experimental and simulation results we can find that the system always achieves the optimal performance under certain threshold values. The experimental results in Fig. 3d show that the system performs the best when the threshold is around 0.1. The simulation results in Fig. 3e show a similar pattern, where the optimal performance region is around the threshold of 0.25. It is hence noted that there is a slight deviation between the optimal threshold values in the experimental and simulation results, which could be mainly attributed to the simplified DM node model that has only three parameters. For example, the output curve near the threshold changes gradually and also the I-O curve beyond the threshold is not completely linear, which are not captured in the simplified DM node model. A detailed

analysis on the dependence of system performance on the node features T and S is presented in **Supplementary Figs. 2 and 3**. In general, the nonlinearity of DM node (i.e. a hard sigmoid like function) moves the feature points outside a high-dimensional cube to its edges or vertices, while the other points inside the cube remain stationary. In order to make them linearly separable, points in different classes would need to be moved to different vertices of the cube. By adjusting the node features T and S , the feature points before nonlinear transformation can be panned and zoomed in the high-dimensional space. Compared with S , the value of T directly controls the position of the feature points before nonlinear transformation and determines whether they can be linearly separated after nonlinear transformation. From the above analysis, we can see that the threshold T is a key feature of the DM node that directly determines the DM-RC system performance.

Furthermore, the experimental and simulation results of system performance as a function of window size are shown in Figs. 3f and 3g, respectively, where the values of T and S remain constant. The results reflect the impact of the dynamic characteristics of DM node on the reservoir performance. Both results show that the system could achieve the optimal performance when the window size W , which to some extent represents the node memory capacity for the input signal, is neither too large nor too small. This can be qualitatively explained as follows: too weak node memory makes the reservoir unable to retain temporal characteristics of the input signal, resulting in a poor system performance; while too strong memory could easily saturate the node state and make the reservoir lose the ability to process subsequent signals, which can also lead to a poor system performance^{16,33}. From a more intuitive point of view, a proper node memory would increase the distance between the points from different classes in the high-dimensional feature space, thereby improving the classification performance of the reservoir. A more detailed analysis on the dependence of system performance on the node features W is presented in **Supplementary Fig. 4**. All these results suggest that the DM-RC system performance can be optimized by carefully tuning the hyperparameters and node features.

Arrhythmia Detection. To evaluate the performance of the DM-RC system on analog signal processing, a typical benchmark temporal task of arrhythmic heartbeat detection is carried out using MIT-BIH heart arrhythmia database⁴⁰, which contains 30-min electrocardiogram (ECG) recordings from 48 subjects. The training and testing processes for the DM-RC system are shown in Fig. 4a. Firstly, the original ECG waveform is re-sampled at a frequency of 72 Hz and split into single heartbeats of 700 ms (i.e., 50 time steps). Each heartbeat, labelled as healthy or arrhythmic according to the health status, is normalized so that its amplitude is in the interval of $[-1, 1]$, as shown in Fig. 4b. In total 10,000 different heartbeats are used as the dataset for this task, of which 5000 heartbeats are healthy and the rest 5000 are arrhythmic. Then, those samples in the dataset are divided into two groups: 1000 randomly selected samples for training and the rest 9000 samples for testing. The input signal is generated by a random combination of single heartbeats in the dataset, and the target signal Y_{target} is generated according to the corresponding labels. Subsequently, the input signal is fed into the DM reservoir module, where one-dimensional temporal signal is applied to 24 DM nodes in parallel after passing through different masks with a sequence length of 5 as shown in Fig. 4c.

For the training process, the outputs of 24 DM nodes are sampled as the reservoir states, as shown in Fig. 4d. Since the mask sequence length is 5, each DM node can produce 5 reservoir states in one time step, so that the number of final reservoir states reaches 120 per time step. In order to improve the system robustness to noise, here a noise-aware linear regression method is used to train the output weight W_{out} . We first combine the reservoir states at all time steps to generate the state matrix X . Then, a random matrix N_{tr} with uniformly distributed values between $-\sigma$ and σ is added to X , where σ is defined as the noise level for training. Subsequently, the weight matrix W_{out} can be calculated by $W_{out} = Y_{target}(X + N_{tr})^T((X + N_{tr})(X + N_{tr})^T)^\dagger$, where the symbol \dagger represents the Moore-Penrose pseudo-inverse^{33,41}. In order to verify the robustness of W_{out} , a simulated test process is carried out, where another random matrix N_{te} with uniformly distributed values between -0.04 and 0.04 is used to simulate the device noise. In such simulated test process, the system output Y can be calculated by $Y = (W_{out} + \max(|W_{out}|)N_{te})X$. A more detailed description of the noise-aware training method can be found in **Supplementary Fig. 5**. The dependence of the training error and simulated test error on different σ values during the training process is shown in Fig. 4e. It can be seen that as the noise level σ increases, the training error increases slowly while the simulated test error (both mean and variance) decreases rapidly. The result shows that the robustness of W_{out} to noise can be improved by adding a certain noise perturbation during the training process. It is worth noting that as σ further increases, the training error continues to increase, which would eventually lead to an increase in the test error as well. Thus, in our experiment, the optimal σ value is set to be 0.06 to get the ideal W_{out} . The last step of the training process is to map the calculated W_{out} to the device conductance of the NVM array. Here we use the differential conductance of two memristors to represent one element in W_{out} , where the conductance of each device is programmed between 0 to $33 \mu S$. A closed-loop programming scheme is applied here to write the device conductance to the target value (see **Method** section for more details). The result of weight mapping is shown in Fig. 4f, which displays the distributions (120×2) of target conductance weights, mapped conductance weights and weight-mapping errors compared with the target values. It can be seen that the trained weights are well mapped with relatively small errors thanks to the excellent analog switching characteristics of our memristor devices⁴².

