The Lynchpin of In-Memory Computing: A Benchmarking Framework for Vector-Matrix Multiplication in RRAMs

Md Tawsif Rahman Chowdhury
Electrical and Computer Engineering
Wayne State University
Detroit, USA
mtawsifrc@wayne.edu

Huynh Quang Nguyen Vo
Industrial Engineering and Management
Oklahoma State University
Stillwater, USA
lucius.vo@okstate.edu

Paritosh Ramanan
Industrial Engineering and Management
Oklahoma State University
Stillwater, USA
paritosh.ramanan@okstate.edu

Murat Yildirim
Industrial and Systems Engineering
Wayne State University
Detroit, USA
murat@wayne.edu

Gozde Tutuncuoglu
Electrical and Computer Engineering
Wayne State University
Detroit, USA
gozde@wayne.edu

***Abstract*—The Von Neumann bottleneck, a fundamental challenge in conventional computer architecture, arises from the inability to execute fetch and data operations simultaneously due to a shared bus linking processing and memory units. This bottleneck significantly limits system performance, increases energy consumption, and exacerbates computational complexity. Emerging technologies such as Resistive Random Access Memories (RRAMs), leveraging crossbar arrays, offer promising alternatives for addressing the demands of data-intensive computational tasks through in-memory computing of analog vector-matrix multiplication (VMM) operations. However, the propagation of errors due to device and circuit-level imperfections remains a significant challenge. In this study, we introduce MELISO (In-Memory Linear Solver), a comprehensive end-to-end VMM benchmarking framework tailored for RRAM-based systems. MELISO evaluates the error propagation in VMM operations, analyzing the impact of RRAM device metrics on error magnitude and distribution. This paper introduces the MELISO framework and demonstrates its utility in characterizing and mitigating VMM error propagation using state-of-the-art RRAM device metrics.**

***Index Terms*—RRAM, crossbar, vector-matrix multiplication, linear solver, error distribution, beyond Von Neumann**

I. Introduction

The proliferating computation complexities and the associated energy consumption patterns are becoming a pressing concern in today's digital era. Projections indicate that by 2040, the cumulative energy consumption for computer operations could soar to an astonishing 10^{27} Joules, surpassing the anticipated capacity for energy production [1]. Conventional computing systems, rooted in the Von Neumann architecture, encounter fundamental obstacles in keeping pace with this exponential growth. The *Von Neumann bottleneck* emerges as a pivotal constraint, wherein the simultaneous execution of fetch and data operations grinds to a halt due to the bottleneck imposed by a shared bus connecting the processing and memory units [2–4]. Consequently, system performance is throttled, exacerbating energy consumption and imposing limitations on computational complexity. A recent study investigating Google server workloads reports that 62.7% of the total energy was allocated to data movements rather than computations in these systems [5]. This inefficiency not only exacerbates energy consumption but also imposes a significant computational overhead, impeding the realization of computational tasks with higher complexity.

Emerging non-volatile memory technologies, such as resistive random access memories (RRAMs) based crossbar arrays, provide a versatile solution to the challenges of data-intensive, large-scale computational tasks [6, 7]. By co-locating memory and computation nodes (i.e. performing in-memory computing), RRAM computing eliminates the need for resource-intensive communication tasks, thereby fundamentally improving computational complexity and energy use.

In a conventional RRAM crossbar array, each word line is connected to the bit line through an RRAM device. This architecture is used to perform vector-matrix multiplication (VMM) operations, the foundational computing block, and lynchpin for a myriad of modern computing tasks. Facilitating an analog VMM, the output current at each column, I_{ij}, is computed as the sum of the products of the input voltages and the corresponding conductance values of each row, expressed mathematically as $I_{ij} = \sum V_i \bullet G_{ij}$. This mechanism underscores the computational efficiency inherent in the RRAM crossbar design for implementing VMM operations, with improved energy efficiency and reduced latency metrics [8–16]. Moreover, RRAM reduces the effective computational complexity of dense VMM operations from $O(n^2)$ to $O(1)$. The efficiency gains achieved in VMM operations can serve as

