Nanoscale

PAPER

View Article Online
View Journal

Cite this: DOI: 10.1039/c9nr06715a

Precision-extension technique for accurate vector–matrix multiplication with a CNT transistor crossbar array†

Sungho Kim,[a] Yongwoo Lee,[b] Hee-Dong Kim[a] and Sung-Jin Choi *[b]

Received 6th August 2019,
Accepted 27th October 2019

DOI: 10.1039/c9nr06715a

rsc.li/nanoscale

Most machine learning algorithms involve many multiply–accumulate operations, which dictate the computation time and energy required. Vector–matrix multiplications can be accelerated using resistive networks, which can be naturally implemented in a crossbar geometry by leveraging Kirchhoff's current law in a single readout step. However, practical computing tasks that require high precision are still very challenging to implement in a resistive crossbar array owing to intrinsic device variability and unavoidable crosstalk, such as sneak path currents through adjacent devices, which inherently result in low precision. Here, we experimentally demonstrate a precision-extension technique for a carbon nanotube (CNT) transistor crossbar array. High precision is attained through multiple devices operating together, each of which stores a portion of the required bit width. A 10 × 10 CNT transistor array can perform vector–matrix multiplication with high accuracy, making in-memory computing approaches attractive for high-performance computing environments.

Introduction

Deep neural networks (DNNs), which are broadly used in recent artificial intelligence applications, achieve outstanding performance when addressing traditionally difficult machine learning problems, such as recognizing hand-written digits, sounds, and images.[1] However, the number of DNN mathematical computations required dramatically increases as the network size increases. Unfortunately, conventional digital computing systems are facing computational-speed limitations owing to unavoidable data transfer inefficiency between processors and off-chip memory. This is referred to as the von Neumann bottleneck. Thus, computing-power efficiency stands as a critical obstacle for DNNs in a broad range of practical applications, especially those related to the Internet of Things and edge computing, which require a drastically lower energy consumption.[2]

DNN computations typically involve a large number of vector–matrix multiplication (VMM) operations, which is a heavy burden on traditional digital computing systems because their computational complexity grows as $O(n^2)$ and cannot be easily reduced.[3] To further accelerate and reduce the energy consumption of VMM computations, resistive networks with a crossbar geometry have been intensively studied using emerging devices capable of analog conductance switching (*e.g.*, memristors).[4–10] It was first demonstrated in the early 1950s[11] that VMM computations can be naturally implemented in a resistive crossbar array in a single readout step based on Ohm's law and Kirchhoff's current law. Building on early conceptual proposals for resistive crossbar arrays,[12,13] recent advances in resistive crossbar arrays, which are generally called dot-product engines (DPEs),[14,15] have enabled a variety of practical calculation tasks, such as sparse coding calculations,[4] *K*-means data clustering,[5] and differential equation solvers[10] (Fig. 1a).

However, many of the precision-related issues of DPEs are not trivial and need to be accounted for, such as the series resistances of wires, sneaky path currents, nonlinearity in the current–voltage relationship of resistive devices, and other unpredictable noise sources. Most importantly, the intrinsic variability of resistive devices, that is, cycle-to-cycle and device-to-device variations in their conductance modulation,[16] leads to the rapid degradation of VMM precision as the array size increases.[17,18] Unfortunately, this variability issue is common to almost all nano-electronic devices, including the two-term-

[a]Department of Electrical Engineering, Sejong University, Seoul 05006, Korea
[b]School of Electrical Engineering, Kookmin University, Seoul 02707, Korea.
E-mail: sjchoiee@kookmin.ac.kr

†Electronic supplementary information (ESI) available: Additional discussions about (1) the electrical properties of the CNT synaptic transistor with the switching mechanism, (2) the update-verify feedback method, (3) the additional explanation of the DPE operation, (4) selective access scheme in the crossbar array, and (5) the experimental setup for the DPE operation. See DOI: 10.1039/c9nr06715a

