Yuan, X, Chai, Z, Zhou, X, Luo, Y, He, Y, Jian, J, Yue, X, Zhang, JF, Zhang, W and Min, T

Hardware Estimation for the Eigenvectors of Stochastic Matrices using Magnetic Tunnel Junctions

Hardware Estimation for the Eigenvectors of Stochastic Matrices using Magnetic Tunnel Junctions

Xihui Yuan, Zheng Chai, Xue Zhou, Yongjie Luo, Yingtong He, Jiajia Jian, Jian Fu Zhang, Weidong Zhang and Tai Min

***Abstract*—Matrices are the foundation of science and engineering. For artificial intelligence (AI) and Internet of Things (IoT) tasks, developing a hardware efficient way to find the eigenvector of stochastic matrix is urgently in need. In this paper, inspired by the divide-and-conquer strategy, we proposed a new hardware architecture, which uses magnetic tunnel junctions (MTJs) to estimate the eigenvector of an n×n SM where n is the power of 2. This approach reduces the required device amount to $\log_2 n$ by converting the larger SM into 2-state SMs which are further represented by stochastic signals generated by MTJs. The validity of this method has been demonstrated and statistically evaluated. This method provides a novel hardware solution to solve mathematic problems using emerging hardware technologies.**

***Index Terms*—Stochastic matrix, eigenvector, Markov chain, magnetic tunnel junction**

I. Introduction

Matrices play a crucial role in the fields of mathematics, science, engineering, and many other disciplines [1]. Among the many matrix operations, finding the eigenvalues and eigenvectors of matrices [2] have practical applications in stability analysis, structural engineering, quantum mechanics, data analysis, etc. In computer science, Google's PageRank algorithm [3] [4] measures the importance of webpages on the Internet by calculating the eigenvector of the "Google matrix," a matrix which represents the links between webpages.

From a mathematical perspective, the Google matrix is a stochastic matrix (SM), a matrix with non-negative elements where the sum of elements in each row is equal to one **(Fig. 1a)**. The maximum eigenvalue of an SM, known as the spectral radius, is one [5]**.** In PageRank, the eigenvector corresponding to this eigenvalue is used to rank the importance of webpages. Apart from the PageRank algorithm, SM is also widely used in data analysis, probability theory, statistics, etc. [6] [7]. In all these applications, finding the eigenvectors of SM is critically important.

So far, eigenvectors are typically calculated using algorithms, including the direct methods such as Gaussian elimination [2], or the iterative methods such as the Power method [8]. Those algorithms are carried out on computers or even super computers for tasks like information processing and image compression. However, due to the speed and energy requirement of artificial intelligence (AI) and Internet-of-Things (IoT) hardware, calculating the eigenvectors of matrices, especially SM, using specifically designed hardware with higher efficiency, is highly desirable. Memristive crossbars [9] [10] [11] have been used to find the eigenvector in SM, but since the crossbar is of the same dimension of the SM, this lead to a polynomial rise (in the n^2 way) in the crossbar size as the SM dimension n increases. Despite the nanometer scale of today's memristive device, the chip's area might become intolerable, considering that the dimension of Google matrix could exceed 10 million [12]. Therefore, an alternative way to obtain the eigenvectors of SM, with architecture innovation to reduce device amount and chip area, is urgently in need.

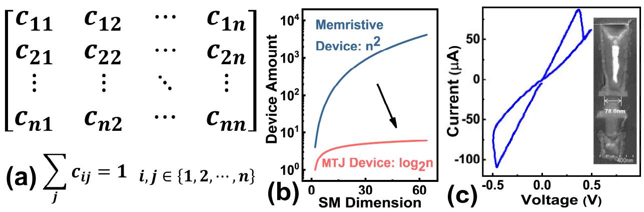

Fig.1. (a) Mathematical definition of an SM of dimension n. (b) The required device amount reduces from n^2 (in the memristive crossbar method) to $\log_2 n$ in the proposed method. (c) I-V curve of the MTJ's magneto-resistive switching, with its cross-section SEM image in the inset.

In this paper, we propose a new hardware architecture to estimate the eigenvector of certain SMs, based on non-volatile memristive devices. After converting the n×n SM (n is the power of 2) into $\log_2 n$ 2×2 SMs and representing each 2×2 SM with a Markov chain signal generated from a memristive device, only $\log_2 n$ devices are needed. As a result, the required device amount reduces from n^2 (in the memristive crossbar method) to $\log_2 n$ in the proposed method (**Fig. 1b**). The functionality of this architecture has been proved, by using spintronic magnetic tunnel junctions (MTJs) [13] [14], a typical non-volatile memristive technology. The proposed architecture provides a promising solution to the development of compact circuits for emerging applications.