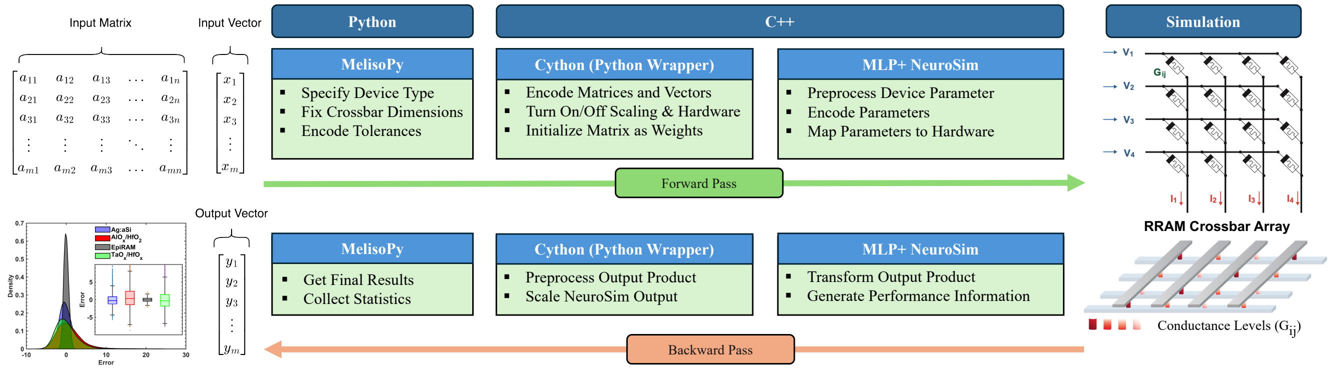

Input Matrix
Input Vector
$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$
$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix}$
Python
C++
Simulation
MelisoPy
Specify Device Type
Fix Crossbar Dimensions
Encode Tolerances
Cython (Python Wrapper)
Encode Matrices and Vectors
Turn On/Off Scaling & Hardware
Initialize Matrix as Weights
MLP+ NeuroSim
Preprocess Device Parameter
Encode Parameters
Map Parameters to Hardware
Forward Pass
V_1
V_2
V_3
V_4
G_{ij}
I_1
I_2
I_3
I_4
RRAM Crossbar Array
Conductance Levels (G_{ij})
Output Vector
$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$
MelisoPy
Get Final Results
Collect Statistics
Cython (Python Wrapper)
Preprocess Output Product
Scale NeuroSim Output
MLP+ NeuroSim
Transform Output Product
Generate Performance Information
Backward Pass
Ag:aSi
AlO_x/HfO_2
EpiRAM
TaO_x/HfO_x
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
0
10
20
30
Error
5
0
-5
Error

Fig. 1: Overview of MELISO: An end-to-end VMM benchmarking framework

the lynchpin and enable cascading benefits across a spectrum of computing tasks. These tasks extend beyond conventional neural network computations, providing a highly configurable computational platform for conducting any linear computation tasks, such as solving linear algebra and optimization problems that demand extensive matrix-vector multiplications and linear equations. The combination of low computational complexity, low latency, and energy-efficient operations provides an ideal computational setting for managing large volumes of data that are typical to these applications [17].

It should be noted that a parallel compounding phenomenon also unfolds concerning the propagation of errors within VMM operations due to issues related to device non-idealities (e.g., low-precision RRAMs, and cycle-to-cycle (C-to-C) variations) and the challenges of CMOS integration. Depending on the algorithmic application, this variability can constitute either a challenge or an asset. For algorithms with strict precision requirements, it becomes an important challenge to devise methods to mitigate this variability and its impact on algorithm performance. For algorithms that require sampling such as Markov Chain Monte Carlo used in Bayesian learning schemes [18], these variabilities can be leveraged as realizations of sampled uncertainties, further reducing the computational burden. It is crucial to understand the relationship between physical device parameters and the algorithm requirements to achieve a holistic algorithm-device co-design framework.

TABLE I: State-of-the-Art Device Metrics

Device Type	***Ag:a-Si***	***TaO_x/HfO_x***	***AlO_x/HfO_2***	***EpiRAM***
CS	97	128	40	64
Non-linearity	2.4/-4.88	0.04/-0.63	1.94/-0.61	0.5/-0.5
R_{ON} (Ω)	26M	100K	16.9K	81K
MW	12.5	10	4.43	50.2
C-to-C (%)	3.5	3.7	5	2