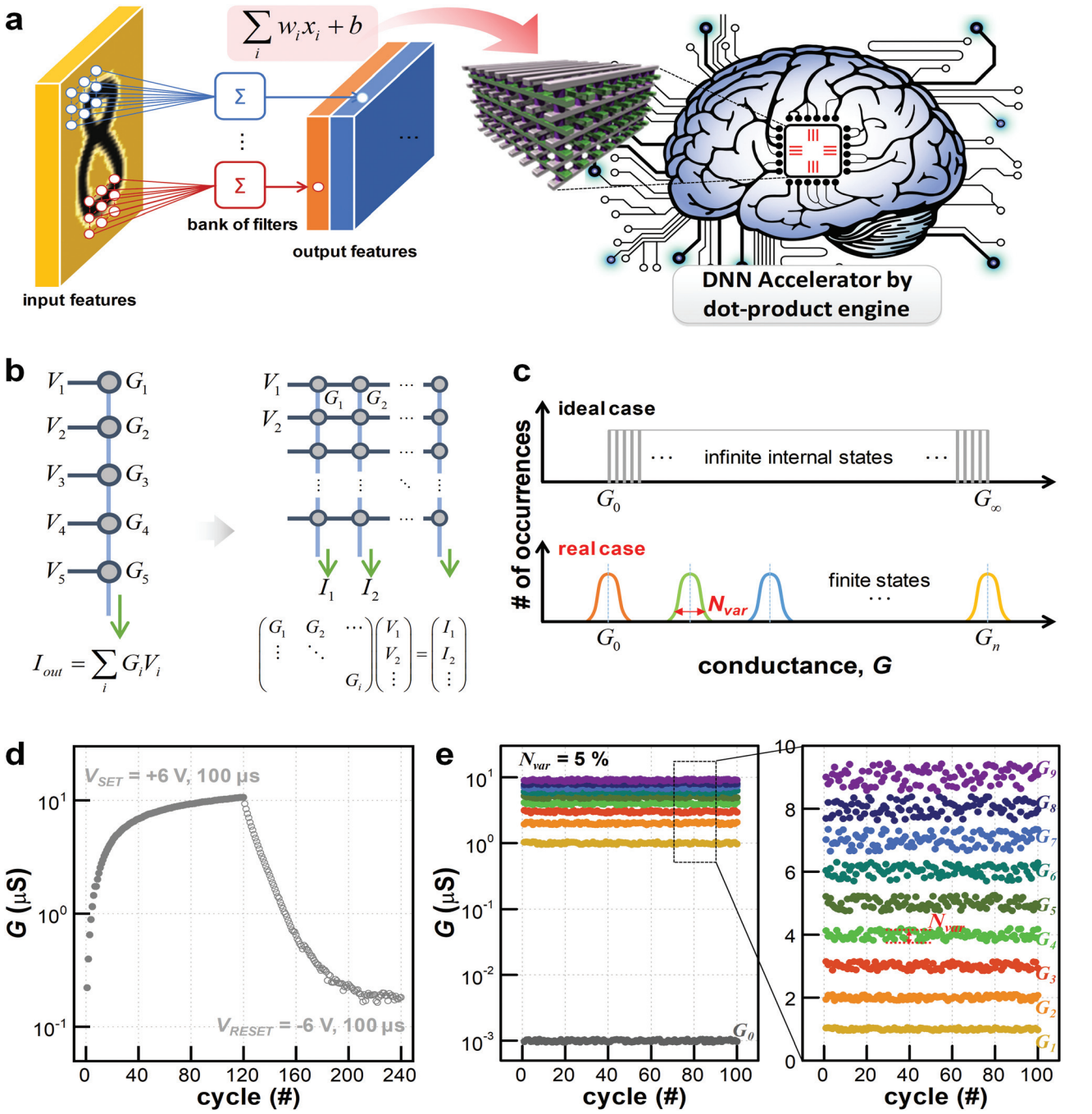

Fig. 1 (a) Conceptual schematic of a dot-product engine. A typical DNN computation involves many VMMs, and the dot-product engine allows for faster VMM computations with a lower energy consumption. (b) Schematic of the DPE operations in a resistive crossbar array. The desired matrix element values are represented by both the device conductance (G_i) and the input vector voltages (V_i). The output currents collected along each column (I_{out}) can yield the results of a VMM computation. (c) Ideal and real analog conductance states in a resistive device, along with their distribution due to device variability. (d) Measured conductance modulation behavior of the analog channel in a CNT transistor. Each pulse train consists of 120 SET and RESET pulses applied to the gate (V_{SET} = 6 V and V_{RESET} = −6 V for 100 µs), followed by small, nonperturbative read voltage pulses (−1 V for 100 µs) within the given intervals. In this case, V_D and V_S are set to V_G/2 and 0 V, respectively (see ESI Note 4†). (e) Ten measured distinguishable conductance states (G_0 to G_9) of one CNT transistor, where 100 iterations of write–read cycles were repeated for each conductance when the target N_{var} = 5% in the write–verify feedback method. Despite leveraging the write–verify process, a certain level of N_{var} cannot be avoided, which is intrinsic to analog switching.

inal resistive switches (*i.e.*, memristors) most commonly used in DPEs. This problem cannot be easily overcome by further optimizing the fabrication process or the materials used because the physical mechanism of conductance modulation is typically an atomic-level random process based on electro/thermodynamics.[19–21] A recent study[10] has presented a smart

breakthrough to the abovementioned problem. The intrinsically low precision of memristors can be extended through the use of multiple crossbars to represent a given number of bits, which can be used to perform high-precision VMM computations. Nevertheless, existing experimental demonstrations of DPEs using passive memristor crossbar arrays have been limited to relatively small sizes ($<16 \times 3$)[10] because eliminating the crosstalk from adjacent devices such as those with sneak path currents, which are a chronic problem of crossbar arrays, is very difficult. Common two-terminal memristors cannot effectively prevent unwanted current flow through unselected devices without additional selector devices; thus, the resulting voltage drops across the parasitic current paths critically reduce the accuracy of the system.[22,23]

A robust mapping scheme that allows for conversion from each matrix element (*i.e.*, real numbers) into device conductance or an input voltage signal is essential for improving the calculation accuracy and efficiency of VMM computations, which should have tolerance to the effects of device variability and the intrinsic drawbacks of crossbar arrays. We recently developed a carbon nanotube (CNT) synaptic transistor[24,25] that can eliminate the abovementioned limitations of the current memristor crossbar array technology. The resulting larger conductance variation margin (ΔG) allows for storing a larger bit width in a single device. In addition, the three individually controllable terminals with a localized carrier-trapping mechanism of the CNT transistor can effectively prevent crosstalk between adjacent devices. High-precision VMM computations can be performed by using multiple devices to store the required bit width and by leveraging a quantization process that uses analog-to-digital circuitry (ADC). We demonstrate VMM computations experimentally in 10×10 CNT transistor crossbar arrays, achieving 1×2 and 2×2 matrix multiplications without error.

