AFRL-RI-RS-TR-2018-248

MEMRISTOR CROSSBAR ARRAYS FOR ANALOG AND NEUROMORPHIC COMPUTING

UNIVERSITY OF MASSACHUSETTS AMHERST

OCTOBER 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2018-248 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
JOSEPH A. CAROLI
Work Unit Manager

/ S /
JOHN MATYJAS
Technical Advisor, Computing
& Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
OCTOBER 2018	FINAL TECHNICAL REPORT	JAN 2015 – MAR 2018

4. TITLE AND SUBTITLE

MEMRISTOR CROSSBAR ARRAYS FOR ANALOG AND NEUROMORPHIC COMPUTING

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-15-2-0044

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

J. Joshua Yang

5d. PROJECT NUMBER
T2BC

5e. TASK NUMBER
MA

5f. WORK UNIT NUMBER
SS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Massachusetts Amherst
Commonwealth Ave.
Amherst, MA 01003-9292

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITB
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2018-248

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This is the final technical report of a project titled Memristor Crossbar Arrays For Analog and Neuromorphic Computing. In this report, we summarize the results of the project on neuromorphic computing using memristive devices and crossbar arrays, including level-based analog computing accelerators and spike-based neuromorphic networks.

15. SUBJECT TERMS
Artificial Neural Network Models, Inference Models, Level-Based Computing, Neuromorphic Computing Algorithms, Spike-Based Computing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					JOSEPH A. CAROLI
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	35	N/A

TABLE OF CONTENTS

LIST OF FIGURES ii

1.0. SUMMARY 1

2.0. INTRODUCTION 2

3.0. METHODS ASSUMPTIONS AND PROCEDURES 2

4.0. RESULTS AND DISCUSSION 2

4.1. LEVEL BASED COMPUTING 2

4.1.1. 128x64 Array Development. *3*

4.1.2. Inferencing and Processing. *5*

4.1.3. Supervised Learning. *10*

4.1.4. Reinforcement Learning. *12*

4.2. SPIKE BASED COMPUTING 14

4.2.1. Diffusive Memristor Development. *15*

4.2.2. Artificial Synapse. *16*

4.2.3. Artificial Neurons. *19*

4.2.4. Fully Memristive Neural Network. *21*

5.0. CONCLUSIONS 25

6.0. REFERENCES 26

APPENDIX A – LIST OF PUBLICATIONS 28

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS 30

LIST OF FIGURES

Figure 1 Chip Layout Design For Integrated 1T1R Array In Different Sizes, Discrete 1T1R Device, Discrete 1T And Discrete 1R, Respectively 3

Figure 2 Illustration of the back-end process for the integration of memristors on the upper layer 4

Figure 3 Images of the integration fabrication result for the (a) 64×64 and (b) 4×4 arrays (c) The design layout for the 4×4 1T1R array 5

Figure 4 Cover article of inaugural issue of Nature Electronics using our memristor array chips. 5

Figure 5 Data stored in a 128×64 1T1R memristor crossbar, demonstrating conductance state linearity, write precision and accuracy, and read stability and reproducibility. 6

Figure 6 Experimental convolution demonstration with differential memristor conductance pairs. 9

Figure 7 Memristive platform for in-situ learning 11

Figure 8 In-situ online training experiment on MNIST handwritten digit recognition. 12

Figure 9 In-memristor reinforcement learning in the cart-pole environment. 13

Figure 10 Timing mechanism of SiOxNy:Ag diffusive memristor 16

Figure 11 Long-term spike-rate-dependent potentiation and bio-realistic spike-timing dependent plasticity (SRDP and true STDP) behavior of a combined device consisting of a diffusive and a drift memristor. 18

Figure 12 Diffusive memristor artificial neuron. 20

Figure 13 Fully integrated memristive neural network for pattern classification 23

Figure 14 Unsupervised training of a fully connected network based on the integrated all-memristive neural network. 24

1.0. SUMMARY

Existing technologies for the current computing systems are approaching their physical limits, and novel computing architecture and concepts are required to fulfill the needs of real-time Intelligence, Surveillance and Reconnaissance (ISR) systems for information processing with both high power efficiency and high speed. Accordingly, memristor based crossbar arrays for analog and neuromorphic computing have been studied in this program. Particularly, our team at the University of Massachusetts Amherst have designed and fabricated of memrsitors and crossbar arrays, with which we have demonstrated novel computing functions.

We have built memristor arrays for both level-based and spike-based computing engines for analog and neuromorphic computing, respectively. These two computing engines pose different challenges for memristor arrays. The analog computing engine requires that memristors have linear current-voltage relationships and can be accurately programmed to different resistance values. We have achieved 128 x 64 1-transistor-1-memristor crossbar arrays. With these arrays, we have successfully demonstrated accurate analog signal and image processing, efficient in-situ learning, reinforcement learning, and long short-term memory. In contrast, the neuromorphic computing engine demands synaptic dynamics to more efficient and faithfully emulate neuromorphic functions. We developed a new type of memristor, termed diffusive memristor, which possesses diffusion dynamics to mimic the synaptic dynamics. Such diffusive memristors have enabled bio-realistic synapses and neurons, with which we have constructed the first fully integrated memristive neural network. Pattern classification and unsupervised learning have been demonstrated with such neural networks. Based on the diffusive memristors, we have also created pseudo-memcapacitive devices, which were utilized to build the first fully capacitive neural network with associative learning and spike signal classification functions.

2.0. INTRODUCTION

The traditional computer systems based on the Von Neumann architecture are facing increasing challenges in the era of big data and autonomy. Completely novel computing architectures and concepts are required to process real time information with high power efficiency and high speed at the same time[1]. Accordingly, AFRL seeks to improve real-time Intelligence, Surveillance and Reconnaissance (ISR) systems and other important applications that require sophisticated high-volume data analysis by orders of magnitude in terms of computational throughput and power requirements.

In order to improve computational power efficiency and speed simultaneously, memristor array-based analog and neuromorphic computing engines have been studied in this program. The brain remains one of the greatest scientific mysteries. The traditional Si-CMOS based computing technology has revolutionized the world, but it is reaching its fundamental limit. In addition, the CMOS devices were not created or optimized for neuromorphic computing. Accordingly, in this program, we have built memristor-based crossbar arrays to demonstrate unconventional computing paradigms, including level-based and spike-based computing architectures.

3.0. METHODS ASSUMPTIONS AND PROCEDURES

The program goal was to build working memristor arrays for both level-based and spike-based computing engines. The level-based computing engine was mainly used to accelerate vector-matrix multiplications while the spike-based computing engine was designed for neuromorphic computing with more neuroscience principles. These two computing engines emphasize different properties of the memristor array. The former requires that memristors have linear current-voltage relation and can be accurately programmed to different resistance values. Array density is not the primary concern for it. In contrast, the latter requires diffusive dynamics. Therefore, the cell stack at each lattice point of the array were different for these two different applications. The sneak path current and half-selected issues necessitate a so-called “access device” at each crossing point in series with the memristor to electrically isolate the cell selected for programming or reading. We used the transistor as the access device for level based computing and developed a diffusive memristor as the selector for spike based computing.

4.0. RESULTS AND DISCUSSION

4.1. LEVEL BASED COMPUTING

The level-based computing is an analog computing engine (ACE). It is mainly used to accelerate vector-matrix multiplications, which are involved directly or indirectly in many important algorithms for deep neural networks (DNNs)[2]. These so-called computation intensive tasks can be implemented by the ACE with orders of magnitude improvement in power efficiency and speed, and thus benefits many computing algorithms.

The two major memristor properties required for ACE are a linear current-voltage relation and a large number of accurately controlled resistance levels. However, the size of a functioning crossbar array is usually small (e.g. $<32 \times 32$) if the memristors are linear because the sneak path current in the array increases with the array size and disables accurate programming operations

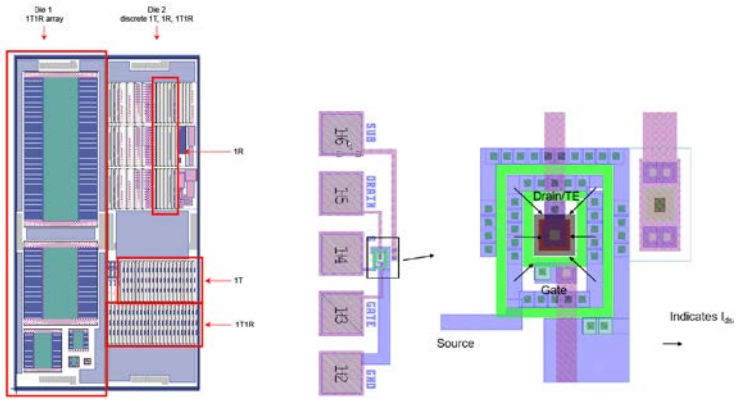
in the array. In this project, the dilemma was solved by adding a transistor in series with a memristor in each crossing point to electrically isolate the target memristor for programming, i.e., memristor switching. This is the 1-transistor-1-memristor (1T1M) scheme[3]. Vector matrix multiplication operation with memristor crossbar arrays is essentially the reading operation of memristors. Accurately programming memristors in the array to different resistance levels is a less frequent but a more challenging operation, especially when the array size is large (e.g. >32 × 32) and the memristors are linear. The transistors used in our 1T1M scheme not only maximized the array size by minimizing the sneak path currents but also maximized the number of the resistance levels of each memristor by providing well-controlled current compliance for the target memristor during programming. Due to the limit of external measurement systems (the number of probes in a probe card), the largest memristor array we designed was 128 x 64, which needs 388 probes in a probe card to build electrically contacts to the memristor array. During programming, the transistor in series with the selected memristor can be set to different current levels by applying different gate voltages, which serves as a current compliance for the memristor to be accurately switched to different resistance levels.

Different from the 1T1M scheme, in the 1-selector-1-memristor (1S1M) scheme, a two terminal nonlinear element, i.e., selector, was used in series with each memristor at the crosspoint. This leads to a nonlinear current-voltage relation for the selector-memristor combined cell and thus reduces the sneak path currents and increases the array size. This scheme is more conveniently used in spike based computing.

4.1.1. 128x64 Array Development.

For level based computing, we integrated Ta/HfO_2/Pt based memristors and transistor arrays to obtain the 1T1M arrays. The chip layout is shown in Figure 1a for 2 dies as the repeating unit on the wafer. The design has four parts: the left part (die 1) includes 6 different sizes of arrays, including 4×4, 8×8, 16×16, 32×32, 64×64 and 64×128. The right die contains discrete 1T1M device, discrete transistor (1T) and discrete memristor (1M) respectively in order to evaluate the difference between discrete devices and integrated devices in the array.

The detailed cell design is shown in Figure 1b for discrete 1T1R device. As we can see in the layout, a ring shaped gate was used, and it is to increase the width to length ratio and hence the output current, which is favorable for memristors when they require a large current to operate.



a b

Figure 1 Chip Layout Design For Integrated 1T1R Array In Different Sizes, Discrete 1T1R Device, Discrete 1T And Discrete 1R, Respectively

The front-end process was carried out in a commercial fab. The top interlayer dielectric is 1 μm thick TEOS silicon oxide. The contact vias was opened by dry etch in order to expose the underlying metal layer which is 500 nm thick AlCuSi alloy.

The following schematic in Figure 2 shows the process to integrate the memristors on the upper layer. First of all, a 2 nm Ti layer and a 20 nm Pt layer were deposited by e-beam evaporator as the adhesion layer and bottom electrode, respectively, and patterned by lift-off. In some cases, a layer of Ag (2-10nm) was used between the Ti and Pt layers in order to further the contact quality between the CMOS layer and the memristor electrodes. The samples were tilt by 45° and rotated during the deposition in order to cover the side walls of the 1 μm deep via. Following that, a switching layer of 5 nm HfO_2 was deposited by atomic layer deposition (ALD) at 250 °C. Reactive ion etch (RIE, with CF_3/O_2) was used to keep the switching layer in the device area to avoid possible crosstalk between devices. Finally, a top electrode of a 100 nm Ta and a 50 nm Pd layer were deposited by sputtering (Ta is easily oxidized).