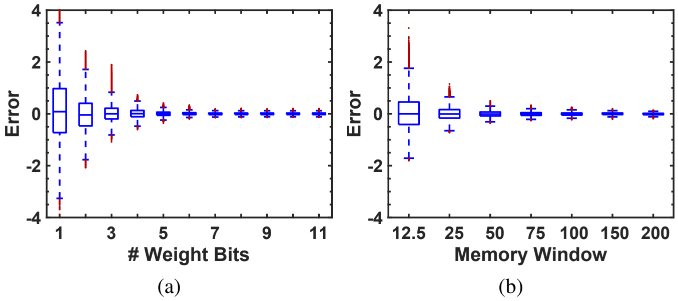
**Ag:a-Si* [27], *TaO_x/HfO_x* [34], *AlO_x/HfO_2* [35], *EpiRAM* [36]
(CS: Conductance States, MW: Memory Window)

Extensive literature exists on benchmarking frameworks designed to evaluate RRAM device performance and integration with CMOS peripheral circuitry for a variety of computational tasks such as image classification with fully connected and convolutional neural networks [19–21], as well as dot product engines [22]. As RRAMs critically suffer from device non-idealities, most prominently non-linear conductance tuning, C-to-C, and device-to-device variations, benchmarking frameworks need to incorporate these key issues [23, 24]. Functioning as a circuit-level macro model and benchmarking tool, NeuroSim+ serves as a prominent example in this regard, designed for evaluating neuro-inspired architectures [25, 26]. This framework quantifies essential circuit-level performance metrics (e.g., chip area, latency) by integrating modifications at the device output, circuit, and algorithm levels.

Realization of the computational benefits of RRAM devices necessitates a concerted effort toward the identification, modeling, and characterization of the propagation of errors within VMM operations. This characterization paves the way for a new generation of hardware and algorithmic methods to contain or harness these error terms to ensure high levels of computational accuracy. In response to this need, this paper presents a benchmarking framework designed to provide a comprehensive analysis of the error propagation in VMM operations for different RRAM devices and device properties. The contributions of this work can be summarized as follows:

- We develop an end-to-end VMM benchmarking framework for RRAMs, called MELISO - *In-Memory Linear Solver*. MELISO builds on NeuroSim+ [25, 26] to provide capabilities for compute-in-memory VMM, a foundational mathematical operation that lies at the core of every computing task.
- We test and analyze the impact of RRAM device parameters and chemistries on the error terms observed in VMM tasks. We implement a comprehensive benchmarking study across a statistically significant population of RRAM devices and VMM operations. We analyze patterns in error distributions with respect to different device chemistries, and device parameters such as C-to-C variability and nonlinearities.
- We analyze and identify the parametric distributions that best represent the errors in RRAM VMM operations to guide the error assumptions of the subsequent algorithm design efforts that aim to mitigate or leverage error propagation for in-memory computing applications.

The rest of the paper proceeds as follows. Section II introduces the methodology used for the realization of the benchmarking framework. An extensive set of results are demonstrated in Section III. Section IV concludes the paper by elucidating the contributions and future research outlook.



(a)

(b)

Fig. 2: Effect of a) Weight Bits b) Memory Window on VMM error term (w/out non-linearity and C-to-C)

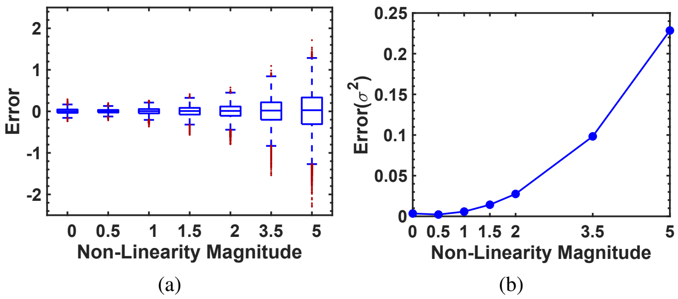
II. Methodology

We propose an end-to-end VMM benchmarking framework called MELISO, which can be described in terms of two distinct stages pertaining to forward and backward computational steps. At the very outset of the forward step, the the matrix and vector inputs are defined in a Python-based environment. Next, device types, crossbar dimensions, and tolerances are specified using a Python module named MelisoPy and a Cython wrapper. The information is then transferred to MLP+NeuroSim, written in C++, to encode the necessary parameters for hardware simulation to compute VMM on RRAM devices. In the backward computational step, the resulting vector of VMM from the forward pass is then scaled and transformed, thus generating the final result as well as performance statistics. Finally, at the output level, the results are collected and analyzed. The entire sequence of steps and the overall design of the MELISO framework is represented in Figure 1.

