In-Memory PageRank Accelerator With a Cross-Point Array of Resistive Memories

Zhong Sun, *Member, IEEE*, Elia Ambrosi, Giacomo Pedretti, Alessandro Bricalli, and Daniele Ielmini, *Fellow, IEEE*

***Abstract*— In-memory computing with cross-point arrays of resistive memory is a promising technique for typical tasks, such as the training and inference of deep learning. Recently, it has been shown that a cross-point array of resistive switching memory (RRAM) with a feedback configuration can be used to solve linear systems, compute eigenvectors, and rank webpages in just one step. Here, we demonstrate the PageRank with a real data set (the Harvard500) and an eight-level RRAM model, describing the conductance update, the standard deviation of each level, and the conductance ratio. By carefully placing each memory conductance value via a program-verify technique, we show that an accuracy of 95% can be achieved for the ranking result. The equivalent throughput of the eigenvector circuit for PageRank is estimated to be 0.183 tera-operations per second (TOPS), while the energy efficiency is 362 TOPS/W. This article supports the feasibility of in-memory PageRank with significant improvements in speed and energy efficiency for practical big-data tasks.**

***Index Terms*— Eigenvector, in-memory computing, PageRank, resistive switching memory (RRAM).**

I. Introduction

THE cross-point array of resistive memories, such as the resistive switching memory (RRAM) and the phase change memory (PCM), has been extensively adopted for in-memory computation to eliminate the memory bottleneck of the conventional von Neumann architecture [1], [2]. In particular, cross-point arrays can accelerate the matrix-vector multiplication (MVM), thus allowing to speed up the solution of various data-centric problems, such as the training and inference of deep neural networks [3], the image and signal processing [4], [5], and the iterative solution of linear systems [6] or differential equations [7]. Recently, a cross-point RRAM circuit with a feedback configuration has been proposed and demonstrated to solve linear systems and evaluate matrix eigenvectors in one step without iterations [8]. The eigenvector computation forms the mathematical basis for the PageRank algorithm [9]; thus, the cross-point RRAM circuit is very promising to accelerate webpage ranking for search engines.

In this article, we demonstrate in-memory eigenvector calculation in one step for the ranking of the Harvard500 data set, which contains 500 relevant webpages of the Harvard University [10]. To simulate the cross-point RRAM circuit for PageRank, we developed a statistical RRAM model based on the experimental observations, featuring eight discrete conductance levels with their corresponding variations. By using a verify algorithm to control the conductance variations in the array, we show that the accuracy of PageRank can reach 95%. Finally, we show the transient simulation of the PageRank circuit, from which the equivalent throughput and energy efficiency of the circuit can be estimated. The results demonstrate improved speed and energy efficiency of the in-memory computing hardware compared with the conventional approach.

II. Experimental Devices and Results

The RRAM device adopted in this work is composed of a Ti/HfO_2/C stack [11]. To fabricate the RRAM device, a HfO_2 film of thickness $t_{ox} = 5$ nm was deposited by the e-beam evaporation on a confined graphitic carbon bottom electrode (BE), and then, a Ti thin top electrode (TE) layer was deposited without breaking the vacuum. To initiate the resistive switching (RS) of the device, the forming process was conducted by applying a dc voltage sweep from 0 to 5 V to the TE, with the BE being grounded. The resulting soft breakdown of the dielectric HfO_2 layer caused the conductive filament formation and the consequent RS behavior. Set and reset transitions took place under positive and negative voltages applied to the TE, respectively.

The RRAM devices had a one-transistor/one-resistor (1T1R) structure to enable the analog programming of the device conductance. To this purpose, the gate voltage (V_G) applied to the transistor was used to control the conductance level of the RRAM device. This is shown in Fig. 1, where four discrete levels of conductance can be obtained with increasing compliance current I_C, i.e., the V_G-dependent transistor's saturation current. The various conductance values can be stored in the cross-point array to execute in-memory matrix

Manuscript received October 15, 2019; revised December 19, 2019; accepted January 8, 2020. Date of publication February 4, 2020; date of current version March 24, 2020. This work was supported by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement 648635. The review of this article was arranged by Editor J. Yang. *(Corresponding author: Daniele Ielmini.)*

