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

In the format provided by the authors and unedited

Supplementary Information

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

Yanan Zhong[1,2,†], Jianshi Tang[1,3,†*], Xinyi Li[1,†], 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], Huaqiang Wu[1,3*]

[1]School of Integrated Circuits, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, 100084, China

[2]Institute of Functional Nano & Soft Materials (FUNSOM), Jiangsu Key Laboratory for Carbon-Based Functional Materials & Devices, Soochow University, Suzhou, Jiangsu, 215123, China

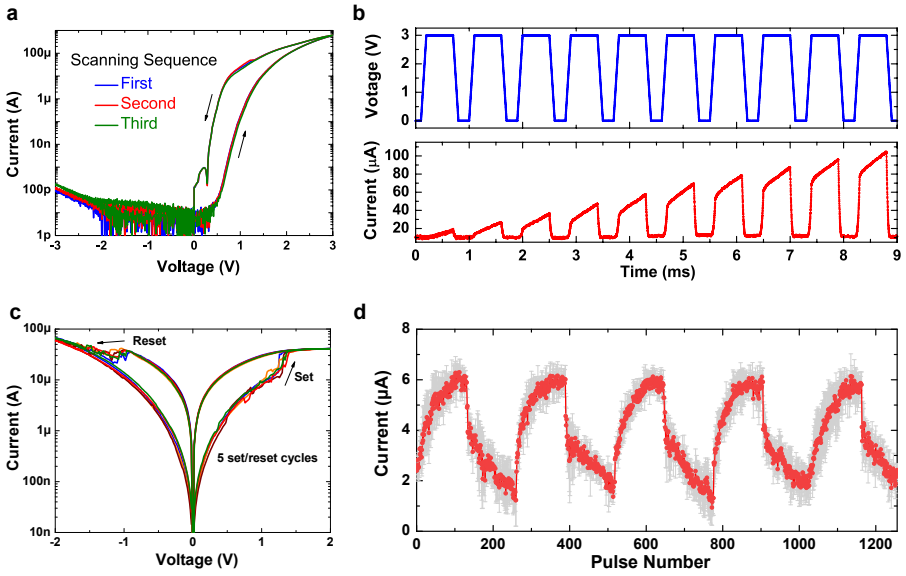
[3]Beijing Innovation Center for Future Chips (ICFC), Tsinghua University, Beijing, 100084, China

[4]Microelectronics Lab, James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom

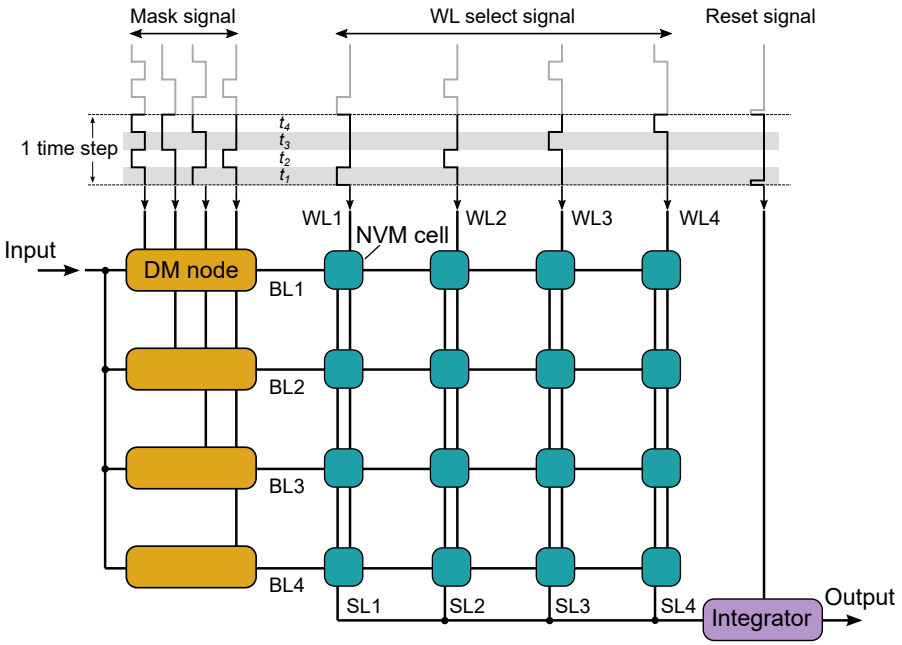
[*]Email: jtang@tsinghua.edu.cn, wuhq@tsinghua.edu.cn.