Results and discussion

VMM computations can be performed in a crossbar array by applying an input vector of voltage signals (V_i) to the rows of the crossbar array (Fig. 1b). The resulting current signals (I_{out}) are collected along the columns; thus, I_{out} reflects the summed results of multiplying the input voltage by the device conductance (G_i) according to Kirchhoff's current law. In the case of our CNT transistor crossbar array (Fig. S1†), the input voltage signal is applied to the drain electrode of each transistor ($V_i = V_D$) in the row direction, and the collected source currents ($\sum I_S$) in the column direction represent the integrated multiplication results between the conductance (G) of each transistor and V_D (*i.e.*, $\sum I_S = \sum V_D \cdot G_i$). During the VMM computation, a constant read voltage is applied to the gate electrode ($V_G = -1$ V), whereas different levels of SET/RESET voltages (V_{SET} and V_{RESET}) are applied to the gate electrode to respectively increase/decrease G when an updating G is required. This adjustable G in the CNT transistor is due to the electron trap states near the valence band provided by CNT-hydroxide complexes at the SiO_2/CNT interface,[26] and details of this mechanism and the associated retention/endurance properties are discussed in ESI Note 1.†

The amplitude of V_i and G will correspond to the elements of the matrix (*i.e.*, the real number) that we want to calculate. The mapping of matrix elements to V_i is a relatively easy task using conventional high-precision digital-to-analog circuits (DACs). On the other hand, the mapping of matrix elements to G poses several issues that have to be addressed. Ideally, resistive devices such as memristors have infinite internal conductance states, and thus any real number can be represented by one of the device's conductance values (Fig. 1c). However, the precise adjustment of conductance is very challenging and although several methods for precise adjustment have been proposed,[27–29] they require impractical and complex peripheral circuitry with limited adjustment accuracy. This control limitation results in inevitable conductance variation (*i.e.*, normalized variation $N_{var} = [G_{max} - G_{min}]/\text{mean}(G)$). Consequently, only finite conductance states that do not overlap with each other (referred to as G_0 to G_n in Fig. 1c) can be used. For example, Fig. 2d shows the analog G modulation behavior of our CNT transistor, in which G can be adjusted gradually by repeatedly applying voltage pulses. However, as noted above, this gradual change cannot guarantee an infinite number of available conductance states. Despite exploiting the write–verify feedback method[27] (see Methods and ESI Note 2†), only a finite number of conductance states, namely from G_0 to G_9 ($n_{state} = 10$), are available owing to unavoidable N_{var}, as shown in Fig. 1e (the reason why G_0 is particularly lower than the other states will be explained later). Although a higher n_{state} could be achieved by lowering the predetermined range of the target N_{var} in the update–verify feedback method, the number of required update–verify pairs would dramatically increase as N_{var} decreased, which would lead to an impractical efficient energy consumption in VMM computations. Accordingly, a robust mapping scheme capable of representing an infinite real number (matrix element) with only a finite number of conductance states is essential for accurate VMM computation.

A previously proposed robust mapping scheme uses a set of multiple devices to store a portion of the required bit width.[10] In principle, this scheme can represent an m-digit number with a base-l number system, $X = [x_m\ x_{m-1}\ \ldots\ x_1]_l$, where x_m is the m^{th} digit, x_{m-1} is the $m-1^{th}$ digit, and so on, as shown in Fig. 2a. Because X is in a base-l system, each digit (x_i) is a number between 0 and $l-1$. X can be expressed as a linear combination of each digit with a digit shifter as follows:

$$X = l^{m-1}x_m + l^{m-2}x_{m-1} + \ldots + l^0 x_1 = \sum_{i=1}^{m} l^{i-1}x_i, \qquad (1)$$

where l is a digit shifter, *e.g.*, l^1 and l^2 denote single- and double-digit shifters, respectively. Here, if the n_{state} of the resistive device is equal to the value of l, then x_i can be mapped to one of the available conductance states. For example, if $X = [1734]_{10}$ and $n_{state} = 10$ (*i.e.*, G_0 to G_9), each digit (x_4, x_3, x_2, and x_1) can be expressed by the conductance state of four

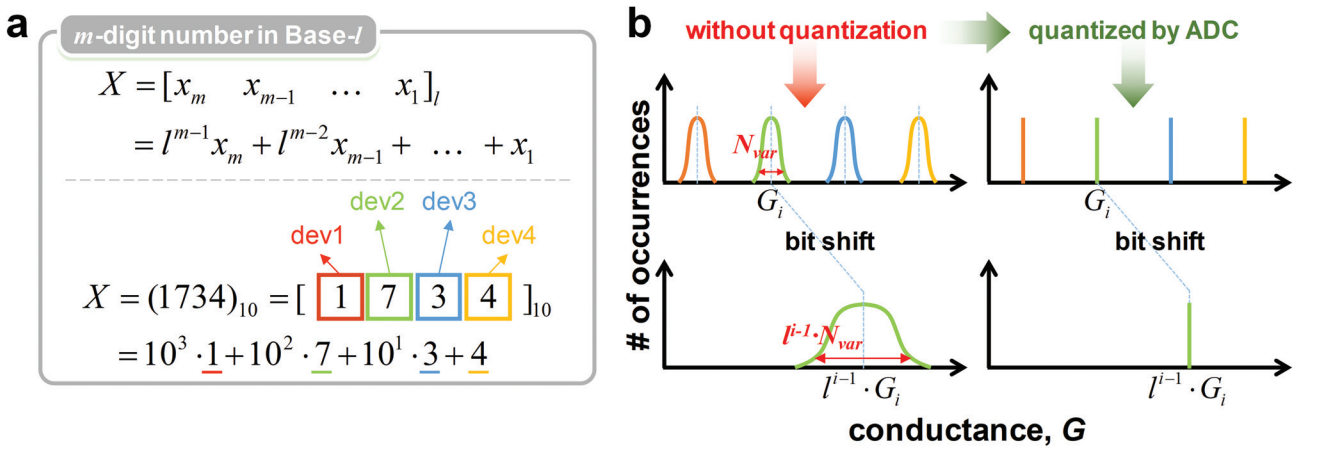

Fig. 2 (a) The proposed mapping scheme consists of a set of multiple devices that store a portion of the required bit width and is applied for an *m*-digit number in a base-*l* number system. (b) Without the quantization process performed by an ADC, the effect of N_{var} is amplified, which leads to error in DPE operation.