The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mail: daniele.ielmini@polimi.it).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TED.2020.2966908

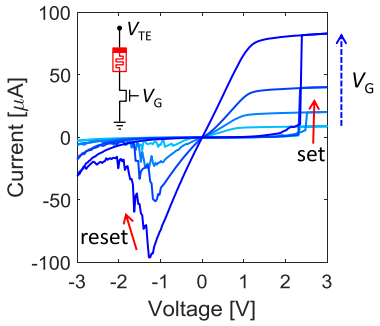

Fig. 1. Current–voltage characteristics of the multilevel operation of the RRAM device. The applied V_G is 0.9, 1.0, 1.2, and 1.4 V for the four curves from the bottom to the top, respectively.

computation in the analog domain. During the computation, a sufficiently high V_G, e.g., 3.5 V, is applied to minimize the voltage drop across the transistor.

The dc conduction and RS characteristics of the RRAM device were collected by a Keysight B1500A Semiconductor Parameter Analyzer, which was conducted in a conventional probe station for electrical characterization.

III. In-Memory PageRank

A. Eigenvector Circuit of a Cross-Point RRAM Array

We addressed the evaluation of the eigenvector, namely, the solution of the matrix equation

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \tag{1}$$

where $\boldsymbol{A}$ is a square matrix, λ is an eigenvalue of $\boldsymbol{A}$, and $\boldsymbol{x}$ is the corresponding unknown eigenvector. This problem can be mapped by a cross-point RRAM circuit with a feedback configuration, as shown in Fig. 2(a), where the conductance matrix $\boldsymbol{G_A}$ of the cross-point array contains the elements of matrix $\boldsymbol{A}$, while the feedback conductance G_λ of the transimpedance amplifiers (TIAs) represents the positive eigenvalue λ. In the circuit, the output voltage vector $\boldsymbol{v}$ multiplied by G_λ yields the cross-point currents that, in turn, are given by the MVM in the cross-point circuit, namely

$$\boldsymbol{G_A}\ \boldsymbol{v} = G_\lambda \boldsymbol{v} \tag{2}$$

which is equivalent to (1). As a result, $\boldsymbol{v}$ yields the solution to the eigenvector equation, thus enabling the eigenvector computation in one step with the circuit.

The circuit of Fig. 2(a) can be viewed as a closed-loop integration of the power iteration algorithm [12], where a single operation is conducted instead of individual discrete iterations. Note that the single-step operation in the circuit allows completing the whole (virtual) power iteration in the analog domain, thus avoiding repeated data transfer from output to input, and the associated analog-digital conversion. Due to the high parallelism of the cross-point architecture, we expect that the MVM computation time does not explicitly depend on the size of the matrix $\boldsymbol{A}$. Due to the positive feedback of the circuit, only the dominant eigenvector, namely, the one corresponding to the largest eigenvalue, can be computed. Also, to guarantee the circuit capability, G_λ should be slightly smaller than the nominal value to provide a loop gain larger than 1 while maintaining a good accuracy [8].

To experimentally demonstrate the circuit of Fig. 2(a), we considered the calculation of the dominant eigenvector of a 3×3 positive matrix. Fig. 2(b) shows the experimental result as a function of the analytical solution, while the matrix $\boldsymbol{A}$ is shown in the inset. The linear relationship in Fig. 2 supports the hardware-based eigenvector computation in one step. Note that the circuit in Fig. 2(a) can also address problems with negative eigenvalues, provided that the analog inverters are removed. Matrices with both positive and negative entries can be addressed by splitting the cross-point array in two arrays [8].

B. RRAM Compact Model

To study the continuous modulation of the RRAM conductance, we conducted 1200 experiments where V_G was increased from 0.8 to 1.9 V with a step $\Delta V_G = 0.05$ V during the set transition. The set transition was induced by applying a triangular voltage pulse to the TE of the device. After each set transition, the device conductance was measured at low voltage. Fig. 3(a) shows the measured conductance value for all the experiments. The mean value μ_G of the conductance shows an almost linear increase from 2 to 32 μS, which supports our RRAM device for implementing analog matrix entries [13]. Note that conductance levels show an intrinsic variation due to the stochastic filament formation during the set transition [14].