To perform VMM experiments for this particular study, we considered a population of crossbar arrays with the size of 32 rows and 32 columns. 1000 of 32×32 matrices, corresponding to vector $\boldsymbol{A}$, and 1000 of 32×1 vectors corresponding to $\boldsymbol{x}$ were randomly generated which have then been multiplied using the crossbar to generate 1000 many dot products $\boldsymbol{A}.\boldsymbol{x}$ of dimension 32×1. The computed values were then compared with the software-calculated dot product to quantify the error. These 32×1 error terms were then concatenated creating a 32000×1 vector accumulating all the errors from a population of identical devices. This method helps the implementation of a statistically significant number of VMM operations to derive reliable insights into error propagation across devices.

III. Results

In Figure 2, we examine the impact of critical device metrics of memory window and weight bit on the error terms observed in VMM operations described in the methodology section. Memory window is defined as the ratio of maximum and minimum conductance levels, G_{max}/G_{min}, while weight bit, also referred to as weight precision, corresponds to the maximum number of RRAM conductance states during weight update. We chose *Ag:a-Si* [27] as a model system due to its reasonable performance in the multi-layer perceptron architecture tested for the MNIST classification task on NeuroSim V3.0 [26]. We performed two critical modifications in the default device properties of the *Ag:a-Si* metrics presented in Table I: i) increased the memory window from the default 12.5 to 100, to accommodate a wider range of conductance states, ii) switched off the C-to-C variation and non-linearity parameters to evaluate the effect of memory window and weight bit metrics independently, without compounding the effects non-linearity and C-to-C variation. These modifications will be rolled back in the subsequent experiments.



(a)

(b)

Fig. 3: Effect of Non-Linearity on VMM error term

Our findings demonstrate a clear dependence of both the magnitude and variance of error terms on the studied device performance metrics. As illustrated in Figure 2a, there is a notable decrease in error magnitude and variance as the weight bits (number of conductance states) increase from 1-bit (2 states) to 11-bit (2048 states). The upper limit of 2048 conductance states was chosen because it represents the current limit in RRAM device technology as the recently reported largest number of conductance states [28]. Similarly, Figure 2b shows a reduction in error as the memory window is increased beyond the initial value of 12.5. These findings are consistent with earlier studies, which showed that an increase in the number of weight bits and memory windows would lead to better precision in encoding synaptic weight and reduce the need for weight quantization [29, 30]. This brings our error rates closer to the results of digital computations that use a standard 32-bit floating-point configuration.

Following our initial examination of primary device metrics, memory window, and weight bits, we further investigated the impact of device non-idealities on performance. Among these non-idealities, weight update non-linearity is particularly critical, as frequently highlighted in the literature [29]. It poses a significant challenge to the deployment of RRAM arrays in online training tasks due to its propensity to cause computational errors through incorrect encoding of synaptic

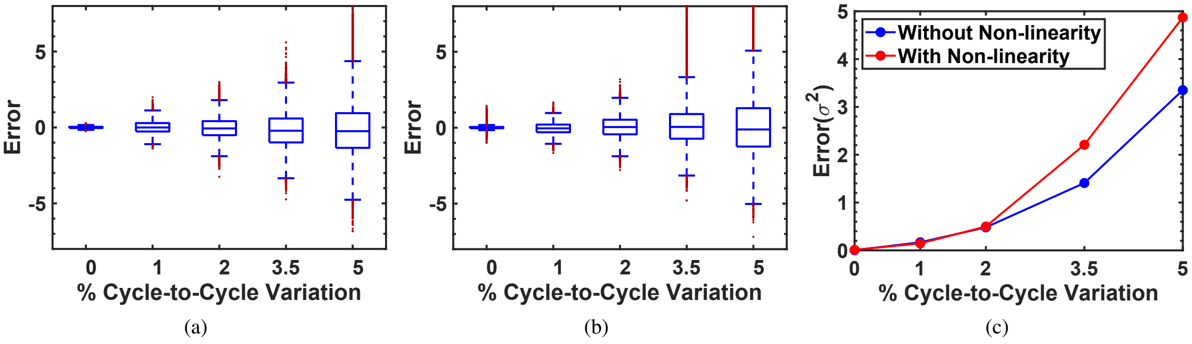


Fig. 4: Effect of the C-to-C variation on VMM error term. a) without considering the non-linearity, b) in the presence of non-linearity, and c) Comparing variance for both cases.

weights. This non-linearity renders the implementation of additional algorithms or circuit-level methodologies, such as write-and-verify techniques, essential to mitigate its effects and ensure the reliable operation of RRAM-based systems in real-world applications.