[†]These authors contributed equally: Yanan Zhong, Jianshi Tang and Xinyi Li

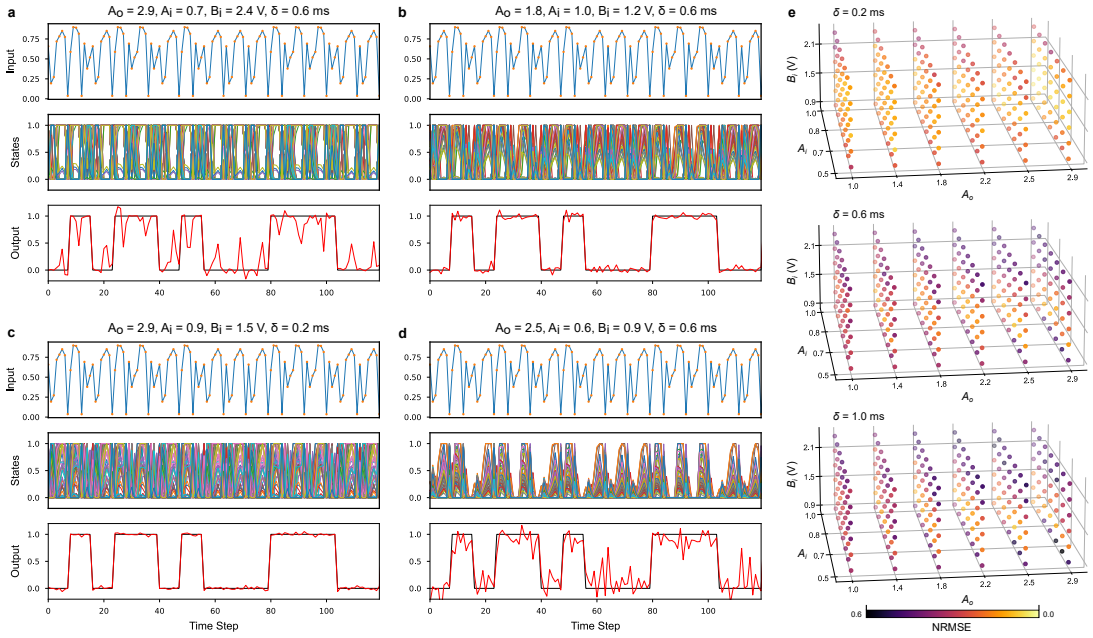
This **Supplementary Information** includes:



