ADVANCED ELECTRONIC MATERIALS Open Access
www.advelectronicmat.de

Random Projection-Based Locality-Sensitive Hashing in a Memristor Crossbar Array with Stochasticity for Sparse Self-Attention-Based Transformer

*Xinxin Wang, Ilia Valov, and Huanglong Li**

Self-attention mechanism is critically central to the state-of-the-art transformer models. Because the standard full self-attention has quadratic complexity with respect to the input's length L, resulting in prohibitively large memory for very long sequences, sparse self-attention enabled by random projection (RP)-based locality-sensitive hashing (LSH) has recently been proposed to reduce the complexity to O(L log L). However, in current digital computing hardware with a von Neumann architecture, RP, which is essentially a matrix multiplication operation, incurs unavoidable time and energy-consuming data shuttling between off-chip memory and processing units. In addition, it is known that digital computers simply cannot generate provably random numbers. With the emerging analog memristive technology, it is shown that it is feasible to harness the intrinsic device-to-device variability in the memristor crossbar array for implementing the RP matrix and perform RP-LSH computation in memory. On this basis, sequence prediction tasks are performed with a sparse self-attention-based Transformer in a hybrid software-hardware approach, achieving a testing accuracy over 70% with much less computational complexity. By further harnessing the cycle-to-cycle variability for multi-round hashing, 12% increase in the testing accuracy is demonstrated. This work extends the range of applications of memristor crossbar arrays to the state-of-the-art large language models (LLMs).

1. Introduction

Attention is an integral part of the cognitive functions. It is a means for animals and humans to quickly select high-value information from massive information with limited information processing resources.[1,2] Its biological significance has inspired researchers to introduce attention mechanisms into artificial intelligence (AI) systems for better performance. Currently, attention mechanisms have become an important part of the compelling large language models (LLMs) in various tasks, allowing handling long-range dependencies between input sequence elements.

AI is now undergoing a paradigm shift with the rise of the Transformer LLMs[3] that have been seen as the foundation models.[4] Unlike the conventional recurrent[5,6] or convolutional[7] neural networks, the Transformer architecture enables parallel processing by solely using attention mechanisms, dispersing with recurrence and convolutions entirely. The success of the Transformer models reasserts the importance of attention mechanisms.

With increasing real-world demands, the Transformer models are being used on increasingly long sequences.[8–10] Unfortunately, the vanilla Transformer model does not scale very well to long sequence lengths because of the quadratic complexity with respect to the length L, rendering it prohibitively memory-intensive and practically trainable only in large industrial research laboratories. Specifically, the standard self-attention used in the Transformer models is full self-attention that requires the query vector to compare to all key vectors (query = key for

X. Wang, H. Li
Department of Precision Instrument, Center for Brain Inspired Computing Research
Tsinghua University
Beijing 100084, P. R. China
E-mail: li_huanglong@mail.tsinghua.edu.cn

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aelm.202300850

DOI: 10.1002/aelm.202300850

I. Valov
Forschungszentrum Jülich, Institute of Electrochemistry and Energy System
WilhelmJohnen-Straße
52426 Jülich, Germany

I. Valov
"Acad. Evgeni Budevski" IEE-BAS
Bulgarian Academy of Sciences (BAS)
Acad. G. Bonchev Str, Block 10, Sofia 1113, Bulgaria

H. Li
Chinese Institute for Brain Research
Beijing 102206, P. R. China

self-attention).[3] It has been estimated that training such a model on sequences of length 64 K even at a batch size of 1 and in 32-bit floats would take 16 GB of memory,[11] which is impractical and has hindered the use of the Transformer models for long sequences. In this context, introducing sparsity in the attention layers has become one of the main solutions to this problem. Among these solutions is the one proposed by Kitaev et al.,[11] using random projection (RP)-based locality-sensitive hashing (LSH) where queries and keys are hashed into several buckets, with similar items falling in the same bucket with high probability. As such, the full self-attention pattern can be approximated by only allowing attention within each bucket, reducing the complexity to O(L log L).

In practice, however, models with sparse self-attention using RP-LSH are quite slow,[12–15] despite their improved memory efficiency. The low speed is mainly attributed to the time taken by hashing.[15] Optimizing the tradeoff between memory usage and speed is of great importance for RP-LSH sparse self-attention if we want it to be useful.

The low speed of RP-LSH sparse self-attention can largely be understood from a hardware perspective. GPU, as the main engine behind the state-of-the-art AI models, still suffers from the so-called von Neumann bottleneck that frequent data shuttling between off-chip memory and processing units is unavoidable during information processing. RP is by nature a matrix multiplication operation. Matrix multiplication is a traditionally intense mathematical operation for the conventional processors. It requires high memory allocations, plus at least one multiply and add per cell. In the specific case of RP, the dot products of a d-dimensional feature vector (query) and k random vectors generated from a certain distribution, respectively, are computed and concatenated to obtain the hash value for this feature vector. These steps are repeated for all feature vectors to obtain all their hash values. Another issue arises from the abovementioned randomness requirement. It is known that digital computers simply cannot generate provably random numbers because they operate deterministically.

In this work, we argue that these two main issues as the roadblock to practical use of RP-LSH sparse self-attention could be addressed by the emerging nonvolatile memory or memristive technology.[16,17] As their name suggests, memristors are resistors with memory, predicted in 1971[18] and connected to physical devices in 2008.[19] Unlike electronic transistors, a majority of memristor devices operate via electrically-driven nanoscale ionic transport and atomic structural changes, thereby enabling changes in resistance states which can maintain for a long time. The suitability of memristors in solving the von Neumann bottleneck issue has been widely reported in literatures where memristors used for storing synaptic weights are integrated in crossbar arrays to perform matrix-vector multiplication in the linear weighted summation steps of neural network processing.[20,21] The computations are performed in one step and at the sites where data is stored, by making use of device physics and other circuit laws,[22] i.e., Ohm's law and Kirchhoff's law that physically govern multiplication and summation, respectively. One of the main challenges for memristor crossbars as neural network accelerators executing linear weighted summation is the intrinsic device-to-device (D2D) variation[23] that is rooted from the stochastic nature of ionic movement. While the matrices of synaptic connections in neural networks are typically fine-tuned on given datasets, the matrices for RP-LSH are, by definition, random ones, which can be naturally embodied in "non-ideal" memristor crossbar arrays. In contrast to the conventional wisdom that D2D variation has to be mitigated, we here actively leverage such randomness for RP. With such a memristor crossbar array, we perform sequence prediction tasks with a sparse self-attention-based Transformer in a hybrid software-hardware approach, achieving a testing accuracy over 70% with much less computational complexity. The accuracy can be further improved by performing multi-round RP-LSH, which takes advantage of the cycle-to-cycle variability.