For the testing process, the outputs of the 24 DM nodes are adjusted to a voltage range of 0 to $0.2 V$ through amplifiers and the corresponding negative voltage signals ($0 \sim -0.2 V$) are then generated by analog inverters. The differential voltage signals are directly fed into the NVM array, on which the devices have been programmed to the appropriate conductance. Subsequently, the output currents of each SL in NVM array are collected by an integrator. The integrator is reset at the beginning of each time step, and its output voltage is then sampled once at the end of each time step. The experimental results of the testing process are shown in Fig. 4g, where the output of the integrator is normalized to $[0, 1]$. It can be seen that the output signal can well fit the target signal and generates a peak when an arrhythmic heartbeat occurs. Finally, a reference value K_{ref} , labelled by the dashed line in Fig. 4g, is used to obtain the overall classification accuracy of the DM-RC system. The optimal system performance is achieved when the hyperparameters are set to be $A_i = 1.0$, $B_j = 2.4 V$, $A_o = 2.2$ and $\delta = 0.2 ms$. The average classification

accuracy we obtained in experiments is 96.6%, which is close to the software baseline of 98.2% achieved by a full digital RC system, as shown in Fig. 4h. This result demonstrates the feasibility of using DM-RC system to effectively process temporal signals with high accuracy.

Dynamic Gesture Recognition. Beyond arrhythmia detection, a more complex spatiotemporal task was implemented to verify the analog computing ability of the DM-RC system. Here we demonstrate the real-time processing of spatiotemporal signals from a three-axis acceleration sensor to classify different hand gestures. The schematic of fully analog computing with DM-RC system for dynamic gesture recognition is shown in Fig. 5a. When the hand performs a certain gesture, the three-dimensional analog signals from the acceleration sensor are directly fed into the DM-based reservoir module, of which the generated reservoir states are calculated in the NVM array-based readout module in a fully analog fashion to yield the final recognition results. In our experiment, the sensor signals of four classes of dynamic gestures (G1: equilateral triangle, G2: capital letter N, G3: inverted triangle, G4: capital letter Z) as shown in Fig. 5a are recorded at a sample frequency of 10 Hz. Typical signals with normalized amplitude (-1 to 1) are displayed in the Fig. 5b. Similar to the previous arrhythmia detection task, the recorded sensor signals are split into single gestures of 3 s (i.e., 30 time steps) to generate the dataset. The gestures in the dataset are then divided into two groups: 600 randomly selected samples for training and the rest 300 samples for testing. Figure 5c shows the implemented network structure by reconfiguring the DM-RC system, where 24 DM nodes are equally divided into three groups to process the input signals from three channels and 4 NVM arrays are used to classify four different types of gestures. The training and testing processes are also similar to those in the previous task. The outputs of 24 DM nodes are sampled as the reservoir states, as shown in Fig. 5d. Here, the mask sequence length is set to be 8, so each DM node can produce 8 reservoir states in one time step, and the final reservoir states reach 192 per time step. The result of weight mapping is shown in Fig. 5e, which displays the distributions (192×8) of target conductance weights, mapped conductance weights and weight-mapping errors compared with the target values.

The experimental results of the system output for the four classes of dynamic gestures are illustrated in the Fig. 5f, where the outputs of four integrators are all normalized to $[0, 1]$. Here, we use two reference values K_{ref} and N_{ref} to obtain the classification accuracy of four independent outputs. During every 30 time steps, we first compare the output signal with value K_{ref} and count the number of times (n) when the output value is greater than K_{ref} . Then, we further compare n with N_{ref} to determine the final classification result. If n is greater than N_{ref} , this segment of signal is detected as true, otherwise it is false. The flow chart of the above process is shown in **Supplementary Fig. 6**. In order to evaluate the advantages of our system in terms of energy efficiency, we compare both classification accuracy and energy consumption of our DM-RC system with a fully digital RC system as shown in Fig. 5g. It can be seen that, compared with the digital RC system, our DM-RC system has an average accuracy loss of only 1.1% but saves more than 99.9% of power consumption. The details of power estimation are described in **Supplementary Table 1**. Such low power consumption can be attributed to two key factors in our system: one is the fully analog signal transmission and processing realized in our system so that the large power consumption overhead caused by ADCs in conventional RC systems can be eliminated; the other one is that the large

recurrent neural network in traditional reservoirs is replaced by a small number of parallel DMs, which largely reduces not only hardware cost but also power consumption.

Discussion

In summary, a fully analog RC system has been realized using a set of parallel DMs for reservoir and NVM arrays for readout. The relationship between the electrical characteristics of DM nodes and DM-RC system performance has been studied in depth by analyzing the I-O curves and system performance under different hyperparameters. We find two key features, threshold and window, that have a significant impact on the quality of DM-based reservoir. The optimal system performance occurs at a specific threshold when the window is in a suitable range. By adjusting the hyperparameters, the implemented DM-RC system can perform different spatiotemporal signal processing tasks real-time with extremely low power consumption. Both temporal task, arrhythmia detection, and more complex spatiotemporal task, dynamic gesture recognition, have been demonstrated using our DM-RC system, where high detection accuracy of 96.6% and recognition accuracy 97.9% have been achieved respectively. Compared with traditional digital RC system, our memristor-based fully analog RC system has achieved more than three orders of magnitudes lower power consumption. Compared with previous works^{21,32,34,35,43,44}, the power consumption of our DM-RC system is also the lowest reported so far (see **Supplementary Table 2** for detailed comparison) owing to the fully analog signal transmission and the advantage of using parallel DMs equivalent to a large recursive network. To further reduce the power consumption and computing latency of our system, the entire DM-RC system can be miniaturized and monolithically integrated on chip in the future, thanks to the excellent CMOS compatibility and scalability of memristors. In addition, a deeper and more sophisticated RC system can be constructed using DM-RC system as a basic unit, which would further enhance the system performance because of richer reservoir states and stronger memory capacity. Our work demonstrates that such fully analog DM-RC system is a promising platform for edge computing with extremely low power consumption, and hence it has tremendous potential in applications such as smart wearable devices and micro robots.