Manuscript received xx. This work is supported by the National Key R&D Program of China (Grant No. 2022YFB4400200), the National Natural Science Foundation of China (Grant No. 62104188 and 12327806). (Corresponding author: Zheng Chai, Tai Min.)

Xihui Yuan, Zheng Chai, Xue Zhou, Yongjie Luo, Yingtong He, Jiajia Jian and Tai Min are with the Center for Spintronic and Quantum Systems, State Key Laboratory for Mechanical Behavior of Materials, and School of Materials Science and Engineering, Xi'an Jiaotong University, Xi'an, China, and also with the Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China, and also with the Pazhou Laboratory (Huangpu), Guangzhou, Guangdong 510555, China (e-mail: zheng.chai@xjtu.edu.cn, tai.min@mail.xjtu.edu.cn).

Jian Fu Zhang and Weidong Zhang are with the School of Engineering, Liverpool John Moores University, L3 3AF Liverpool, U.K.

II. Device and Experiment

The experiments are based on a bottom-pinned perpendicular magnetization anisotropy (PMA) MTJ with a diameter of 78 nm. The magneto resistive switching between its low-resistance parallel (P) state and high-resistance antiparallel (AP) state is illustrated in **Fig. 1c,** along with the cross section scanning electron microscopy (SEM) image of the MTJ (**inset of Fig. 1c**). All electrical measurements were performed using the pulse measurement units (PMUs) in a Keithley 4200 semiconductor analyzer.

III. Results and Discussions

In mathematics, an n×n SM describes the transitions of an n-state Markov chain (MC) [15] [16]. The stationary distribution of the MC equals to the eigenvector corresponding to the eigenvalue 1 of the SM. In our earlier works [17] [18], we have developed methods to physically generate and modulate 2-state MC signals using a single MTJ, and obtained their stationary distributions, based on MTJ's spintronic probabilistic switching nature [19]. However, for SMs larger than 2×2, how to use MTJs to estimate eigenvector remains a question.

It is worth noting that the divide-and-conquer strategy [20], which involves breaking down a complex problem into smaller and more manageable parts, solving each part individually, and then combining the solutions to solve the original problem, is the basis of many efficient algorithms in mathematics and computer science. For example, in the famous Strassen algorithm [21] [22], an n×n matrix is divided into sub-matrices of size n/2×n/2. This is done iteratively, thereby reducing the complexity of matrix multiplication from $O(n^3)$ to $O(n^{log_2 7})$.

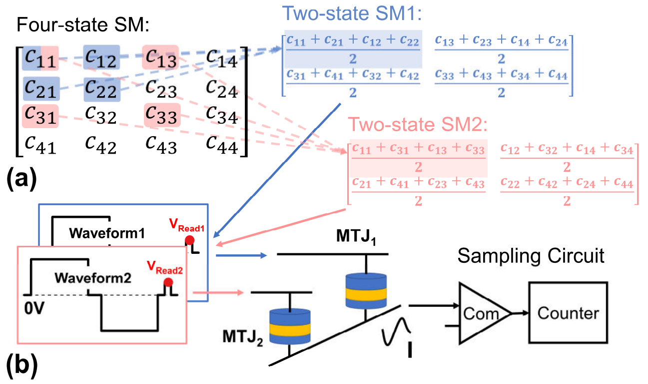

Fig.2. (a) The way that the 16 elements in a 4×4 SM are re-grouped in two different ways and averaged to form two 2×2 SMs. (b) The circuit schematic showing how the 2×2 SMs represented by MTJs and how the stationary distribution of the 4×4 SM can be obtained.

Following the divide-and-conquer strategy, we developed a hardware approach to estimate the eigenvector of SMs, which includes converting a larger SM (with dimension being the power of 2) into 2×2 SMs, encoding the 2×2 SMs in the 2-state MC signals generated by MTJs, re-merging the 2-state MC signals, and finally reading its stationary distribution to estimate the eigenvector of the original larger SM. Take a 4×4 SM for example: first of all, the 16 elements in the 4×4 SM are re-grouped in two different ways and averaged respectively to form two 2×2 matrices, in the way as **Fig. 2a** shows. Obviously, the 2×2 matrices are also SMs as their rows sum to one, which enables them to be individually represented by an MTJ device probabilistically switched by the 1st and 2nd pulse in the three-pulse waveform. In the circuit schematic of **Fig. 2b**, the read pulses (the 3rd in the three-pulse waveforms) with same width are synchronously applied onto the top electrodes of the two MTJs. The read pulses are designed with different amplitudes ($V_{Read1} > V_{Read2}$) to make the 4-state signal distinguishable by the sampling circuit. At the same time, a merged readout current, which has 4 levels, can be obtained from the connected bottom electrodes of the 2 MTJs. After digitization, this 4-level current is transferred into a 4-state MC signal and finally, its stationary distribution can be further obtained by simply counting the occurrence of the 4 states, which is an estimated eigenvector of the original 4×4 SM.