2. Results and Discussion

In order to reduce the complexity of attention computation, RP-LSH is used to hash queries and keys for clustering similar vectors into the same bucket with high probability, thereby approximating the full self-attention pattern by attention only within each bucket, as shown in **Figure 1**. RP-LSH in memristive crossbar arrays for interactive attention[24,25] has been reported in previous works.[26,27] Unlike interactive attention, self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence.[3] RP is mathematically expressed by the dot product of the input vector and a random normal vector. To implement RP-LSH physically for sparse self-attention, we exploit the intrinsic randomness of our memristor crossbar array. The component cells are made from stacking layers of V/HfO_2/Ta_2O_5/Ta (see Experimental Section), as schematically shown in **Figure 2b**. Figure 2c,d shows the cross-sectional transmission electron microscopy (TEM) image of this device and the elemental distribution maps, respectively. The energy dispersive X-ray spectroscopy (EDS) line scan reveals the distributions of compositional elements along the stacking direction, as shown in Figure S1 (Supporting Information), from which the 10 nm thick Ta bottom electrode, the oxide bilayer made of 8 nm thick Ta_2O_5 and 8 nm thick HfO_2, and the 10 nm thick top V electrode can be distinguished according to the EDS signals.

Figure 3a shows the I–V curves (see Experimental Section) of the device obtained from 20 cycles of set-reset resistive switching, from which several noteworthy characteristics can be observed. Unlike many reported devices whose repeatable switching behavior can only be elicited after electroforming,[28–30] a one-time application of significantly higher voltage or current than the ones in the subsequent operating cycles, our device is electroforming-free. From its pristine high-resistance state, the first positive voltage ramping from 0 to 5 V with respect to the Ta electrode is performed. Before an observable increase in current occurs, the current remains at an ultralow level about several pA (close to the detection limit of the instrument), forming a prominent current plateau in the voltage range between 0 and ≈2.5 V. Further increase of the voltage till 5 V results in gradual current increase from pA level to a few tens of nA. The programming current of our device is among the lowest reported in the literature,[31–34] which is desired for low-power memristor applications. It is also worth mentioning that this device operates in a self-compliant way that no external current limitation is needed. After reaching the stop voltage (V_{stop}) of 5 V, the ramping direction is

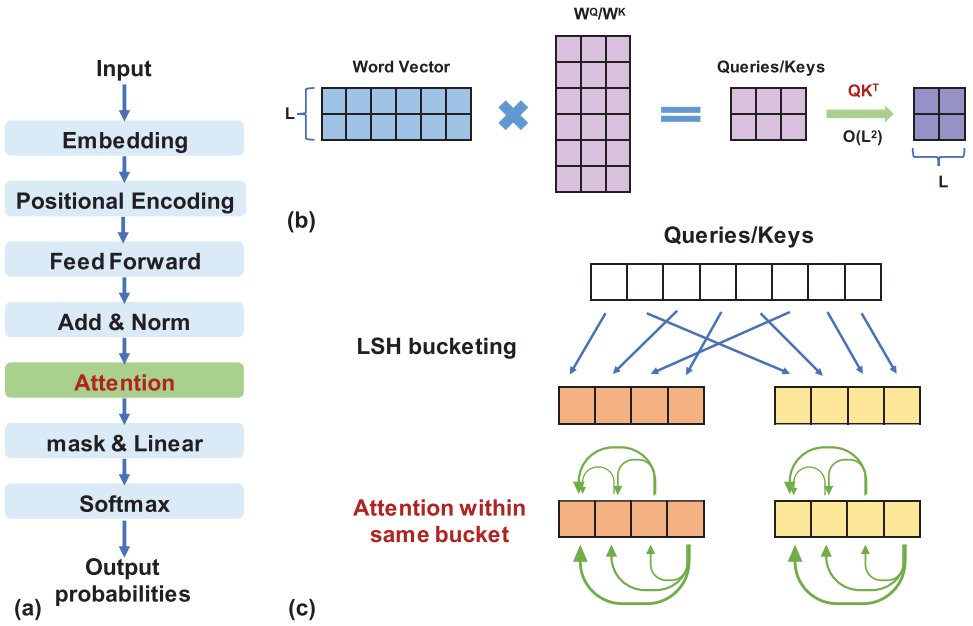

Figure 1. a) Architecture of a standard Transformer model that is based solely on attention mechanism, dispensing with recurrence and convolutions entirely. b) Schematic diagram of the traditional full-attention mechanism. c) Schematic diagram of the sparse-attention mechanism based on RP-LSH.

reversed from 5 V back to 0. It is seen that the evolution of current does not follow the same trajectory as that during positive voltage ramping. Instead, a clear counterclockwise hysteresis loop emerges, which is a key fingerprint of the memristive effect[16,35] and an indication of the switching of the device from a high-resistance state (HRS) to a low-resistance state (LRS), or simply, set switching. Interestingly, this loop is pinched at the voltage ≈2.5 V where the current increase becomes detectable during the positive ramping phase. Below ≈2.5 V, the current evolves across a plateau that looks overlapping (at the detection limit of our instrument) with the one formed in the positive ramping phase. This biased hysteresis loop (not pinched at the origin of the

