nature electronics

Article

https://doi.org/10.1038/s41928-022-00838-3

A memristor-based analogue reservoir computing system for real-time and power-efficient signal processing

Received: 25 April 2022

Accepted: 18 August 2022

Published online: 26 September 2022

Check for updates

Yanan Zhong[1,2,5], Jianshi Tang[1,3,5] ✉, Xinyi Li[1,5], Xiangpeng Liang[1,4], Zhengwu Liu[1], Yijun Li[1], Yue Xi[1], Peng Yao[1], Zhenqi Hao[1], Bin Gao[1,3], He Qian[1,3] and Huaqiang Wu[1,3] ✉

Reservoir computing offers a powerful neuromorphic computing architecture for spatiotemporal signal processing. To boost the power efficiency of the hardware implementations of reservoir computing systems, analogue devices and components—including spintronic oscillators, photonic modules, nanowire networks and memristors—have been used to partially replace the elements of fully digital systems. However, the development of fully analogue reservoir computing systems remains limited. Here we report a fully analogue reservoir computing system that uses dynamic memristors for the reservoir layer and non-volatile memristors for the readout layer. The system can efficiently process spatiotemporal signals in real time with three orders of magnitude lower power consumption than digital hardware. We illustrate the capabilities of the system using temporal arrhythmia detection and spatiotemporal dynamic gesture recognition tasks, achieving accuracies of 96.6% and 97.9%, respectively. Our memristor-based fully analogue reservoir computing system could be of use in edge computing applications that require extremely low power and hardware cost.

With the rapid development of artificial intelligence, artificial neural networks (ANNs) are becoming increasingly larger and deeper in scale, placing considerable strain on hardware based on the von Neumann architecture on which they are run[1,2]. To break the von Neumann bottleneck—the performance and energy limits imposed by having to transfer data between memory and processing units—brain-inspired neuromorphic computing based on devices such as memristors[3,4] could be used[5–7]. For example, computing-in-memory architecture with emerging resistive switching devices (including non-volatile memristors (NVMs)) has shown orders of magnitudes higher energy efficiency than traditional computing platforms, such as central processing units and graphics processing units[2]. This is primarily because parallel multiply–accumulate computations can be implemented in a purely analogue fashion via fundamental physics laws on a memristor array by making use of its analogue resistive switching characteristics.

Most of NVM-based ANN demonstrations have been feedforward networks, such as multilayer perceptron and convolutional neural network[8–11], which have difficulty handling temporal tasks. ANNs with recursive connections—typically called recurrent neural networks (RNNs)—are designed to deal with temporal inputs[12–14]. However,

[1]School of Integrated Circuits, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China. [2]Institute of Functional Nano & Soft Materials (FUNSOM), Jiangsu Key Laboratory for Carbon-Based Functional Materials & Devices, Soochow University, Suzhou, China. [3]Beijing Innovation Center for Future Chips (ICFC), Tsinghua University, Beijing, China. [4]Microelectronics Lab, James Watt School of Engineering, University of Glasgow, Glasgow, UK. [5]These authors contributed equally: Yanan Zhong, Jianshi Tang, Xinyi Li. ✉e-mail: jtang@tsinghua.edu.cn; wuhq@tsinghua.edu.cn

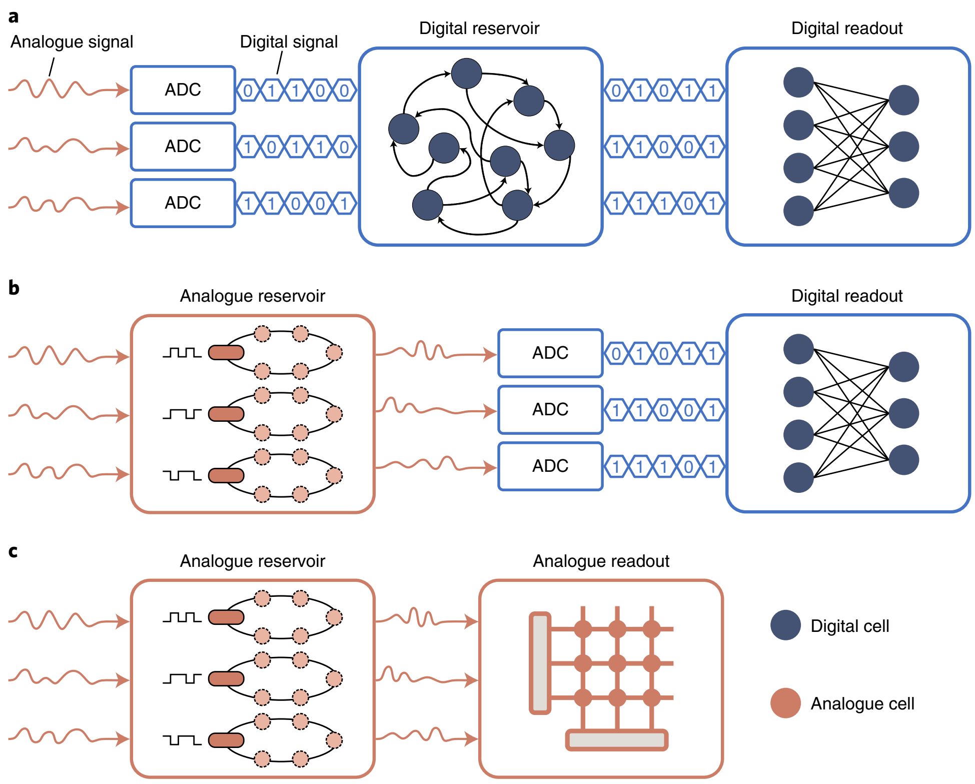

Fig. 1 | Different types of RC system. a, Fully digital RC system, where the analogue input signals are first converted into digital signals by ADCs and then fed into the reservoir and readout layer based on digital cells. **b**, Hybrid RC system, where the analogue signals are directly input to the reservoir based on analogue cells, and the analogue reservoir states are converted into digital signals that are then fed into the readout layer based on digital cells. **c**, Fully analogue RC system, where the analogue input signals are directly fed into the reservoir and readout layer based on analogue cells.

training RNNs is challenging as the process suffers from exploding or vanishing gradient-related problems. They are also difficult to implement with NVM arrays in a fully analogue fashion because variations in NVM conductance and noise in analogue circuits can cause error accumulation during recursive computations, leading to large output errors.