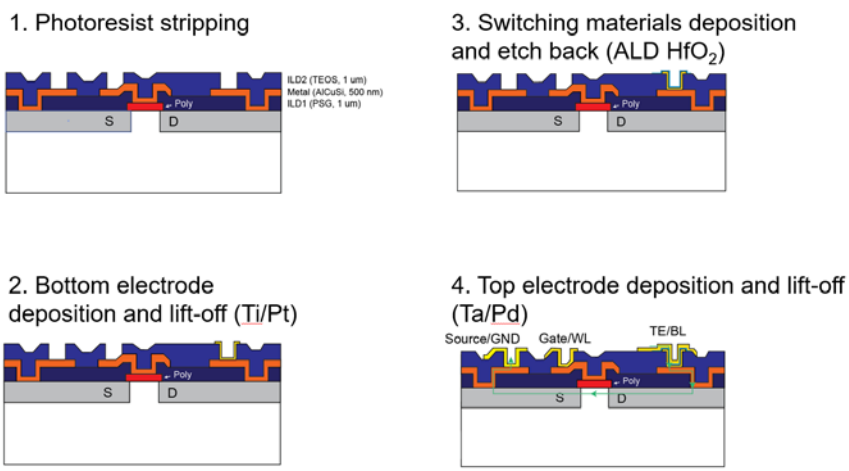
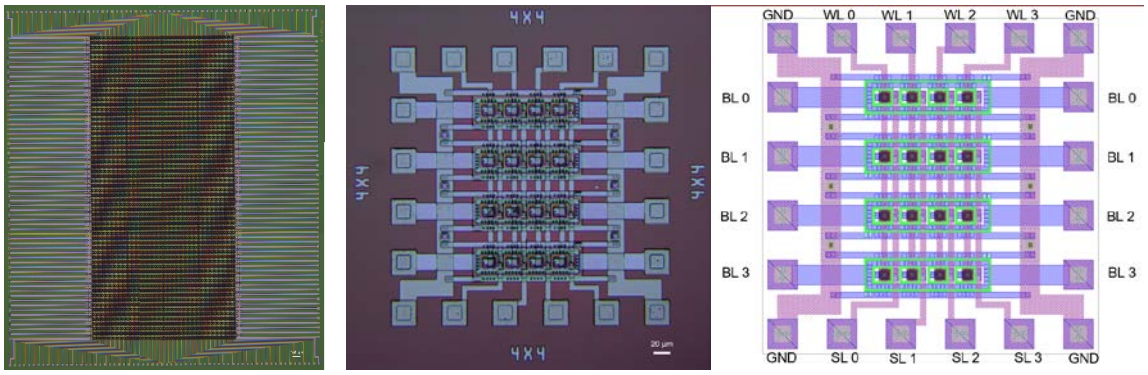


Figure 2 Illustration of the back-end process for the integration of memristors on the upper layer

The representative fabrication result for the 1T1R array is shown in Figure 3. The left image shows a large 64×64 array with no defect, and the right image shows the 4×4 array, from which one sees the detailed connections of the array. The left and right pads are connected with bit lines (Figure 3c), while top pads are with word lines except that the leftmost and rightmost ones are the shielding ground. The bottom pads are with source lines and the arrangement is the same as corresponding word lines. The resulted 128×64 memristor arrays were used for a number of analog computing demonstrations and selected as the cover image of the first issue of *Nature Electronics*[4], as shown in Figure 4.



(a) (b) (c)

Figure 3 Images of the integration fabrication result for the (a) 64×64 and (b) 4×4 arrays (c) The design layout for the 4×4 1T1R array

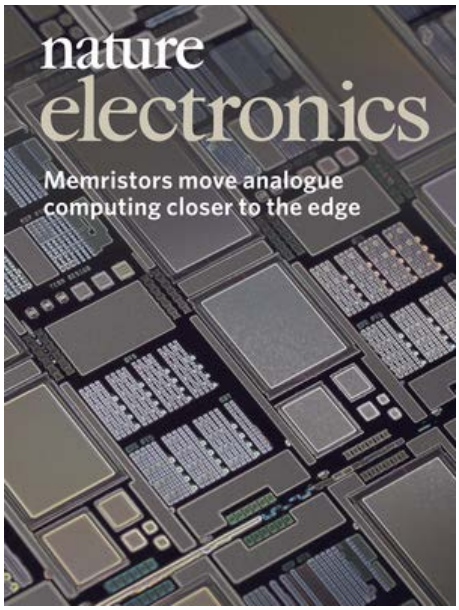


Figure 4 Cover article of inaugural issue of Nature Electronics using our memristor array chips

4.1.2. Inferencing and Processing.

Many of the inferencing and processing computations in DNNs can be expressed as a vector-matrix multiplication (VMM), which in principle can be performed in the analog domain by a memristor crossbar array using Ohm's law for multiplication and Kirchhoff's current law for summation[4]. Such VMMs are being developed as accelerators for DNNs, and they may also be used as reconfigurable analog processors for edge computing. A vector of voltage outputs from a sensor can be applied directly to the rows of a memristor crossbar, in which the values of the appropriate matrix elements have been stored as conductances of the cells. The currents that appear on the columns of the array in real time represent the output vector of the multiplication if the series resistance of the interconnection wires is negligible compared with the memristor resistances. To read out the results in parallel, the current signal from each column is converted to a voltage signal through a transimpedance amplifier (TIA), which also serves as a virtual ground.

To date, demonstrations of this concept have been limited to binary signal input and/or binary matrix weights. Recently, pulse width instead of amplitude was used to represent the analog input signals[5-8], but this scheme requires more readout time and more complicated integrated circuits. The previous experimental demonstrations of an analog-voltage-amplitude-vector by analog-conductance-matrix product, to the best of our knowledge, was limited to a 1×3 system, which is not strictly a VMM implementation. In this program, we have obtained a completely analog VMMs with adequate accuracy and high speed-energy efficiency that were based on up to 128×64 crossbars of hafnium oxide (HfO_2) memristors, and experimentally demonstrated the important IoT and network edge applications of signal spectrum analysis, image compression and convolutional filtering[4].

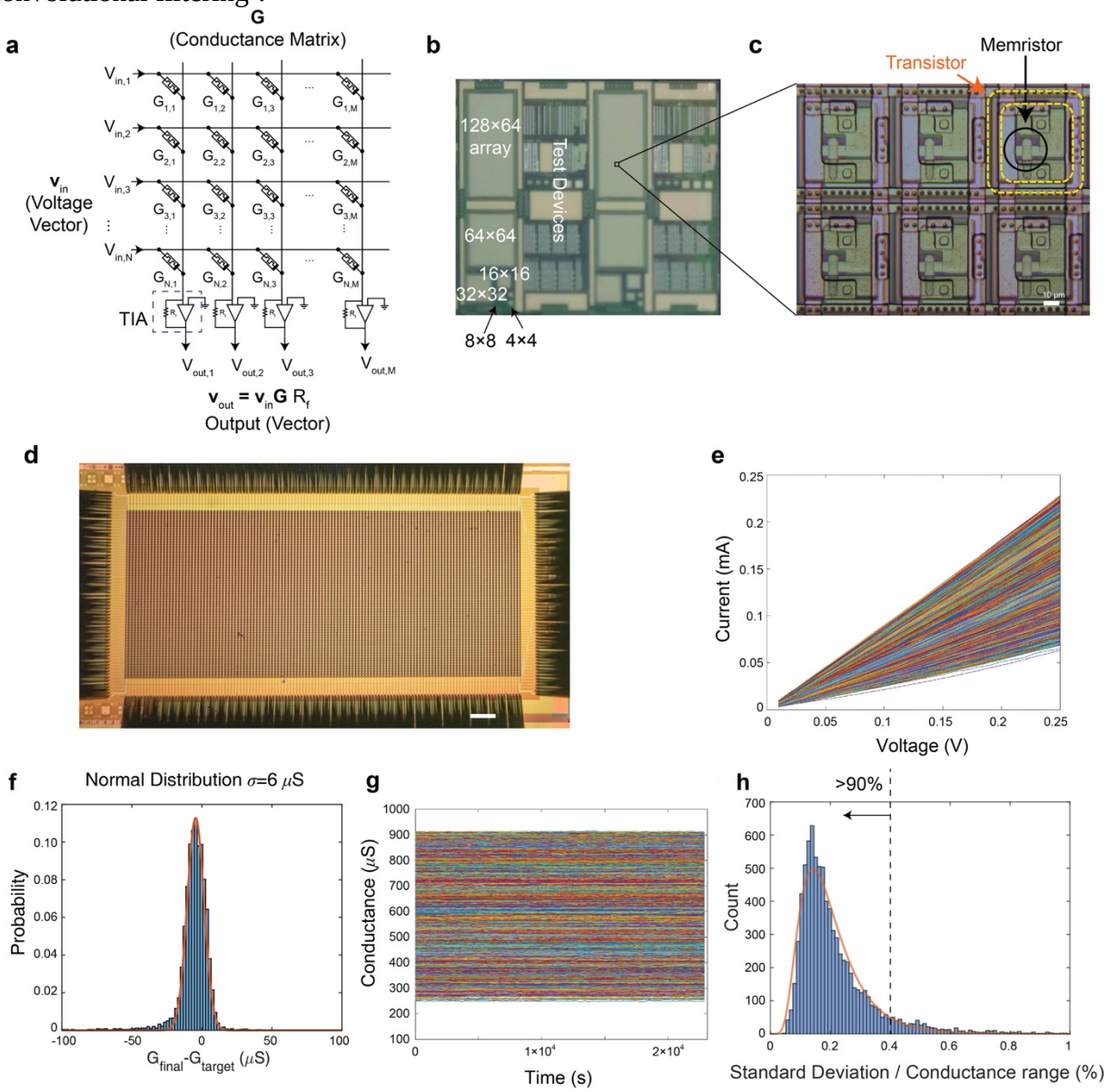


Figure 5 Data stored in a 128×64 1T1R memristor crossbar, demonstrating conductance state linearity, write precision and accuracy, and read stability and reproducibility a. Schematic of the VMM operation. Multiplication is performed via Ohm's law as the product of the voltage applied to a row times the conductance of a cross-point cell to yield a current injected onto a column, and the currents on each

column are summed according to the Kirchoff current law. The total current from each column is converted to a voltage by a TIA, which also provides a virtual ground for the column wires. b. A 2cm×2cm detail from a photograph showing 2 die of 1T1R memristor crossbars, each of which contained array sizes from 4×4 to 128×64 cells, along with various test devices. c. A microscope image of six cells in a 1T1R array, with a 10 μm scale bar. The crosses are memristors, and the transistors are ring-shaped. d. Photograph of a probe card in contact with an operational 128×64 1T1R array, with a 500 μm scale bar. e. Quasi-DC IV curves for all the devices, showing good linearity over the read interval. f. Histogram of the initial difference between the target and measured conductances written into a 128×64 array. A fit of the peak to a normal distribution yielded a standard deviation of 6 μS, with the peak maximum located at -5 μS. g. Room temperature state retention and read disturb of the device states. The DC conductance states of all the devices were measured with a 0.2 V bias for 1000 cycles, or a total of 6.4 hours, showing no discernable drift in the plots. h. Histogram of the normalized standard deviation (SD), defined as the SD per conductance range (100 – 900 μS), for all measured states, which was fitted to a lognormal distribution. It shows that there are fluctuations during the read operation that can occasionally degrade the effective precision of an individual memristor. 90% of the device states has normalized SD less than 0.39%.

To precisely tune the conductance of each memristor in a crossbar, we monolithically integrated a memristor on top of a metal-oxide-semiconductor (MOS) transistor as an access device in each cell, i.e., the '1T1R' architecture. The integration was conducted at UMass Amherst by building Ta/HfO_2/Pd memristors[9] on top of a CMOS chip fabricated by a commercial vendor. Figure 5b shows part of the integrated chip consisting of 1T1R arrays with sizes ranging from 4×4 to 128×64. The detailed structure of some cells is shown in Fig. 5c. The source wires of the transistors are rotated by 90 degree so that when all the transistors are turned on, the array converts into a fully connected memristor crossbar array. The programing and computing were achieved through a custom-built testing system connected to the chip by a probe card. Figure 5d shows a 128 × 64 array with probes touching the contact pads. With the 1T1R scheme, the array size can be much larger than 128 × 64, which was chosen for this demonstration mainly because of the constraint of the maximum number of probes (388 as shown in Fig. 5d) available on the commercial probe card used for the testing. With the transistors as access devices, we were able to program the conductance of most of the memristors to an arbitrary value within a pre-defined conductance range. We wrote MATLAB scripts to control the resistance tuning by communicating with the testing system. With the Ta/HfO_2/Pd memristors, the current-voltage (IV) relation of the cells was very linear once the conductance was larger than the quantum conductance (77.5 μS), as shown in Fig. 5e for conductances ranging from 300 to 900 μS, an important feature for accurate analog computing. Among the 8,192 devices in a 128 × 64 array, there were only 3 stuck ON and 15 stuck OFF devices after programming, leading to a responsive device yield of 99.8%. The histogram of the writing error, defined as the initial difference between the target conductance value and the measured written value of the responsive memristors, is plotted in Fig. 5f. The peak of the writing error conformed to a normal distribution with a standard deviation σ of 6 μS when the writing tolerance was set to ±10 μS, and could be further reduced by defining a narrower tolerance in the MATLAB script and/or using a larger number of closed-loop iterations, at the expense of increased programming time. If for the moment we discount the tail of the distribution, which represents a small number of 'sticky' cells, and define the interval between states as $\pm\sigma$, we effectively demonstrated more than 64 levels of conductance or 6 bits of digital precision over the conductance range 100-900 μS, which has been proven to be sufficient for many tasks in machine learning algorithms. The accuracy error δG of the memristor programing operation is taken to be the median value of the writing error, which was -4.7 μS. To explore

the read stability and reproducibility, we measured the conductances of the responsive 8,174 devices in the 128×64 array with 0.2 V read pulses for more than 6 hours and did not see any detectable state drift (Fig. 5g). There were fluctuations in the read operations of individual cells but these were small enough to have little impact for column current measurements over multiple memristors. This fluctuation is the limiting factor for the conductance writing precision given the zero writing tolerance and indefinite close-loop iterations. The statistics shows that 90% devices states have (Fig. 5g) fluctuation with normalized standard deviation less than 0.39%, providing the potential for the writing precision of 128 states or 7 bits in the conductance range of 100-900 μS.