Methods

Measurement setup. The entire test system mainly consists of three parts, i.e., a host personal computer (PC), a STM32 (MCU) development board and a user customized PCB as test board. The host PC operates as the upper computer for data loading, command sending and user interface which is coded in Python. The STM32 operates as the lower computer to communicate with the upper computer and generate the specific control signals which are transmitted to the test board through the general-purpose input/output (GPIO) ports. In addition, the STM32 is also responsible for generating and reading voltage signals through the integrated 12-bit digital-to-analog converters (DACs) and ADCs, respectively. It needs to be pointed out here, in addition to collecting the output of the integrators, the ADCs are only used in the steps of sampling states and mapping weights during the training process. The test board integrates 4 NVM chips, 24 DMs (connected by a probe card) and several commercial chips such as low-dropout

regulators (LDOs), MUXs, amplifiers and digital potentiometers (DPs). LDOs are used to provide stable positive and negative DC voltage sources for other chips in test board. MUXs are used to control the signal flow in the system and realize the mask process. Amplifiers are used to adjust the voltage signals and realize TIAs and integrators. DPs are used to adjust the gain of the amplifiers and adjust the bias voltage (B_i).

Simplified DM node model. The simplified DM node model is defined as:

$$V_o(k) = f(S(V_i(k) - V_t(k)))$$

1

where $V_i(k)$, $V_o(k)$ and $V_t(k)$ are the input voltage, output voltage and threshold voltage at k -th time step, respectively, while S and f are the slope of I-O curve and nonlinear function, respectively. Here, f is defined as:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

2

In Eq. (1), V_t changes dynamically with the input signal V_i . We consider the simplest case where the dynamic process is linear and time-invariant. In other words, V_t is the result of a linear filtering on V_i . The differential function of such linear filtering process is defined as:

$$V_t(k+1) = (1 + \alpha)V_t(k) - \alpha(V_i(k) - 2T)$$

3

where α is the filter weight which determines the time scale of the system memory, and T is the initial value of V_t which also corresponds to the threshold extracted from the I-O curve. The other feature extracted from the I-O curve is the window (W) which can be calculated as follows:

$W = \max(V_t) - \min(V_t)$, where the sequence of V_t is obtained by Eq. (3) with a dual sweep sequence of V_i . The simulation of such simplified DM node model is performed in Python with Numpy package by solving Equations (1)-(3).

Weight mapping process. We use a closed-loop programming scheme to map the target weight (W_{target}) onto the device conductance of the NVM array⁹. The basic loop of such programming scheme is run in the PC and coded in Python. In the PC program, there are three core functions: f_{read} , f_{set} , f_{reset} and f_{comp} . First, the function f_{read} is executed and a command is sent from the PC to the MCU to read the conductance state (W_{real}) of the NVM, where the read voltage is set to be 0.2 V. Then, the function f_{comp} is

executed, where the difference between W_{real} and W_{target} is calculated as $\Delta W = W_{\text{real}} - W_{\text{target}}$ and a maximum error weight ΔW_{max} is used to compare with ΔW to determine the next step of programming: 1) if $|\Delta W|$ is smaller than ΔW_{max} , then W_{target} is considered to be successfully mapped and the programming step is finished; 2) if ΔW is smaller than $-\Delta W_{\text{max}}$, the function f_{set} is executed and a SET operation is performed, where two voltage pulses with amplitudes of 5 V and V_{wl} (varying from 0.5 to 2.8 V) are applied to the BL and word line (WL) of the 1T1R cell, respectively; 3) if ΔW is larger than ΔW_{max} , the function f_{reset} is executed and a RESET operation is performed, where two voltage pulses with amplitudes of V_{sl} (varying from 1.5 to 3.0 V) and 5 V are applied to the SL and WL of the 1T1R cell, respectively. Steps 2) and 3) are repeated until the target weight W_{target} is successfully mapped.

Data Availability

Data that support the findings of this study are available from the corresponding authors upon reasonable request.

Code Availability

The code that supports the DM-RC system simulations in this study is available at <https://github.com/Tsinghua-LEMON-Lab/Reservoir-computing>. Other codes that support the findings of this study are available from the corresponding authors upon reasonable request.

Declarations

Acknowledgements

This work was supported in part by China key research and development program (2021ZD0201205), National Natural Science Foundation of China (91964104, 61974081, 62025111, 92064001, 62104126), and the XPLOERER Prize.

Author contributions

Y.Z. and J.T. conceived and designed the experiments. X.L. contributed to the device preparation and material analysis. Y.Z. performed the experiments and data analysis. Y.Z. and J.T. wrote the paper. All authors discussed the results and commented on the manuscript. J.T., H.W. and H.Q. supervised the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information is available in the online version of the paper. Reprints and permissions information is available at www.nature.com/reprints. Correspondence and requests for materials should be addressed to J. T., and H. W.

References

1. Zidan M. A., Strachan J. P. & Lu W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).
2. Zhang W. *et al.* Neuro-inspired computing chips. *Nat. Electron.* **3**, 371–382 (2020).
3. Chua L. Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* **18**, 507–519 (1971).
4. Strukov D. B., Snider G. S., Stewart D. R. & Williams R. S. The missing memristor found. *Nature* **453**, 80–83 (2008).
5. Cai F. *et al.* Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **3**, 409–418 (2020).
6. Li X. *et al.* Power-efficient neural network with artificial dendrites. *Nat. Nanotechnol.* **15**, 776–782 (2020).
7. Kumar S., Williams R. S. & Wang Z. Third-order nanocircuit elements for neuromorphic engineering. *Nature* **585**, 518–523 (2020).
8. Yao P. *et al.* Face classification using electronic synapses. *Nat. Commun.* **8**, 15199 (2017).
9. Yao P. *et al.* Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–646 (2020).
10. Dalgaty T. *et al.* In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling. *Nat. Electron.* **4**, 151–161 (2021).
11. Cai F. *et al.* A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nat. Electron.* **2**, 290–299 (2019).
12. Hochreiter S. & Schmidhuber J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
13. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
14. Salmela L. *et al.* Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network. *Nat. Mach. Intell.* **3**, 344–354 (2021).
15. Wright L. G. *et al.* Deep physical neural networks trained with backpropagation. *Nature* **601**, 549–555 (2022).