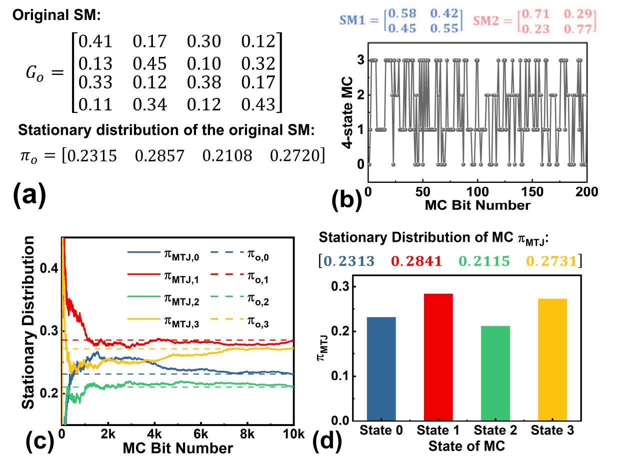

Fig.3. (a) An original 4×4 SM G_o with eigenvector π_o for the demonstration of the proposed method. (b) The two 2×2 SMs, and the 4-state MC generated from the MTJs. (c) As the length of MC increases, (d) the experimental eigenvector π_{MTJ} gradually approaches the eigenvector π_o calculated with MATLAB.

Fig. 3 shows a practical demonstration of this method, with an original 4×4 SM G_O in **Fig. 3a** converted into two 2×2 SMs and then represented by two MTJs in the form of MC signals, which are later merged into a 4-state MC including state 0, 1, 2 and 3 (**Fig. 3b**). Since $V_{Read1} > V_{Read2}$, the state 1 in the 4-state signal corresponds to MTJ_1 in P state and MTJ_2 in AP state, while the state 2 corresponds to MTJ_1 in AP state and MTJ_2 in P state. The stationary distribution of the 4-state MC leads to an eigenvector π_{MTJ}, which gradually approaches to the original SM's eigenvector π_o calculated with MATLAB, as the length of MC increases (**Fig. 3c**). This trend agrees with the many computation paradigms that involves stochasticity. Finally, with 10,000 bits, the stationary distribution of the 4-state MC is very close to the original one (**Fig. 3d**).

To evaluate the accuracy of this estimation methods, the similarity of the two eigenvectors, π_{MTJ} and π_o, is evaluated using the cosine similarity, a mathematical metric frequently used in statistics and machine learning to compared two vectors [23] [24]. Cosine similarity is defined in (1):

$$cosim = \frac{\pi_{MTJ}\pi_r^T}{\|\pi_{MTJ}\|\|\pi_r\|} \tag{1}$$

where $\|\cdot\|$is the Euclidean norm. When its value is closer to one, it means the two vectors are more similar. The cosine similarity in the example in **Fig. 3** is 99.999%, supporting the validity of our method.

Beyond the 4×4 SM, this method is applicable to SMs with larger dimensions. For example, an 8×8 SM can be divided into three 2×2 SMs as shown in **Fig. 4a**. It can be inferred that for any n×n SM, where n is a power of 2, the original SM can be converted into $\log_2 n$ 2-state SMs and represented by $\log_2 n$ MTJs. The effectiveness and generalizability of this method for larger dimension SMs is evaluated. An SM is randomly generated, then its eigenvector is estimated and compared to the mathematical value. For statistical reason, this process is repeated for 1000 times for a certain n. **Fig. 4b** demonstrates the averaged cosine similarity between the π_{MTJ} and π_o with n = 4, 8, 16 and 32, which gradually increases with larger matrix dimension, approaching 99% for 32×32 SMs.