We also experimentally demonstrated 2D convolution for image filtering. We employed 10 different convolutional filters: Gaussian, Disk and Average to smooth out noisy images, Laplacian of Gaussian (LoG) with three different parameters, Sobel (both x- and y- gradient) to extract the edges, and Motion (2 directions) to mimic the motion blur effect. We added artificial Gaussian white noise on the original 128×128 *Lena* image to show how the these convolutions damp out the noise and how well the edge location routines work. The noisy Lena image was used as the input, the image intensity of which was converted into voltages applied on the rows of the crossbar, as illustrated in Figs. 6a and 6b. Each pixel in the filtered image was generated by the dot product of the 25-dimensional voltage vector mapped from a 5×5 input sub-image and the 25-dimensional conductance vector mapped from a 5×5 convolution matrix. We scanned the 5×5 sub-image with a stride of one, and did not use zero-padding, so the dimension of the filtered images was 124×124 (=128-5+1). The negative values of the convolution matrices were mapped to memristor cell conductances by the differential approach described earlier, but the differential pairs were arranged in the neighboring columns rather than rows (Fig. 6b). Thus, the 10 different convolution maps were generated in parallel from 20 columns of current output. The experimental results are presented in Fig.6c, showing the performance of the crossbar in smoothing images and extracting the edges out of the images. The edge extraction described in this step is also the convolution layers of the convolutional neural networks (CNNs or ConvNets), which is the most computation expensive step in the networks. Compared to previously reported convolutions operating with binary inputs, binary weights and series readout, our image filtering procedure included both analog convolution matrices and analog inputs, as well as parallel readout of 10 features maps.

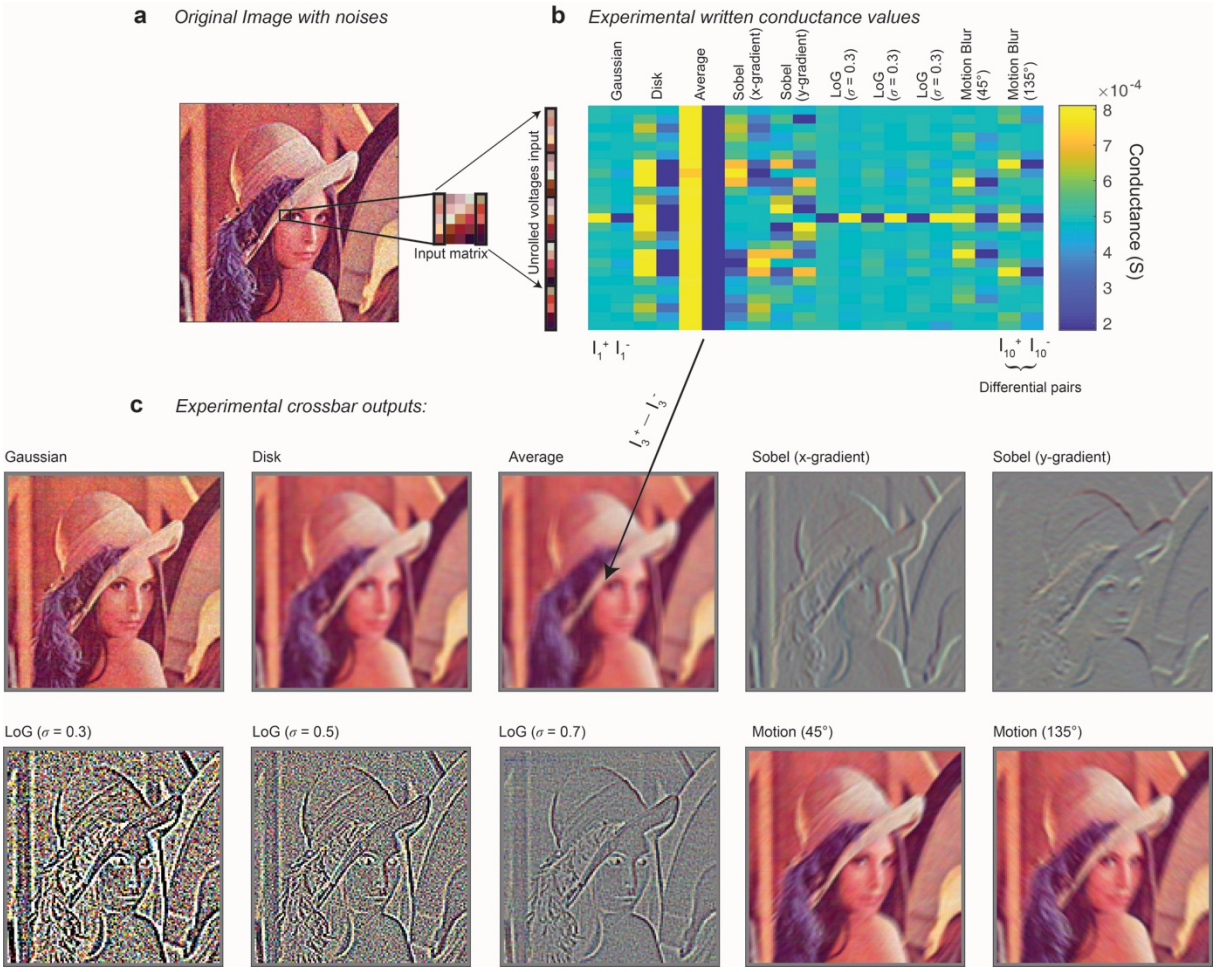


Figure 6 Experimental convolution demonstration with differential memristor conductance pairs a. The image input we used for the image filtering with crossbar. The image was standard Lena image with artificially added Gaussian white noise. Each color channel of the image is represented by a float point number between 0 to 1, while added noises are with standard deviation of 0.004 and mean of zero. b. Readout conductances after programming 10 convolution kernels into the 25×20 crossbar array. The pixel intensities, represented by two voltages with equal amplitude and opposite polarity, were input into the crossbar onto a pair of differential memristor conductances in adjacent columns representing one matrix element of a convolution. c. 10 different filtered images obtained in parallel by the convolution operation: Gaussian, disk and average reduce noise by smoothing the image, Laplacian of Gaussian with various parameters, Sobel (x- and y- gradient) were used to detect edges, and Motion to generate motion blur.

The key advantages of our hardware VMM are the reconfigurability of the memristor crossbar, the reasonable accuracy and precision of the physical computation, and the efficiency both in speed and energy consumption. We analyzed the performance and the energy efficiency of the system. Since one 128-dimensional *vector* and 128-by-64 *matrix* multiplication is done by single current read process on the column wires, so readout time within 10 ns gives 1.6 TOPS (Tera Operations Per Second). We performed a simulation of the power consumption for the image compression task with our experimental parameters, including readout conductances after programming, wire resistances and input patterns, and found the power consumed in the 128×64 crossbar array was about 13.7 mW, or an efficiency of about 119.7 TOPS/W. As a comparison, the highly optimized digital system with the application specific integrated circuit (ASIC) in 40

nm technology node for 4-bit 100-dimensinal *vector* and 4-bit 100-by-200 *matrix* multiplication, of which the accuracy is comparable with our solution, has the energy efficiency of 7.02 TOPS/W. While not a direct comparison, our system is 17× more energy efficient than the ASIC solution. More importantly, our memristor crossbar hardware VMM handles the analog signals acquired from the sensor directly, without the need of extra peripherals such as analog-to-digital converters (DACs), which is mandatory for the digital ASIC solution and consumes extra time and energy, but was not considered in above energy estimation calculation. The ADCs is also not always necessary if only specific features need to be detected within signals, which can be provided with threshold-gate circuits at much lower cost both in latency and energy. This feature, along with its low latency and energy efficiency, makes the demonstration of the system ideal for the edge computation.

4.1.3. Supervised Learning.

We used Ta/HfO_2/Pt memristors developed to achieve stable tunable multilevel behavior with a linear current-voltage (IV) relationship[10,11]. The memristors were monolithically integrated with foundry-produced transistor arrays on a six-inch wafer, as discussed before. Each memristor was connected with a series transistor in a '1T1R' configuration, also mentioned before (Figures 7a-e show the integrated memristor array from wafer scale to nanometer scale)[12]. To electronically increase the conductance of a given cross point, we applied synchronized positive voltage pulses from a driving circuit board to the memristor top electrode and the gate of the series transistor. The gate voltage, which specifies a compliance current, determines the resulting memristor conductance. We decreased the conductance by first applying a sufficient positive pulse to the memristor bottom electrode to initialize the state, and then used a conductance increase scheme to set the memristor to the desired level. With this scheme, we achieved linear and symmetric conductance increase and decrease with minimal cycle-to-cycle (Fig. 7f) and device-to-device (Fig. 7g) variation. We were able to set the conductance values across the entire 128×64 array, except for the stuck devices, with only two electrical pulses to each memristor with reasonably high accuracy (Fig. 7h). The rapidity and reliability of the conductance-update scheme make it possible to arbitrarily train the network *in-situ* with almost any standard algorithm. Here, the network was trained using stochastic gradient descent (SGD) to subsequently perform Bayesian inference on the MNIST classification task. For each new sample of training data, the network first performs Bayesian inference to get the log-probability of the label for each output by the softmax function, and then the weights in each layer are updated.

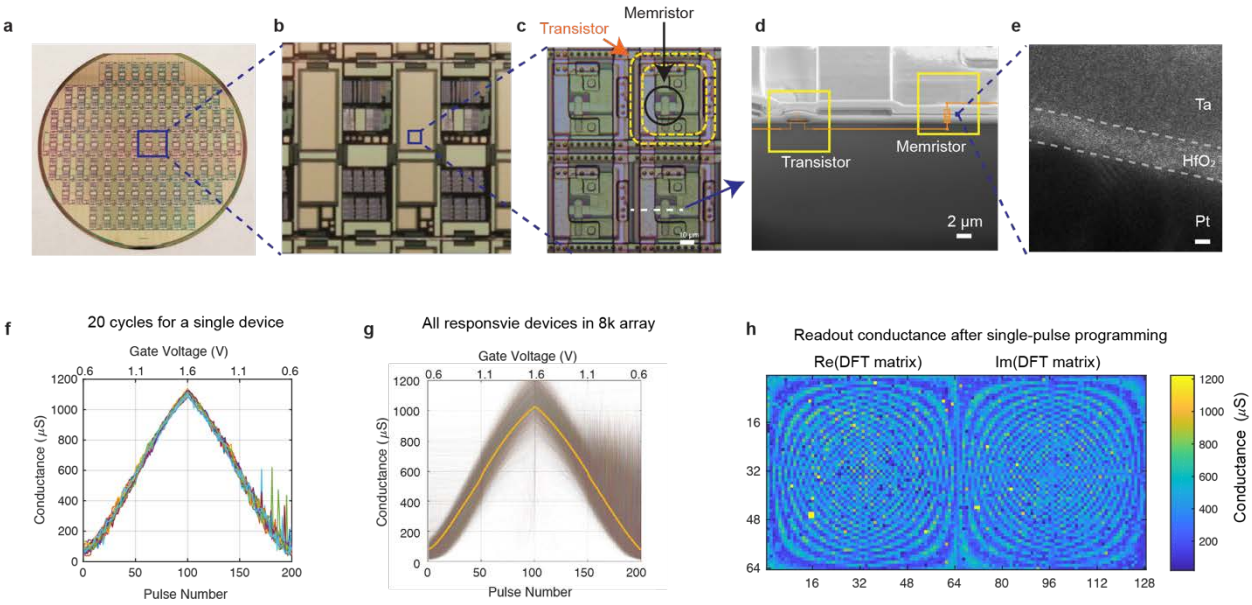


Figure 7 Memristive platform for in-situ learning a, An optical image of a wafer with transistor arrays. b, Close-up of chip image showing arrays of various sizes. c, Microscope image showing the 1T1R (one transistor one memristor) structure of the cell. Scale bar, 10 µm. d, Cross-sectional scanning electron microscopic image of an individual 1T1R cell, which is cut in a focused ion beam microscope from the dashed line in c. Scale bar, 2 µm. e, Cross-sectional transmission electron microscopic image of the integrated Ta/HfO_2/Pt memristor. Inset: electron dispersive spectroscopic elemental mapping. Scale bar, 2 µm. f, Evolution of conductance during 20 cycles of full potentiation and depression for a single cell with 200 pulses per cycle, showing low cycle-to-cycle variability. g, Evolution of conductance over one 200-pulse cycle of full potentiation and depression for all responsive devices in the array, with median conductance indicated by the yellow line. h, Conductance of a 128×64 array after single-pulse conductance writing of the discrete Fourier transform matrix. Several stuck devices are visible (in yellow).