16. Appeltant L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
17. Shastri B. J. *et al.* Photonics for artificial intelligence and neuromorphic computing. *Nat. Photonics* **15**, 102–114 (2021).
18. Jaeger H. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* **148**, 13 (2001).
19. 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).
20. Gauthier D. J., Bollt E., Griffith A. & Barbosa W. A. S. Next generation reservoir computing. *Nat. Commun.* **12**, 5564 (2021).
21. Alomar M. L. *et al.* Digital Implementation of a Single Dynamical Node Reservoir Computer. *IEEE Transactions on Circuits and Systems II: Express Briefs* **62**, 977–981 (2015).
22. Wang Q. *et al.* Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA. *Neurocomputing* **221**, 146–158 (2017).
23. Kleyko D., Frady E. P., Kheffache M. & Osipov E. Integer Echo State Networks: Efficient Reservoir Computing for Digital Hardware. *IEEE Trans Neural Netw Learn Syst*, 1–14 (2020).
24. Torrejon J. *et al.* Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428–431 (2017).
25. Riou M. *et al.* Neuromorphic computing through time-multiplexing with a spin-torque nano-oscillator. In *2017 IEEE International Electron Devices Meeting (IEDM)* 36.33.31–36.33.34 (IEEE, 2017).
26. Nakane R., Tanaka G. & Hirose A. Reservoir Computing With Spin Waves Excited in a Garnet Film. *IEEE Access* **6**, 4462–4469 (2018).
27. Martinenghi R. *et al.* Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **108**, 244101 (2012).
28. Vandoorne K. *et al.* Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, 3541 (2014).
29. Antonik P., Marsal N., Brunner D. & Rontani D. Human action recognition with a large-scale brain-inspired photonic computer. *Nat. Mach. Intell.* **1**, 530–537 (2019).
30. Hochstetter J. *et al.* Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nat. Commun.* **12**, 4008 (2021).
31. Fu K. *et al.* Reservoir Computing with Neuromemristive Nanowire Networks. In *2020 International Joint Conference on Neural Networks (IJCNN)* 1–8 (IEEE, 2020).
32. Milano G. *et al.* In materia reservoir computing with a fully memristive architecture based on self-organizing nanowire networks. *Nat. Mater.* **21**, 195–202 (2022).
33. Zhong Y. *et al.* Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun.* **12**, 408 (2021).

34. Moon J. *et al.* Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2**, 480–487 (2019).
35. Yu J. *et al.* Energy efficient and robust reservoir computing system using ultrathin (3.5 nm) ferroelectric tunneling junctions for temporal data learning. In *2021 Symposium on VLSI Technology* 1–2 (IEEE, 2021).
36. Tanaka G. *et al.* Recent advances in physical reservoir computing: A review. *Neural Netw.* **115**, 100–123 (2019).
37. Gao B. *et al.* Oxide-based analog synapse: Physical modeling, experimental characterization, and optimization. In *2016 IEEE International Electron Devices Meeting (IEDM)* 7.3.1–7.3.4 (IEEE, 2016).
38. Li C. *et al.* Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **1**, 52–59 (2017).
39. Li C. *et al.* Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* **9**, 2385 (2018).
40. Moody G. B. & Mark R. G. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine* **20**, 45–50 (2001).
41. Lukosevicius M. & Jaeger H. Survey: Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
42. Wan W. *et al.* 33.1 A 74 TMACS/W CMOS-RRAM Neurosynaptic Core with Dynamically Reconfigurable Dataflow and In-situ Transposable Weights for Probabilistic Graphical Models. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)* 498–500 (IEEE, 2020).
43. Brunner D., Soriano M. C., Mirasso C. R. & Fischer I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
44. Liang X. *et al.* Rotating neurons for all-analog implementation of cyclic reservoir computing. *Nat. Commun.* **13**, 1549 (2022).