devices, *i.e.*, G_1, G_7, G_3, and G_4, respectively. As a result, $X = [1734]_{10}$ can be represented as the total conductance of four devices, *i.e.*, $X \rightarrow 10^3G_1 + 10^2G_7 + 10^1G_3 + 10^0G_4$. This scheme provides a precise way to represent any real number using a set of resistive devices with only a finite number of conductance states. However, one more critical issue should be addressed. As mentioned above, there is no way to accurately adjust the device conductance as desired; any measured conductance state, such as G_1, is actually $G_1 + N_{var}$. Note that the effect of N_{var} is dramatically amplified by the digit shifter l^{i-1}. Therefore, the measured conductance value needs to be quantized before the digit-shift operation is carried out (Fig. 2b). Otherwise, the error introduced by N_{var} would degrade the precision of the final computation. Fortunately, the quantization operation can be readily implemented in the existing circuitry through the ADC, which enables this robust mapping scheme to perform properly.

Fig. 3a shows a simple example of a multiplication operation using the proposed mapping scheme; two decimal integers, *X* and *Y*, are multiplied. In the following discussion, all *X* and *Y* are assumed to be positive values; the sign determination process for the negative matrix element is discussed in ESI Note 3.† Each number has a single digit (*e.g.*, $x_1 = 8$ and $y_1 = 4$). Because these two numbers (*X* and *Y*) are assumed to be decimal values in this example (*i.e.*, $l = 10$), n_{state} should be 10. In step 1, the amplitude of the input voltage (V_i) applied to the row of the crossbar array is determined by an integer multiple of the amplitude when $y_1 = 1$. For example, if V_i is 0.1 V when $y_1 = 1$, then V_i will be 0.4 V when $y_1 = 4$. In our CNT transistor crossbar array, the maximum V_i is limited to 1 V because V_i is applied to the drain electrode of each transistor. Accordingly, $y_1 = 0, 1, \dots 9$ correspond to $V_i = 0$ V, 0.1 V, ... 0.9 V, respectively. Similarly, the conductance of the device is adjusted according to x_1; G_2 to G_9 should be integer multiples of G_1. For example, if G_1 is 1 μS when $x_1 = 1$, then G_8 will be 8 μS when $x_1 = 8$. As for the conductance variation margin ($\Delta G = G_{max}/G_{min}$) of our CNT transistor and as shown in Fig. 1d, $x_1 = 1, 2, \dots$ and 9 correspond to $G_i = 1$ μS, 2 μS, ... and 9 μS, respectively. It should be noted that G_0 (10^{-3} μS) should be as low as possible. Although G_0 represents $x_1 = 0$, because the device conductance cannot be exactly zero, lowering the value of G_0 as much as possible can accordingly reduce the resulting error. In step 2, the multiplication result is obtained by measuring the output current (I_{out}), which is the source current (I_S) of the CNT transistor. In this example, a current of $0.4(G_8 + N_{var})$ is measured. The maximum measurable current is theoretically $0.9G_9$, which is the result of $9 \times 9 = 81$, when $x_1 = 9$ and $y_1 = 9$. Based on this fact, the linear relationship between $x_1 \cdot y_1$ and I_{out} can be determined, which determines the scale ratio *R* (see also ESI Note 3†). In step 3, the actual multiplication result can be inferred from the measured value of I_{out}, by rescaling I_{out} with *R*. Unfortunately, owing to the several unpredictable noise sources, such as series resistances in wires, sneaky path currents, and N_{var}, the inferred multiplication result is still inaccurate. In this example, the inferred result is $32 \pm \alpha$, where α refers to the error. Thus, in the last step, the error is eliminated by using the ADC, and an accurate multiplication result can finally be obtained.

This single-digit multiplication process can be easily expanded to enable the multiplication of multidigit numbers. We now present another example of a DPE operation (shown in Fig. 3b) involving the multiplication of two decimal integers *X* and *Y*, with each number having three digits. The only difference with the one-digit-number multiplication discussed above is that the digit-shift operation is also performed. Because the ADC eliminates the error during one-digit-number multiplication operations, subsequent digit-shift operations cannot produce any error. To physically implement this multiplication operation in the crossbar array (Fig. 3c), a set of resistive devices representing the number *X* repeatedly occupy the rows of the crossbar array as many times as the number of digits of *X*, and each set is shifted by one column. Then,