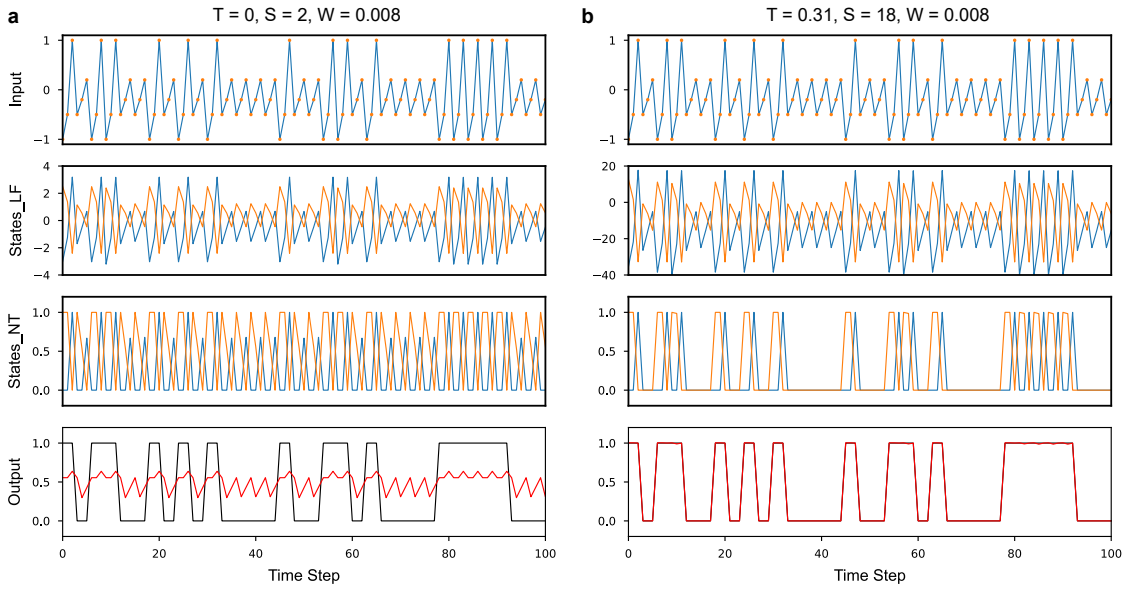
Supplementary Figure 1. **The electrical characterizations of two types of memristors.** (**a**) The typical I-V characteristic of the DM device. Three scans were repeated, and the arrows indicate the direction of the voltage scan. (**b**) The current response of the DM under a set of consecutive voltage pulses. (**c**) The typical I-V characteristic of the NVM device. The Set/Reset cycle is repeated 5 times, and the arrows indicate the direction of the voltage scan. (**d**) The typical analog switching characteristic of 5 NVM devices, where the red line plots the mean value and the grey line plots the variance. The set, reset and read pulses are (1.5 V, 50 ns), (-1.5 V, 50 ns) and (0.15 V, 4 ms), respectively.



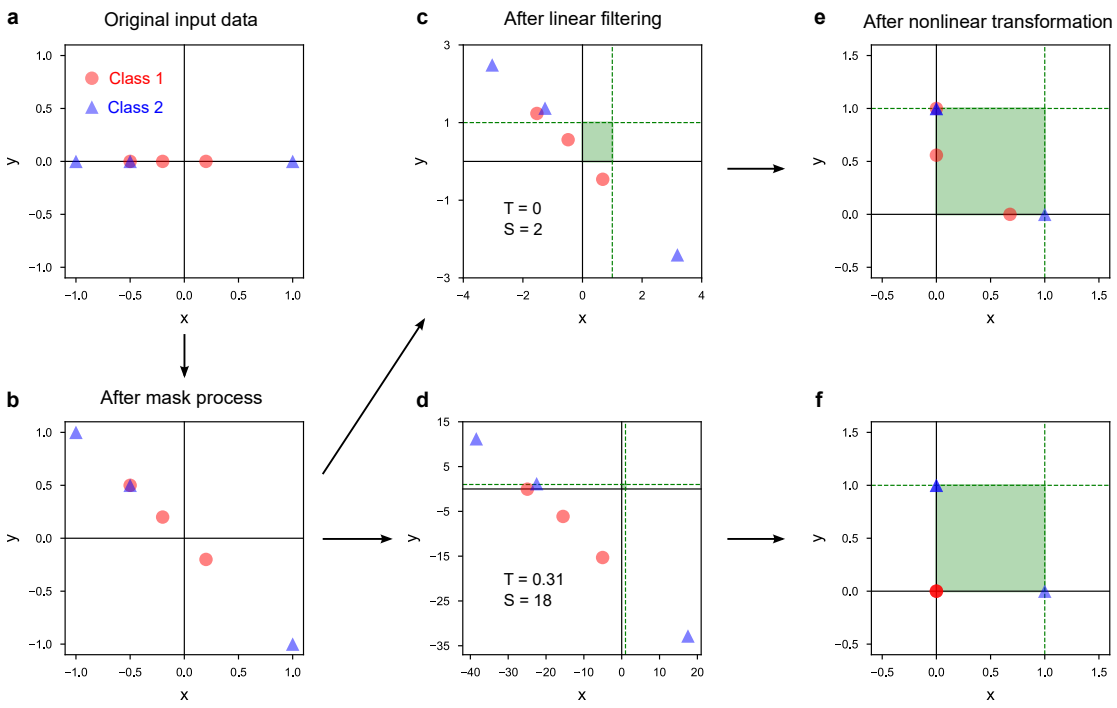
Supplementary Figure 2. **Schematic diagram of signal processing flow in the DM-RC system.** During one time step, the input signal is time-multiplexed by the mask process, which divides the original time step into N small steps (here N = 4 shown in the figure). Therefore, the output of the DM node (i.e., reservoir states) also has N small steps within one time step. The WL select signals are used to enable only one column of the NVM array at each small step. The SLs of N columns are connected to an integrator, which sums the MAC results of N columns at different times. The integrator is reset at the beginning of each time step, and its output is then sampled at the end of each time step.