We partitioned a single 128×64 array and constructed a two-layer perceptron with 64 input neurons, 54 hidden neurons, and ten output neurons to be trained on the MNIST dataset of handwritten digits '0' through '9', which has become the standard benchmark by which to gauge new neural network approaches. Each input image was rescaled to 8 pixels by 8 pixels to match our network. The intensities of the grayscale images were unrolled into 64-dimensional input feature vectors, which were duplicated to produce 128 analog voltages to enable negative effective weights. The two-layer network used 7,992 memristors, each of which was initialized with a single pulse with a 1.0 V gate voltage. The network was then trained on 80,000 images drawn from the training database (some images were drawn more than once), with a minibatch size B=50 for a total of 1,600 training cycles. The accuracy during online training is shown in Fig. 8c. The inset of Fig. 8c shows the linear relationship between the conductance and the applied gate voltage during each update cycle, which was critical for this demonstration. After utilizing the entire training database, the network correctly classified 91.71% of the 10,000 images in the separate test set (Fig. 3c). Many of the misclassified images are in fact difficult for humans to identify at the available resolution.

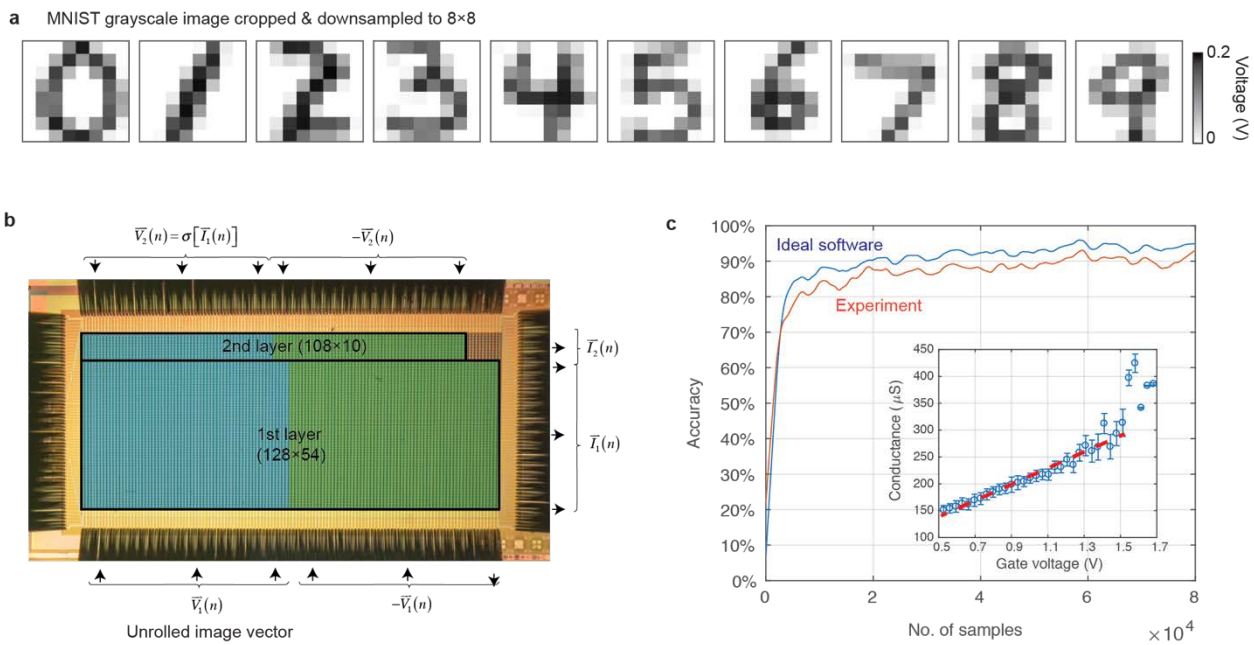


Figure 8 In-situ online training experiment on MNIST handwritten digit recognition a, Typical handwritten digits from the MNIST database. b, Photo of the integrated 128×64 array during measurement. The array was partitioned into two parts for the 1st and 2nd layer, respectively. 54 hidden neurons were used, so the 1st layer weight matrix is 64×54 (implemented using 6,912 memristors) and the 2nd layer matrix is 54×10 (implemented using 1,080 memristors). The blue and green false-colored areas are the positive and negative parts of the differential pairs. c, Minibatch accuracy increases over the course of training. Experimental data followed the simulation closely, with a consistent 2-4% gap. Inset shows the conductance-gate voltage relation extracted from data collected during training.

4.1.4. Reinforcement Learning.

In addition to the above demonstrated supervised learning, we have also demonstrated reinforcement learning using the 128×64 array, which is essentially used as a vector matrix multiplication accelerator as well. Reinforcement learning has been adopted in AlphaGo[13,14].

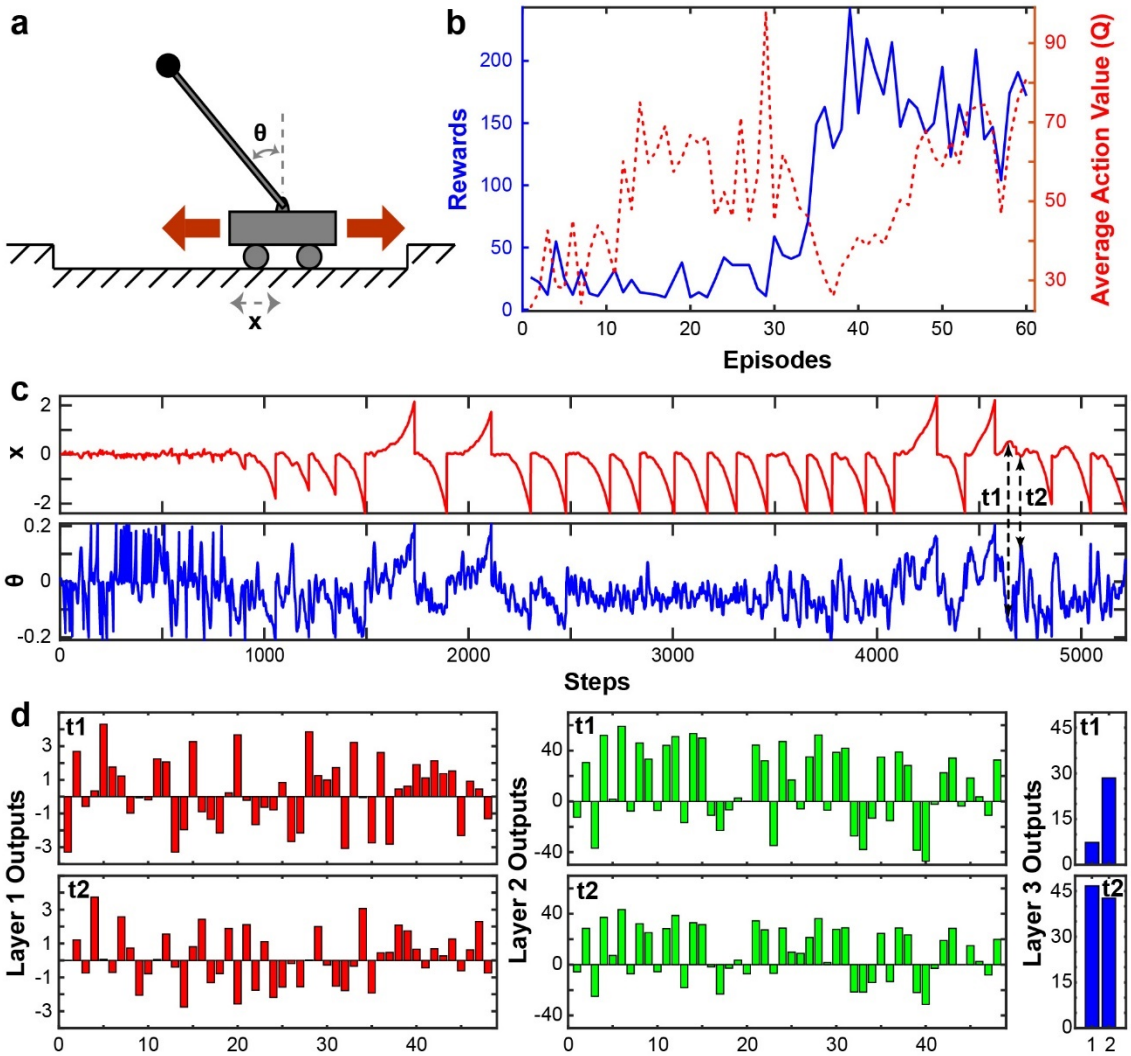


Figure 9 In-memristor reinforcement learning in the cart-pole environment a, Schematic illustration of the cart-pole environment. The cart is free to move along the track while supporting a hinged pole. The learning agent can make a left or right push at each discrete time step to avoid the pole falling or the cart driving beyond the bounds. b, Training curves tracking the number of agent rewards per epoch (blue) and the average predicted action-value (red). Each point of the latter is the average Q-value of all actions taken per epoch. c, The time evolution of the cart position $\boldsymbol{x}$ and pole angle $\boldsymbol{\theta}$. The failures of the early game epochs are mainly due to the pole falling. The pole was kept upright (i.e. $\boldsymbol{\theta}$ rarely hit the upper/lower bounds.) for many more steps in the later phases of the game. d, Output of all layers of the memristor Q-network at time t1 and t2 specified in c. At time t1, the pole was tilted counter-clockwise. The second neuron of the output layer, representing the left push (push from the right), was larger. On the contrary, at time t2, the pole was tilted clockwise. The first neuron of the output layer, representing the right push (push from the left), was larger.

We use the cart-pole problem[15] as a testbed for the in-memristor reinforcement learning system. the single 128×64 1T1R array was partitioned to construct the 3-layer Q-network with 4 input neurons, 48 neurons in each hidden layer, and 2 output neurons. The network consisted of 2592 weights or differential pairs, implemented on 5184 memristors of the 1T1R network.

Fig. 9a shows a schematic representation of a cart-pole scenario. A rigid pole is hinged to a cart which is free to move within the bounds of a one-dimensional track. The pole rotates in the vertical plane above the track due to both the gravitational force and the motion of the cart. The learning agent could either apply an impulsive 'left' or 'right' push of fixed force magnitude to the cart at discrete time intervals. The dynamics of the cart-pole environment was simulated in software. The environment was abstracted as a 4-dimensional Markov state vector $\left(x, \frac{dx}{dt}, \theta, \frac{d\theta}{dt}\right)$, where x and θ are the cart position and pole angle, respectively, which also serves as the input to the memristive Q-network. The learning was model free. Therefore, the agent was unaware of the equations of motion of the cart-pole system. The only feedback for evaluating performance was a binary reward signal which was 0 if the pole fell past a certain angle from the vertical or the cart reached an end of the track, or 1 otherwise. If a '0' reward signal is seen by the agent, it ends the current game epoch and resets the cart-pole environment to start another game. This forms an "avoidance control problem" so the reward "1" makes the agent try to avoid failure for as long as possible. As shown in Fig. 9b, the memristor backend learning agent scored poorly in the first 30 epochs of the game, as it had not yet learned a good Q-function to approximate the dynamics of the environment. Fig. 9c shows the time evolution of cart position and pole angle, which reveals that most failures in this phase were triggered by the fact that the pole was no longer upright while the cart was still within the legal range. The failure signal usually occurred after a long sequence of individual actions, which is why the agent needed a thousand steps to figure out what was responsible for the failure. In the second half of the learning course, the agent gradually constructed associations between the input and output, which was also reflected by the rising of the average action value in Fig. 9b, based on the reinforcement feedback. The agent achieved an average performance of ~166 per game (i.e. the pole could be balanced for 166 time steps with a legal cart position) from epoch 40 to 60. The pole was almost upright in this phase as depicted in Fig. 9c, which reveals that the memristor Q-network had gained the capability to balance the pole. Sample responses of the network in different scenarios are illustrated in Fig. 9d. At time t1, the angle of the pole was negative, indicating the pole tilted towards the left end of the track. The Q-network suggested the action of a left push, which could physically make the cart accelerate leftwards to restore the balance of the pole. On the contrary, at time t2, the pole experienced a clockwise rotation. A right push was made by the learning agent to avoid further tilting. To show that the same system can be applied to another control problem, we applied the same memristor Q-network architecture to the mountain car game while keeping the hyperparameters and learning procedures the same. This demonstrates that memristor crossbar arrays can be used to implement reinforcement learning much more efficiently, owning to its in-memory comping nature.