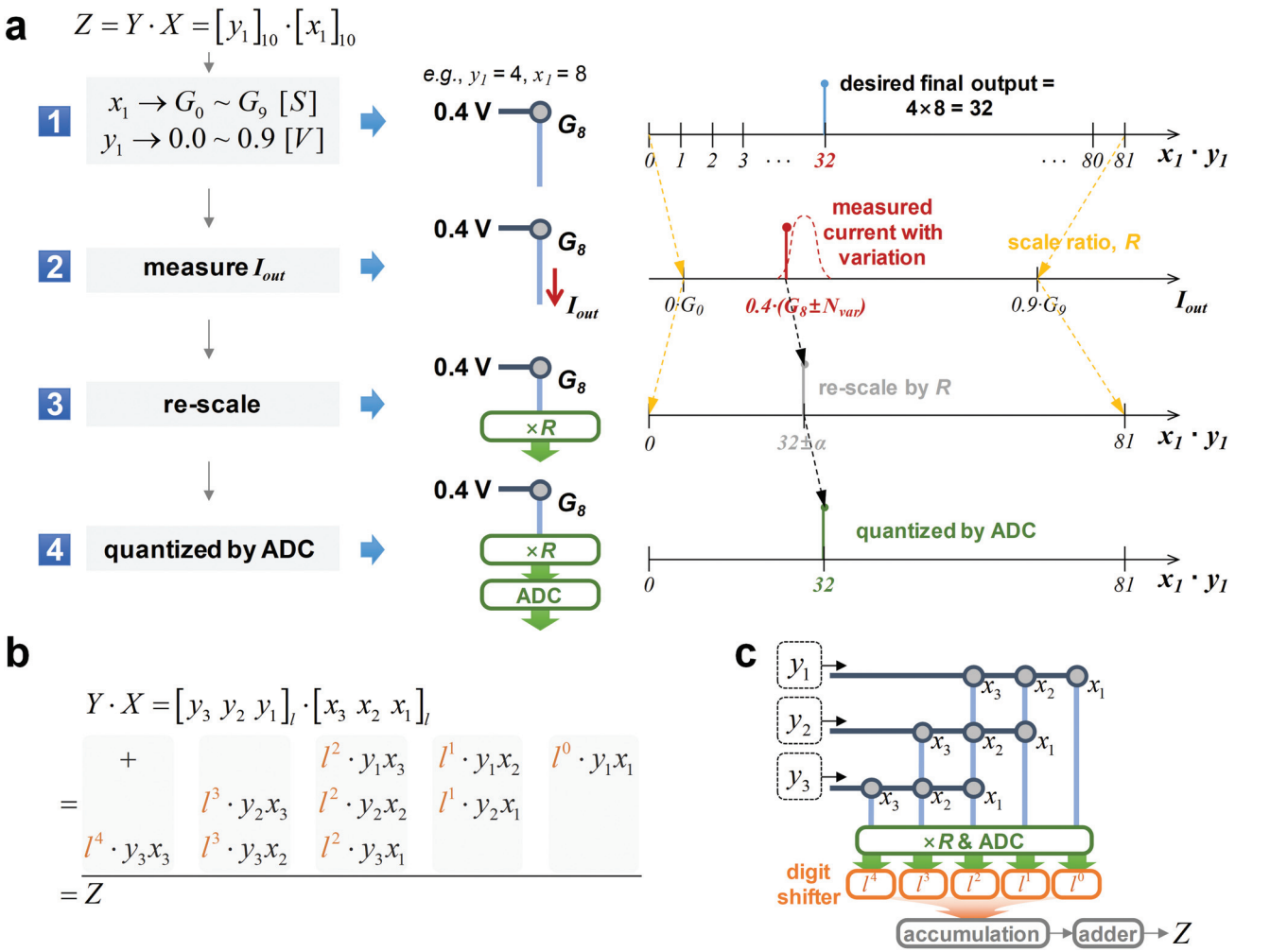

Fig. 3 (a) First example of a DPE operation, in which two decimal integers *X* and *Y* are multiplied, with each number having a single digit (*e.g.*, $x_1 = 8$ and $y_1 = 4$). (b) Second example of a DPE operation, in which the multiplication of two decimal integers *X* and *Y* can be expressed as a combination of the multiplications of each digit using a digit shifter. (c) Schematic of the physical implementation of the DPE in a crossbar array with peripheral circuitry. The input voltages corresponding to y_1, y_2, and y_3 are sequentially applied. The outputs after rescaling and passing through the ADC and the digit shifter are accumulated along each column. Finally, the multiplication result (*Z*) can be inferred from the summation of all the accumulated results.

voltage V_1 corresponding to y_1 is applied to the 1st row, and the output currents in each column are accumulated *via* rescaling, the ADC, and the digit shifter. When the equivalent process is sequentially performed for the 2nd and 3rd rows, the final multiplication result *Z* can be obtained by adding the accumulated results through the adder.

In fact, the above-discussed mapping scheme is applicable to any type of resistive crossbar array; in other words, it is independent of the device structure or conductance switching mechanism of the resistive devices used in the crossbar array. Nonetheless, our three-terminal CNT transistor has more advantages for this mapping scheme than existing two-terminal memristors because of its larger ΔG and its capability to eliminate crosstalk among adjacent devices. First, because memristors generally have a ΔG below 10,[30–33] n_{state} is inevitably small and the subsequent gap between each conductance state should be narrowed, which would result in N_{var} being more susceptible to DPE operation. Alternatively, when using three-terminal resistive devices such as our CNT transistor or NAND flash memory, ΔG is typically greater than 50 with higher reliability. This larger ΔG allows for a greater n_{state}, and thus a larger bit width can be stored in a single device. Therefore, three-terminal resistive devices enable more accurate and reliable VMM computations by simply lowering the number of resistive devices needed. Secondly, the three individually controllable terminals of the CNT transistor possess a localized carrier trapping mechanism and can effectively prevent crosstalk between adjacent devices (see ESI Note 4†). By properly controlling the gate, drain, and source voltages, it is possible to prevent unselected devices from being disturbed by the update/reading processes of *G* in the adjacent selected device (Fig. S4†); thereby, the three-terminal structure of the transistor can easily eliminate unwanted current flows through unselected devices without requiring an additional selector