Supplementary Figure 3. **Waveform classification with the DM-RC system**. (**a-d**) The waveform classification results when using different hyperparameters (A_i, B_i, A_o and δ). The first panel of each subfigure shows the input waveform which contains two types of random sequences, each consisting of 8 random numbers uniformly distributed between 0 and 1. The second panel of each subfigure shows the reservoir states which are sampled from the normalized current response of 24 dynamic memristors. The third panel of each subfigure shows the classification results, where the red and black lines represent the system output and target output, respectively. (**e**) The performance of DM-RC system under different sets of hyperparameters, where the system performance is measured by NRMSE on waveform classification task.

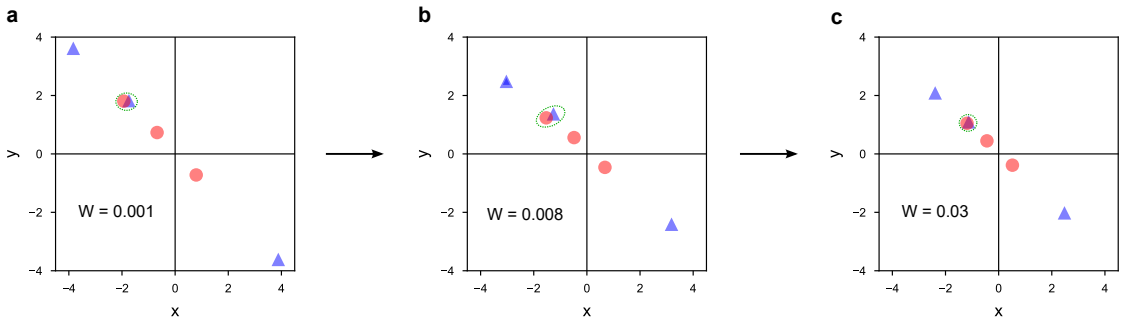


Supplementary Figure 4. **Three-point waveform classification simulated with DM model.** (**a-b**) The simulated results when using different values of threshold (*T*) and slope (*S*). The first panel of each subfigure shows the input waveform which is a random combination of sequence s_1: [-0.5, -0.2, 0.2] and s_2: [-1, -0.5, 1]. The second and third panels of each subfigure show the reservoir states obtained from the dynamic memristor model without and with nonlinear transformation, respectively. The fourth panel of each subfigure shows the classification results, where the red and black lines represent the system output and target output, respectively.