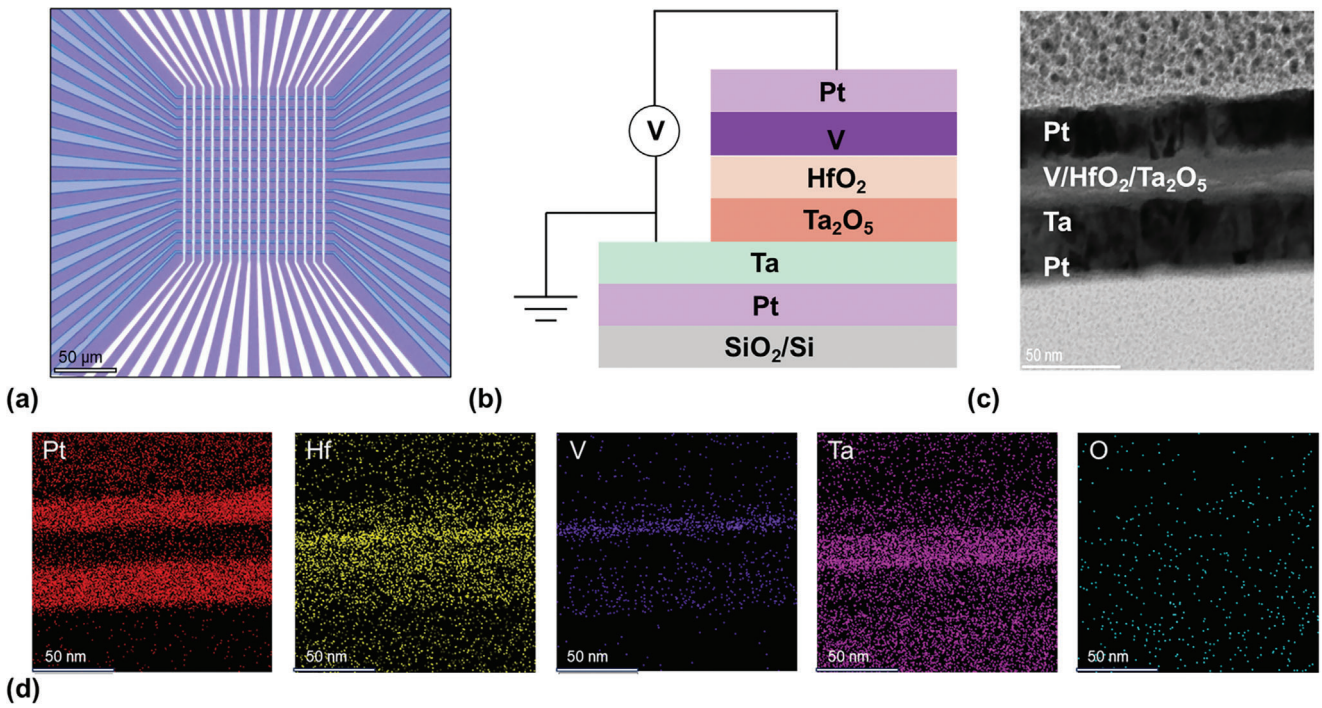

Figure 2. a) The optical image of the crossbar array of the size of 16 × 16. b) Schematic structure of the Pt/V/HfO_2/Ta_2O_5/Ta self-selective cell. c) The cross-sectional TEM image of the device. d) EDS mapping images of the device.

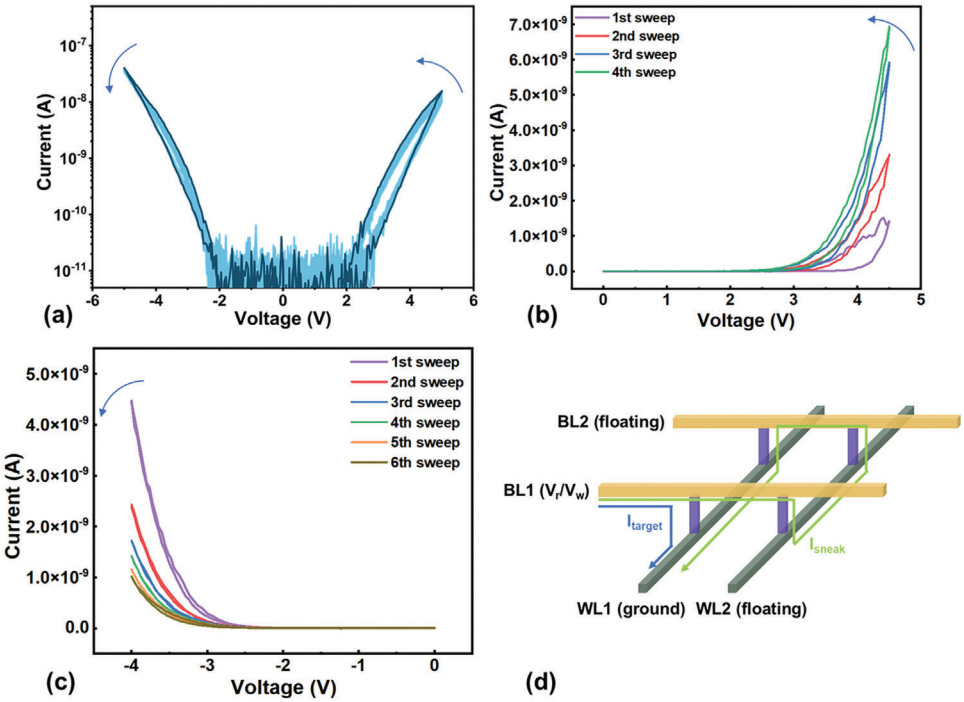

Figure 3. a) Self-selective analog resistive switching behavior observed in the V/HfO_2/Ta_2O_5/Ta device. b) Four consecutive set processes with a stop voltage of 4.5 V. c) Six consecutive reset processes with a stop voltage of −4 V. d) Schematic of the sneak path current in a 2 × 2 crossbar array. The blue and green lines indicate the current through the target cell and the sneak path current, respectively.

I–*V* coordinates) is reminiscent of those of the one-selector-one-memristor (1S1R) cells, which are first commercialized by Intel and Micron.[36] We name the voltage of 2.5 V in our case as the hold voltage (V_h), following the naming convention for 1S1R.

In addition to the nonvolatile memristor, an access device in series is normally required in each crosspoint to avoid undesired current leakage in a large memristor crossbar array.[37] Two-terminal selectors are superior in scalability compared to transistors as access devices but currently still suffer from many manufacturing challenges, such as materials selection[38] and device performance optimization.[39] Future applications require even higher-density memristor crossbar arrays like 3D vertical memristor crossbar arrays. In this case, a separate selector is not allowed to be integrated with the memristor due to fundamental reasons,[40,41] leaving memristors with built-in selectivity (self-selective memristors) as the only choice.

As an example, the leakage suppression function of our self-selective memristor in a two-by-two crossbar array is schematically shown in Figure 3d. Assuming that the unselected bitline (BL) and wordline (WL) are floating and the selected cell is biased to its operating voltage (read voltage, V_r, or write voltage, V_w), the path of leakage current (sneak path) is labeled by the green curve. This sneak path passes through three crosspoint cells. It is not hard to see that sneak paths like this are the shortest in crossbar arrays of any sizes. Leakage current through this sneak path can be significantly suppressed if V_h is greater than one third of the biasing voltage across the selected cell. This can be understood as due to the fact that at least one cell on any sneak path is biased in the current plateau region where the device is of significantly high resistance (inaccessible state).

Our self-selective memristor exhibits bipolar switching characteristics and bidirectional selectivity, as shown in Figure 3a. After the first half cycle of up and down-ramping of the positive voltage, a successive negative voltage ramping begins. It can be seen that there is also a current plateau at the pA level in this negative branch of the *I*–*V* curve. This plateau terminates at the hold voltage of ≈−2 V, beyond which ramping up the negative voltage results in apparent increase in current till a maximum current of the order of 10 nA is reached at the stop voltage of −5 V. Ramping down the voltage from −5 V gives rise to a counterclockwise hysteresis |I|-V loop, which is pinched at the negative V_h of −2 V. The direction of this hysteresis loop indicates that the device is switched from the LRS back to a HRS (reset switching). The device then remains in a state of significantly high resistance (inaccessible state) till the voltage drops to zero. The set-reset switching *I*–*V* curve of the device is reproducible in the successive cyclic voltage ramping measurements. The nonvolatile and analog switching properties of our device are also investigated. As shown in Figure 3b,c, ramping the positive (negative) voltage back and forth leads to gradual increase (decrease) in current at the stop voltage and, in the meantime, gradual rotation of the |I|-V hysteresis loop counterclockwise about the point of V_h, which indicate that the resistance state can be stabilized even after voltage removal and it can also be continuously programmed. The *I*–*V* characteristics of five more devices obtained by consecutively sweeping the positive and negative voltages are shown in Figure S2 (Supporting Information), where it is seen that all these devices exhibit similar analog resistive switching characteristics. This analog-type of conductance is vital for enabling RP-LSH.[26,27]