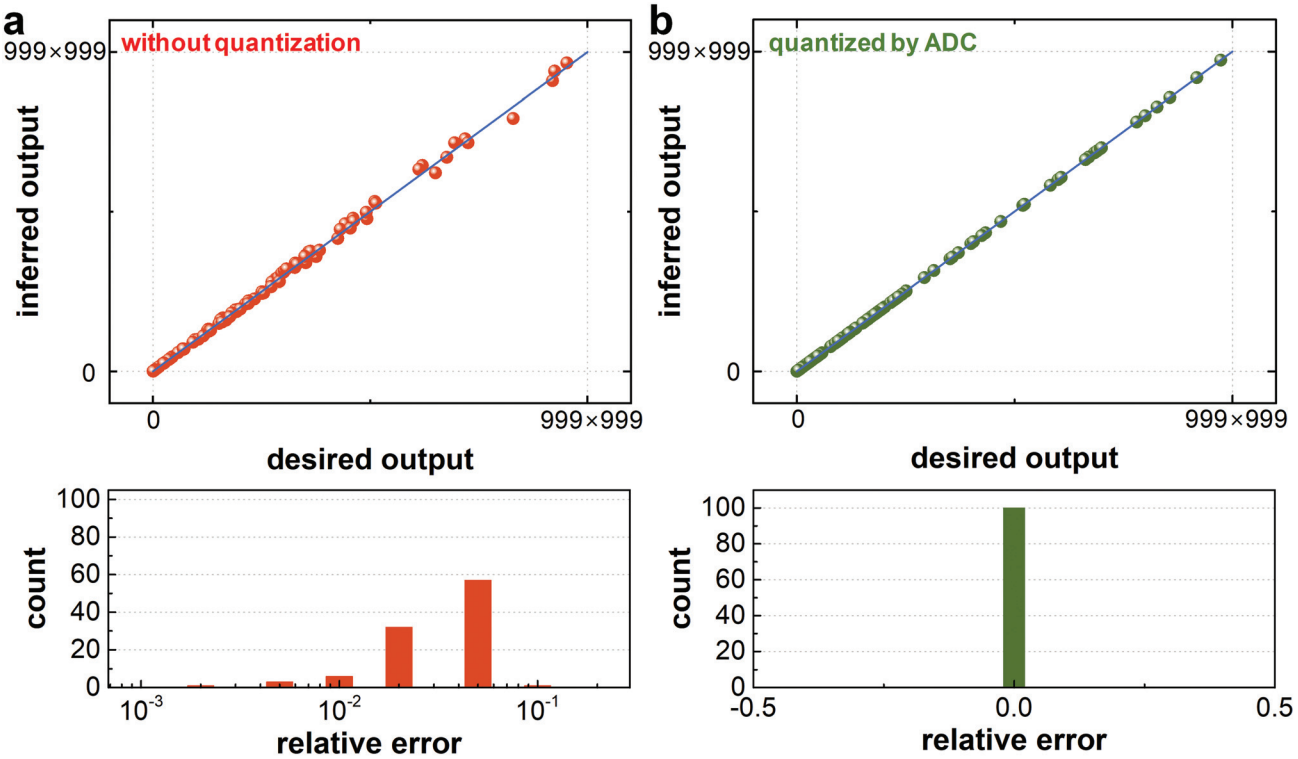

Fig. 4 Experimental result of (1 × 2) and (2 × 2) matrix multiplication. The output matrix has a size of (1 × 2); thus two matrix elements, Z_1 and Z_2, are obtained. (a) The 100 examples of Z_1 and Z_2 obtained without the quantization process. (b) The 100 examples of Z_1 and Z_2 obtained with the quantization process. Relative error is eliminated completely.

device or any intrinsic rectifying behavior in the resistive devices, unlike the schemes presented in previous studies.[34,35]

We now demonstrate VMM computation experimentally using a 10 × 10 CNT transistor crossbar array, achieving (1 × 2) and (2 × 2) matrix multiplication (see ESI Note 5†). Because n_{state} = 10 in our CNT transistor, all matrix elements are randomly assigned decimal integers with three digits (the number of digits was limited by the array size). Fig. 4a and b show the experimental results of the VMM computation, in which a total of 100 multiplication operations were performed. The relative error of each multiplication was obtained. Note that without the quantization process (Fig. 4a), the error always occurred in the multiplication operations. This error can be completely eliminated *via* the quantization process (Fig. 4b).

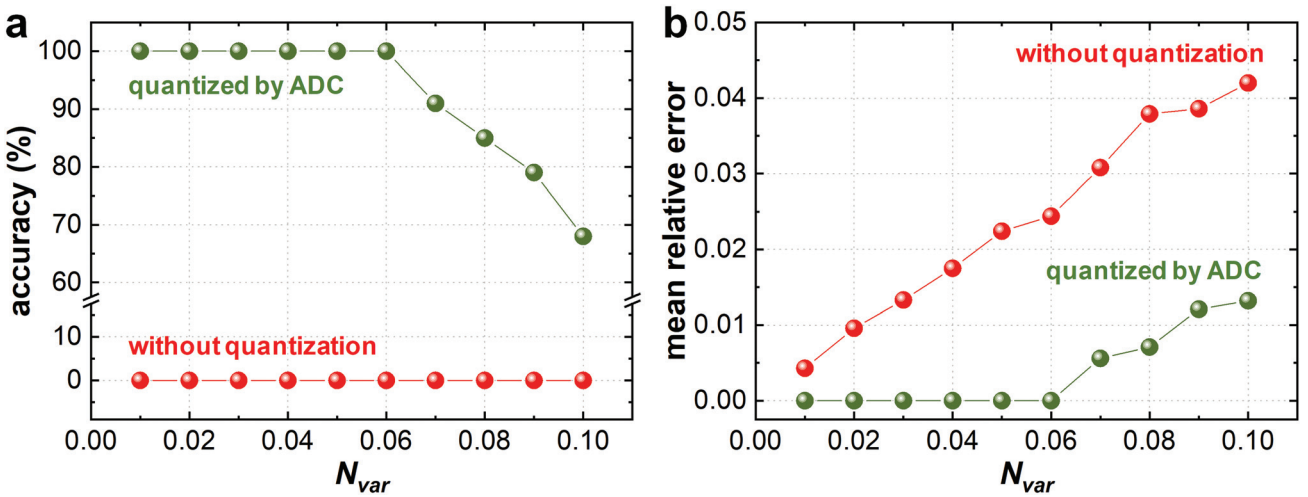

Fig. 5 (a) Percentage of accurate computations for 100 multiplication operations for different N_{var} values. (b) Mean relative error for 100 multiplication operations for different N_{var} values.

Consequently, the proposed precision-extension technique for DPEs based on the above-described robust mapping scheme and quantization process could be employed as a promising DNN accelerator by performing high-precision VMM computations.