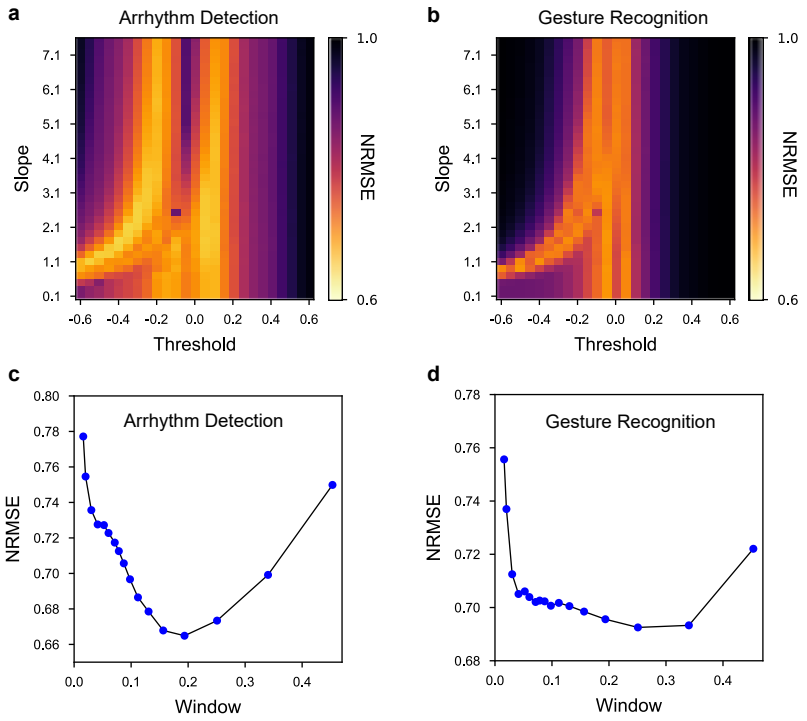


Supplementary Figure 5. Analysis of the working mechanism of temporal signal classification in the DM-RC system. (**a**) Two classes of original input data points, which are 1-D time sequence distributed along the x-axis. (**b**) The intermediate data points after the mask process, where the original 1-D data is projected into a 2-D space and distributed along the line $y = -x$ because of the mask sequence [1, -1]. (**c-d**) The intermediate data points after linear filtering, where the values of $\{T, S\}$ are set to be $\{0, 2\}$ and $\{0.31, 18\}$, respectively. The value of window size remains constant (W = 0.008). After linear filtering, the distance between data points of different classes will be enlarged. (**e-f**) The final results after nonlinear transformation of the results in (**c**) and (**d**), respectively. The nonlinear transformation moves the data points outside a particular rectangle (the green square area) to its edges or vertices, while the other points inside the rectangle remain stationary. In order to make them linearly separable, points in different classes would need to be moved to different vertices of the rectangle.

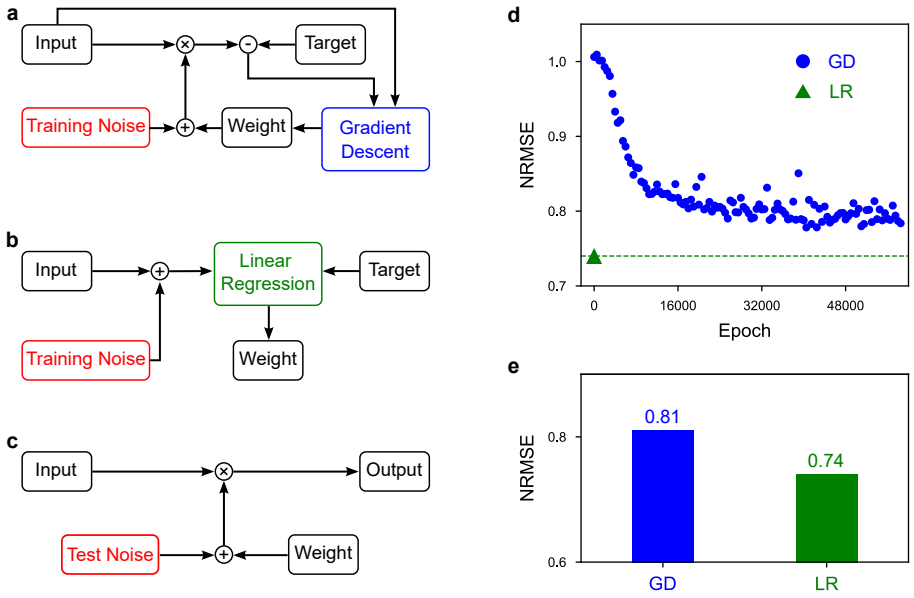
Therefore, the points in different classes before nonlinear transformation need distribute outside the rectangle and in different quadrants. By adjusting the node features *T* and *S*, the points before nonlinear transformation can be panned and zoomed in the 2-D plane. Compared with *S*, the value of *T* directly controls the position of the points before nonlinear transformation and determines whether they can be linearly separated after nonlinear transformation.