Figures

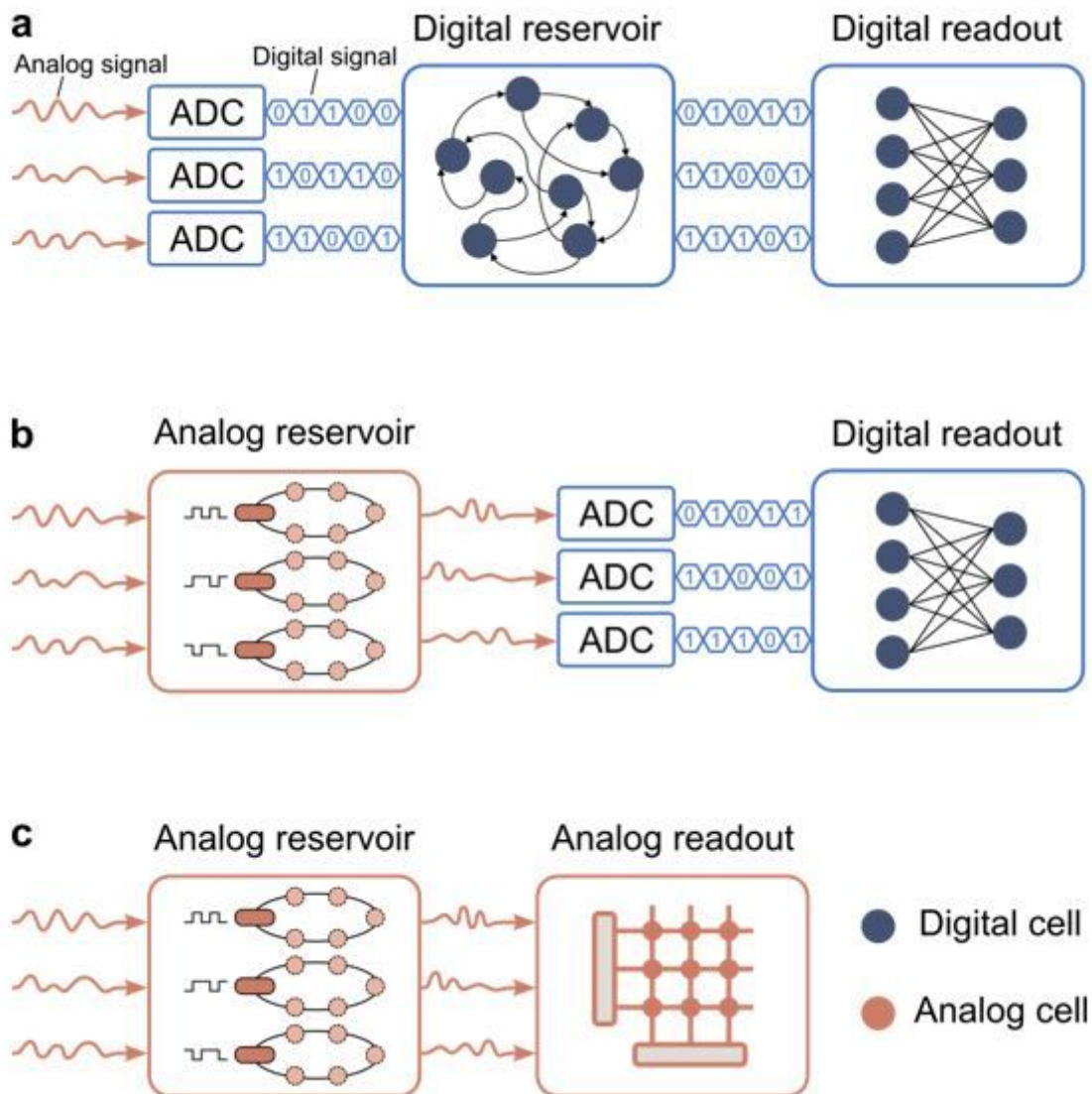


Figure 1

Different types of RC systems. (a) Fully digital RC system, where the analog input signals are first converted into digital signals by ADCs, and then fed into the reservoir and the readout layer based on digital cells. (b) Hybrid RC system, where the analog signals are directly input to the reservoir based on analog cells, and the analog reservoir states are converted into digital signals which are then fed into the readout layer based on digital cells. (c) Fully analog RC system, where the analog input signals are directly fed into the reservoir and the readout layer based on analog cells.