We have also measured the endurance and retention of these devices. On applying cyclic voltage sweep with positive and negative stop voltages of 4 and −4 V, respectively, each of these device can be reversibly switched back and forth between the HRS and LRS for at least 10^5 cycles, as shown in Figure S3a (Supporting Information). The C2C and D2D variations of the on/off ratio are shown in Figure S4 (Supporting Information). For retention performance measurements, the conductances of each device in its HRS and LRS are read out at the voltage of 3 V. As shown in Figure S3b (Supporting Information), both HRS and LRS can be retained for over 10^4 s, indicating the non-volatility of resistive switching. Figure S5 (Supporting Information) shows the resistive switching *I*–*V* curve of the device after being reversibly switched 200 times, which is still similar to that obtained in the 10th cycle.

To investigate the switching rate of the device, we have performed several sets of pulse measurements, each using a 4.7 V triangular stimulating pulse with a different width, preceded and followed by 3 V square pulses with the widths of 1 µs to read out the device conductances before and after stimulation, respectively. It can be seen that only when the simulating pulse is longer than 400 ns can the device be switched, as exhibited by the significant difference between the readout currents shown in Figure S6 (Supporting Information). In other words, our device could be switched by a 4.7 V triangular pulse in a relatively short timescale of the order of 400 ns. Further increase in the switching speed is possible by using stronger pulses.[42,43]

In order to clarify the nanoscale mechanism of resistive switching, we fabricate devices with different electrode areas and perform additional electrical measurements of these devices. Figure S7 (Supporting Information) shows the relationship between the conductance in different states and the area of the device. It is seen that there is no obvious dependence between the conductance of device in its LRS and the area, indicating filamentary conduction. On the other hand, the conductance of device in the HRS and current-plateauing state is highly area-dependent, indicating an interface-limited conduction mechanism, such as Schottky thermionic emission. The bipolarity of resistive switching implies that electric field effect is the primary driving force. To investigate the effect of heating, we have performed conductivity-temperature (σ−T) measurements, using a hot plate to thermally equilibrate the device at various temperatures in a cyclic manner. The σ−T curve shown in Figure S8 (Supporting Information) does not exhibit any discernible hysteresis, indicating that heating effect alone is unable to induce resistive switching. Nevertheless, field effect and Joule heating effect are intertwined in a complicated way, and it is unfounded to assert that resistive switching is solely due to any one of these effects. The relationship of these two effects may vary from synergetic to adversarial, depending on the stage of switching as well as the materials system.[39,44]

As discussed above, our self-selective memristors in a crossbar array can block the sneak paths if V_h is greater than one third of the biasing voltage across the selected cell. To determine a suitable read voltage V_r for our devices, one must also ensure that the resistance state of our device is unaffected by V_r, or in other words, V_r does not elicit resistive switching. To this end, we carry out further quasi-DC voltage sweeping measurements with reduced stop voltages. We identify 3 V as suitable for reading because the *I*–*V* curves obtained by sweeping the voltage between 0 and 3 V back and forth are single-valued (Figure S9, Supporting Information), indicating that no resistive switching occurs. Accordingly, V_r is set to 3 V in this study. As for writing operation, we choose 4 V as the V_w.

While the most common application of memristive crossbar arrays as the accelerators for computing weighted sums of neural activations (also dot-product operations) requires the mitigation of D2D variability,[25] RP-LSH function is instead enabled by employing such a non-ideal factor.[28,29] The conductance map (read at 3 V) of a 16 × 2 self-selective memristive crossbar array is shown in **Figure 4a**. Here, all devices in the array have been subjected to one-time voltage sweep from 0 to −4 V and back to 0, reset to their HRSs. This pre-treatment helps reduce the current and thus the energy consumption in the subsequent RP-LSH dot-product operations. It is seen from Figure 4c that the conductances of these devices follow a lognormal distribution, with a mean value of 0.125 nS and a standard deviation of 0.008 µS. To perform RP, Indyk and Motwani,[45] and Dasgupta and Gupta[46] have shown that the entries of the random matrix can be independent random variables with the Gaussian distribution. Intriguingly, the distribution of the row-wise differences between the conductances of these two columns of memristors in the array can be approximated by the Gaussian distribution with a zero mean value, as shown in Figure 4e,f. Similar observations are reported previously.[26,27]

With this physical random matrix, RP-LSH is conducted by applying the input voltage vector to the row wires of the crossbar and comparing the output current from the adjacent column wires. To obtain the hash, the current difference vector and its opposite are concatenated, whose argmax is defined as the hash value.[11,47] The main advantage of the hardware RP-LSH is that the computation can be performed in just one step.

In addition to D2D variability, cycle-to-cycle variability is also an intrinsic nature of memristors, which has been exploited for nonconventional computing functionalities.[48–52] To investigate such variability in our devices in the crossbar array, we perform varying numbers of cycles of set-reset switching operations for each device and measure the conductance difference statistics after the last reset operation has ended. As shown in Figure 4d,f and Figure S10 (Supporting Information), though the conductance differences still follow a Gaussian distribution irrespective of the number of switching cycles, the standard deviation of the distribution does depend on the cycle number. We will demonstrate the potential benefits of harnessing this variability for sparse attention later.

To evaluate the distance-preserving capability of our hardware-implemented RP-LSH, we conduct four experimental trials in which twenty 16-dimensional random vectors (**Figure 5a**) are hashed into two buckets following the above-introduced steps. In the *n*th trial, the memristive crossbar array has been pre-treated by set-reset operations for *n* times. For comparison, four trials of simulations of hashing these vectors are also carried out. The entries of the simulated random matrix used in the *n*th trial follow the conductance difference distribution in the experimental memristive crossbar array. Figure 5b,c shows the classification results of the software- and hardware-based hashing, respectively. It is shown that the experimental results are consistent with the simulation results for 19 out of 20 vectors. In order to further