4.2. Spike Based Computing

The level based computation does not bare much bio-inspiration in it, while spike based computation is believed to be closer to how brain functions and thus implement neuromorphic computing more faithfully, which hopefully leads to more energy efficiency[16]. In order to better emulate the bio-intelligent systems, the artificial synapses[17] and neurons need to have the critical dynamics associated with ion diffusions (e.g. Ca^{2+} diffusion) observed in the bio-counterparts.

4.2.1. Diffusive Memristor Development.

Diffusive memristors[18] were developed to equip the artificial neural networks with diffusion dynamics. CMOS circuits have been employed to mimic synaptic Ca^{2+} dynamics, but three-terminal devices bear limited resemblance to bio-counterparts at the mechanism level and require significant numbers and complex circuits to simulate synaptic behavior. A substantial reduction in footprint, complexity and energy consumption can be achieved by building a two-terminal circuit element, such as a memristor directly incorporating Ca^{2+}-like dynamics. Various types of memristors based on ionic drift (drift-type memristor) have recently been utilized for this purpose in neuromorphic architectures[19-26]. Although qualitative synaptic functionality has been demonstrated, the fast switching and non-volatility of drift memristors optimized for memory applications do not faithfully replicate the nature of plasticity. Similar issues also exist in MOS-based memristor emulators, although they are capable of simulating a variety of synaptic functions including spike-timing-dependent plasticity (STDP). A device with similar physical behavior as the biological Ca^{2+} dynamics would enable improved emulation of synaptic function and broad applications to neuromorphic computing. In this program, we developed such an emulator using Ag doped oxide, which was a memristor based on Ag atom diffusion and spontaneous nanoparticle formation, as determined by *in situ* high-resolution transmission electron microscopy (HRTEM) and nanoparticle dynamics simulations. The dynamical properties of the diffusive memristors were confirmed to be functionally equivalent to Ca^{2+} in bio-synapses, and their operating characteristics were experimentally verified by demonstrating both short- and long-term plasticity, including mechanisms that have not been unambiguously demonstrated previously. In addition, significant similarities exist between the Ag dynamics and that of synaptic Ca^{2+}, not only in the diffusion mechanism but also in their dynamical balance of concentration and regulating roles in their respective systems. Ca^{2+} dynamics is responsible for initiating both short- and long-term plasticity of synapses, forming the basis of memory and learning.

The dynamical properties of diffusive memristors were studied by applying voltage pulses and measuring resulting currents. Under an applied pulse, the device exhibited threshold switching to a low resistance state after an incubation period τ_d, as shown in Fig. 10a. This τ_d is related to the growth and clustering of silver nanoparticles to eventually form conduction channels. Upon channel formation, the current jumped abruptly by several orders of magnitude, and then slowly increased further under bias as the channel thickened. As the voltage pulse ended, the device relaxed back to its original high resistance state over a characteristic time τ_r. As shown in Fig. 10b, τ_r decreased as the ambient temperature increased, consistent with a diffusion activation energy of 0.27eV (inset of Fig. 10b), and the characteristic time was on the same order as the response of bio-synapses, i.e., tens of ms. In addition to the temperature, τ_d and τ_r were also functions of the voltage pulse parameters, operation history, Ag concentration, host lattice, device geometry, humidity, and other factors, which alone or combined could be used to tune the desired dynamics for neuromorphic systems.

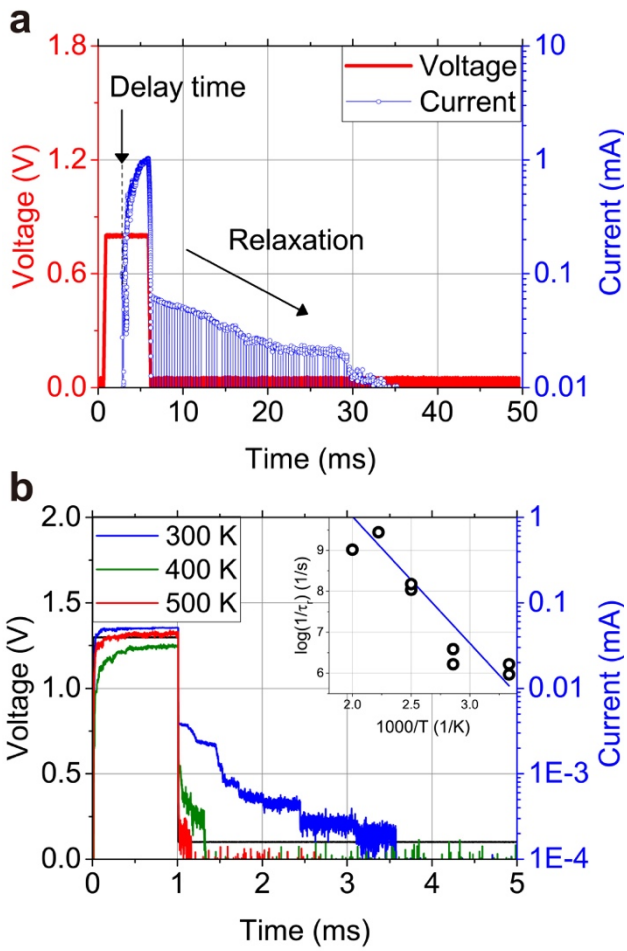


Figure 10 Timing mechanism of SiOxNy:Ag diffusive memristor a. Delay and relaxation characteristics of the device showing variation of current (blue) with applied voltage (red) pulses. Multiple read voltage pulses of (0.05V, 10µs) are used to study the device relaxation current after the switching pulse (0.75V, 5ms). The device requires a finite delay time to turn ON and has a finite relaxation time before it goes to the high resistance state after the switching pulse is removed. b. Device relaxation performance showing the variation of current with applied voltage at different temperatures. The relaxation time decreases with increasing temperature. Inset shows the Arrhenius plot of the temperature dependence of the relaxation time. Each data point (black circles) is an average over 10-15 measured relaxation times, and are fitted to the blue line. The activation energy for the material system is calculated to be 0.27eV.

4.2.2. Artificial Synapse.

We used the diffusive memristor with a non-volatile element, i.e., a drift-type memristor, to build a more faithful artificial synapse, which naturally exhibited long-term plasticity[27] following the spike-rate-dependent plasticity and STDP learning rules. For demonstration purposes, we created a combined circuit element using a diffusive memristor in series with a Pt/TaO_x/Ta/Pt drift memristor. This combined element was connected to pulsed voltage sources similar to a synapse between pre- and post-synaptic neurons (Fig. 11a). The spike-rate-dependent potentiation demonstration is illustrated in Fig. 11b, where the drift memristor weight (conductance) change is a function of the frequency of the applied pulses. A shorter t_{zero} resulted in a greater increase in the conductance of the diffusive memristor and thus a larger voltage drop across the drift

memristor, which thereby switched due to the voltage divider effect[16]. A longer t_{zero} resulted in a smaller increase in the diffusive memristor conductance and thus a smaller voltage drop across the drift memristor, leading to a smaller or non-detectable resistance change in the drift memristor. To demonstrate STDP learning rules with non-overlapping spikes, pre and post-synaptic spikes (Fig. 11c) were applied to the combined element. The two spikes were separated by a time difference Δt, which determined how much conductance change was programmed in the drift memristor. Each spike consisted of two parts, a high voltage short pulse and a low voltage long pulse. The pre-spike and post-spike were equal in magnitude but opposite in voltage polarity (Fig. 11c). In the combined element, the resistance of the diffusive memristor in its OFF state is much larger than that of the drift memristor, while the resistance of its ON state is much smaller than that of the drift memristor. Because the diffusive memristor has a finite delay time, the short high voltage pulse will not turn it ON. In contrast, the long voltage pulse with a lower amplitude will turn ON the diffusive memristor. The drift memristor is not switched by the first spike, because the majority of the voltage drops across the diffusive memristor and turns it ON first. After the spike ends, the resistance of the diffusive memristor gradually increases from its ON state over time, regulated by the diffusive dynamics. The second spike occurs at a time Δt from the end of the first spike, and it may or may not switch the drift memristor depending on how much voltage drops on the drift memristor, which is determined by the conductance of the diffusive memristor at that moment, a function of Δt. A smaller Δt corresponds to a smaller diffusive memristor resistance and results in a greater resistance change in the drift memristor and vice versa (Fig. 11d). If the pre-spike appears before the post-spike, the drift memristor conductance increases (potentiation). If the pre-spike follows the post-spike, depression occurs. Because the dynamics of the diffusive memristor provides an intrinsic timing mechanism for the combined element, the spike-rate-dependent plasticity and STDP do not require complex pulse engineering or spike overlapping. This substantially reduces the complexity of both circuit and algorithm design and enables low-energy operations. In addition, depending on the application, any non-volatile memristor (low/high retention, analog/digital) can be used along with the diffusive memristor, allowing a significantly broader choice of materials rather than relying on the properties of the drift memristor when used alone.

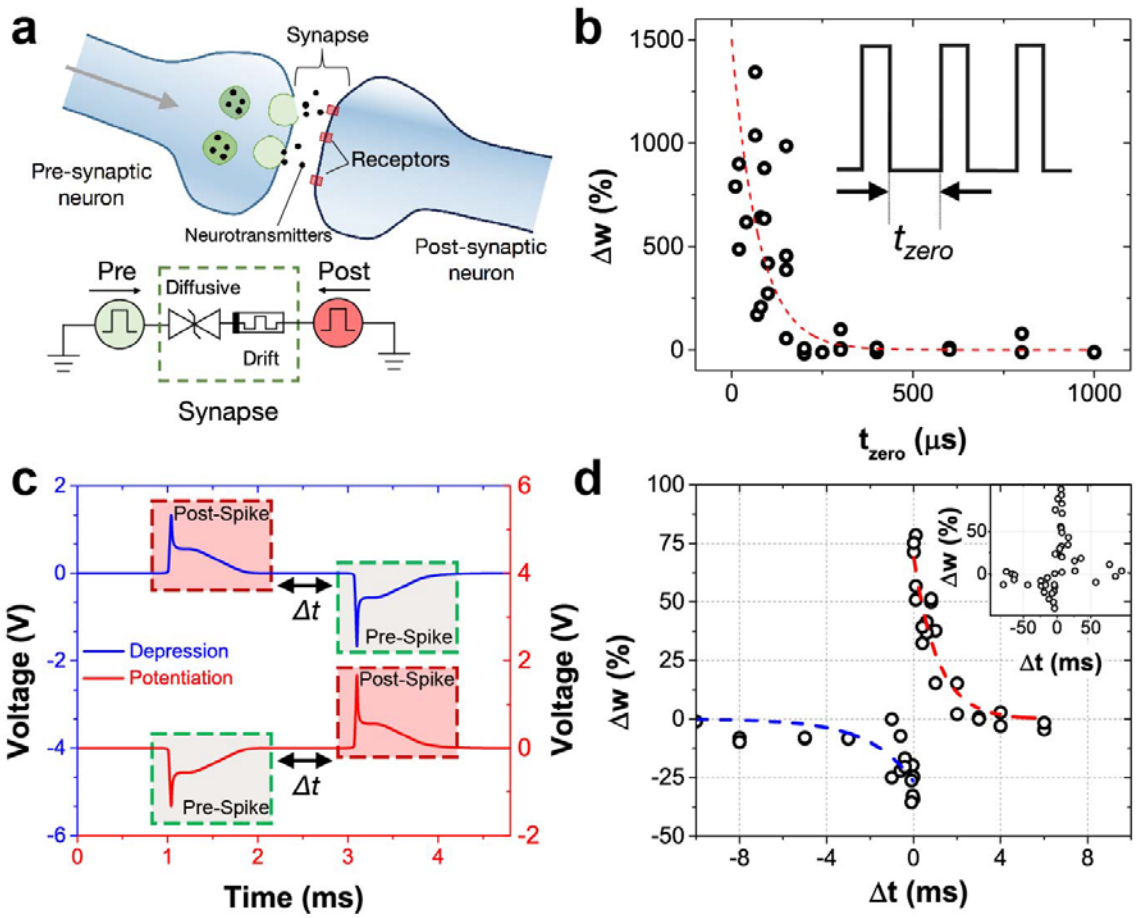


Figure 11 Long-term spike-rate-dependent potentiation and bio-realistic spike-timing dependent plasticity (SRDP and true STDP) behavior of a combined device consisting of a diffusive and a drift memristor a. Illustration of a biological synaptic junction between the pre- and post-synaptic neurons. Also shown is the electrical implementation, a circuit diagram of the electronic synapse consisting of the Si-O_xN_y:Ag diffusive memristor connected in series with the TaO_x drift memristor and between pulsed voltage sources, which act as neurons that send voltage spikes to the synaptic junction. b. Spike-rate-dependent potentiation showing the change in the conductance (weight) of the drift memristor in the electronic synapse with change in the duration t_{zero} between the applied pulses. For long t_{zero}, the change in the conductance of the diffusive memristor is lower (see Fig. 5b), resulting in a lower weight change of the drift memristor. As the t_{zero} decreases, the weight change increases. The dotted red line represents a fit of the average conductance change with change in t_{zero}. c. Schematic of the pulses applied to the combined device for STDP demonstration. The long low voltage pulse in each spike turns the diffusive memristor ON, and the short high voltage pulse switches the drift memristor. When the post-spike precedes the pre-spike, the device is reset (depressed), and when the pre-spike precedes the post-spike, the device is set (potentiated). The timing (Δt) between the two spikes determines the voltage drop across the drift memristor. d. Plot of the conductance (weight) change of the drift memristor with variation in Δt showing the spike-timing dependent plasticity of the electronic synapse. This response is characteristic of the timing-dependent response of biological synapses. The inset shows the spike-timing-dependent plasticity of a typical chemical synapse.