The RRAM conductance levels follow a normal distribution, as indicated by the seven equally spaced levels (L_1, $L_2, \ldots, L_7$) in Fig. 3(b) with increasing μ_G from 2 to 32 μS. The standard deviation σ_G of conductance is about 3.8 μS for each discrete level, in line with previous reports [14], [15]. Due to the clear overlaps between adjacent levels, it might be difficult to identify the state of a device after programming. However, assuming that all devices have strong retention, once the cross-point array is programed to solve a specific problem, the device conductance values are fixed and no more recognition of the device state is needed, thus eliminating the confusion caused by state overlaps. The inset of Fig. 3(b) shows the distribution of the reset state L_0, indicating a log-normal distribution with $\mu_G = 0.019$ μS and $\sigma_{\log G} = 0.29$. The eight conductance levels allow for a 3-bit RRAM to implement the analog elements. Note the large ratio between the maximum and the minimum conductance values, namely, $(G_{\max}/G_{\min}) = 1684$.

C. PageRank of Harvard500

The cross-point eigenvector circuit shown in Fig. 2 can be straightforwardly used to implement the PageRank algorithm, as the citation matrix contains only positive entries by definition [9]. To demonstrate the circuit at a relatively large scale, we solved the PageRank with the Harvard500 data set of 500 webpages [10].

In PageRank, each page is ranked based on the number of citations by other pages. The citation matrix $\boldsymbol{C}$ elements are defined as follows: If page j contains a link to page i,

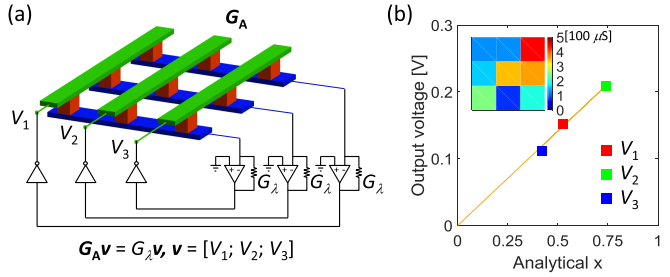

Fig. 2. (a) Eigenvector circuit of a cross-point RRAM array, which maps the eigenvector equation. (b) Experimental eigenvector solution of a positive matrix, in comparison with the normalized analytical eigenvector. The matrix is shown in the inset, whose largest eigenvalue is 6.5.

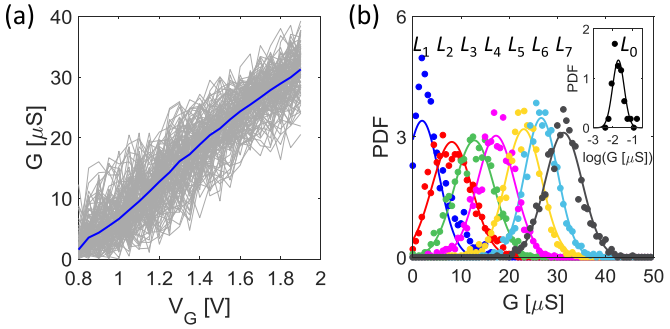

Fig. 3. (a) 100 traces of RRAM conductance G as a function of V_G. The blue line is the mean conductance value which increases almost linearly with V_G. (b) Distributions of G for an eight-level RRAM from experimental measurements and the compact model. Seven high conductance levels are extracted from (a). The inset shows the distribution of the reset state L_0.

the element C_{ij} is set to 1; otherwise, $C_{ij} = 0$. More pages citing one page indicates that the latter is more important. Also, citation by important pages gives rise to the importance of the page. Fig. 4(a) shows the citation matrix of Harvard500. To rank the webpages by their importance, a transition matrix $\boldsymbol{T}$ is defined according to

$$T_{ij} = \begin{cases} \dfrac{pC_{ij}}{\sum_i C_{ij}} + \delta, & \text{if } \sum_i C_{ij} \neq 0 \\ 1/N, & \text{if } \sum_i C_{ij} = 0 \end{cases} \tag{3}$$

where $N = 500$ is the number of pages, $p = 0.85$ is the random walk probability, and $\delta = (1 - \mathrm{p})/\mathrm{N}$ is the probability of randomly picking a page. A uniform probability $1/N$ is assigned if a page gets no link. The transition matrix can be viewed as a stochastic matrix with the largest eigenvalue always being 1 and the dominant eigenvector giving the importance scores of the webpages.