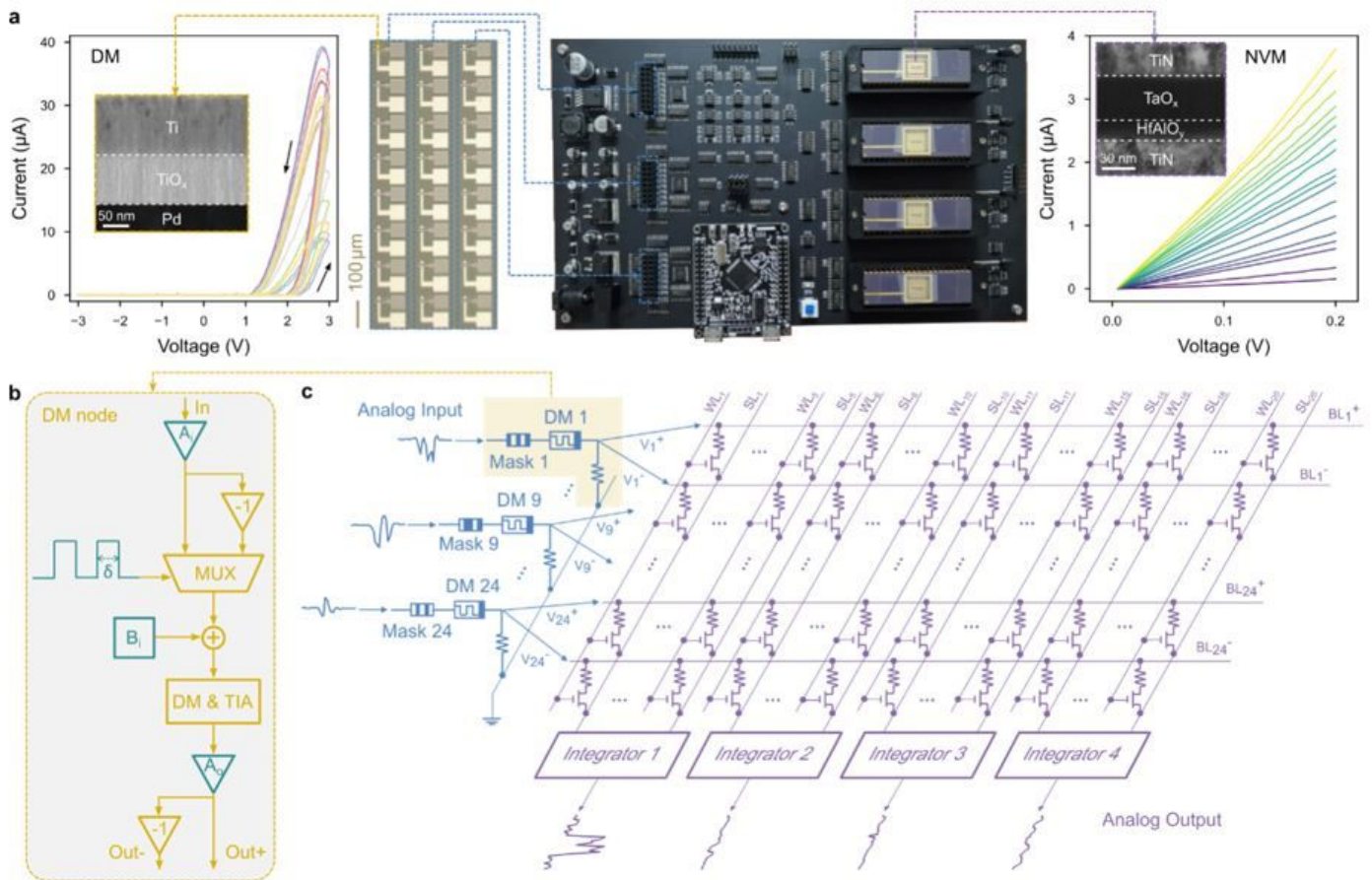


Figure 2

Device characteristics of memristors and the architecture of DM-RC hardware system. (a) Left panel, optical image and I-V hysteresis curves of 24 dynamic memristors (DM) and cross-sectional transmission electron microscope (TEM) image of the fabricated dynamic memristor, consisting of a vertically stacked structure of Ti/TiO_x/Pd (110 nm/80 nm/50 nm). Middle panel, photograph of the integrated PCB system, which consists of a power module, a reservoir module, a readout module (including 4 NVM chips consisting of a 2,048-memristor array) and an MCU. Right panel, I-V curves of non-volatile memristors (NVM) with different resistance states and cross-sectional TEM image of the fabricated non-volatile memristor, consisting of a material stack of TiN/TaO_x/HfAlO_y/TiN (30 nm/45 nm/8 nm/30 nm). (b) The details of signal processing in a DM node. First, the input signal is amplified by a factor of A_i through the input amplifier. Then, the amplified signal and its inverted signal are connected to a MUX which is controlled by a special mask sequence with a pulse width of δ . The masked signal is added by an input voltage bias B_i and then applied to the DM. The output current of DM is first converted to voltage signal by a trans-impedance amplifier (TIA) and then amplified by a factor of A_o through the output amplifier. The final output signal and its inverted signal are fed to the readout module. (c) Circuit diagram of the hardware RC system. The analog input signals are directly fed into multiple parallel DM nodes with different masks. The output signals of the reservoir module are applied to the BLs of the NVM array in the

form of differential pairs, in order to realize both positive and negative weights. The current signals output from the SLs are collected by the integrators and the final output signals are sampled from the output of these integrators.

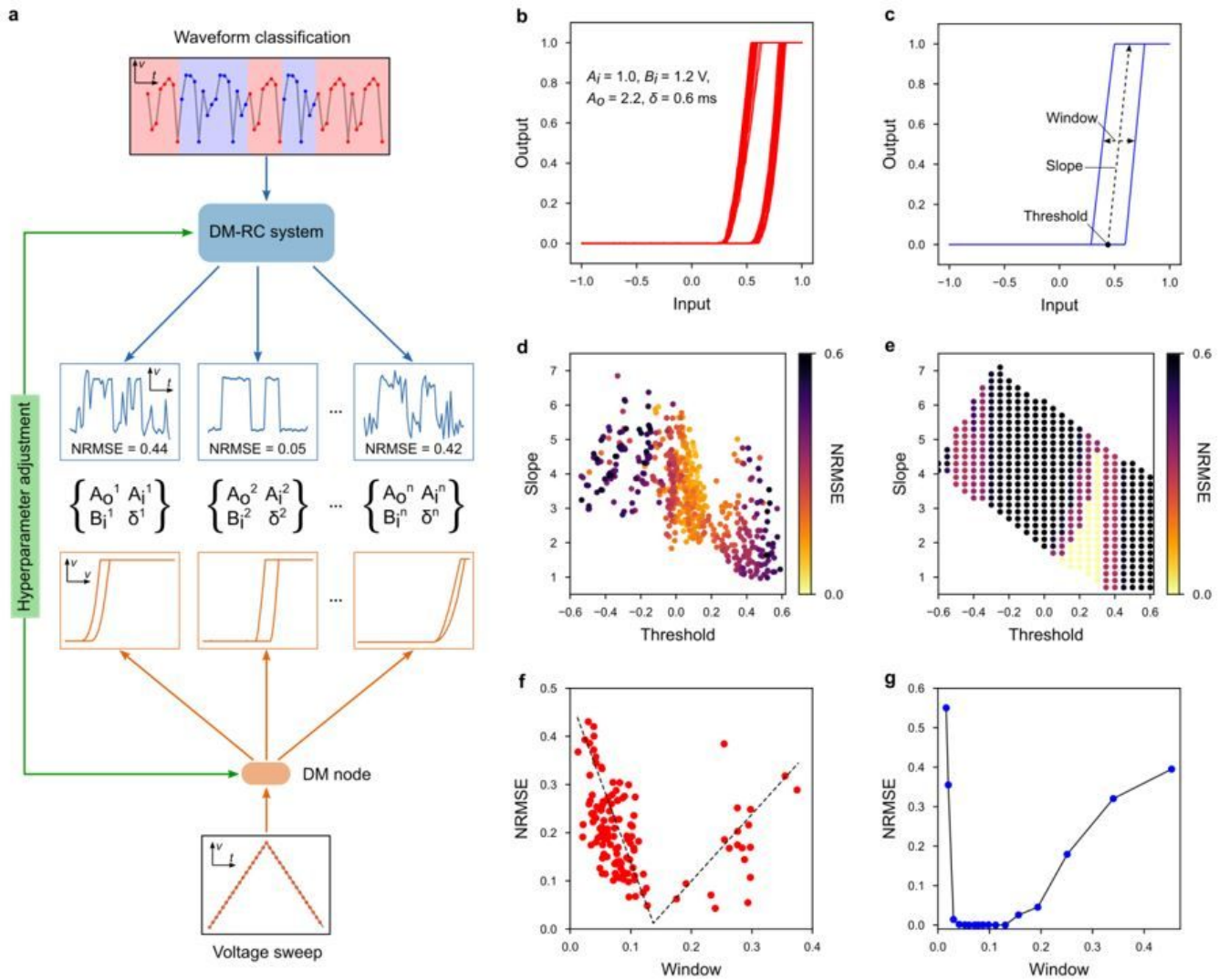


Figure 3

Hyperparameter adjustment for improving the reservoir performance. (a) The experimental approach of exploring the relationship between the DM node behavior and the system performance, where the DM node behavior is characterized by I-O curve and the system performance is measured by the normalized root mean square error (NRMSE) on a waveform classification task. Different hyperparameters (A_i, B_i, A_0 and δ) are used when obtaining waveform classification results and the corresponding I-O curves. (b) The experimental results of the normalized I-O curves under a set of hyperparameters ($A_i = 1.0, B_i = 1.2 \text{ V}, A_0 = 2.2$ and $\delta = 0.6 \text{ ms}$), where the input voltage is normalized to $[-1, 1]$ and the output voltage is normalized to $[0, 1]$. (c) The simulated results of the normalized I-O curves under a set of simulation parameters,