In Figure 3, we explore the effect of weight update non-linearity using the modified model system of *Ag:a-Si* [27]. This device system has the reported non-linearity metrics, $2.4/\text{-}4.88$, by default, but we varied the non-linearity magnitude from 0 to 5 in this particular study. Our findings indicate that increases in non-linearity metrics substantially exacerbate the error terms in VMM operations. Additionally, the relationship between error variance and the degree of non-linearity demonstrates an exponential dependency. This dependency reflects the underlying exponential non-linear synaptic weight encoding methodology employed in NeuroSim+ [25].

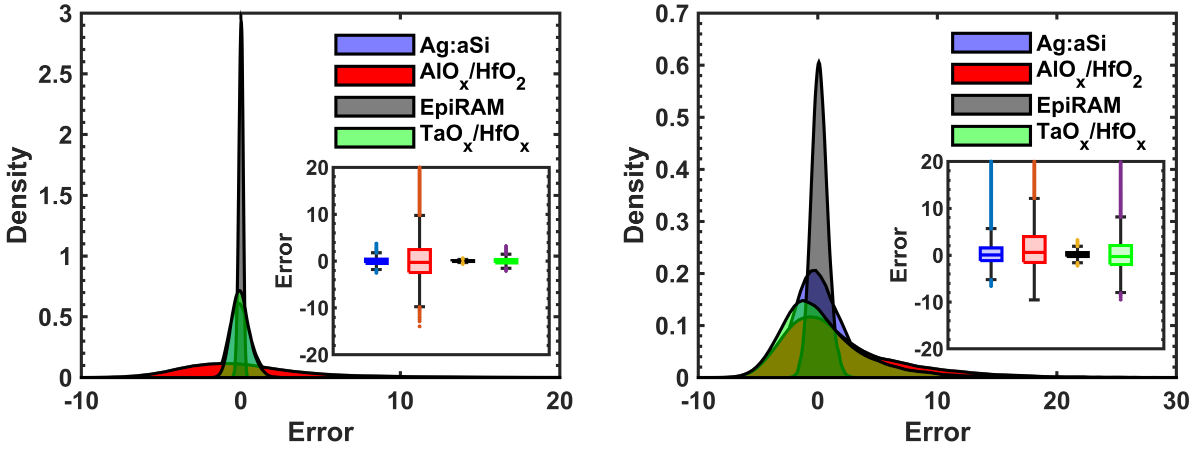
Next, we investigated the impact of C-to-C variation, another significant non-ideality in RRAM devices. C-to-C variation has been demonstrated to be particularly challenging to reduce below certain limits [33]. This variability introduces additional errors each time synaptic weights are updated (re-encoded) within the RRAM array. State-of-the-art devices studied in this section (see Table I) exhibit C-to-C values ranging from 2% to 5%. This range represents the performance limitations of the devices in the literature, as C-to-C is often reported as a critical problem. Despite potential mitigations provided by materials, device design, or operational level modifications, RRAMs are inherently more susceptible to C-to-C variations due to the stochastic nature of atomic-level chemical and physical resistive switching mechanisms that are omnipresent in RRAM technology [31].

Figure 4 illustrates the relationship between VMM error terms and C-to-C variation in the modified *Ag:a-Si* model system, comparing two configurations with (Figure 4a) and without (Figure 4b) considering non-linearity. Fig. 4c presents the variance comparison of both cases. The data spans a range of C-to-C standard deviations from 0% to 5%. We observe a significant increase in the error term as a function of C-to-C, with the highest error rates recorded in this study corresponding to the largest C-to-C values. Even the baseline C-to-C metric of 3.5% of our model system introduces a substantial level of error, underscoring the critical influence of C-to-C variation on device performance. As expected, the introduction of non-linearity exacerbates the VMM error term, evidenced by the larger variance presented in Figure 4c.

Finally, we conducted a benchmarking study of four different RRAM crossbar systems, with device metrics extracted from the literature (see Table I) as reported in NeuroSim+ V3.0 [26]. We evaluated these systems for identical VMM tasks, both with and without considering the effects of device non-idealities such as non-linearity and C-to-C variability. The results of these experiments are depicted in Figure 5, where a and b illustrate the error distributions for scenarios without and with non-idealities, respectively. Additionally, insets in each figure present the VMM error terms as box plots.