An alternative approach is to replace the recursive network with a dynamic physical system[15–17]. This approach can circumvent the error accumulation problem in the recursive network as well as reduce hardware cost. A representative example of this idea is reservoir computing (RC)[18–20], which was initially considered as a specialized variant of an RNN. Unlike a 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 created 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.

Hardware implementations of RC systems can be roughly divided into three types: fully digital systems, hybrid systems and fully analogue systems. In fully digital RC systems (Fig. 1a), the reservoir layer and readout layer use digital components, such as field-programmable gate arrays or digital signal processors[21–23]. Although these systems can minimize noise, additional analogue-to-digital converters (ADCs) are usually needed to convert the analogue signals into digital inputs, which require power and introduce latency. These inefficiencies are worsened when floating-point-based approaches are used. To address these problems, hybrid RC systems, which have an analogue reservoir layer and a digital readout layer, can be used (Fig. 1b). These systems can directly receive analogue signals and process them in the reservoir, and they have been demonstrated using spintronic oscillators[24–26], photonic modules[27–29], nanowire networks[30–32] and memristors[33–35]. These devices can be considered as special physical systems or nodes[36] that function as a reservoir and can further reduce both energy consumption and hardware area. However, these hybrid RC systems still require ADCs to convert analogue reservoir states into digital signals for the digital readout layer. Recently, NVM arrays have been used to create the analogue readout layer[32,35]. However, the systems reported so far still need digital units, such as ADCs and registers, between the reservoir module and readout module for data conversion and buffering. In fully analogue RC systems (Fig. 1c) with analogue reservoir and readout layers, signals can be directly transmitted and processed throughout the system without any conversion and buffering. Such systems could perform real-time spatiotemporal signal processing with lower power consumption and hardware cost compared with digital or hybrid digital–analogue systems, but they have not yet been demonstrated.

In this Article, we report a memristor-based fully analogue RC system. The approach is based on two distinct types of memristor: dynamic memristors (DMs) that are used as physical nodes to construct parallel reservoirs, and NVM arrays that are used as the readout layer. We examine the relationship between the node features and system performance by carefully adjusting the hyperparameters. Two key node features—threshold and window—are found to have a large impact on system performance. The threshold represents the smallest input signal that is required to turn on the physical node, and the window characterizes the size of the hysteresis window that a physical node exhibits under the dual sweep of the input signal. By engineering DM-based reservoirs, our RC system can be used for real-time spatiotemporal signal processing with more than three orders of magnitude lower power consumption than a digital RC system equivalent. We evaluate the performance of the approach via arrhythmia detection and dynamic gesture recognition tasks, achieving accuracies of 96.6% and 97.9%, respectively.