Fig. 4(b) shows the resulting transition matrix for Harvard500. Most entries in the transition matrix have small values, as the citation matrix is sparse in Fig. 4(a). In particular, 74.5% of the entries are equal to 3×10^{-4}, while 24.4% of the entries are equal to 2×10^{-3}. These small values are implemented by using the reset state of the RRAM devices, thus simplifying the programming of the cross-point array. The unique nature of the transition matrix also supports the low power consumption of the circuit, thus further improving the energy efficiency of the in-memory PageRank.

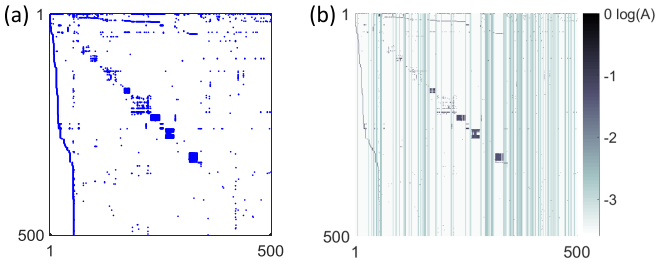

Fig. 4. (a) Citation matrix of Harvard500, with dots indicating a value of 1, while the rest of the matrix is 0. (b) Logarithmic plot of the transition matrix for the Harvard500 data set.

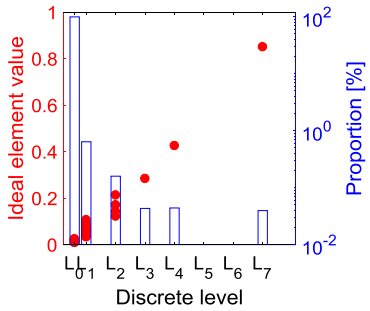

Fig. 5. Mapping result of the transition matrix of Harvard500 by the discrete levels of the RRAM, namely, the ideal value in the transition matrix as a function of the discrete level, and the corresponding proportion of discrete levels in the matrix implementation.

To simulate PageRank of the Harvard500 webpages within the eigenvector circuit, the transition matrix $\boldsymbol{T}$ was mapped in a cross-point array where RRAM devices were assumed to obey the 3-bit compact model of Section III-B. Fig. 5 shows the amplitude and proportion of each level in the discretized matrix $\boldsymbol{T}$. The top 0.04% largest entry in $\boldsymbol{T}$ were mapped by L_7, while other entries were mapped by lower levels. Due to the unique structure of $\boldsymbol{T}$, only six levels were needed to map the whole matrix, with 99% of the matrix entries falling within L_0.

Fig. 6(a) shows the dominant eigenvector of the discretized transition matrix, namely, the importance score of each Harvard500 webpage, obtained by the SPICE simulation of the cross-point circuit. Fig. 6(b) shows the correlation between the computed score and the analytical solution of the original transition matrix. The cosine similarity, defined as $\mathrm{Cosim} = (\boldsymbol{x_1} \cdot \boldsymbol{x_2})/(\|\boldsymbol{x_1}\| \cdot \|\boldsymbol{x_2}\|)$, where vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ represent, respectively, the ideal and simulated scores and $\|\cdot\|$ is the Euclidean norm, is equal to 0.98. Cosim appears a suitable figure of merit for the correctness of the webpage ranking, as it is more affected by the more important pages with high score. Fig. 6(c) illustrates the ranking result of the top-10 pages from the ideal analytical solution (left) and the simulation results (right). Compared with the ideal solution, only one page is missed out, which is ranked in the 14th place in the simulation, while the 11th page in the ideal solution is ranked in eighth place in simulation. Some of the positions of the other nine pages are interchanged, which would not impose an obvious influence in the search results. These results support the feasibility of implementing the PageRank with discrete conductance levels of RRAM device.