4.2.3. Artificial Neurons.

In addition to artificial synapses, we have also designed and built artificial neurons[28]. We physically emulated the leaky integrate-and-fire neuron model with a diffusive memristor, fabricated by sandwiching a dielectric material (e.g. SiO_xN_y or SiO_x) carefully doped with Ag nanoclusters between two electrodes. This discrete device, schematically illustrated in Figure 12a, was characterized by applying voltage pulses across the artificial neuron in series with resistors to represent synapses and recording the resulting output current versus time. Figures 12b-e compare experimentally measured data with corresponding physics-based simulation results. The temporal behavior of the artificial neuron was observed during and after the input of a single super-threshold voltage pulse followed by a train of smaller pulses. There was a distinct delay time (τ_d) between the arrival of the voltage pulse and the rise of the output current, which was caused by the interaction of the RC time constant of the circuit with the internal Ag dynamics of the memristor. With a relatively large circuit capacitance, the RC time constant, that is the time for establishing the switching voltage of the diffusive memristor, dominates the delay time. With a smaller capacitance, the RC time becomes shorter and the internal Ag dynamics of the memristor dominates the delay time and thus the integrate-and-fire behavior, as shown in Figure 12. The internal Ag dynamics of diffusive memristors originates from complicated multi-physics effect including field induced Ag mass transport from the electrodes (e.g. Ag diffusion and redox reaction) and the formation of an electrical conducting path. We have constructed a physics-based model that agrees well with the microscopic observation of Ag filament growth and rupture during threshold switching as well as the measured temporal response to voltage signals (e.g. Figures 12) [29-31]. After the fall of the voltage pulse, the memristor conductance relaxed with a characteristic time (τ_r) determined within our model by the Ag diffusive dynamics to dissolve the nanoparticle bridge and return the neuron to its resting state. The relaxation dynamics also leads to the leakiness of the internal Ag dynamics, which gradual dissolves partially formed Ag filament driven by the minimization of interfacial energy between Ag and dielectrics, or Thomson-Gibbs effect. When a sequence of sub-threshold pulses was applied to the device, as shown in Figures 12b and 12d, the device fired after some number of pulses and relaxed back to the resting state after the end of the pulse train. Shown in Figures 12c and 12e are the corresponding experimentally measured and simulated histograms of the firing statistics, respectively, which show that the threshold is not sharp but has an associated probability distribution function, providing the stochastic behavior commonly observed in actual neurons. Since the internal memristor dynamics depend on the behavior of nanoparticles, the leaky integrate and fire mechanism observed here should scale to very small device sizes.

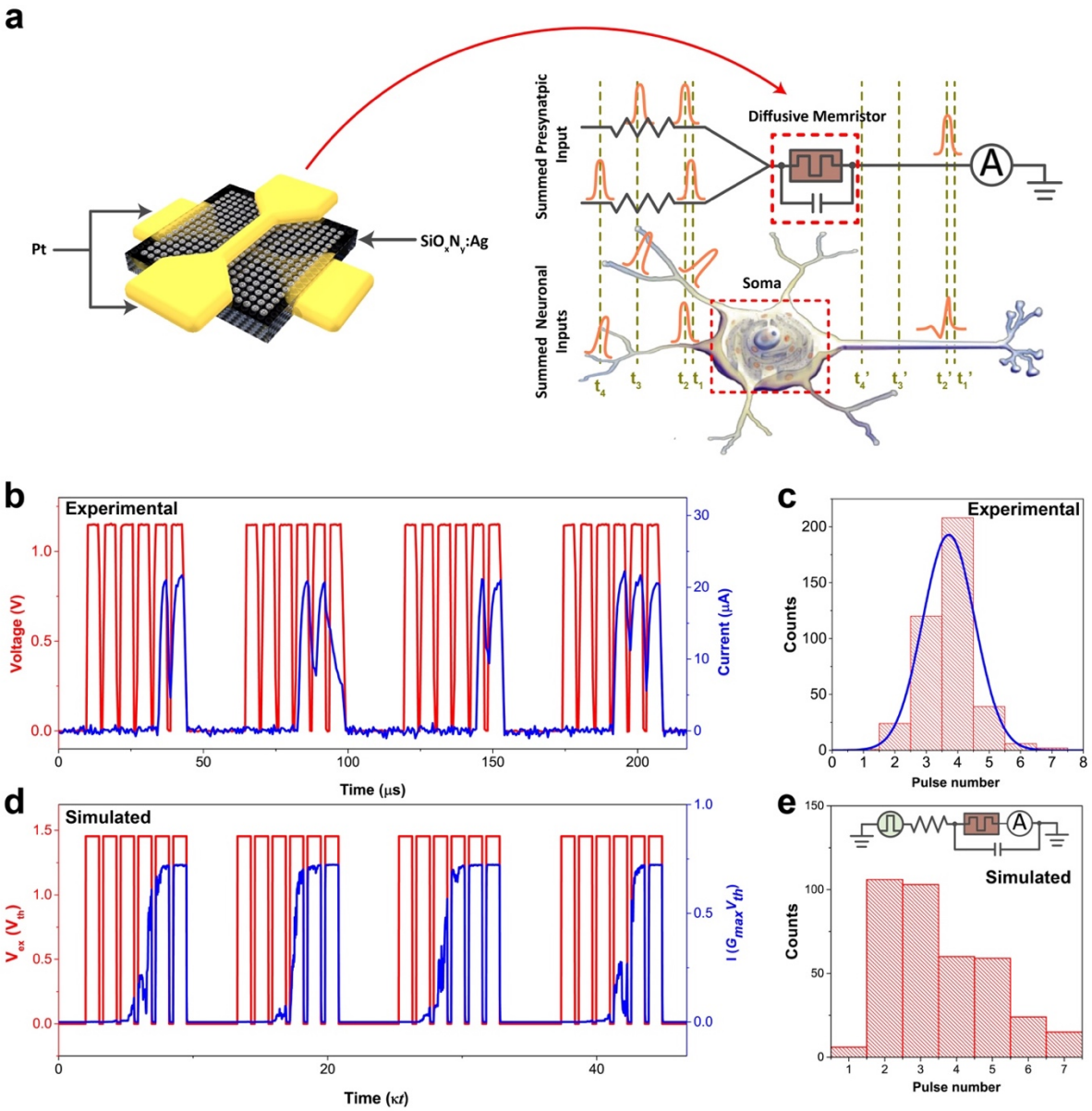


Figure 12 Diffusive memristor artificial neuron a. Schematic illustration of a crosspoint diffusive memristor, which consists of a SiO_xN_y:Ag layer between two Pt electrodes. The surrogate neuron receives software summed weighted presynaptic inputs via a pulsed voltage source and an equivalent synaptic resistor (e.g. 20μS in this case). (See Supplementary Note 3 for the principle of software spatial summation.) Both the surrogate and biological neurons integrate input stimuli (orange) beginning at t_1 and fire when the threshold condition is reached (i.e. at t_2'). The integrated signal decays over time such that input stimuli spaced too far apart will fail to reach threshold (i.e. the delay between t_3 and t_4). b. Experimental response of the device to multiple subthreshold voltage pulses followed by a rest period of 200 μs (only 20 μs is shown for convenience). The device required multiple pulses to reach the threshold and 'fire'. c. Histogram of the number of subthreshold voltage pulses required to successfully fire the artificial neuron (red) compared to a Gaussian distribution (blue). d. Simulated response of the device to multiple subthreshold voltage pulses as in b and showing similar behaviour to experiment, with the resting time between pulse trains chosen to allow the Ag in the device to diffuse back to the OFF state. (Only 10% of the rest period is shown

for convenience) e. Simulated switching statistics with respect to pulse numbers (within each train), consistent with the experimental results in c. The inset illustrates the circuit diagram used in the simulation.

4.2.4. Fully Memristive Neural Network.

We then went a step further to demonstrate inference on a prototype chip of fully integrated memristive neural network[28]. Figure 13a shows the overview of the integrated chip consisting of synaptic array and diffusive memristor neurons. The synapses were built by integrating drift memristors with foundry-made transistor arrays using back-end-of-the-line (BEOL) processes. Each Pd/HfO_2/Ta memristor is connected to a series transistor. Figure 13b shows the detailed structure of a single 1T1R cell and associated connections. When all the transistors are turned on, the 1T1R array works as a fully connected memristor crossbar. Structural analysis using high-resolution transmission electron microscopy was performed on the integrated memristors, which reveals an amorphous HfO_2 layer sandwiched between Pd and Ta electrodes in Figure 13c. Figure 13d illustrates the junction of a single diffusive memristor. A transmission electron micrograph of its cross-section shows the amorphous nature of the background SiO_x dielectric lattices and the nano-crystalline Ag layer in Figure 13e.

Pre-synaptic signals could be classified by such a fully memristive neural network. For demonstration purpose, the synapses were pre-programmed to have different weights, which could be the result of any kind of learning process. Four letter patterns “U”, “M”, “A”, and “S” with artificially added noise were used as example inputs. The red and blue squares in Figure 13f represent the input differential voltages fed to the rows of the synaptic array. For example, a red square means a +0.8V/-0.8V input pair and a light blue square means a -0.6V/+0.6V input pair. The input pattern is divided into 4 sub-images of a 2×2 size, with a stride of two. Each sub-image is unrolled into 1 column input vector (8 voltages) and fed into the network (8 rows) at each time. For each possible sub-image there is a corresponding convolutional filter implemented by 8 memristor synapses in a column, with a total of 8 filters (8 columns) in the 8×8 array. The measured weights were depicted in Figure 13g after programming. The negative values of the convolution matrices are mapped to the conductance of memristor cells by grouping memristors from adjacent rows to form a differential pair. The result of the convolution of the 8 filters to each sub-image are concurrently revealed by the firing of their corresponding diffusive memristor artificial neurons which serve the role of the ReLUs. This network can produce unique response for each input pattern, as illustrated in the Figure 13h and i, in the form of integration time and the maximum fire current. We have also verified the repeatability of the network by feeding the 8 noise-free patterns in cycles to the network and record the average firing delay and current of neurons. The integration time of a noisy input is generally longer due to smaller inputs and thus smaller convolution results. Correspondingly, inputs with positive additive noise will usually fire faster. This proof of principle demonstration of the fully integrated memristive neural network comprising memristor-based artificial synapses and artificial neurons can be expanded to implement learning systems of larger complexity in an energy efficient manner, such as multilayer neuron networks.

Due to the efficient interplay between the artificial synapses and neurons, the fully memristive neuron network is capable of unsupervised learning using STDP learning rule. STDP is the prevalent protocol of synaptic weights update in spiking neural networks. In this project, we derived a simple STDP scheme to train a fully connected layer in an unsupervised way, which naturally complements the convolution and ReLU layers and further enables a functional deep convolutional network. Since the drift memristor synapses encode the conditional probability[32], the

neurons will tend to respond to the means of inputs associated with fire events, essentially carrying out clustering of the inputs. This is experimentally demonstrated in Figure 14. Software pooling and signal conversion are used to fit the output of ReLU layer to the input of the fully connected layer. (See Figure 14a) Lateral inhibition is deployed, which is typical in fully connected feedforward networks to enhance the discrimination of the inputs and make the self-adapting network energy efficient. After a few cycles of uncertainty where the conductance of synapses concentrates around the initial values (~100µS), the synapses are quickly attuned by the simple STDP rules. As shown in Figure 14d, undergoing either potentiation or depression, patterns of synapses of the N1, N2, and N3 neurons quickly gain similarities to the prototypical patterns in Figure 14a. (i.e. '11110000', '11000011', '00001100') It is also noted that synapses may show different response to the learning rules. For instance, the 3rd synapse of N1 and the 7th synapse of N2 are much less potentiated, which may be due to the device-to-device variation of the threshold conditions of the drift memristors. The quick divergence of conductance of drift memristors indicates a fast learning rate which is dependent on the firing time or pulse width of diffusive memristor neurons. Such convergence is also reflected by the magnitude (or threshold) of input patterns in Figure 14b. The magnitude of a specific pattern reduces in the first few cycles and then becomes stable. This is because diverged conductance of drift memristors tend to saturate so that further increment (decrement) in conductance will become more and more ineffective when they are close to the upper (lower) bound of the conductance range. Unsupervised learning was demonstrated for the first time in memristive neural network.