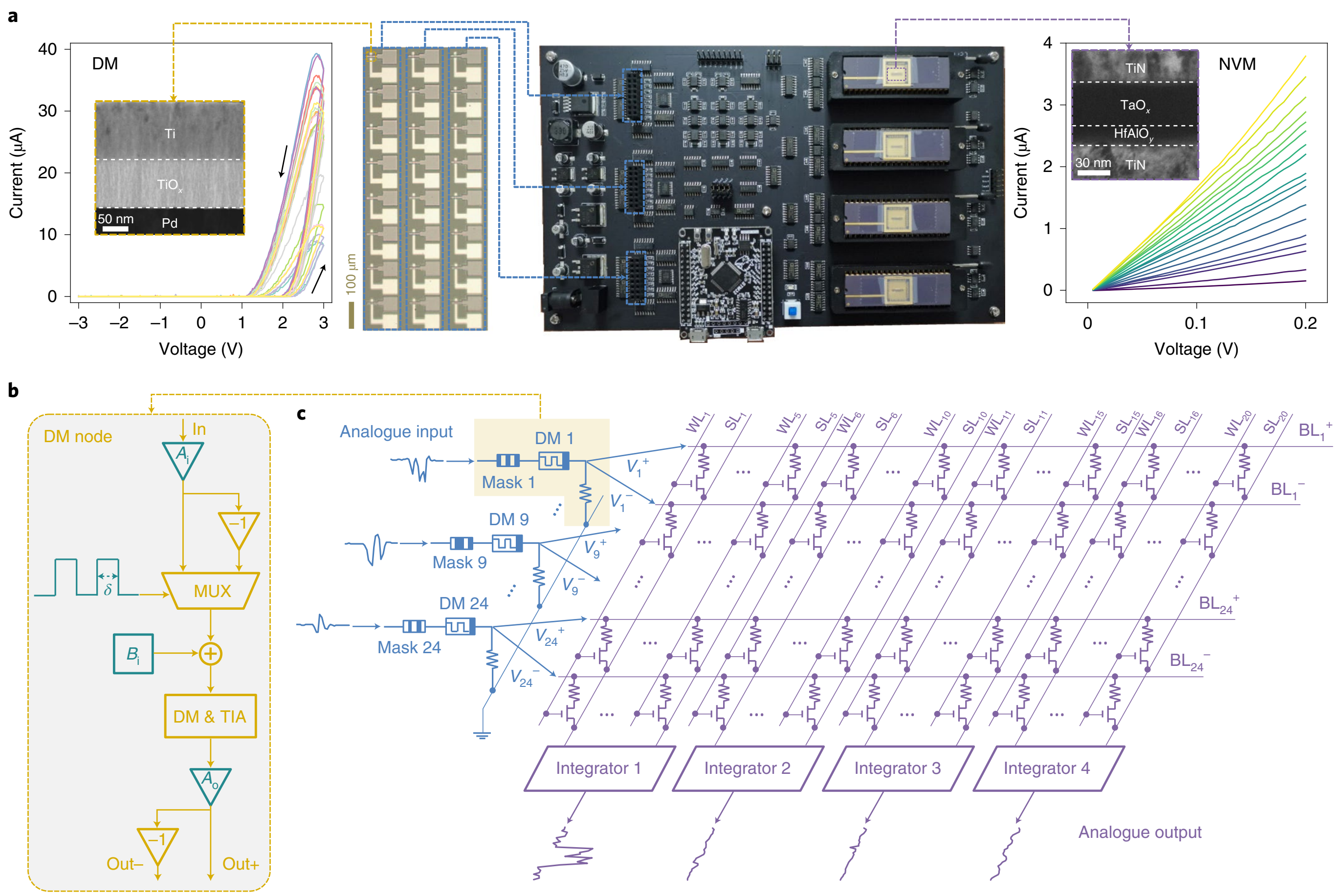

Fig. 2 | Device characteristics of memristors and architecture of DM-RC hardware system. a, Optical image and *I*–*V* hysteresis curves of 24 DMs and cross-sectional TEM image of the fabricated DM, consisting of a vertically stacked structure of Ti/TiO$_x$/Pd (110/80/50 nm) (left). Photograph of the integrated PCB system, which consists of a power module, reservoir module, readout module (including four NVM chips consisting of a 2,048 memristor array) and an MCU (middle). *I*–*V* curves of NVM with different resistance states and cross-sectional TEM image of the fabricated NVM, consisting of a material stack of TiN/TaO$_x$/HfAlO$_y$/TiN (30/45/8/30 nm) (right). **b**, Details of signal processing in a DM node. First, the input signal is amplified by a factor 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 pulse width δ. The masked signal is added by an input voltage bias B_i and then applied to the DM. The output current of the DM is first converted to a voltage signal by a TIA and then amplified by a factor 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 analogue 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, to realize both positive and negative weights. The current-signal outputs from the SLs are collected by the integrators and the final output signals are sampled from the output of these integrators.

System architecture and device characteristics

The hardware implementation of a fully integrated dynamic-memristor-based reservoir computing (DM-RC) system is shown in Fig. 2a, which consists of a 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 the data transfer. The reservoir module maps the features of low-dimensional temporal input signals to high-dimensional space. Such high-dimensional features (that are reservoir states) are usually linearly separable; therefore, they can be classified using a simple fully connected readout layer. The core of the reservoir module is a set of DMs that 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 (110/80/50 nm), and the cross-sectional transmission electron microscopy (TEM) image of the device is shown in Fig. 2a, left inset. Here 24 DMs are used in our system to form parallel reservoirs, and the memristive current–voltage (*I*–*V*) curves along with the optical image of these devices are shown in Fig. 2a, left. All the 24 DMs show similar *I*–*V* curves with a strong rectification feature, which offers the desired nonlinearity for designing the reservoir. More electrical characterization results of the DM are shown in Supplementary Fig. 1a,b. To build a complete reservoir module, peripheral circuits, such as amplifiers and multiplexers (MUXs), are employed to realize the mask process[33] (Methods) and signal transmission. One DM, along with its peripheral circuits, is treated as one nonlinear physical node, which we call a DM node; therefore, 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 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 pulse width 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 the DM is first converted to a voltage signal through a transimpedance amplifier (TIA) and then amplified by a factor 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 the reservoir states to obtain the final system output.

The readout module includes four NVM chips, each one consisting of an array of 2,048 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 $HfAlO_y$ acts as the resistive switching layer and the TaO_x layer serves as the thermal enhanced layer to improve the analogue switching characteristics[8,37]. The cross-sectional TEM image of the NVM device is shown in Fig. 2a, right inset. The *I*–*V* curves of NVM at different resistance states are shown in Fig. 2a, right, 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 analogue voltages as inputs. More electrical characterization results of the NVM are shown in Supplementary Fig. 1c,d. The circuit diagram of the complete DM-RC system is shown in Fig. 2c. The analogue input signals are directly fed into the reservoir module that 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 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 (Supplementary Fig. 2). In this way, the DM-RC system completes the signal processing in a fully analogue fashion (Fig. 1c).

Hyperparameter analysis

To optimize the performance of the DM-RC system, four hyperparameters, namely, 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 behaviour and system performance. The DM node behaviour is characterized by the input–output (*I*–*O*) curve measured from the voltage sweep, and the DM-RC system performance is measured by the normalized root mean square error (NRMSE) on a waveform classification task (Supplementary Fig. 3). Different sets of hyperparameters are used when optimizing both DM node behaviours and 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 as $A_i = 1.0$, $B_i = 1.2$ V, $A_o = 2.2$ and $\delta = 0.6$ ms. To facilitate the analysis of the impact of the DM node's behaviour on the RC system performance, three key features, namely, 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, and the details of this model are described in Methods. 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 system performance. Through the analysis of both experimental and simulation results, we can then clarify the relationship between the DM node features and DM-RC system performance.