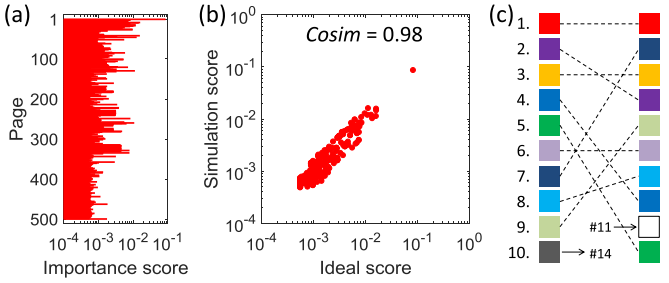

Fig. 6. PageRank results for the discretized transition matrix. (a) Normalized importance scores of the 500 pages. (b) Simulated score as a function of the ideal solution, with a Cosim of 0.98. (c) Comparison between the ideal (left) and the simulated (right) ranking results of the top-10 positions.

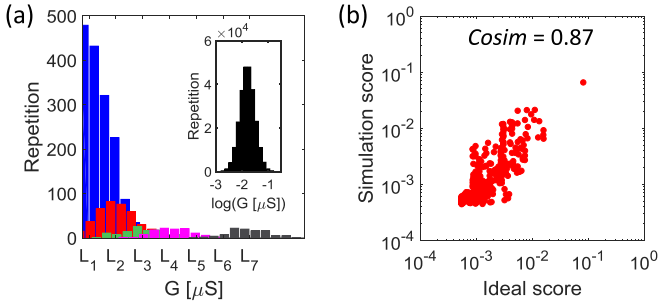

Fig. 7. (a) Conductance level distribution of the transition matrix mapping according to the RRAM statistical model considering conductance variations. (b) PageRank results from the RRAM, as a function of the ideal solution, showing a relatively low Cosim of 0.87.

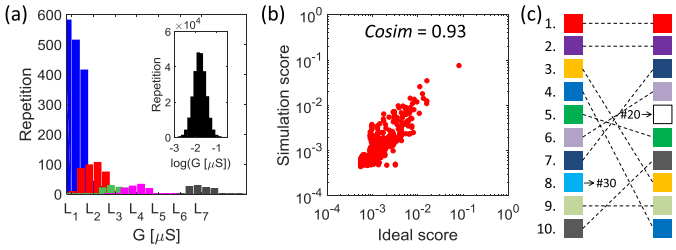

Fig. 8. PageRank results with a program-verify algorithm. (a) Conductance level distribution of the matrix mapping result by the RRAM model with a single verify pulse. (b) PageRank results from the program-verify RRAM model transition matrix, as a function of the ideal solution, showing a significantly improved Cosim. (c) Comparison between the ideal (left) and the simulated (right) ranking results of the top-10 positions.

While simulations in Fig. 6 assume idealized distributions with zero standard deviations, a more realistic study should consider the conductance variation of each level in the cross-point array. Fig. 7(a) shows the conductance level distribution of mapping the transition matrix of Harvard500, assuming that the experimental deviation $\sigma_G = 3.8$ μS in Fig. 3(b). The dominant eigenvector of the transition matrix $\boldsymbol{T}$ mapped by the discrete and statistical RRAM model was simulated in the circuit. Fig. 7(b) shows the correlation between the simulated scores and the ideal values, indicating a relatively low Cosim of 0.87. To improve the ranking accuracy, we adopted a program-verify algorithm aimed at reducing the device variation for matrix implementation. In the program-verify technique, if the device conductance falls out of the central $\pm\sigma$ range of the target level, an additional programming operation

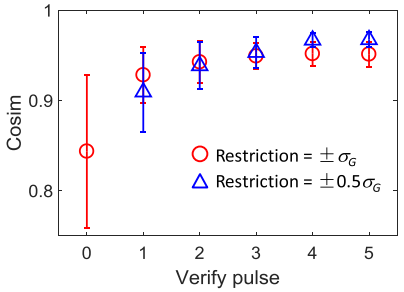

Fig. 9. Average Cosim as a function of the number of verify pulses, with corresponding standard deviations. Ten simulations were conducted for each verify scheme. The targeted conductance is restricted in the $\pm\sigma$ or $\pm 0.5\sigma$ range.

is executed. Fig. 8(a) shows the resulting conductance distribution for the program-verify technique. Fig. 8(b) shows the simulated dominant eigenvector, indicating a significantly improved Cosim = 0.93. The webpage ranking in Fig. 8(c) shows that nine pages appear in the top-10 positions though the missed page is more deviated from the correct position.