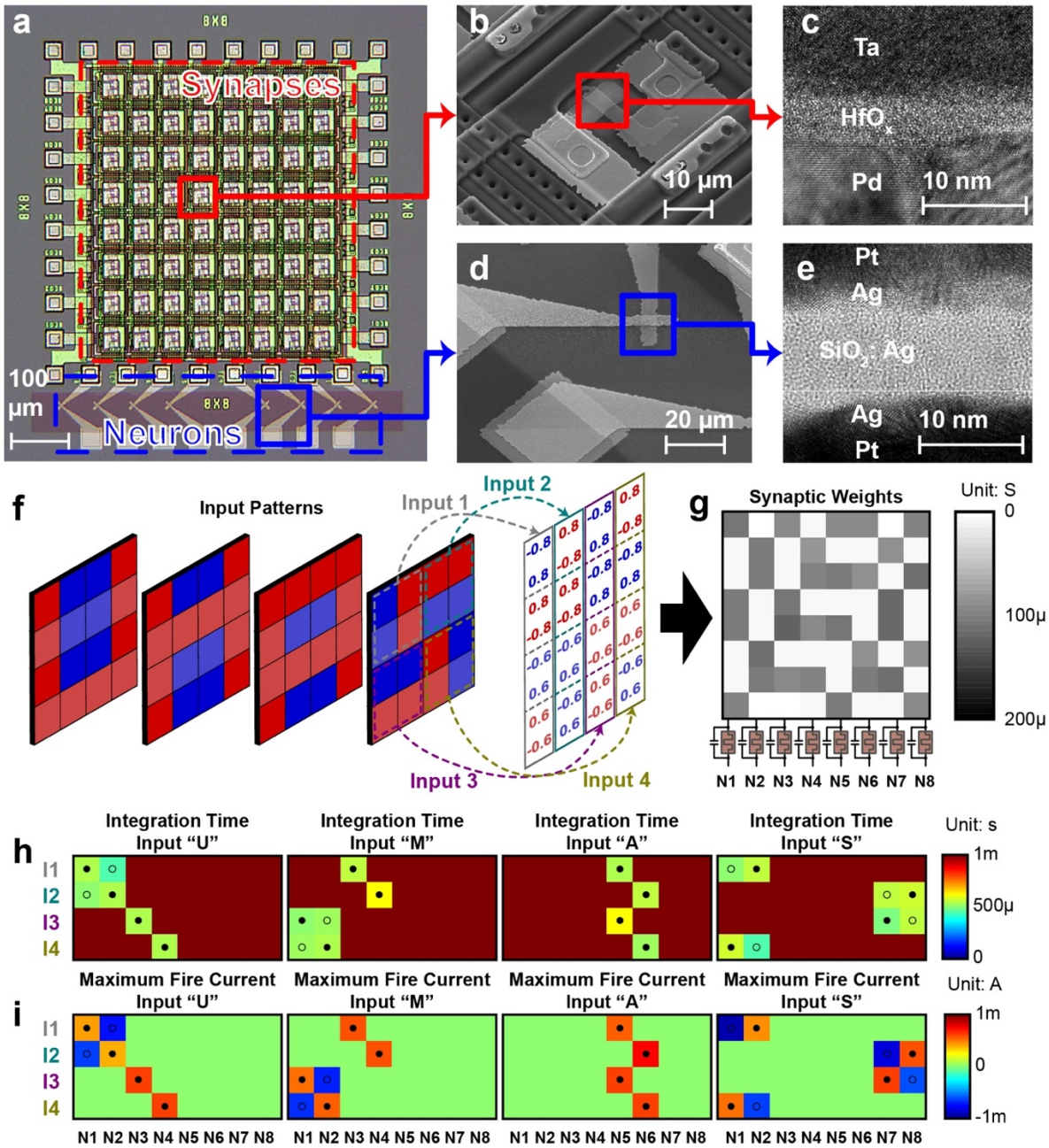


Figure 13 Fully integrated memristive neural network for pattern classification a. Optical micrograph of the integrated memristive neural network, consisting of an 8×8 1T1R memristive synapse crossbar interfacing with 8 diffusive memristor artificial neurons (Each neuron used in this demonstration has an external capacitor.). b. Scanning electron micrographs of a single 1T1R cell. Memristive synapses of the same row share bottom electrode lines while those of the same column share top electrode and transistor gate lines. c. Cross-sectional transmission electron microscopy image of the integrated Pd/HfO_x/Ta drift memristor prepared by focused-ion-beam cutting. d. Scanning electron micrograph of a single diffusive memristor junction. e. High-resolution transmission electron micrograph of the cross-section of the Pt/Ag/SiO_x:Ag/Ag/Pt diffusive memristor showing amorphous background SiO_x with nano-crystalline thin Ag layers. f. The input pattern consists of 4 letters 'UMAS' with artificially added noise. Each input pattern consists of 4×4 pixels which are divided into four inputs (Input 1, Input 2, Input 3, and Input 4). Each input

covers a sub-array of 2×2 size (4 pixels) of the original pattern using differential pairs as listed. Triangular voltage waveforms are fed to the 8 rows of synapses of the network. g. Measured conductance weights of the memristors after programming the 8 convolutional filters (1 filter per column) onto the 8×8 array using a differential pair scheme. Each of the 8 columns is interfacing with a diffusive memristor neuron at the end of the column. h-i. Measured integration time and maximum amplitude of fire current of the artificial neurons as responses to the 'UMAS' input patterns. Each individual input pattern is associated with its unique firing pattern of the 8 artificial neurons. The ideal output patterns are marked by the dots (circles) for neurons with positive (negative) fire current flowing out of the network.

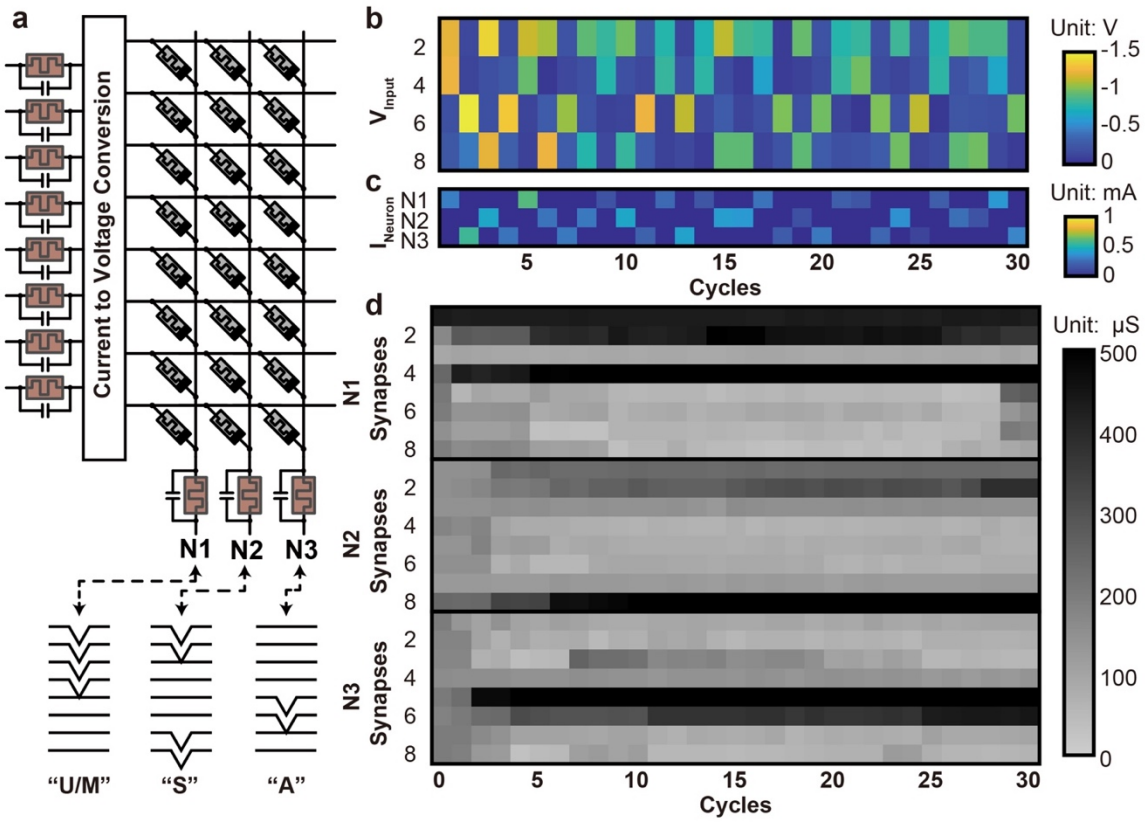


Figure 14 Unsupervised training of a fully connected network based on the integrated all-memristive neural network a, The schematic diagram of the 8×3 network with inputs based on the outputs of the neurons in Figure 4. The prototypical patterns of neurons after training correspond to the input letters "U/M", "S", and "A" in Figure 4, respectively. b-d, The input patterns (peak voltages of triangular waveforms), peak neuronal currents, and synaptic weights at each training cycle. The synapses of the N1, N2, and N3 neurons quickly diverge from the initial 100μS and evolve to patterns with increasing similarities to the prototypical patterns in a. The magnitude of input patterns in b reduces in the first few cycles and becomes stable due to conductance saturation of the diverged drift memristor synapses.

5.0.CONCLUSIONS

We have demonstrated the first analog-vector and analog-matrix VMM utilizing crossbars with over 8,000 memristors, with an equivalent six bit or 64 level precision and 99.8% device yield. The device resistance states were precisely tuned and the IV characteristics were linear, ideal for analog computing. We successfully implemented some important applications for IoT and edge computation, including signal processing, image compression and convolutional filtering. The energy efficiency of the system was over 119.7 trillion equivalent OPS/W operation given the readout process can be done in 10 ns, and is expected to increase significantly at shorter readout time. The results here represent an encouraging advance in hardware implementation of computing using emerging devices and provide a promising path towards energy-efficient analog computing based on memristors.

In addition to inference, we have demonstrated the *in-situ* and self-adaptive learning capability of a multilayer neural network built by monolithically integrating memristor arrays onto a foundry-made CMOS substrate. The transistors enabled reliable, linear and symmetric synaptic weight updates, allowing the network to be trained with standard machine learning algorithms. After training with a stochastic gradient descent algorithm on 80,000 images drawn from the MNIST training set, we achieved 91.71% accuracy on the complete 10,000-image test set. This accuracy is only 2.4% lower than an idealized simulation despite an 11% defect rate for the memristors used. The demonstrated performance with *in-situ* online training and inference suggests that memristor crossbars are a promising high speed and energy efficiency technology for artificial intelligence applications. The software neurons used in this demonstration indicate that a hybrid digital processor and neuromorphic analog approach for DNNs can be effective, but all of the software functions used in the present demonstration can be integrated as hardware onto a full-function chip in the near future. Furthermore, reinforcement learning has also been demonstrated with memristor crossbar array to accelerate the vector matrix multiplications.

Towards spiking neural networks, we have constructed and demonstrated a new class of memristors as synaptic emulators that function primarily based on diffusion (rather than drift) dynamics. The microscopic nature of both the threshold switching and relaxation of the diffusive memristor is revealed for the first time by *in situ* HRTEM and explained by nanoparticle dynamics simulation. The Ag dynamics of the diffusive memristors functionally resemble the synaptic Ca^{2+} behavior in chemical synapses and lead to a direct and natural emulation of multiple synaptic functions for both short-term and long-term plasticity, such as PPF, PPD, PPD following PPF, SRDP and STDP. In addition to providing a synapse emulator, the diffusive memristor can also serve as a selector with a large transient nonlinearity that is critical for the operation of a large crossbar array as a neural network. The results here provide an encouraging pathway toward synaptic emulation using diffusive memristors for neuromorphic computing.

In addition to artificial synapse, we have demonstrated a stochastic leaky integrate-and-fire artificial neuron based on a discrete scalable diffusive memristor, featuring Ag dynamics similar to that of actual neuron ion channels, representing to date the simplest and yet faithful realization of electronic neural functionality, in contrast to traditional approaches requiring tens to hundreds of CMOS devices. Physics-based simulations reproduce our experimental observations and thus enhance our understanding of the interplay between memristor dynamics and circuit RC effects.