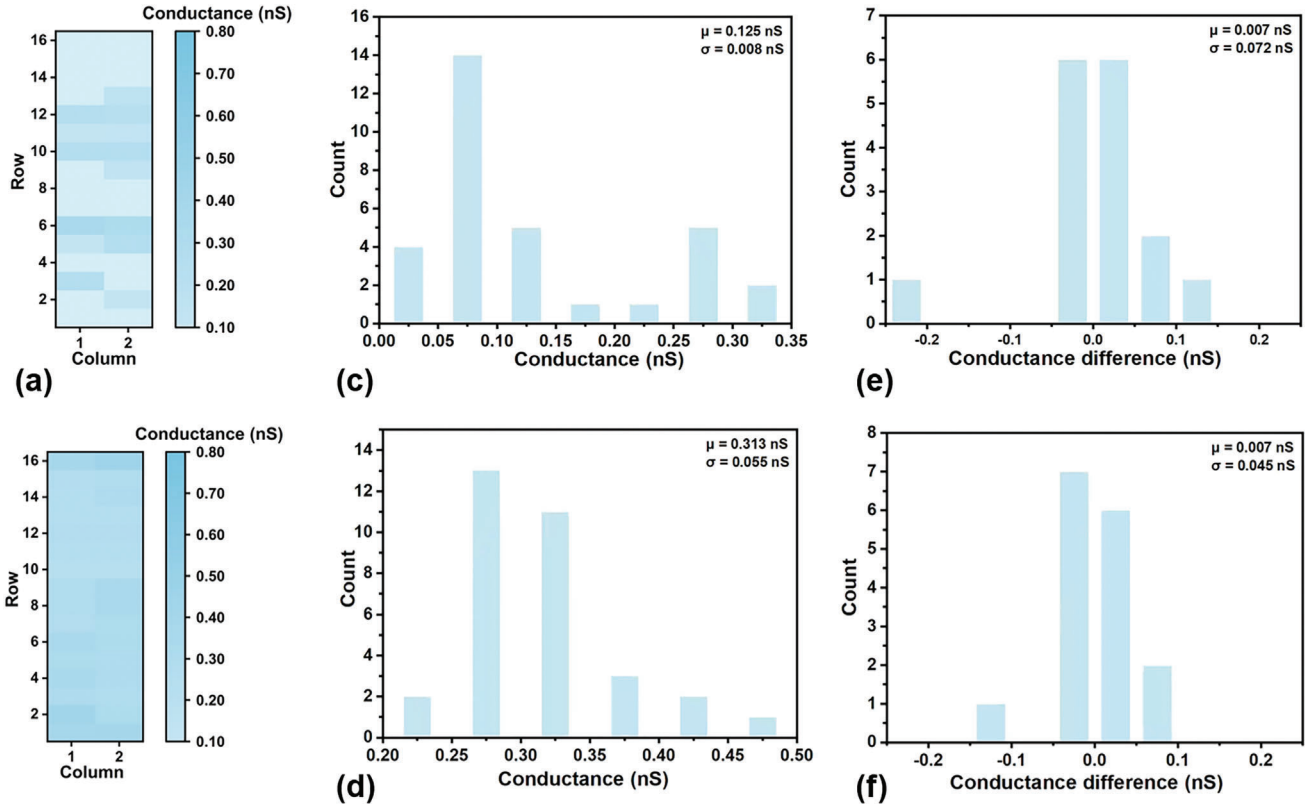

Figure 4. The HRS conductance map of a 16 × 2 memristor crossbar array after a) being reset for once and b) five set-reset switching operations. c,d) Distribution of the device conductances corresponding to (a,b). e,f) Distribution of the conductance differences corresponding to (a,b).

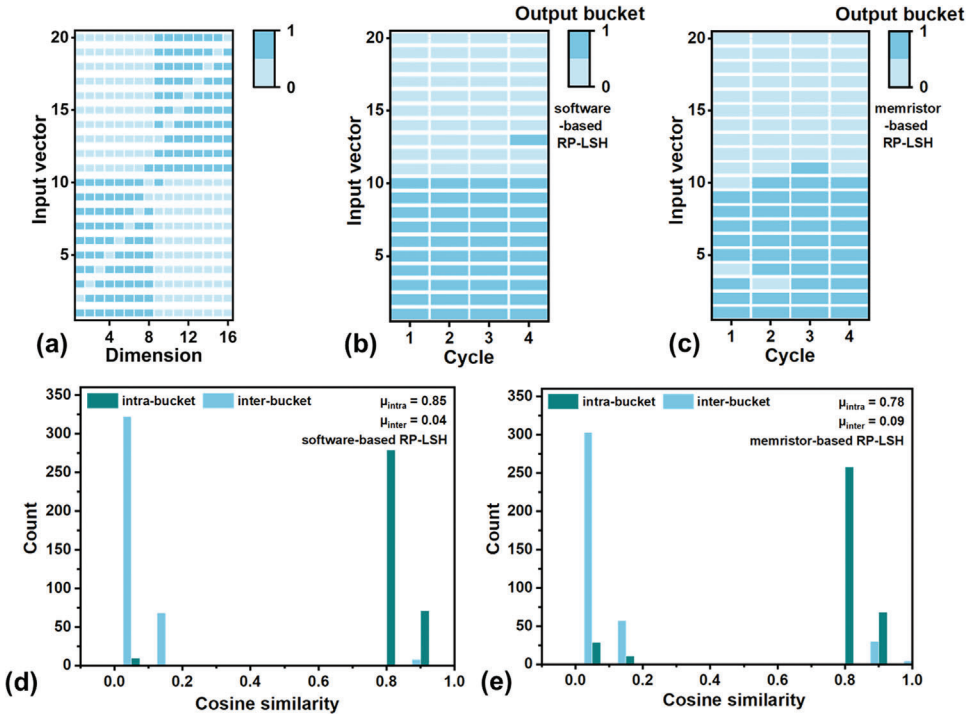

Figure 5. a) Twenty 16-dimensional random input vectors. The classification results of the input vectors based on b) software and c) hardware hashing in four trials, respectively. Statistics of the cosine similarity between the intra-bucket (light blue) and inter-bucket (dark blue) vectors obtained by d) software-based RP-LSH and e) memristor-based RP-LSH.