Our findings reveal significant differences between the two configurations. In the absence of non-idealities (Figure 5a), the error distributions across the devices are relatively narrow, with the exception of *AlO_x/HfO_2*. These devices exhibit similar performance profiles, while *EpiRAM* [36] stands out with an exceptionally narrow distribution. Conversely, when we accounted for the non-idealities of C-to-C variability and non-linearity (Figure 5b), there was a noticeable increase in both the spread and magnitude of the error distributions. Under these conditions, while the *EpiRAM* still remains the best-performing device, the performance disparity among the other devices becomes more significant. The *Ag:a-Si* [27] and *TaO_x/HfO_x* [34] systems exhibit similar device performances, with the latter presenting an increased 25 and 75 percentile errors. Both systems clearly outperform *AlO_x/HfO_2* [35].

These results can be explained by considering the device metrics presented in Table I. The *EpiRAM* device with the largest memory window (50.2), lowest cumulative weight update non-linearity (0.5/-0.5), and C-to-C variation (2%) exhibits the smallest error magnitude in VMM operations, thus achieving the highest performance. The second-best performing device, *Ag:a-Si*, demonstrates significant weight update non-linearity (2.4/-4.88) but benefits from a larger memory window of 12.5 and lower C-to-C variation (3.5%), surpassing both the *TaO_x/HfO_x* and *AlO_x/HfO_2* systems according to the error box plot in Fig. 5b. Similar error distributions of *Ag:a-Si* and *TaO_x/HfO_x* can be attributed to their comparable device



(a)

(b)

Fig. 5: Effect of non-idealities on VMM performance of different device types, a)Without non-linearity and C-to-C variation, b)With non-linearity and C-to-C variation

metrics: i.e. although *Ag:a-Si* exhibit a higher non-linearity term, memory window, and C-to-C characteristics across these devices are comparable. At the lower end of the performance spectrum, the *AlO_x/HfO_2* system exhibits the smallest memory window, coupled with the lowest number of conductance states (weight bit) and C-to-C variation, alongside considerable non-linearity. These device metrics manifest themselves by significantly reducing device performance compared to the other benchmarked devices.

It should be noted that the performance of *EpiRAM*, *Ag:a-Si* and *TaO_x/HfO_x* deteriorates with the introduction of non-linearity and C-to-C variation. These observations are consistent with our earlier findings illustrated in Figure 2, 3, and 4, where we systematically explored the impact of memory window, non-linearity, and C-to-C variability on error dynamics. An interesting pattern emerges when we closely study the performance trends of *AlO_x/HfO_2*. Due to low conductance states, poor memory window, and high C-to-C variation, the variability introduced through device non-idealities slightly improves device performance. This is specifically prominent when we study the outliers: i.e. as shown in Figure 5, the 25 and 75 percentile errors increase slightly while shifting from ideal to non-ideal *AlO_x/HfO_2* performances, however, the noise introduced by nonidealities reduces the span of outliers, resulting in a slight improvement to distributional metrics of the errors, as shown in Table II.

Table II presents an overview of the empirical distribution of the errors with best fitting models and the corresponding distribution moment characteristics for *Ag:a-Si*, *TaO_x/HfO_x*, *AlO_x/HfO_2* and *EpiRAM*, across two key device non-ideality parameters: e.g. non-linearity and C-to-C. Our results indicate that the errors typically follow one of the following distributions: Johnson S_u, Normal-3-Mixture, Normal-2-Mixture, and Sinh-Arc-Sinh (SHASH). The error distributions do not follow a typical normal distribution due to the presence of asymmetry and heavy tail behavior, which are measured through skewness and kurtosis metrics, respectively. The analysis shows that among the four devices studied in this paper, *AlO_x/HfO_2* exhibits the highest variance, which is followed by *Ag:a-Si* and *TaO_x/HfO_x*. This trend is seen in our experiments with both ideal and non-ideal device parameters.