Figure 3d,e shows the experimental and simulation results of system performance as a function of threshold and slope, respectively, where the value of window remains constant ($W = 0.15$). The results shown in Fig. 3d,e reveal the effect of node nonlinearity on the performance of the reservoir. From both experimental and simulation results, we can find that the system always achieves the optimal performance under certain threshold values. The experimental results (Fig. 3d) show that the system performs the best when the threshold is around 0.1. The simulation results (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. 4 and 5. In general, the nonlinearity of the DM node (that is a hard sigmoid-like function) moves the feature points outside a high-dimensional cube to its edges or vertices, whereas the other points inside the cube remain stationary. 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 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 Fig. 3f,g, 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 the temporal characteristics of the input signal, resulting in poor system performance; however, 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 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 of the dependence of system performance on node feature *W* is presented in Supplementary Fig. 6. 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 analogue signal processing, a typical benchmark temporal task of arrhythmic heartbeat detection is carried out using the MIT-BIH heart arrhythmia database[40], which contains 30 min electrocardiogram recordings from 48 subjects. The training and testing processes for the DM-RC system are shown in Fig. 4a. First, the original electrocardiogram waveform is re-sampled at a frequency of 72 Hz and split into single heartbeats of 700 ms (that is, 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 [−1, 1] (Fig. 4b). In total, 10,000 different heartbeats are used as the dataset for this task, out of which 5,000 heartbeats are healthy and the remaining 5,000 are arrhythmic. Then, those samples in the dataset are divided into two groups: 1,000 randomly selected samples for training and the remaining 9,000 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 a one-dimensional temporal signal is applied to 24 DM nodes in parallel after passing through different masks with a sequence length of 5 (Fig. 4c).

For the training process, the outputs of 24 DM nodes are sampled as the reservoir states (Fig. 4d). Since the mask sequence length is 5, each DM node can produce five reservoir states in one time step; therefore, the number of final reservoir states reaches 120 per time step.