Utilizing the integrate-and-fire function, the artificial neurons have performed unsupervised synaptic weight updating and pattern classification for the first time on deep integrated neural networks comprising only memristors.

6.0.REFERENCES

[1] Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. Nature Nanotechnology 8, 13-24 (2013).

[2] Burr, G. W. et al. Neuromorphic computing using non-volatile memory. Advances in Physics: X 2, 89-124 (2016).

[3] Hu, M. et al. in Proceedings of DAC.

[4] Li, C. et al. Analogue signal and image processing with large memristor crossbars. Nature Electronics 1, 52 (2017).

[5] 5 Ma, W. et al. in Int El Devices Meet 436-439 (IEEE, San Francisco, CA, USA, 2016).

[6] Yao, P. et al. Face classification using electronic synapses. Nat Commun 8, 15199, doi:10.1038/ncomms15199 (2017).

[7] Choi, S., Shin, J. H., Lee, J., Sheridan, P. & Lu, W. D. Experimental Demonstration of Feature Extraction and Dimensionality Reduction Using Memristor Networks. Nano Lett 17, 3113-3118, doi:10.1021/acs.nanolett.7b00552 (2017).

[8] Sheridan, P. M. et al. Sparse coding with memristor networks. Nature Nanotechnology, doi:10.1038/nnano.2017.83 (2017).

[9] Jiang, H. et al. Sub-10 nm Ta Channel Responsible for Superior Performance of a HfO2 Memristor. Scientific Reports 6 (2016).

[10] Jiang, H. et al. Sub-10 nm Ta Channel Responsible for Superior Performance of a HfO2 Memristor. Sci Rep 6, 28525, doi:10.1038/srep28525 (2016).

[11] Li, C. et al. Analog signal and image processing with large memristor crossbars. (2018, accepted).

[12] Li, C. et al. Efficient and self-adaptive in-situ learning in multilayer memristive neural networks. Nature communications 9, 2385 (2018).

[13] Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484-489 (2016).

[14] Silver, D. et al. Mastering the game of Go without human knowledge. Nature 550, 354 (2017).

[15] Barto, A. G., Sutton, R. S. & Anderson, C. W. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE transactions on systems, man, and cybernetics, 834-846 (1983).

[16] Wang, Z. et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. Nature Materials 16, 101-108 (2017).

[17] Shi, Y. et al. Electronic synapses made of layered two-dimensional materials. Nature Electronics 1, 458 (2018).

[18] Midya, R. et al. Anatomy of Ag/Hafnia-Based Selectors with 1E10 Nonlinearity. Advanced Materials 29, 1604457 (2017).

[19] Jo, S. H. et al. Nanoscale Memristor Device as Synapse in Neuromorphic Systems. Nano Letters 10, 1297-1301, doi:10.1021/nl904092h (2010).

[20] Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D. & Wong, H. S. P. An Electronic Synapse Device Based on Metal Oxide Resistive Switching Memory for Neuromorphic Computation. IEEE Trans. Electron Devices 58, 2729-2737, doi:10.1109/TED.2011.2147791 (2011).

[21] Ohno, T. et al. Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. Nat. Mater. 10, 591-595, doi:10.1038/nmat3054 (2011).

[22] Wang, Z. Q. et al. Synaptic Learning and Memory Functions Achieved Using Oxygen Ion Migration/Diffusion in an Amorphous InGaZnO Memristor. Adv. Funct. Mater. 22, 2759-2765, doi:10.1002/adfm.201103148 (2012).

[23] Lim, H., Kim, I., Kim, J. S., Hwang, C. S. & Jeong, D. S. Short-term memory of TiO2-based electrochemical capacitors: empirical analysis with adoption of a sliding threshold. Nanotechnology 24, 384005, doi:10.1088/0957-4484/24/38/384005 (2013).

[24] La Barbera, S., Vuillaume, D. & Alibart, F. Filamentary Switching: Synaptic Plasticity through Device Volatility. ACS Nano 9, 941-949, doi:10.1021/nn506735m (2015).

[25] Prezioso, M. et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature 521, 61-64, doi:10.1038/nature14441, http://www.nature.com/nature/journal/v521/n7550/abs/nature14441.html - supplementary-information (2015).

[26] Kim, S. et al. Experimental Demonstration of a Second-Order Memristor and Its Ability to Biorealistically Implement Synaptic Plasticity. Nano Letters 15, 2203-2211, doi:10.1021/acs.nanolett.5b00697 (2015).

[27] Bi, G.-q. & Poo, M.-m. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. The Journal of neuroscience 18, 10464-10472 (1998).

[28] Wang, Z. et al. Fully memristive neural networks for pattern classification with unsupervised learning. Nature Electronics 1, 137-145, doi:10.1038/s41928-018-0023-2 (2018).

[29] Wang, Z. et al. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. Nat. Mater. 16, 101-108, doi:10.1038/nmat4756 (2016).

[30] Midya, R. et al. Anatomy of Ag/Hafnia-Based Selectors with 1010 Nonlinearity. Adv. Mater. 29, 1604457-n/a, doi:10.1002/adma.201604457 (2017).

[31] Jiang, H. et al. A novel true random number generator based on a stochastic diffusive memristor. Nat. Commun. 8, doi:10.1038/s41467-017-00869-x (2017).

[32] Serb, A. et al. Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses. Nat. Commun. 7, 12611, doi:10.1038/ncomms12611 (2016).

Appendix A – List of publications under the project

[1] Z. Wang, M. Rao, J.-W. Han, J. Zhang, P. Lin, Y. Li, C. Li, W. Song, S. Asapu, R. Midya, Y. Zhuo, H. Jiang, J. H. Yoon, N. K. Upadhyay, S. Joshi, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, Q. Qiu, R. S. Williams, Q. Xia*, and J. Joshua Yang*, "Capacitive neural network with neuro-transistors", NATURE COMMUNICATIONS 9, 3208 (2018).

[2] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. Joshua Yang*, and Q. Xia*, "Efficient and self-adaptive in-situ learning in multilayer memristive neural networks", NATURE COMMUNICATIONS 9, 2385 (2018).

[3] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, Y. Zhuo, H. Jiang, P. Lin, C. Li, J. H.. Yoon, N. K. Upadhyay, J. Zhang, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, R. Stanley Williams, Q. Xia, and J. Joshua Yang*, "Fully memristive neural networks for inference and unsupervised learning", NATURE ELECTRONICS 1, 137-145 (2018).

[4] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, Z. Li, J. P. Strachan*, P. Lin, W. Song, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. Joshua Yang*, Q. Xia*, "Analogue signal and image processing with large memristor crossbars", NATURE ELECTRONICS 1, 52-59 (2018).

[5] J. H. Yoon, Z. Wang, K. M. Kim, H. Wu, V. Ravichandran, Q. Xia*, C. S. Hwang and J. Joshua Yang*, "An Artificial Nociceptor Based on a Diffusive Memristor", NATURE COMMUNICATIONS 8, 417 (2018).

[6] M. Wang, S. Cai, C. Pan, C. Wang, X. Lian, K. Xu, Y. Zhuo, J. Joshua Yang*, P. Wang*, F. Miao*, "Ultra-robust memristors based on fully layered two-dimensional materials", NATURE ELECTRONICS 1, 130-136 (2018).

[7] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, Y. Li, P. Lin, M. Hu, N. Ge, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. Joshua Yang*, and Q. Xia*, "Long short-term memory networks in memristor crossbars", NATURE MACHINE INTELLIGENCE, Accepted (2018).

[8] H. Jiang, C. Li, R. Zhang, P. Yan, P. Lin, Y. Li, J. Joshua Yang*, D. Holcomb*, and Q. Xia*, "Provable Key Destruction with Large Memristor Crossbars", NATURE ELECTRONICS, Accepted (2018).

[9] Z. Wang, M. Rao, R. Midya, S. Joshi, H. Jiang, P. Lin, W. Song, S. Asapu, Y. Zhuo, C. Li, H. Wu*, Q. Xia*, and J. Joshua Yang*, "Threshold Switching of Ag or Cu in dielectrics: Materials, Mechanism, and Applications", ADVANCED FUNCTIONAL MATERIALS 28, 1704862 (invited feature article, 2018).

[10] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, M. Barnell, G-L Li, H. L. Xin, R. S. Williams, Q. Xia, and J. Joshua Yang*, "Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing", NATURE MATERIALS 16, 101-108 (2017).

[11] J. Joshua Yang* and Q. Xia, "Battery-like artificial synapses", *NATURE MATERIALS* 16, 396-397 (2017).

[12] R. Midya, Z. Wang, J. Zhang, C. Li, S. Joshi, H. Jiang, P. Lin, K. Norris, N. Ge, Q. Wu, M. Barnell, Z. Li, R. S. Williams, Q. Xia*, and J. Joshua Yang*, "Anatomy of Ag/hafnia based selectors with 10^{10} nonlinearity", ADVANCED MATERIALS 29, 1604457 (2017).

[13] J. H. Yoon, J. Zhang, X. Ren, Z. Wang, H. Wu, Z. Li, M. Barnell, Q. Wu, L. J. Lauhon, Q. Xia and J. Joshua Yang*, "Truly Electroforming-Free and low- Energy Memristors with

Pre-conditioned Conductive Tunneling Paths", ADVANCED FUNCTIONAL MATERIALS 27, 1702010 (2017).

[14] H. Jiang, D. Belkin, S. Savel'ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao, M. Barnell, Q. Wu, J. Joshua Yang*, Q. Xia*, "A novel true random number generator based on a stochastic diffusive memristor", NATURE COMMUNICATIONS 8, 882 (2017).

[15] C. Li, L. Han, H. Jiang, M. Jang, J. Joshua Yang, H. L. Xin and Q. Xia, "3-Dimensional Crossbar Arrays of Self-rectifying Si/SiO2/Si Memristors", NATURE COMMUNICATIONS 8, 15666 (2017).

[16] Z. Wang, H. Jiang, M. Jang, P. Lin, A. Ribbe, Qing Wu, Mark Barnell, Qiangfei Xia, and J. Joshua Yang* "Electrochemical Metallization Switching with a Platinum Group Metal in Different Oxides", NANOSCALE 8, 14023-14030 (2016).

[17] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, J. Joshua Yang, H. Xin, and Q. Xia, "Sub-10 nm Ta channel responsible for superior performance of a HfO_2 memristor", SCIENTIFIC REPORTS 6, 28525 (2016).

[18] N. Ge*, J. H. Yoon, M. Hu, E. J. Merced-Grafals, Z. Li, H. Holder, Q. Xia, R. S. Williams, X. Zhou, J. Joshua Yang*, "An efficient analog Hamming distance comparator based on a diagonal memristive crossbar array" SCIENTIFIC REPORTS 7, 40135 (2016).

[19] R. Zhang, H. Jiang, Z. Wang, P. Lin, Ye. Zhuo, D. Holcomb, D. Zhang, J. Joshua Yang, Q. Xia, "Nanoscale Diffusive Memristor Crossbars as Physical Unclonable Functions", NANOSCALE 10, 2721, (2018).

[20] Y. Li, Z. Wang, R. Midya, Q. Xia and J. Joshua Yang* "Review of memristor devices in neuromorphic computing: materials sciences and device challenges" invited review for special issue in Journal of Physics D: Applied Physics on brain-inspired pervasive computing 51,503002 (2018).

LIST OF SYMBOLES, ABBREVIATIONS, AND ACRONYMS

ACE	Analog Computing Engine
ADC	Analog-to-Digital Converter
ALD	Atomic Layer Deposition
ASIC	Application Specific Integrated Circuit
BEOL	Back End Of the Line
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
DAC	Digital-to-Analog Converter
DNN	Deep Neural Network
HRTEM	High-Resolution Transmission Electron Microscopy
ISR	Intelligence, Surveillance and Reconnaissance
IV	Current-Voltage
LoG	Laplacian of Gaussian
MNIST	Modified National Institute of Standards and Technology
MOS	Metal-Oxide-Semiconductor
PPD	Paired Pulse Depression
PPF	Paired Pulse Facilitation
RC	Resistance-Capacitance
ReLU	Rectified Linear Unit
RIE	Reactive Ion Etch
SD	Standard Deviation
SGD	Stochastic Gradient Descent
Si	Silicon
SRDP	Spike Rate Dependent Plasticity
STDP	Spike Timing Dependent Plasticity
TEOS	Tetraethyl Orthosilicate
TIA	Trans-Impedance Amplifier
TOPS/W	Tera Operations Per Second per Watt
VMM	Vector Matrix Multiplication
1T1M	One-Transistor-One-Memristor
1S1M	One-Selector-One-Memristor