We have systematically studied the impact of the verify algorithm on ranking accuracy. Fig. 9 shows the calculated Cosim as a function of the number of verify pulses. For each verify, simulations were carried out. The single-pulse verify scheme provides the most significant improvement for the ranking result, with the average Cosim increasing from 0.85 to 0.93. As the number of verify pulses increases, the accuracy reaches a saturation region of Cosim = 0.95. The program-verify algorithm, thus, improves the accuracy of the eigenvector circuit to implement the PageRank algorithm. If the device conductance is constrained within the central $\pm 0.5\sigma$ range for the verify scheme, an even higher Cosim up to 0.97 can be achieved, which, however, increases the time and energy consumptions for device programming, thus implicating an accuracy-time tradeoff. Another important concern about the cross-point circuit is the wire resistance, especially for large array implementations. To alleviate its impact on the results of the ranking, optimized memory devices operating with relatively high resistance values can be used [5], [8]. Also, adopting intermediate interconnect technology nodes for cross-point arrays is helpful to keep the wire resistance sufficiently low though in contrast with the aggressive downscaling of conventional high-density memory [16].

IV. Performance Analysis

To study the throughput and energy efficiency of the eigenvector circuit for PageRank computation, we simulated the transient behavior of the circuit to compute with SPICE. Fig. 10 shows the simulated output voltages as a function of time for all 500 webpages, indicating a response time of 43.8 μs to reach the norm of error of 0.1% compared with the steady-state solution. In the simulation, a supply voltage V_{DD} of ± 1 V was provided to the amplifiers, while the maximum output voltage was limited to 0.5 V.

The performance of the in-memory computing circuit can be compared with conventional digital computers, where PageRank is solved by the power iteration method [17].

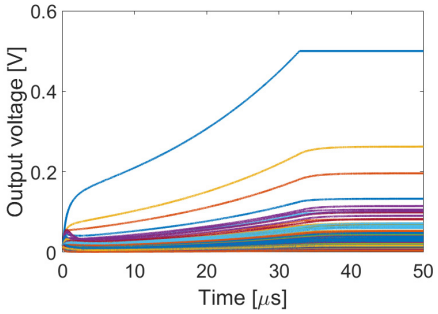

Fig. 10. Transient behavior the eigenvector values in Fig. 8 according to SPICE simulations of the circuit of Fig. 1. The maximum output voltage was limited to be 0.5 V in the simulation. The normalized result of the steady-state output voltages gives the importance scores given in Fig. 8.

In this case, for a network of N pages, the required number of floating-point operations is zN^2, where z is the number of iterations. We simulated the power iteration method in MATLAB for the case in Fig. 8, resulting in 32 iterations to achieve the same accuracy as the eigenvector circuit. During the power iteration simulation, the same initial solution as in Fig. 10 was assumed, which is the noise voltages of inverters in SPICE. As a result, the equivalent throughput of the in-memory circuit is $(32 \cdot 500^2)/(43.8 \cdot 10^{-6}) = 0.183$ tera-operations per second (TOPS).

The power consumption of the circuit was estimated by summing the product of the output current and the supply voltage for each amplifier, which is the lower bound for power consumption of analog circuits [18], and applies to both the analog inverters and the TIAs in Fig. 2(a). The resulting power consumed by the cross-point array and the inverters is given by

$$P_1 = \sum_{i,j=1}^{500} G_{T_{ij}} V_j V_{\mathrm{DD}} = 252.5\ \mu\mathrm{W} \tag{4}$$

while the power consumed by the TIAs is given by

$$P_2 = \sum_{i=1}^{500} G_\lambda V_i V_{\mathrm{DD}} = 252.5\ \mu\mathrm{W}. \tag{5}$$