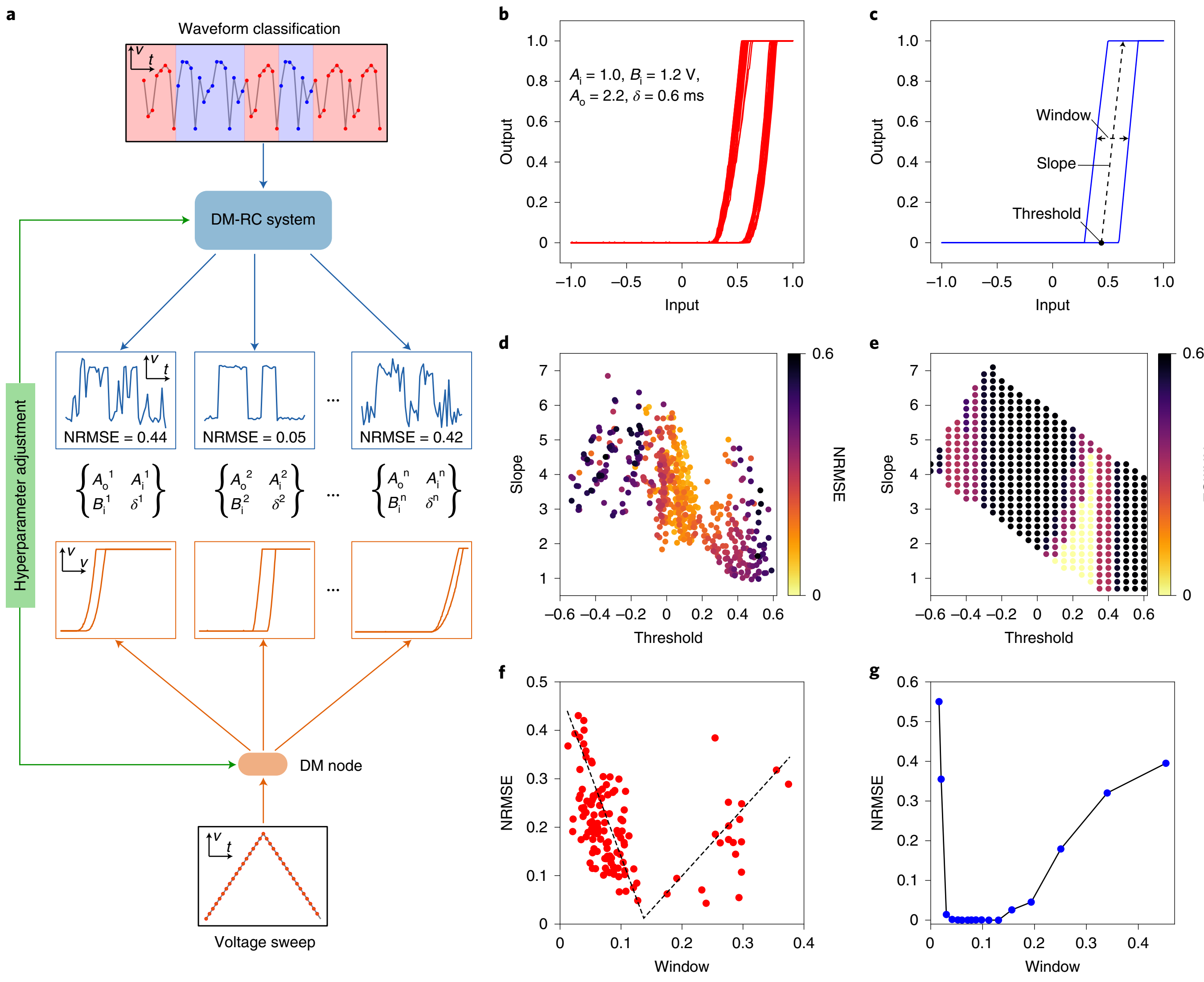

Fig. 3 | Hyperparameter adjustment for improving the reservoir performance. a, Experimental approach for exploring the relationship between the DM node behaviour and system performance, where the DM node behaviour is characterized by the *I*–*O* curve and the system performance is measured by the NRMSE on a waveform classification task. Different hyperparameters (A_i, B_i, A_o and δ) are used when obtaining waveform classification results and the corresponding *I*–*O* curves. **b**, Experimental results of the normalized *I*–*O* curves under a set of hyperparameters ($A_i = 1.0$, $B_i = 1.2$ V, $A_o = 2.2$ and $\delta = 0.6$ ms), where the input voltage is normalized to [−1, 1] and the output voltage is normalized to [0 1]. **c**, Simulated results of the normalized *I*–*O* curves under a set of simulation parameters, where three features (*T*, *S* and *W*) are extracted for subsequent analysis. **d**, Experimental results of system performance changing with the threshold and slope (extracted from the normalized *I*–*O* curve). **e**, Simulation results of system performance changing with the threshold and slope (simulation parameters). **f**,**g**, Experimental (**f**) and simulated (**g**) results of system performance changing with the window size.

To improve the system robustness to noise (which is mainly derived from the stochastic nature of NVM devices), here a noise-aware linear regression method is used to train the output weight W_{out}. We first combine the reservoir states at all the 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 symbol '†' represents the Moore–Penrose pseudo-inverse[33,41]. 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. The noise level of 0.04 is estimated from the experimental data of NVM devices (Methods). In such a simulated test process, the system output Y can be calculated by $Y = (W_{out} + \max(|W_{out}|)N_{te})X$, where the term $\max(|W_{out}|)N_{te}$ is used to quantify the noise of W_{out}. A more detailed description of the noise-aware training method can be found in Supplementary Fig. 8. The dependence of training error and simulated test error on different σ values during the training process is shown in Fig. 4e. Evidently, as the noise level σ increases, the training error increases slowly whereas 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} value. 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 and 33 µS. A closed-loop programming scheme is applied here to write the device conductance to the target value (Methods).