The remaining issues to be addressed are how to control N_{var} and the effect of N_{var} on the accuracy of VMM computations. As shown in Fig. 5a and b, error occurs even if the ADC is used when N_{var} becomes larger than 6% because each conductance state (G_i + N_{var}) is overlapped with other states. There are several solutions to this issue. The easiest approach is to lower the target N_{var} value of the write–verify process. However, as mentioned above, lowering the target N_{var} requires a larger number of write–verify pairs, which results in greater energy consumption. Another approach is the optimization of the conductance modulation behavior at the device level. In two-terminal memristors, cycle-to-cycle or device-to-device variation can be improved *via* engineering of the electrode material, interfacial layer, doping, or resistive switching material in order to reduce N_{var}.[16,36] However, further optimizing the fabrication process or the materials used in memristors cannot entirely eliminate N_{var} because of the random physical mechanism of analog conductance modulation. A more practical approach is to increase ΔG. When using three-terminal resistive devices, ΔG can be increased *via* engineering of the floating gate.[25] Thus, three-terminal resistive devices have promising potential for accurate and energy-efficient DPE operation.

Conclusion

We have experimentally demonstrated DPE operation in a CNT transistor crossbar array that can compute vector–matrix multiplications. Despite the intrinsic low precision owing to device variability, the precision-extension techniques discussed here can effectively lead to completely error-free computations. Furthermore, although the proposed techniques are equally applicable to any type of resistive crossbar array, our three-terminal CNT transistor exhibits more promising potential. Although recent advances in both 1 transistor–1 memristor structure-based arrays (1T1R arrays)[37–39] and 1 selector–1 memristor structure-based arrays (1S1R arrays)[40–42] can solve chronic problems in the existing passive memristor crossbar array, a unique advantage of our CNT transistor is its simpler structure, as fewer materials and fabrication process steps are required than when implementing 1T1R or 1S1R structure. Therefore, although the existing memristors or 1T1R or 1S1R may have better performance at the device level, our three-terminal synaptic transistor will be more advantageous in implementing and operating highly integrated DPEs. The demonstrated computing accuracy is acceptable for practical machine learning applications; the present work provides an experimental baseline for future analog computing systems and demonstrates their potential for accelerating machine learning operations while requiring a lower energy consumption.

Methods

Fabrication of CNT transistor crossbar array

CNT transistors were fabricated on p-doped rigid silicon substrates with a thermally grown 50 nm thick SiO_2 layer. We used the local back-gate structure for the modulation of the channels in the CNT transistors. To form the local back-gate, a 20 nm thick Ti layer was deposited by e-beam evaporation and patterned by a subsequent lift-off process. Next, a 40 nm thick Al_2O_3 layer and a 10 nm thick SiO_2 layer were deposited sequentially as a gate insulator by atomic layer deposition. Then, the top surface of the SiO_2 layer was functionalized with a 0.1 g mL^{-1} poly-L-lysine solution for 20 min to form an amine-terminated layer, which acted as an effective adhesion layer for the deposition of the CNTs. Then, the CNT network channel was formed by immersing the chip into a 0.01 mg mL^{-1} 99%-semiconducting CNT solution (NanoIntegris, Inc.) for 8 min at an elevated temperature of 100 °C. Next, the source/drain electrodes consisting of Ti and Pd layers (each 2 nm and 30 nm, respectively) were deposited and patterned using conventional thermal evaporation and a lift-off process, respectively. Finally, additional photolithography and oxygen plasma etching steps were conducted to remove unnecessary CNTs other than in the channel area, thus isolating the devices from one another.

For the crossbar array, 80 nm thick Cu and 150 nm thick SiO_x were sequentially deposited and patterned for the metal line and interlayer dielectric layer (ILD), respectively.

Update–verify process

To reduce the device variability, we used an update–verify technique to write and update the conductance of each device in the crossbar. Specifically, each write operation is based on a sequence of update–read pulse pairs, each pair including an updating (SET or RESET) pulse and a subsequent READ pulse (V_G = −2 V, 100 μs) for verification purpose. Current from the READ operation on a target cell is used to compare with a target value and calculate an error. If the error is below a predefined threshold, the operation is considered complete and the process stopped, otherwise operations are taken based on the sign of the error. For positive errors, a SET pulse (V_G = 6 V, 100 μs) is applied to increase the device conductance, while for negative errors, a RESET pulse (V_G = −6 V, 100 μs) is applied to decrease the device conductance. The procedure is then repeated until the conductance reaches within a predetermined range of the target value (for example N_{var} = 5%). In the experimental implementation, the updating of device conductance typically requires around 20 update–verify pairs in a sequence.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This research was supported by the Nano Material Technology Development Program (2016M3A7B4910430) funded by the Ministry of Science, ICT and Future Planning, research programs supported by the National Research Foundation of Korea (NRF) grant (2019R1A2C1002491, 2019R1A2B5B01069988, and 2016R1A5A1012966), and the Future Semiconductor Device Technology Development Program (Grant 10067739) funded by MOTIE (Ministry of Trade, Industry & Energy) and KSRC (Korea Semiconductor Research Consortium).