Note that P_1 and P_2 are equal since the feedback conductance is identical to the equivalent conductance of the RRAM cross-point array, as expressed by (2). The overall power consumption is, thus, 505 μW, corresponding to equivalent energy efficiency of $(0.183/(505 \cdot 10^{-6})) = 362$ TOPS/W. The low power and, hence, the high energy efficiency of the circuit are attributed to the relatively low-conductance range of the memory device, as well as the special structure of the transition matrix of PageRank, where most elements are represented by the reset state of the device. Compared with the energy efficiency of 2.3 TOPS/W of the tensor processing unit (TPU) [19], the cross-point circuit provides 157 times better performance, thus supporting the high efficiency of in-memory computing for practical big-data tasks.

V. Conclusion

This article demonstrates an in-memory PageRank accelerator based on the cross-point array of RRAM devices. The RRAM device was experimentally characterized to provide parameters for a compact statistical model of RRAM. By using the RRAM model and a program-verify algorithm, the PageRank of the Harvard500 data set was computed by the circuit, with a cosine similarity of 95% for the ranking result with respect to the floating-point solution. Due to the one-step computing approach and the sparse nature of the citation matrix in PageRank, the cross-point eigenvector circuit demonstrates clear advantages over conventional computing in terms of throughput and energy efficiency. These results support in-memory computing as a fast, low-power architecture to accelerate PageRank and other big data tasks.

Acknowledgment

This work was partially carried out at Polifab, Micro- and Nanofabrication Facility, Politecnico di Milano.

References

[1] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on Memristive systems," *Nature Electron.*, vol. 1, no. 1, pp. 22–29, Jan. 2018.
[2] D. Ielmini and H.-S.-P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, Jun. 2018.
[3] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Front. Neurosci.*, vol. 10, pp. 1–13, Jul. 2016.
[4] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu, "Sparse coding with Memristor networks," *Nature Nanotechnol.*, vol. 12, no. 8, pp. 784–789, Aug. 2017.
[5] C. Li *et al.*, "Analogue signal and image processing with large Memristor crossbars," *Nature Electron.*, vol. 1, no. 1, pp. 52–59, Jan. 2018.
[6] M. Le Gallo *et al.*, "Mixed-precision in-memory computing," *Nature Electron.*, vol. 1, no. 4, pp. 246–253, Apr. 2018.
[7] M. A. Zidan *et al.*, "A general Memristor-based partial differential equation solver," *Nature Electron.*, vol. 1, no. 7, pp. 411–420, Jul. 2018.
[8] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 10, pp. 4123–4128, Mar. 2019.
[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. SIDL-WP-1999-0120, 1998.
[10] C. B. Moler, *Numerical Computing With MATLAB*. Philadelphia, PA, USA: Indian Mathematical Society, 2004, pp. 23–30.
[11] Z. Sun, E. Ambrosi, A. Bricalli, and D. Ielmini, "Logic computing with stateful neural networks of resistive switches," *Adv. Mater.*, vol. 30, no. 38, Sep. 2018, Art. no. 1802554.
[12] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980.
[13] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018.
[14] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Statistical fluctuations in HfO_x resistive-switching memory: Part I–Set/reset variability," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2912–2919, Aug. 2014.
[15] D. Ielmini, "Resistive switching memories based on metal oxides: Mechanisms, reliability and scaling," *Semicond. Sci. Technol.*, vol. 31, no. 6, Jun. 2016, Art. no. 063002.
[16] S. Yu, P.-Y. Chen, Y. Cao, L. Xia, Y. Wang, and H. Wu, "Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect," in *IEDM Tech. Dig.*, Dec. 2015, p. 17.
[17] K. Bryan and T. Leise, "The $25,000,000,000 eigenvector: The linear algebra behind Google," *SIAM Rev.*, vol. 48, no. 3, pp. 569–581, Jan. 2006.
[18] C. Svensson and J. J. Wikner, "Power consumption of analog circuits: A tutorial," *Analog. Integr. Circ Signal Process.*, vol. 65, no. 2, pp. 171–184, Nov. 2010.
[19] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2017, pp. 1–12.