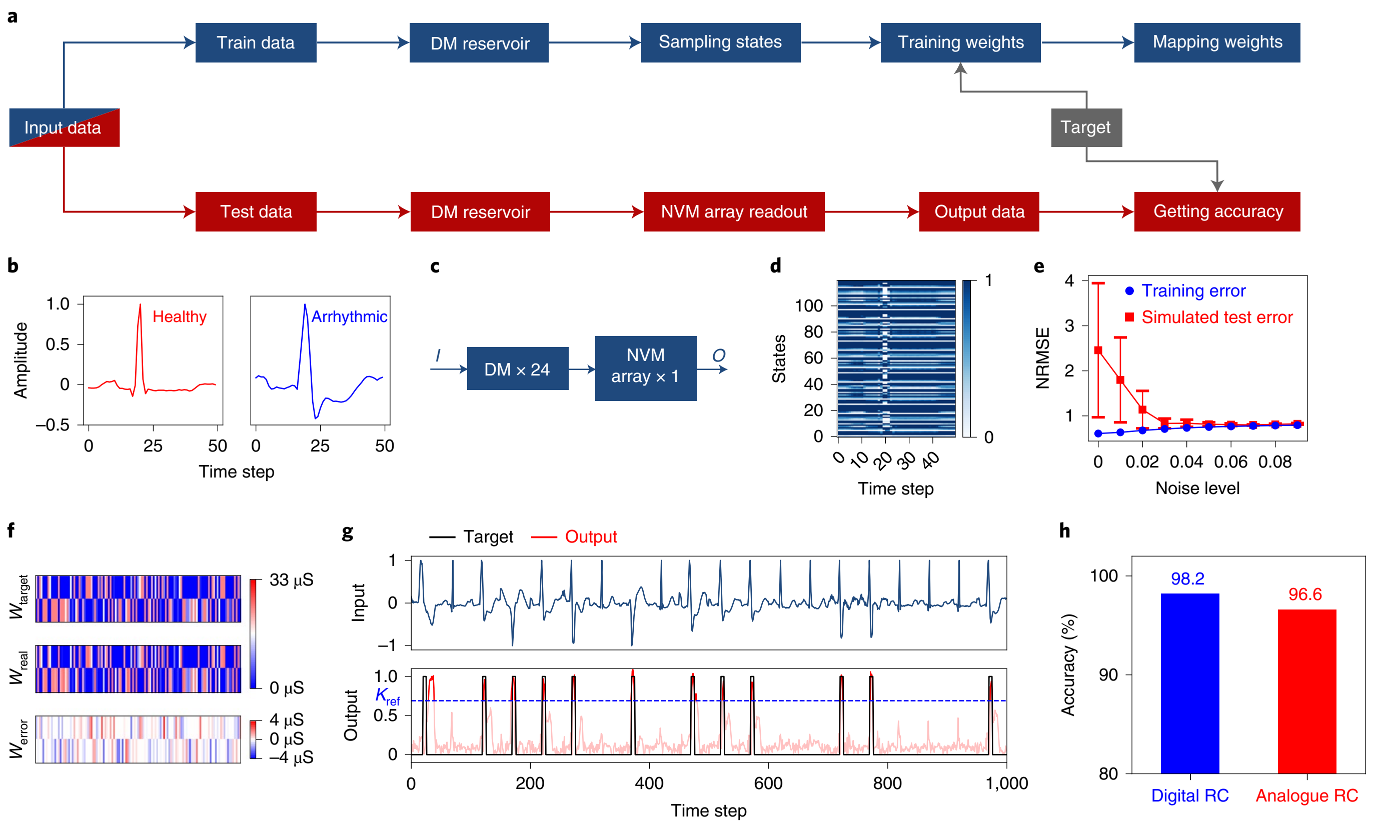

Fig. 4 | Demonstration of arrhythmia detection with the DM-RC system. **a**, Schematic of the training and testing processes in the DM-RC system. **b**, Two classes (healthy and arrhythmic) of heartbeat signals selected from the MIT-BIH arrhythmia database. **c**, Network structure used for the arrhythmia 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 an NVM array for readout. **d**, Normalized reservoir states corresponding to the input signal. **e**, Weight training process using linear regression as a function of noise, where the training 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**, Input (combination of healthy and arrhythmic 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**, Comparison of the classification accuracy between digital and fully analogue RC systems, where the accuracy loss of our fully analogue DM-RC system is only 1.6%.

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. Evidently, the trained weights are well mapped with relatively small errors owing to the excellent analogue switching characteristics of our memristor devices[42].

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 to −0.2 V) are then generated by the analogue 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 the 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]. Evidently, the output signal can effectively fit the target signal and generates a peak when an arrhythmic heartbeat occurs. Finally, a reference value K_{ref} (Fig. 4g, 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. The average classification accuracy obtained in our experiments is 96.6%, which is close to the software baseline of 98.2% achieved by a fully digital RC system (Fig. 4h). This result demonstrates the feasibility of using the 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 analogue 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 analogue computing with a DM-RC system for dynamic gesture recognition is shown in Fig. 5a. When the hand performs a certain gesture, the three-dimensional analogue signals from the acceleration sensor are directly fed into the DM-based reservoir module, out of which the generated reservoir states are calculated in the NVM array-based readout module in a fully analogue 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) (Fig. 5a) are recorded at a sample frequency of 10 Hz. Typical signals with a normalized amplitude (−1 to 1) are displayed in Fig. 5b. Similar to the previous arrhythmia detection task, the recorded sensor signals are split into single gestures of 3 s (that is, 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 remaining 300 samples for testing. Figure 5c shows the implemented network structure by reconfiguring