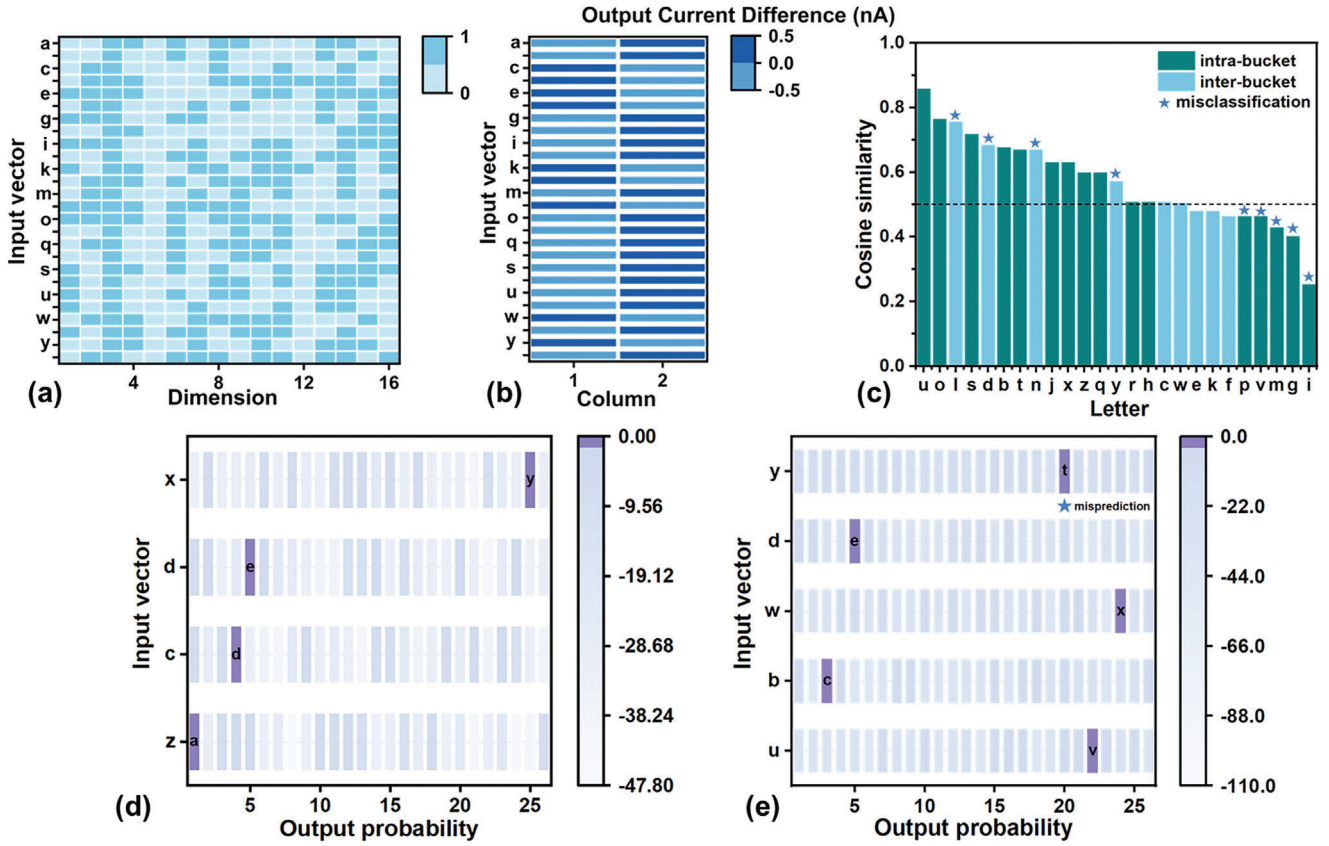

Figure 6. a) The input vectors corresponding to the 26 letters. b) Hashing bits (differential currents) for the input vectors generated by performing RP-LSH within the memristor crossbar array. c) Cosine similarities between letter "a" and the other letters, where the stars represent misclassification. The probability distributions of the predicted letters given the input sequences d) "xdcz" and e) "ydwbu".

verify the effectiveness of memristor-based RP-LSH, we calculate the cosine similarities between the input vectors hashed to the same and different buckets, respectively. It is seen that the cosine similarities (Figure 5d,e) between vectors in the same buckets are normally greater than those between vectors in different buckets, with a few exceptions though. This indicates reasonably satisfactory distance-preserving capabilities of both the software- and hardware-based hashing.

Next, we will demonstrate the potential of memristor crossbar array for use in sparse self-attention-based Transformer. The task we test is sequence prediction, in which the input string should lead to a shift of each letter to its successive one, for example, "xdcz→yeda". The task is performed in a hybrid software-hardware approach where the training is performed entirely in software (see Experimental Section) while in the testing phase the random matrix for RP-LSH is implemented in our memristor crossbar array and the rest of the operations are still performed in software. Here, each letter in an input string is embedded to a 64-dimensional vector and they are packed together into an embedding matrix X. With a single attention head, the Q, K and V matrices are then computed by linear projection of X using 64×16 projection matrices W_Q, W_K and W_V ($W_Q = W_K$ in our experiments), respectively, and binarization subsequently using thresholding. The obtained 16-dimensional binarized query vectors (also key vectors) are shown in **Figure** 6a. In the testing phase, queries q_i and keys k_j are mapped to their respective voltage vectors (element "0": 0 V; element "1": 3 V) which are hashed in the memristor crossbar array into two buckets. Figure 6b shows the current differences obtained according to the aforementioned approach and their grouping for all 26 letters. Letter "a" and the other 16 letters are hashed to the same bucket. We calculate the cosine distances between "a" and other 25 letters in either the same or a different bucket, as shown in Figure 6c. It can be seen that letters in the same (a different) bucket are normally more similar (dissimilar) to "a" than letters in a different (the same) bucket, as expected. A small number of letters similar to "a" are still divided into a different bucket, including letters "d", "l", "n" and "y". This problem can be alleviated by performing multiple rounds of hashing as will be discussed later. By allowing attention only within each bucket to approximate the full-attention, we still achieve 72% testing accuracy in predicting the output sequences based on the inputs, regardless of the sequence length and the order of letters. Two test cases are shown in Figure 6d,e.

With hashing, there is always a small probability that similar items nevertheless fall in different buckets, leading to inaccurate prediction. In Figure 6e, for example, this hybrid software-hardware Transformer fails to correctly predict the successive letter of "y" in string "ydwbu". The prediction can become more accurate as the number of hashes increases, each with a distinct hash function.[11] To implement multi-round RP-LSH attention,

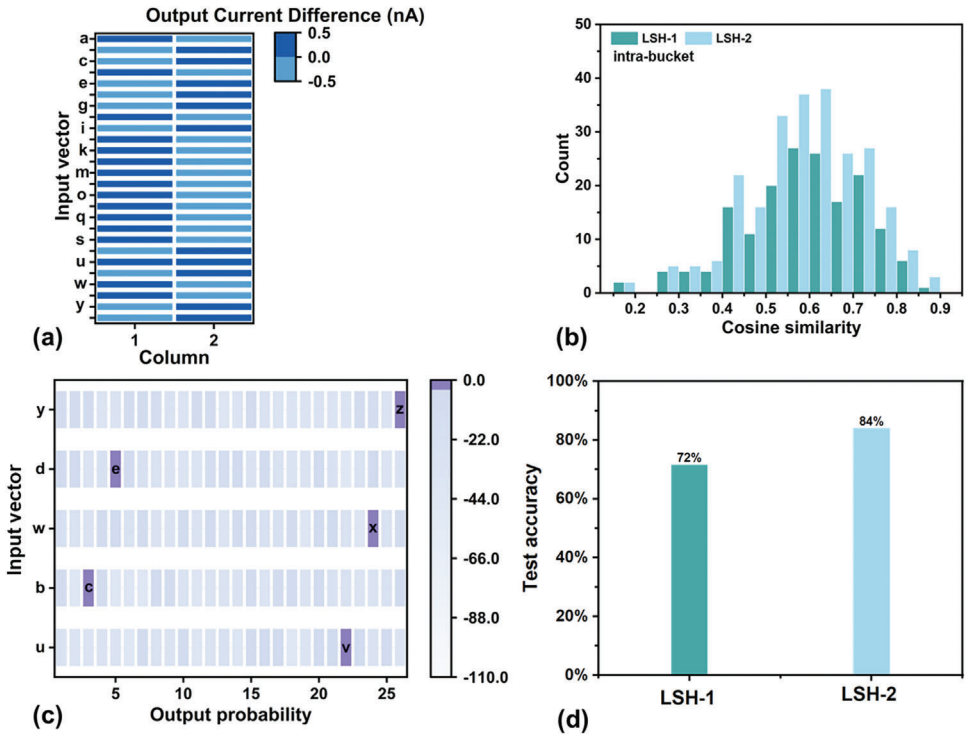

Figure 7. a) Grouping of the input vectors after two-round hashing. b) Cosine similarities between the intra-bucket vectors after one-round (green part) and two-round (light blue) hashing. c) The probability distributions of the predicted letters given the input sequence "ydwbu". d) Comparison of the prediction accuracies of the sparse-attention-based Transformer model based on single and two rounds of hashing in hardware.