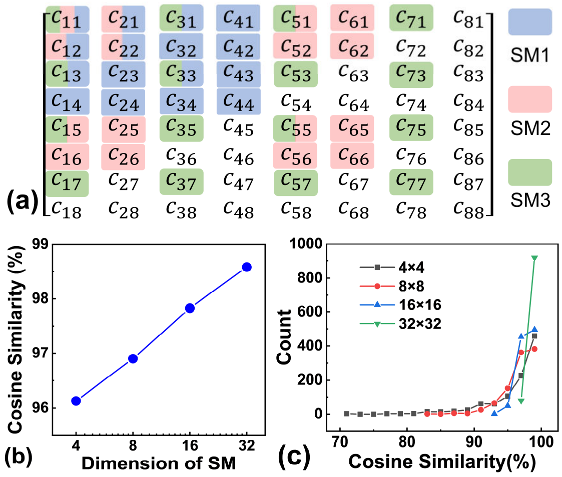

Fig.4. (a) An 8×8 SM is divided into three 2×2 SMs. (b) The averaged cosine similarity between π_o and π_{MTJ} with n = 4, 8, 16 and 32 gradually increases with larger matrix dimension, approaching 99% for 32×32 SMs. (c) As the 1000 estimates for each dimension show, the majority of the cosine similarities of SMs are greater than 90%. Matrices with larger dimensions show higher estimation cosine similarity.

While the overall averaged cosine similarity is good, for some SMs with smaller dimension (e.g. 4×4), the cosine similarity could be lower, as demonstrated in **Fig. 4c**. We assume that it might be attributed to the fact that this method only keeps the sums of partial elements from the high-dimensional SM in 2×2 SMs, thereby the information of individual elements is, to various extent, lost. This is somehow similar to the convolution or average pooling in convolution neural networks [25]. Nevertheless, the improved estimation accuracy with larger matrix dimension is probably due to the more significant averaging effect among the larger amount of elements. That might be why the proposed method works better on larger SMs which are more common in practical operations. In addition, at this stage, this method is only applicable to SM whose dimension is power of 2. We will further expand this method for SMs and even a wider scope of matrices with arbitrary dimension. In addition, the high latency of this method, a drawback common in many stochastic-related computing paradigms, could be alleviated by introducing more devices to carry out this method in parallel.

IV. Conclusions

In this paper, inspired by the divide-and-conquer strategy, we proposed a new hardware architecture using MTJs to estimate the eigenvector of an n×n SM where n is the power of 2. This approach reduces the required device amount to $\log_2 n$ by converting the SM into 2-state SMs which are further represented by MC signals generated by MTJs. The validity of this method has been demonstrated and statistically evaluated. This method provides a novel hardware solution to solve mathematic problems using emerging hardware technologies.

Reference

[1] A. Jeffrey, Matrix Operations for Engineers and Scientists, Dordrecht, Netherlands: Springer, 2010, doi: 10.1007/978-90-481-9274-8.

[2] C. Woodford and C. Phillips, "Eigenvalues and Eigenvectors," in *Numerical Methods with Worked Examples: Matlab Edition*, Dordrecht, Springer, 2012, pp. 215-229, doi: 10.1007/978-94-007-1366-6_10.

[3] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking : Bringing Order to the Web," in *The Web Conference*, California, USA, 1999, doi: 10.1145/1060745.1060828.

[4] A. N. Langville and C. D. Meyer, Google's PageRank and Beyond: The Science of Search Engine Rankings, Princeton, NJ, USA: Princeton University Press, 2006, doi: 10.1007/s10791-008-9063-y.

[5] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society,* vol. 7, p. 733, 1963, doi: 10.2307/2034984.

[6] G. Ghoshal and A.-L. Barabási, "Ranking stability and super-stable nodes in complex networks," *Nature Communications,* vol. 2, no. 394, pp. 1-7, 2011, doi: 10.1038/ncomms1396.

[7] J. L. Morrison, R. Breitling, D. J. Higham and D. R. Gilbert, "GeneRank: Using search engine technology for the analysis of microarray experiments," *MC Bioinformatics,* vol. 6, no. 233, pp. 1-14, 2005, doi: 10.1186/1471-2105-6-233.

[8] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet M athematics,* vol. 1, no. 3, pp. 335-380, 2004, doi: 10.1080/15427951.20 04.10129091.

[9] 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, 2019, doi: 10.1073/pnas.1815682116.

[10] G. Zoppo, A. Korkmaz, F. Marrone, S. Palermo, F. Corinto and R. S. W illiams, "Analog Solutions of Discrete Markov Chains via Memristor C rossbars," *IEEE Transactions on Circuits and Systems I: Regular Paper s,* vol. 68, no. 12, pp. 4910-4923, 2021, doi: 10.1109/tcsi.2021.312647 7.

[11] Z. Sun, E. Ambrosi, G. Pedretti, A. Bricalli, and D. Ielmini, "In-memory PageRank accelerator with a cross-point array of resistive memories," *IEEE Trans. Electron Devices,* vol. 67, no. 4, pp. 1466-1470, 2020, doi: 10.1109/ted.2020.2966908.