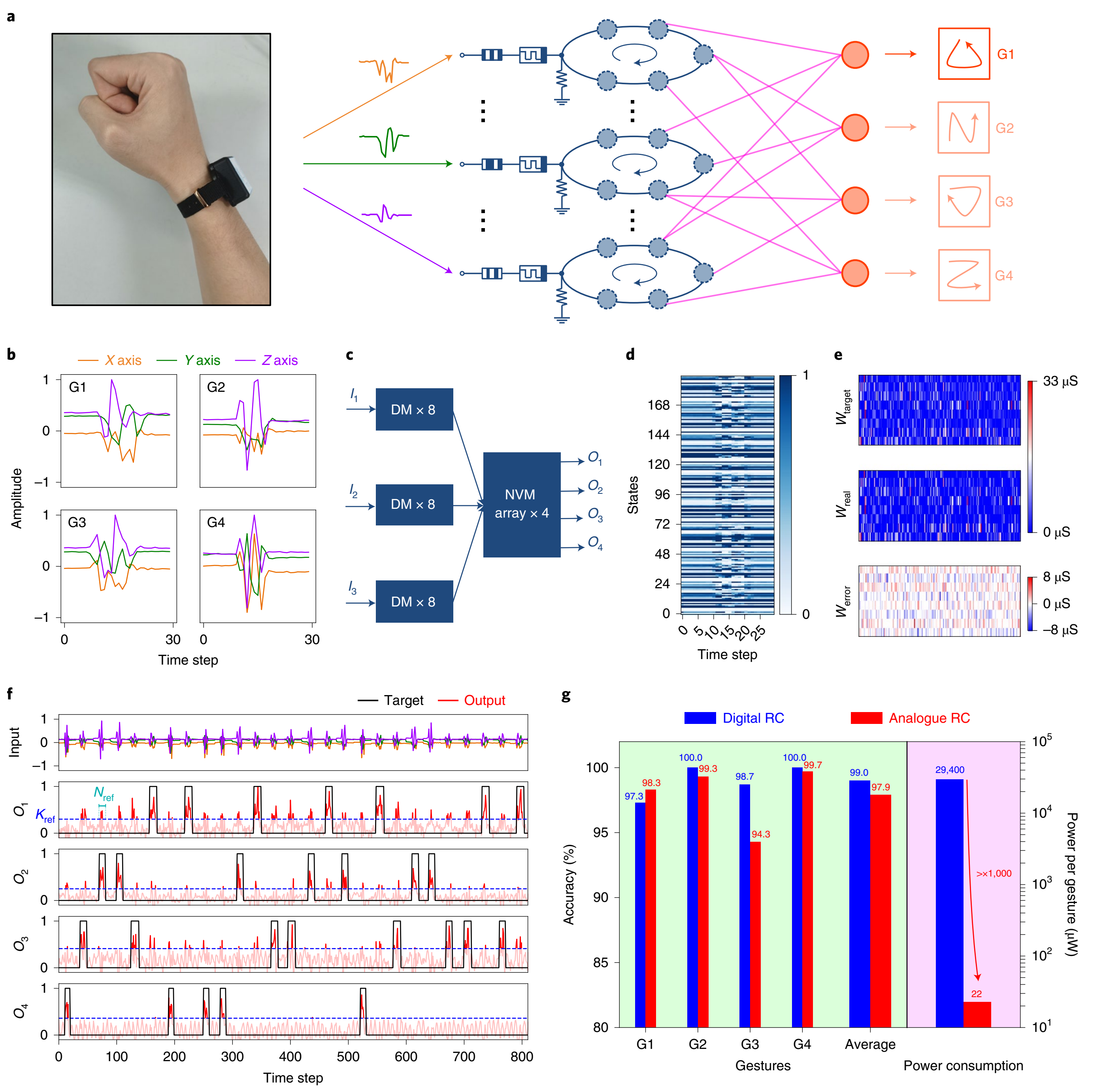

Fig. 5 | Demonstration of dynamic gesture recognition with the DM-RC system. a, Conceptual schematic of fully analogue computing with a DM-RC system in a dynamic gesture recognition task. The arrows indicate the direction of hand movement. **b**, Four classes (G1, equilateral triangle; G2, capital letter N; G3, inverted triangle; G4, capital letter Z) of dynamic gesture signals collected from a three-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 three groups of DM nodes (each group has eight DM nodes) and the output signals of the reservoir module are simultaneously input to four NVM arrays. **d**, Normalized reservoir states corresponding to a gesture signal. **e**, Distributions (192 × 8) of the target conductance weight, mapped conductance weight and weight-mapping error compared with the target values. **f**, Input (combination of four classes of gesture signals) and output signals of the DM-RC system, where all the sampled outputs of the four integrators are normalized to [0, 1]. Two reference values (K_{ref} and N_{ref}, shown as dashed lines) are 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, same as the previous arrhythmia detection task. **g**, Comparison of the classification accuracy and power consumption between digital and fully analogue RC systems, where the overall reduced accuracy and power consumption for our DM-RC system are 1.1% and 99.9%, respectively.

the DM-RC system, where 24 DM nodes are equally divided into three groups to process the input signals from three channels and four NVM arrays are used to classify four different types of gesture. The training and testing processes are also similar to those in the previous task. The outputs of the 24 DM nodes are sampled as the reservoir states (Fig. 5d). Here the mask sequence length is set to be 8; therefore, each DM node can produce eight reservoir states in one time step, and the final number of reservoir states reaches 192 per time step. The result of weight mapping is shown in Fig. 5e, which displays the distributions (192×8) of the 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 Fig. 5f, where all the outputs of the four integrators are normalized to [0, 1]. Here we use two reference values, namely, 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 K_{ref} and count the number of times (n) 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 the signal is detected as true; otherwise, it is false. The flow chart of the above process is shown in Supplementary Fig. 9. The values of K_{ref} and N_{ref} are searched within a certain range (that is, $K_{\mathrm{ref}} \in [0.21, 0.80]$ and $N_{\mathrm{ref}} \in [1, 10]$) to find the optimal ones. Different sets of K_{ref} and N_{ref} values are used to obtain the classification accuracy and we choose the set of values that yields the highest accuracy. 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 (Fig. 5g). Evidently, 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 analogue signal transmission and processing realized in our system such that the a 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 the hardware cost but also the power consumption.

Conclusions

We have reported a fully analogue RC system that is fabricated 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 by analysing the *I*–*O* curves and system performance under different hyperparameters. We find two key features—threshold and window—that have a notable impact on the quality of DM-based reservoirs, and the optimal system performance occurs at a specific threshold when the window is within a suitable range. By adjusting the hyperparameters, the DM-RC system can perform different spatiotemporal signal processing tasks in real time with low power consumption.

Compared with previous works[21,32,34,35,43,44], the power consumption of our DM-RC system is lower (Supplementary Table 2 provides a detailed comparison) due to the fully analogue signal transmission chain and the use of parallel DMs. To further reduce the power consumption and computing latency in the system, the entire DM-RC system could be miniaturized and monolithically integrated on chip, as the technologies used in our work are complementary metal–oxide–semiconductor compatible and scalable. A more sophisticated RC system could also be constructed using our DM-RC architecture as a basic unit—this would further enhance the system performance due to richer reservoir states and better memory capacity. Our work offers a promising platform for edge computing with low power consumption, and could be of use in applications such as smart wearable devices and microrobots.

Methods

Device fabrication

The DM devices were fabricated with a Pd/TiO_x/Ti crossbar structure on a Si substrate with 200 nm thermally grown SiO_2 on it, and the device size was 5 µm × 5 µm. First, 50-nm-thick Pd was electron-beam evaporated and patterned as the bottom electrode. Then, the functional 80-nm-thick TiO_x was deposited by reactive sputtering in the mixed atmosphere of Ar and O_2. The sputtering power was 400 W, the process pressure was 10 mtorr and the O_2 partial pressure was 17%. Finally, about 110-nm-thick Ti was electron-beam evaporated and patterned as the top electrode. The NVM array was fabricated using a standard 0.13 µm complementary metal–oxide–semiconductor process. The NVM array had a 1T1R cell structure in which the NVM material stack was TiN/TaO_x/$HfAlO_y$/TiN. The device size was 0.5 µm × 0.5 µm. The 8-nm-thick $HfAlO_y$ resistive switching layer was deposited by atomic layer deposition. The 45-nm-thick TaO_x thermal enhanced layer was deposited by sputtering. The top and bottom electrodes were both 30-nm-thick TiN deposited by sputtering.