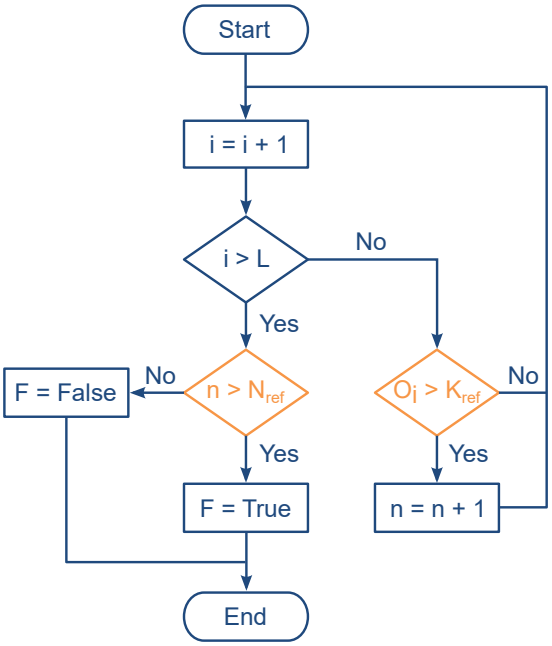
Supplementary Figure 6. **The effect of window size on the system performance.** The intermediate data after linear filtering when using different values of window (W) in the dynamic memristor model: (**a**) $W = 0.001$; (**b**) $W = 0.008$; (**c**) $W = 0.03$. A proper value of W can increase the distance between the feature points from different classes, which improves the classification performance of the DM-RC system.



Supplementary Figure 7. The simulated results of the DM-RC system performance changing with node features for different tasks. (**a-b**) Simulated system performance changing with the threshold and slop values on the arrhythm detection and dynamic gesture recognition tasks, respectively. (**c-d**) Simulated system performance changing with the window size on the arrhythm detection and dynamic gesture recognition tasks, respectively.



Supplementary Figure 8. Comparison of two training methods for the output layer in consideration of noise. (**a**) The training process of the output layer when using the standard gradient descent (GD) algorithm. (**b**) The training process of the output layer when using the linear regression (LR) algorithm. (**c**) The test process of the output layer. (**d**) Training results when using the GD (blue circle) and LR algorithms (green triangle) respectively. (**e**) Comparison of the final test results of these two methods.



Supplementary Figure 9. **The flow chart of obtaining the final classification result in the DM-RC system.** Here *i*, L, K_{ref}, *n*, N_{ref} and F are the *i*th time step, the length of one gesture signal, the reference value compared with output signal amplitude, the number of times when the output value is greater than K_{ref}, the reference value compared with *n* and the flag to judge whether the input signal belongs to the right class, respectively.