where three features (threshold T , slope S and window W) are extracted for subsequent analysis. **(d)** The experimental results of system performance changing with the threshold and slope (extracted from normalized I-O curve). **(e)** The simulation results of system performance changing with the threshold and slope (simulation parameters). **(f-g)** The experimental **(f)** and simulated **(g)** results of system performance changing with the window size, respectively.

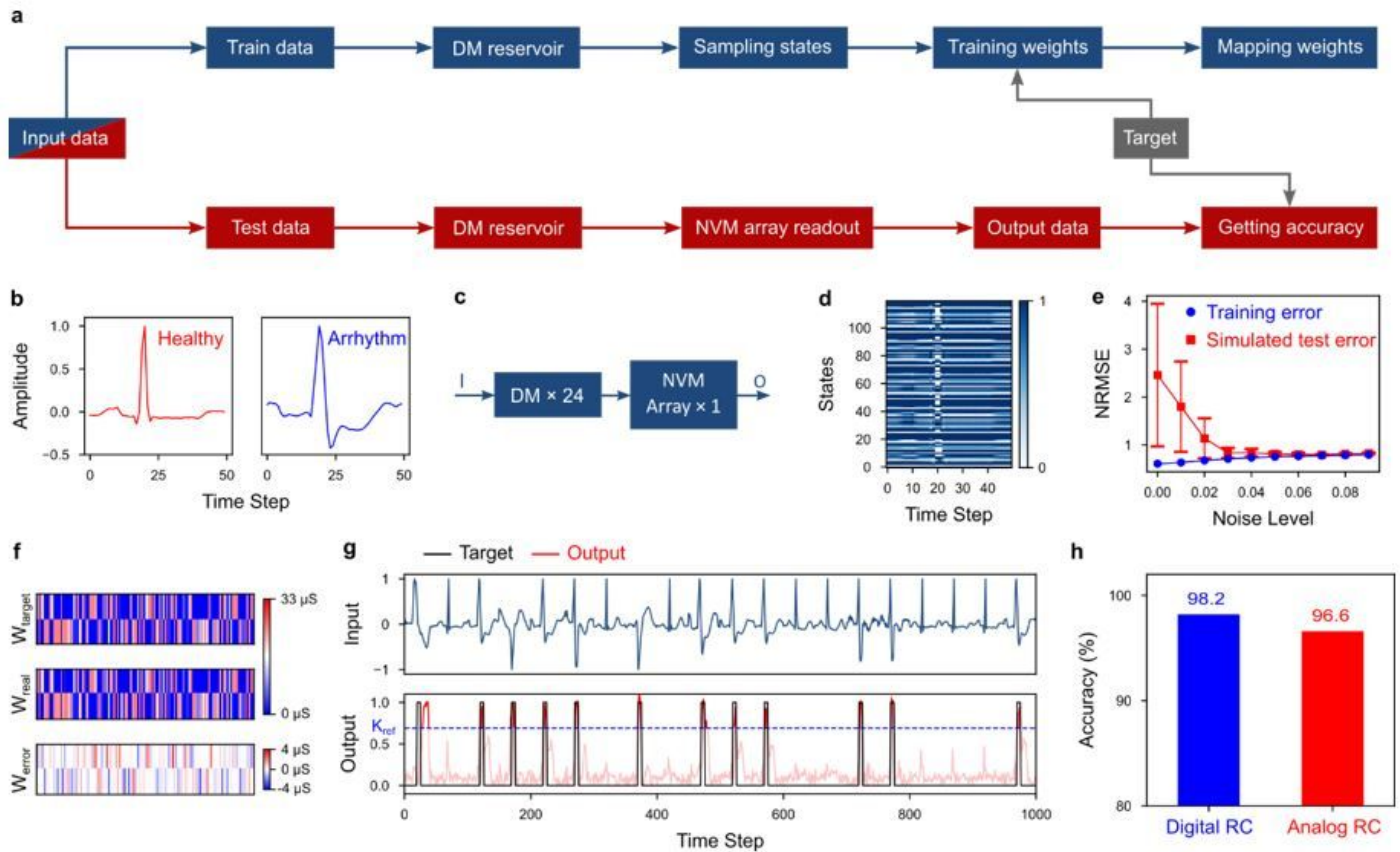


Figure 4

Demonstration of arrhythmia detection with the DM-RC system. **(a)** Schematic of the training and testing processes in the DM-RC system. **(b)** Two classes (healthy and arrhythmia) of heartbeat signals selected from MIT-BIH arrhythmia database. **(c)** Network structure used for the arrhythmia detection task, where one-dimensional heartbeat signals are simultaneously fed into 24 independent DM nodes with different masks and the output signals of reservoir module are input to a NVM array for readout. **(d)** The normalized reservoir states corresponding to the input signal. **(e)** Weight training process using a linear regression as a function of noise, where the train error and simulated test error vary with the noise level. **(f)** Distributions (120×2) of target conductance weight, mapped conductance weight and weight-mapping error compared with the target values. **(g)** The input (combination of healthy and arrhythmia heartbeats) and output signals of the DM-RC system, where the system output is normalized to [0, 1]. A reference value K_{ref} , shown in the figure as a dashed line, is used to obtain the overall classification accuracy of the

DM-RC system. The optimal system performance is achieved when the hyperparameters are set to be $A_i = 1.0$, $B_i = 2.4$ V, $A_o = 2.2$ and $\delta = 0.2$ ms. **(h)** The comparison of the classification accuracy between digital and fully analog RC systems, where the accuracy loss of our fully analog DM-RC system is only 1.6%.