Mask process

The mask process in our system is a time-multiplexing method that can increase the number of reservoir states by dividing the original time step into N small steps. Usually, the mask is a binary sequence of length N composed of −1 and 1, which is repeatedly multiplied to the input signal at each time step. The masked input signal is then applied to the DM node to generate the reservoir states. To obtain different reservoir states from each DM node, the mask sequences used here must be different from each other. Since the mask sequences are binary, the number of different mask sequences can be calculated as 2^N. In this work, we randomly select 24 of these mask sequences for the DM-RC system. In this way, the mask process using different mask sequences can enhance the richness of the reservoir state and system performance.

Measurement setup

The entire test system mainly consists of three parts, namely, a host personal computer (PC), an STM32 (MCU) development board and a user-customized PCB as the test board. The host PC operates as the upper computer for data loading, command sending and user interface coded in Python. The STM32 operates as the lower computer to communicate with the upper computer and generate the specific control signals that are transmitted to the test board through the general-purpose input/output ports. In addition, the STM32 board is also responsible for generating and reading voltage signals through the integrated 12-bit digital-to-analogue converters and ADCs, respectively. It needs to be pointed out here that 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 four NVM chips, 24 DMs (connected by a probe card) and several commercial chips such as low-dropout regulators, MUXs, amplifiers and digital potentiometers. Low-dropout regulators are used to provide stable positive and negative d.c. voltage sources for other chips in the test board. The MUXs are used to control the signal flow in the system and realize the mask process. All the MUXs used in our system are analogue MUXs with a digital control port. More specifically, we used CD4053 chips (Texas Instruments) as the single-pole-double-throw analogue switches to realize the mask process, where the mask signal is directly fed into the digital pin of the CD4053 MUX. Amplifiers are used to adjust the voltage signals and realize TIAs and integrators. Digital potentiometers 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

$$V_{\mathrm{o}}(k) = f\left(S\left(V_{\mathrm{i}}(k) - V_{\mathrm{t}}(k)\right)\right), \quad (1)$$

where $V_{\mathrm{i}}(k)$, $V_{\mathrm{o}}(k)$ and $V_{\mathrm{t}}(k)$ are the input, output and threshold voltages at the kth time step, respectively, whereas S and f are the slope of the *I*–*O* curve and nonlinear function, respectively. Here f is defined as

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \le x \le 1 \\ 1, & x > 1 \end{cases} \quad (2)$$

In equation (1), V_t changes dynamically with 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 a linear filtering process is defined as

$$V_t(k+1) = (1+\alpha)V_t(k) - \alpha(V_i(k) - 2T), \tag{3}$$

where α is the filter weight that determines the timescale of the system memory and T is the initial value of V_t that 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 using equation (3) with a dual-sweep sequence of V_i. The simulation of such a simplified DM node model is performed in Python3.8 with the Numpy package by solving equations (1)–(3). The simulation results suggest that other devices can also be used to implement the DM node for RC as long as the behaviour of such a DM node can be modelled by the above three parameters, which may need to be optimized for specific devices based on their device characteristics.

Weight-mapping process

We use a closed-loop programming scheme to map the target weight (W_{target}) onto the device conductance of the NVM array[9]. The basic loop of such a programming scheme is run in the PC and coded in Python. In the PC program, there are four 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 $\Delta W = W_{\mathrm{real}} - W_{\mathrm{target}}$ and the maximum error weight $\Delta W_{\max}$ is used to compare with ΔW to determine the next step of programming. (1) If $|\Delta W|$ is smaller than $\Delta W_{\max}$, then W_{target} is considered to be successfully mapped and the programming step is finished. (2) If ΔW is smaller than $-\Delta W_{\max}$, 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 $\Delta W_{\max}$, 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 5.0 V are applied to the SL and WL of the 1T1R cell, respectively. Steps (2) and (3) above are repeated until the target weight W_{target} is successfully mapped. As evident from the above weight-mapping process, the smaller the $\Delta W_{\max}$ value is, the closer the value of W_{real} is to W_{target}. In practice, the minimum value of $\Delta W_{\max}$ is limited by the intrinsic noise of the NVM devices, below which the mapping procedure would be difficult to converge. In our experiment, the minimum value of $\Delta W_{\max}$ is found to be $0.04 \times \max(W_{\mathrm{target}})$. Therefore, we use 0.04 as the noise level to take into consideration both mapping deviation and noise of NVM devices.

Data availability

Data that support the findings of this study are available from the corresponding authors upon reasonable request. Source data are provided with this paper.

Code availability

The code that supports the DM-RC system simulations in this study is available via GitHub 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.

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. Photon.* **15**, 102–114 (2021).
18. Jaeger, H. The 'echo state' approach to analysing and training recurrent neural networks—with an erratum note. *Bonn., Ger.: Ger. Natl Res. Cent. Inf. Technol. GMD Tech. Rep.* **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 Trans. Circuits Syst., II, Exp. 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.* **33**, 1688–1701 (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)* 36.3.1–36.3.4 (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 Eng. Med. Biol. Mag.* **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).

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 and 62104126) and the XPLORER Prize.

Author contributions

Y.Z. and J.T. conceived and designed the experiments. Xinyi Li contributed to device preparation and material analysis. Y.Z. performed the experiments and data analysis. Y.Z. and J.T. wrote the paper. All the 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 The online version contains supplementary material available at https://doi.org/10.1038/s41928-022-00838-3.

Correspondence and requests for materials should be addressed to Jianshi Tang or Huaqiang Wu.

Peer review information *Nature Electronics* thanks Xiaobing Yan and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.