we take advantage of the cycle-to-cycle variation of the conductance difference distribution in our memristor crossbar array, as shown above in Figure 4b and Figure S7 (Supporting Information). Specifically, we perform two-round hashing in the testing phase (the settings of the training phase remain the same). After cycling each memristor five times, the memristor crossbar array is used to implement the new random matrix (Figure 4b) for the second round of hashing. Sets that a query q_i attends to obtained by each RP-LSH are aggregated to form a union. We regard q_i collides with q_j if q_j is an element of the union. In this way, more similar items are grouped into the same bucket after two-round hashing than after single-round hashing, as shown in **Figure 7b**. With two-round hashing, more correct predictions are obtained (Figure 7c) and the accuracy is increased by 12% (Figure 7d).

3. Conclusion

To conclude, we have experimentally demonstrated the feasibility of using the emerging memristor crossbar arrays as the physical embodiments of RP matrices for LSH. Thanks to the in-memory computing architecture for matrix-vector-multiplication and the intrinsic D2D variability of the memristor crossbar array, RP-LSH algorithm can be executed more locally so that data does not need to be shipped from place to place, being desired for reducing energy consumption. With PR-LSH, we have further performed sequence prediction tasks with a sparse self-attention-based Transformer in a hybrid software-hardware approach, achieving a testing accuracy over 70% with much less computational complexity. To increase the collision probability of similar items, which is essential for sparse self-attention, we have exploited the C2C variability of the memristor crossbar array for multi-round hashing, resulting in further improvement in the testing accuracy. This work presents a new paradigm for accelerating the state-of-the-art Transformer LLMs.

4. Experimental Section

Device Fabrication: The memristor crossbar array with a 2 × 2 μm^2 junction area was fabricated on a SiO_2/Si wafer, patterned through photolithography and lift-off processes. The bottom Ta layer was deposited to a thickness of 10 nm by 50 W direct current (DC) magnetron sputtering under an argon pressure of 3 mTorr. Subsequently, an 8 nm Ta_2O_5 layer and an 8 nm HfO_2 layer were deposited through 50 W radio frequency (RF) sputtering using the respective ceramic targets under an argon pressure of 4 mTorr. Then, a 10 nm thick V top electrode was fabricated by 50 W DC sputtering under an argon pressure of 4 mTorr. Finally, a 30 nm Pt protection layer was deposited by 50 W DC sputtering under an argon pressure of 3 mTorr. All the films were deposited at room temperature.

Electrical Measurements: Cyclic quasi-DC voltage sweep with a sweep rate of 0.2 V μs^{-1} and pulse measurements were performed by an Agilent B1500A semiconductor analysis system. Using a high-frequency semi-automatic probe station Summit 12000B-M and probe card configuration, the DC and pulsed voltages were applied to the target bit line (BL), the corresponding word line (WL) was grounded and other electrodes were left floating. The switching matrix B2200 was used to select BLs and WLs. All the measurements were performed at room temperature and under ambient atmosphere.

Simulations: PyTorch 1.13.0 was used as the deep learning framework. During the training process, the weights of the embedding layer, the feed-forward layer, and the linear layer were updated to minimize the

cross-entropy loss function using stochastic gradient descent. The learning rate was set to 0.001.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

The authors acknowledge funding from the National Key R&D Program of China (2021ZD0200300 and 2018YFE0200200), the National Natural Science Foundation (grant nos. 61974082, 61704096, and 61836004), the Youth Elite Scientist Sponsorship (YESS) Program of China Association for Science and Technology (CAST) (no. 2019QNRC001), the Key Laboratory of Luminescence Analysis and Molecular Sensing (Southwest University), Ministry of Education, Southwest University, Chongqing, 400715, P. R. China, Tsinghua-IDG/McGovern Brain-X program, the Beijing science and technology program (grant nos. Z181100001518006 and Z191100007519009), the Suzhou-Tsinghua innovation leading program 2016SZ0102, and CETC Haikang Group-Brain Inspired Computing Joint Research Center.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

attention, large language models, locality-sensitive hashing, memristor crossbar arrays, self-selective memristors, transformer