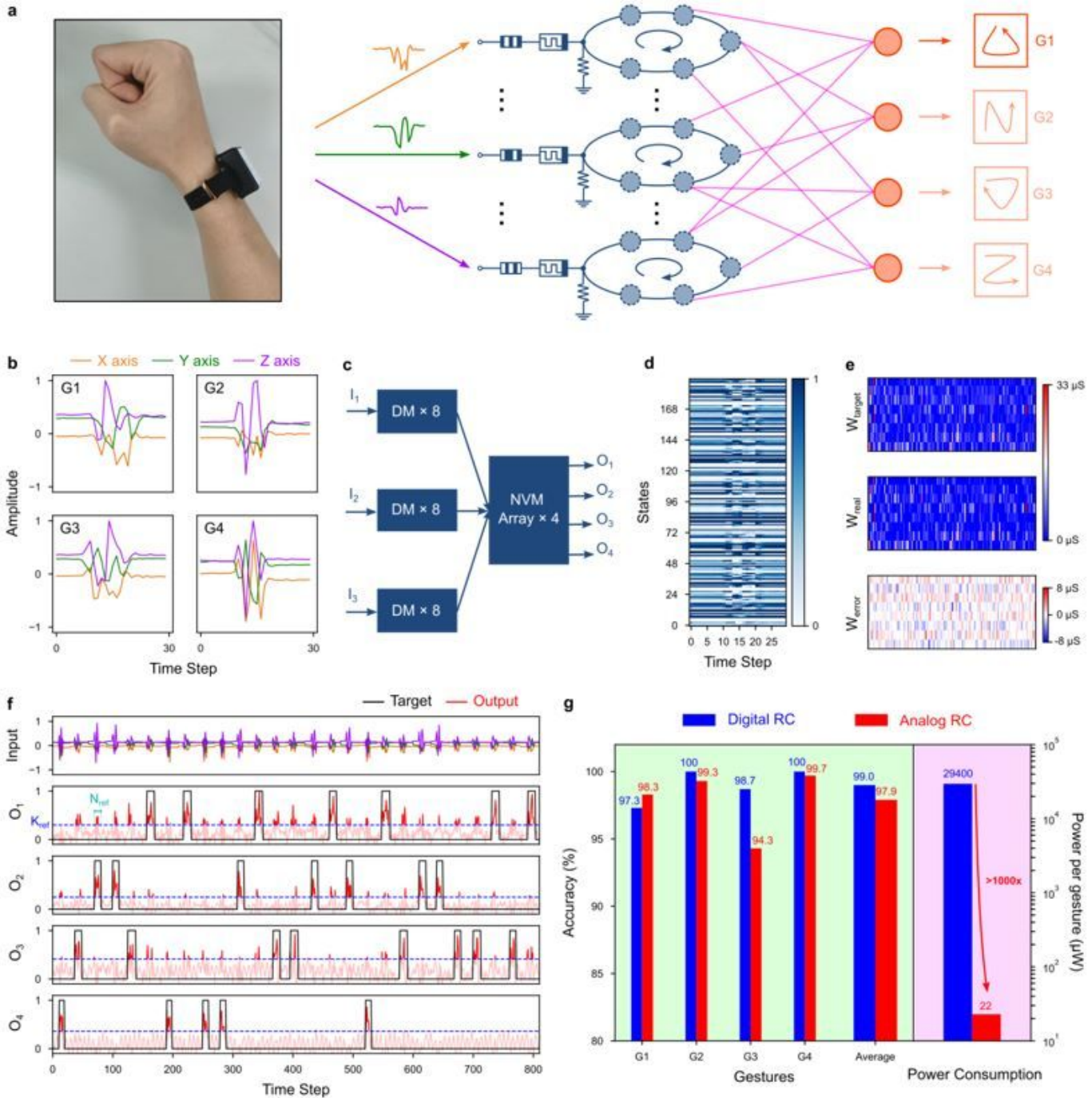


Figure 5

Demonstration of dynamic gesture recognition with the DM-RC system. **(a)** A conceptual schematic of fully analog computing with DM-RC system in a dynamic gesture recognition task. Arrows indicate the

direction of hand movement. **(b)** Four classes (G1: equilateral triangle, G2: capital letter N, G3: inverted triangle, G4: capital letter Z) of dynamic gesture signals collected from a 3-axis acceleration sensor. The signal amplitude is normalized to $[-1, 1]$. **(c)** Network structure used for the dynamic gesture recognition task, where the three-dimensional gesture signals are parallelly fed into 3 groups of DM nodes (each group has 8 DM nodes) and the output signals of reservoir module are simultaneously input to 4 NVM arrays. **(d)** The normalized reservoir states corresponding to a gesture signal. **(e)** Distributions (192×8) of target conductance weight, mapped conductance weight and weight-mapping error compared with the target values. **(f)** The input (combination of 4 classes of gesture signals) and output signals of DM-RC system, where the sampled outputs of 4 integrators are all normalized to $[0, 1]$. Two reference values K_{ref} and N_{ref} shown in the figure as dashed lines, are used to obtain the overall classification accuracy of DM-RC system. The optimal system performance is achieved when the hyperparameters are set to be $A_i = 1.0$, $B_i = 2.4 \text{ V}$, $A_o = 2.2$ and $\delta = 0.2 \text{ ms}$, same as the previous arrhythm detection task. **(g)** Comparison of the classification accuracy and power consumption between digital and fully analog RC systems, where the overall reduced accuracy and power consumption for our DM-RC system are 1.1% and 99.9 % respectively.

Supplementary Files

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

- [SupplementaryInformation.docx](#)
- [SupplementaryVideo.mp4](#)