Module	Power per gesture (μW)
NVM	0.002
DM	19.2
Amplifier	2.68
MUX	0.32
Sum	**22.202**

Supplementary Table 1. Detailed power consumption of each circuitry module in the DM-RC system for the dynamic gesture recognition task. The power estimation for each module is explained as follows. For the NVM module, we assume that all devices in the array are in a high conductance state (G_{max} = 33 μS), while the input read voltage is at its maximum value (V_{max} = 0.2 V). Then we can estimate the maximum power consumption of the NVM module as $P_{NVM} = V_{max}^2 \times G_{max} \times N_{NVM} \times t_r \times f_s$ = 0.002 μW, where N_{NVM}, t_r and f_s are the number of NVM devices used in this task (N_{NVM} = 192×8), the pulse width of the input voltage (t_r = 100 ns) and the sampling rate of the input waveform (f_s = 10 Hz), respectively. For the DM module, given the power of single DM device of P_d = 50 μW[1], we can then calculate the power consumption of the whole DM module as $P_{DM} = P_d \times N_{DM} \times \delta \times L \times f_s$ = 19.2 μW, where N_{DM}, δ and L are the number of DM devices used in the task (N_{DM} = 24), the pulse width of the mask process (δ = 0.2 ms) and the mask length (L = 8), respectively. For the amplifiers, given the power of a single commercial amplifier of P_a = 8.8 μW, we can then calculate the power consumption of the whole amplifier module as $P_{AMP} = P_a \times N_{AMP} \times \delta \times L \times f_s$ = 2.68 μW, where N_{AMP} is the number of amplifiers used in the task. For the MUXs, the power consumption consists of two parts which are static power $P_{st} = V_s \times I_s \times N_{MUX}$ = 0.02 μW

and dynamic power $P_{dy} = V_s \times I_d \times N_{MUX} \times t_s \times f_s = 0.3$ μW, where V_s, I_s, I_d, N_{MUX} and t_s are the supply voltage ($V_s = 5$ V), static current ($I_s = 40$ pA), dynamic current ($I_d = 100$ μA), the number of transmission gate ($N_{MUX} = 120$) and the switching time ($t_s = 1$ ns), respectively. Thus, the total power of MUXs can be calculated as $P_{MUX} = P_{st} + P_{dy} = 0.32$ μW. As a result, the total power consumption for the DM-RC system is $P_{DM\text{-}RC} = P_{NVM} + P_{DM} + P_{AMP} + P_{MUX} = 22.202$ μW. In comparison, we choose Intel Core i5-4460T CPU for estimating the power consumption of a typical digital RC system. The total power of the CPU is 35 W and the CPU usage and running time when recognizing 900 dynamic gestures are 27 % and 8.4 s, respectively. Thus, the power consumption of the digital RC can be calculated as $P_{digital} = 35$ W × 27 % × 8.4 s / 900 / 3 s = 29400 μW, where 3 s is the time scale of single dynamic gesture.

Works	Reservoir implementation	Total states	Readout implementation	Tasks	Power consumption
Brunner et. al.[2]	Optoelectronic	388	On computer	Spoken digit recognition	150 W (whole system)
Alomar et. al.[3]	FPGA	50	FPGA	Waveform classification	83 mW (whole system)
Moon et. al.[4]	DM (×50)	400	On computer	Spoken digit recognition	300 μW (single input)
Yu et. al.[5]	FTJ (×28)	112	RRAM (binary, 196×64×256×10)	Handwritten digit classification	70 μW (single input)
Milano et.al.[6]	NW networks	3	RRAM (analog, 4×4)	4 × 4 pattern recognition	75 mW (single input)
Liang et.al.[7]	Analog circuits	64	RRAM (analog, 64×10)	Handwritten letter recognition	32.7 μW (whole system)
This work	DM (×24)	192	RRAM (analog, 192×8)	Dynamic gesture recognition	22.2 μW (whole system)

Supplementary Table 2. Comparison with literature-reported reservoir computing (RC) systems.

Supplementary reference

1. Zhong Y. *et al.* Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun.* **12**, 408 (2021).
2. 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).
3. 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).
4. Moon J. *et al.* Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2**, 480-487 (2019).
5. 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).
6. 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).
7. Liang X. *et al.* Rotating neurons for all-analog implementation of cyclic reservoir computing. *Nat. Commun.* **13**, 1549 (2022).