References

1 Y. LeCun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436–444.
2 R. S. Williams, *Comput. Sci. Eng.*, 2017, **19**, 7–13.
3 G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Univ Press, 2013.
4 P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang and W. D. Lu, *Nat. Nanotechnol.*, 2017, **12**, 784–789.
5 Y. Jeong, J. Lee, J. Moon, J. H. Shin and W. D. Lu, *Nano Lett.*, 2018, **18**, 4447–4453.
6 A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams and V. Srikumar, in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, IEEE, 2016, pp. 14–26.
7 M. Hu, R. S. Williams, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge and J. J. Yang, in *Proceedings of the 53rd Annual Design Automation Conference on - DAC '16*, ACM Press, New York, New York, USA, 2016, pp. 1–6.
8 Y. Jeong, M. A. Zidan and W. D. Lu, *IEEE Trans. Nanotechnol.*, 2018, **17**, 184–193.
9 X. Guo, F. M. Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do and D. B. Strukov, in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, IEEE, 2017, pp. 1–4.
10 M. A. Zidan, Y. Jeong, J. Lee, B. Chen, S. Huang, M. J. Kushner and W. D. Lu, *Nat. Electron.*, 2018, **1**, 411–420.
11 G. Liebmann, *J. Appl. Phys.*, 1950, **1**, 92–103.
12 D. B. Strukov and K. K. Likharev, *Nanotechnology*, 2005, **16**, 888–900.
13 J. Hasler and B. Marr, *Front. Neurosci.*, 2013, **7**, 118.
14 M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia and J. P. Strachan, *Adv. Mater.*, 2018, **30**, 1705914.
15 M. Hu, J. P. Strachan, Z. Li and R. S. Williams, in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, IEEE, 2016, pp. 374–379.
16 D. Kuzum, S. Yu and H.-S. Philip Wong, *Nanotechnology*, 2013, **24**, 382001.
17 P. Gu, B. Li, T. Tang, S. Yu, Yu Cao, Y. Wang and H. Yang, in *The 20th Asia and South Pacific Design Automation Conference*, IEEE, 2015, pp. 106–111.
18 M. Hu, H. Li, Q. Wu and G. S. Rose, in *Proceedings of the 49th Annual Design Automation Conference on - DAC '12*, ACM Press, New York, New York, USA, 2012, p. 498.
19 D. B. Strukov and R. S. Williams, *Appl. Phys. A*, 2009, **94**, 515–519.
20 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. J. Yang, *Nat. Mater.*, 2017, **16**, 101–108.
21 S. Kim, S. Choi and W. Lu, *ACS Nano*, 2014, **8**, 2369–2376.
22 C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang and Q. Xia, *Nat. Commun.*, 2018, **9**, 2385.
23 C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang and Q. Xia, *Nat. Electron.*, 2018, **1**, 52–59.
24 S. Kim, J. Yoon, H. D. Kim and S. J. Choi, *ACS Appl. Mater. Interfaces*, 2015, **7**, 25479–25486.
25 S. Kim, B. Choi, M. Lim, J. Yoon, J. Lee, H.-D. Kim and S.-J. Choi, *ACS Nano*, 2017, **11**, 2814–2822.
26 D. L. Duong, S. M. Lee and Y. H. Lee, *J. Mater. Chem.*, 2012, **22**, 1994–1997.
27 F. Alibart, L. Gao, B. D. Hoskins and D. B. Strukov, *Nanotechnology*, 2012, **23**, 75201–75207.
28 L. Gao, F. Alibart and D. B. Strukov, in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, IEEE, 2012, pp. 88–93.
29 L. Gao, P.-Y. Chen and S. Yu, *IEEE Electron Device Lett.*, 2015, **36**, 1157–1159.
30 S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. H. R. Lee, B. H. R. Lee, B. H. R. Lee and H. Hwang, in *Technical Digest - International Electron Devices Meeting, IEDM*, IEEE, 2013, pp. 25.6.1–25.6.4.
31 G. W. Burr, R. M. Shelby, C. di Nolfo, J. W. Jang, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. Kurdi and H. Hwang, in *2014 IEEE International Electron Devices Meeting*, IEEE, 2014, pp. 29.5.1–29.5.4.
32 P. Y. Chen, B. Lin, I. T. Wang, T. H. Hou, J. Ye, S. Vrudhula, J. S. Seo, Y. Cao and S. Yu, in *2015 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2015*, IEEE, 2016, pp. 194–199.
33 S. Yu, P.-Y. Y. Chen, Y. Cao, L. Xia, Y. Wang and H. Wu, in *Technical Digest - International Electron Devices Meeting, IEDM*, IEEE, 2015, pp. 17.3.1–17.3.4.
34 E. J. Fuller, S. T. Keene, A. Melianas, Z. Wang, S. Agarwal, Y. Li, Y. Tuchman, C. D. James, M. J. Marinella, J. J. Yang, A. Salleo and A. A. Talin, *Science*, 2019, **364**, 570–574.
35 P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong and H. Qian, *Nat. Commun.*, 2017, **8**, 15199.
36 D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi and C. S. Hwang, *Adv. Electron. Mater.*, 2016, **2**, 1600090.
37 C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan,

M. Barnell, Q. Wu, R. S. Williams, J. J. Yang and Q. Xia, *Nat. Commun.*, 2018, **9**, 2385.
38 F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn and W. D. Lu, *Nat. Electron.*, 2019, **2**, 290–299.
39 Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P. Strachan, N. Ge, M. Barnell, Q. Wu, A. G. Barto, Q. Qiu, R. S. Williams, Q. Xia and J. J. Yang, *Nat. Electron.*, 2019, **2**, 115–124.
40 C. C. Hsieh, Y. F. Chang, Y. Jeon, A. Roy, D. Shahrjerdi and S. K. Banerjee, *IEEE Electron Device Lett.*, 2017, **38**, 871–874.
41 Y. C. Chen, S. T. Hu, C. Y. Lin, B. Fowler, H. C. Huang, C. C. Lin, S. Kim, Y. F. Chang and J. C. Lee, *Nanoscale*, 2018, **10**, 15608–15614.
42 L. Sun, Y. Zhang, G. Han, G. Hwang, J. Jiang, B. Joo, K. Watanabe, T. Taniguchi, Y.-M. Kim, W. J. Yu, B.-S. Kong, R. Zhao and H. Yang, *Nat. Commun.*, 2019, **10**, 3161.