[12] L. Ermann, A. D. Chepelianskii and D. L. Shepelyansky, "Toward two-dimensional search engines," *J. Phys. A: Math. Theor.,* vol. 45, pp. 1-20, 2012, doi: 10.1088/1751-8113/45/27/275101.

[13] B. Dieny, I. L. Prejbeanu, K. Garello, P. Gambardella, P. Freitas, R. Lehndorff, W. Raberg, U. Ebels, S. O. Demokritov, J. Akerman, A. Deac, P. Pirro, C. Adelmann, A. Anane, A. V. Chumak, A. Hirohata, S. Mangin, et al., "Opportunities and challenges for spintronics in the microelectronics industry," *Nature Electronics,* vol. 3, p. 446–459, 2020, doi: 10.1038/s41928-020-0461-5.

[14] H. Zhao, A. Lyle, Y. Zhang, P. K. Amiri, G. Rowlands, Z. Zeng, J. Katine, H. Jiang, K. Galatsis, K. L. Wang, I. N. Krivorotov and J.-P. Wang, "Low writing energy and sub nanosecond spin torque transfer switching of in-plane magnetic tunnel junction for spin torque transfer

random access memory," *Journal of Applied Physics* , vol. 109, p. 07C720, 2011, doi: 10.1063/1.3556784.

[15] W.-K. Ching, X. Huang, M. K. Ng and T.-K. Siu, Markov Chains: Models, Algorithms and Applications, Berlin: Springer Sci. Bus. Media, 2013, doi: 10.1007/978-1-4614-6312-2.

[16] H. Tian, X.-F. Wang, M. A. Mohammad, G. -Y. Gou, F. Wu, Y. Yang and T.-L. Ren, "A hardware Markov chain algorithm realized in a single device for machine learning," *Nature Communications,* vol. 9, no. 4305, pp. 1-11, 2018, doi: 10.1038/s41467-018-06644-w.

[17] X. Yuan, J. Jian, Z. Chai, S. An, Y. Gao, X. Zhou, J. F. Zhang, W. Zhang and T. Min, "Markov Chain Signal Generation based on Single Magnetic Tunnel Junction," *IEEE Electron Device Letter,* vol. 44, no. 12, pp. 1963 - 1966, 2023, doi: 10.1109/led.2023.3322194.

[18] X. Yuan, J. Jian, Z. Chai, H. Wei, Y. Wen, Y. Liu, W. Yan, Y. He, X. Zhou, J. F. Zhang, W. Zhang and T. Min, "Arbitrary Modulation of Average Dwell Times in Discrete-Time Markov Chains based on Tunneling Magnetoresistance Effect," *IEEE Electron Device Letter,* 2024, doi: 10.1109/led.2024.3394711.

[19] W. F. Brown, "Thermal Fluctuations of a Single-Domain Particle," *Phys. Rev,* vol. 130, no. 5, pp. 1677-1686, 1963, doi: 10.1103/physrev.130.1677.

[20] D. R. Smith, "The design of divide and conquer algorithms," *Sci. Comput. Program.,* vol. 5, pp. 37-58, 1985, doi: 10.1016/0167-6423(85)90003-6.

[21] Strassen, V., "The asymptotic spectrum of tensors and the exponent of matrix multiplication," in *27th Annual Symposium on Foundations of Computer Science*, Toronto, ON, Canada, 1986, doi: 10.1109/sfcs.1986.52.

[22] Strassen V., "Gaussian elimination is not optimal," *Numer. Math.,* vol. 13, p. 354–356, 1969, doi: 10.1007/bf02165411.

[23] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications,* vol. 68, pp. 13-18, 2013, doi: 10.5120/11638-7118.

[24] A. Huang, "Similarity Measures for Text Document Clustering," in *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, vol. 4, pp. 9-56, Christchurch, New Zealand, 2008.

[25] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech and time series," in *Handbook of Brain Theory Neural Networks*, MA, USA, MIT Press, 1998, pp. 255-258, doi: 10.5555/303568.303704.

[26] D. F. Gleich, "PageRank Beyond the Web," *SIAM Review,* vol. 57, no. 3, pp. 321-363, 2015, doi: 10.1137/140976649.

[27] W. Cai, Y, Huang, X. Zhang, S. Wang, Y. Pan, J. Yin, K. Shi and W. Zhao, "Spintronics intelligent devices," *Sci. China Phys. Mech. Astron.,* vol. 66, no. 11, pp. 1-27, 2023, doi: 10.1007/s11433-022-2012-2.