Received: December 3, 2023
Revised: May 9, 2024
Published online: June 17, 2024

[1] Z. Niu, G. Zhong, H. Yu, *Neurocomputing* **2021**, *452*, 48.
[2] M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, S. H. Zhang, R. R. Martin, M. M. Cheng, S. M. Hu, *Comput. Vis. Media* **2022**, *8*, 331.
[3] F. A. Furfari(tony), *IEEE Ind. Appl. Mag.* **2002**, *8*, 8.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, presented at NIPS **2017**, *30*.
[5] S. Hochreiter, J. Schmidhuber, *Neural Comput.* **1997**, *9*, 1735.
[6] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, presented at NIPS December, **2014**, https://doi.org/10.48550/arXiv.1412.3555.
[7] S. Albawi, T. A. Mohammed, S. Al-Zawi, *Proc. 2017 Int. Conf. Engineering and Technology ICET 2017*, Antalya, Turkey, August, **2017**.
[8] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, Ł. Kaiser, N. Shazeer, presented at Proc. of ICLR January, **2018**, https://doi.org/10.48550/arXiv.1801.10198.
[9] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, D. Eck, presented at CoRR December, **2018**, https://doi/org/10/48550/arXiv.1809.04281.
[10] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, *35th Int. Conf. Mach. Learn. ICML* **2018**, *80*, 4055.
[11] N. Kitaev, Ł. Kaiser, A. Levskaya, presented at Proc. of ICLR, February, **2020**, https://doi.org/10.48550/arXiv.2001.04451.
[12] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, D. Metzler, presented at Proc. of ICLR, November, **2020**, https://doi.org/10.48550/arXiv.2011.04006.
[13] Q. Fournier, G. M. Caron, D. Aloise, *ACM Comput. Surveys* **2023**, *55*, 1.
[14] R. Soleymani, J. Beaulieu, J. Farret, *Sens. Transducers* **2021**, *249*, 110.
[15] H. Kitano, D. Taide, Applying and Adapting the Reformer as a Computationally Efficient Approach to the SQuAD 2.0 Question-Answering Task, https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/default/report07.pdf, (accessed: May 2020).
[16] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.
[17] D. S. Jeong, R. Thomas, R. S. Katiyar, J. F. Scott, H. Kohlstedt, A. Petraru, C. S. Hwang, *Reports Prog. Phys.* **2012**, *75*, 076502.
[18] L. Chua, *IEEE Trans. Circuits Syst.* **1971**, *18*, 507.
[19] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.
[20] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, R. S. Williams, in *2016 53rd ACM/EDAC/IEEE Design Automation Conf. (DAC)*, IEEE, Piscataway, NJ, **2016**.
[21] Q. Xia, J. J. Yang, *Nat. Mater.* **2019**, *18*, 309.
[22] D. Ielmini, H. S. P. Wong, *Nat. Electron.* **2018**, *1*, 333.
[23] H. Kim, M. R. Mahmoodi, H. Nili, D. B. Strukov, *Nat. Commun.* **2021**, *12*, 5198.
[24] A. Graves, G. Wayne, I. Danihelka, presented at CoRR December, **2014**, https://doi.org/10.48550/arXiv.1410.5401.
[25] F. Meng, Z. Lu, H. Li, Q. Liu, presented at Proc. of COLING October, **2016**, https://doi.org/10.48550/arXiv.1610.05011.
[26] R. Mao, B. Wen, A. Kazemi, Y. Zhao, A. F. Laguna, R. Lin, N. Wong, M. Niemier, X. S. Hu, X. Sheng, C. E. Graves, J. P. Strachan, C. Li, *Nat. Commun.* **2022**, *13*, 6284.
[27] L. Yang, X. Huang, Y. Li, H. Zhou, Y. Yu, H. Bao, J. Li, S. Ren, F. Wang, L. Ye, Y. He, J. Chen, G. Pu, X. Li, X. Miao, *InfoMat* **2023**, *5*, e12416.
[28] J. Joshua Yang, F. Miao, M. D. Pickett, D. A. A. Ohlberg, D. R. Stewart, C. N. Lau, R. S. Williams, *Nanotechnology* **2009**, *20*, 215201.
[29] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H. S. P. Wong, G. Cauwenberghs, *Nature* **2022**, *608*, 504.
[30] M. Rao, H. Tang, J. Wu, W. Song, M. Zhang, W. Yin, Y. Zhuo, F. Kiani, B. Chen, X. Jiang, H. Liu, H. Y. Chen, R. Midya, F. Ye, H. Jiang, Z. Wang, M. Wu, M. Hu, H. Wang, Q. Xia, N. Ge, J. Li, J. J. Yang, *Nature* **2023**, *615*, 823.
[31] K. M. Kim, J. Zhang, C. Graves, J. J. Yang, B. J. Choi, C. S. Hwang, Z. Li, R. S. Williams, *Nano Lett.* **2016**, *16*, 6724.
[32] P. Bousoulas, I. Michelakaki, E. Skotadis, M. Tsigkourakos, D. Tsoukalas, *IEEE Trans. Electron Devices* **2017**, *64*, 3151.
[33] M. Xiao, K. P. Musselman, W. W. Duley, Y. N. Zhou, *ACS Appl. Mater. Interfaces* **2017**, *9*, 4808.
[34] H. Song, Y. S. Kim, J. Park, K. M. Kim, *Adv. Electron. Mater.* **2019**, *5*, 1800740.
[35] S. P. Adhikari, M. P. Sah, H. Kim, L. O. Chua, *IEEE Trans. Circuits Syst.* **2013**, *60*, 3008.
[36] F. T. Hady, A. Foong, B. Veal, D. Williams, *Proc. IEEE* **2017**, *105*, 1822.
[37] G. W. Burr, R. S. Shenoy, K. Virwani, P. Narayanan, A. Padilla, B. Kurdi, H. Hwang, *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.* **2014**, *32*, 040802.
[38] H. Li, J. Robertson, *Sci. Rep.* **2019**, *9*, 1867.
[39] Y. Yang, M. Xu, S. Jia, B. Wang, L. Xu, X. Wang, H. Liu, Y. Liu, Y. Guo, L. Wang, S. Duan, K. Liu, M. Zhu, J. Pei, W. Duan, D. Liu, H. Li, *Nat. Commun.* **2021**, *12*, 6081.

[40] Q. Luo, X. Xu, H. Liu, H. Lv, T. Gong, S. Long, Q. Liu, H. Sun, W. Banerjee, L. Li, J. Gao, N. Lu, M. Liu, *Nanoscale* **2016**, *8*, 15629.
[41] Q. Huo, Y. Yang, Y. Wang, D. Lei, X. Fu, Q. Ren, X. Xu, Q. Luo, G. Xing, C. Chen, X. Si, H. Wu, Y. Yuan, Q. Li, X. Li, X. Wang, M. F. Chang, F. Zhang, M. Liu, *Nat. Electron.* **2022**, *5*, 469.
[42] B. J. Choi, S. Choi, K. M. Kim, Y. C. Shin, C. S. Hwang, S. Y. Hwang, S. S. Cho, S. Park, S. K. Hong, *Appl. Phys. Lett.* **2006**, *89*, 2000815.
[43] S. Siegel, C. Baeumer, A. Gutsche, M. von Witzleben, R. Waser, S. Menzel, R. Dittmann, *Adv. Electron. Mater.* **2021**, *7*, 012906.
[44] S. Kim, S. Choi, W. Lu, *ACS Nano* **2014**, *8*, 2369.
[45] P. Indyk, R. Motwani, in *Proceedings of the thirtieth annual ACM Symp. on Theory of Computing (STOC)*, ACM, New York, **1998**.
[46] S. Dasgupta, A. Gupta, *Random Struct. Algorithms* **2003**, *22*, 60.
[47] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, L. Schmidt, *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1225.
[48] T. Dalgaty, N. Castellani, C. Turck, K. E. Harabi, D. Querlioz, E. Vianello, *Nat. Electron.* **2021**, *4*, 151.
[49] B. Gao, B. Lin, Y. Pang, F. Xu, Y. Lu, Y. C. Chiu, Z. Liu, J. Tang, M. F. Chang, H. Qian, H. Wu, *Sci. Adv.* **2022**, *8*, eabn7753.
[50] S. Balatti, S. Ambrogio, Z. Wang, D. Ielmini, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2015**, *5*, 214.
[51] H. Jiang, D. Belkin, S. E. Savel'Ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao, M. Barnell, Q. Wu, J. J. Yang, Q. Xia, *Nat. Commun.* **2017**, *8*, 882.
[52] S. Dutta, G. Detorakis, A. Khanna, B. Grisafe, E. Neftci, S. Datta, *Nat. Commun.* **2022**, *13*, 2571.