Studying the skewness and kurtosis allows us to understand trends in error distribution that cannot be explained through box plots or mean and variance metrics alone. As indicated in Table II, *Ag:a-Si* (with non-idealities) and *AlO_x/HfO_2* (without non-idealities) exhibit high positive skewness indicating a longer tail on the right side of the distribution. *Ag:a-Si* with non-linearity and C-to-C variability also has the highest kurtosis indicating a heavier tail compared to *EpiRAM* or *TaO_x/HfO_x*. This analysis suggests that the skewness and kurtosis metrics are most sensitive to the device non-linearity properties: Although *Ag:a-Si* has better memory window and C-to-C properties compared to *TaO_x/HfO_x*, high non-linearity in *Ag:a-Si* causes its skewness and kurtosis metrics to be higher. More specifically, we observe that the non-linearity impacts are becoming more prominent only after going into the third and fourth moments (e.g. skewness and kurtosis) of the error distributions.

IV. Conclusion and Outlook

Successful deployment of RRAM devices for in-memory computing applications relies on conducting accurate VMM operations with improved energy efficiency and reduced latency metrics. This study introduces a comprehensive benchmarking framework, MELISO, designed to methodically analyze and model the error landscape of VMM operations conducted on RRAM crossbar systems, with the ultimate goal of guiding the development of more robust RRAM technologies and algorithms. Our extensive benchmarking reveals significant insights into the impact of RRAM de-

TABLE II: Statistical analysis of error distributions for each device material

Device Type	Non-linearity	C-to-C	Summary Statistics				
			Best Fit	**Mean**	**Variance**	**Skewness**	**Kurtosis**
Ag:a-Si	No	No	Normal-3-Mixture	-0.00084	0.4607	0.4639	0.4369
	Yes	Yes	Johnson S_u	0.7059	13.0763	3.3405	15.6567
AlO_x/HfO_2	No	No	Normal-3-Mixture	0.8311	32.0761	2.7935	13.3362
	Yes	Yes	Normal-3-Mixture	0.5247	13.9694	1.5065	3.7796
EpiRAM	No	No	SHASH	0.0044	0.0179	-0.2463	0.0256
	Yes	Yes	Normal-2-Mixture	0.1453	0.4630	0.1927	0.1744
TaO_x/HfO_x	No	No	Normal-3-Mixture	-0.0001	0.3336	0.4314	0.3761
	Yes	Yes	Normal-3-Mixture	0.4117	12.5167	1.2150	2.2775

vice parameters and non-idealities on error propagation. Key findings include distinct patterns in error distributions related to specific device characteristics, such as C-to-C variability and non-linear responses. Furthermore, we have reported the parametric distributions that most accurately represent the observed errors, providing a solid foundation for future algorithmic strategies aimed at either mitigating or harnessing these errors to enhance computational accuracy in memory-centric computing environments.

Future research will focus on enhancing our understanding of error propagation dynamics and their computational implications. Specifically, we plan to investigate the aspects of neuromorphic device virtualization and parallelization primitives for RRAM arrays. Additionally, our future work also involves the development of a suite of computationally efficient, general-purpose optimization libraries for convex and non-convex solvers, to address low-latency problems in power systems engineering. We will also conduct a more detailed study on optimizing RRAM device metrics for targeted computational tasks while considering performance and energy consumption benchmarking metrics. Through such concerted efforts, RRAM-based systems are expected to realize their full potential, contributing substantially to the next generation of computational technology.

V. Acknowledgment

Gozde Tutuncuoglu would like to acknowledge NSF-CRII grant 2153177.

References

[1] S. Das and E. Mao, “The global energy footprint of information and communication technology electronics in connected internet-of-things devices,” in *Sustainable Energy, Grids and Networks*, vol. 24, p. 100408, Dec. 2020. doi: 10.1016/j.segan.2020.100408

[2] X. Zou, S. Xu, X. Chen, L. Yan, and Y. Han, “Breaking the Von Neumann bottleneck: Architecture-level processing-in-memory technology,” in *Science China Information Sciences*, vol. 64, no. 6, Apr. 2021. doi: 10.1007/s11432-020-3227-1

[3] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland and D. Glasco, ”GPUs and the Future of Parallel Computing,” in *IEEE Micro*, vol. 31, no. 5, pp. 7-17, Sept.-Oct. 2011, doi: 10.1109/MM.2011.89.

[4] D. Kimovski et al., ”Beyond Von Neumann the Computing Continuum: Architectures, Applications, and Future Directions,” in *IEEE Internet Computing*, doi: 10.1109/MIC.2023.3301010.

[5] A. Boroumand et al., ”Google workloads for consumer devices: Mitigating data movement bottlenecks.” *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems* 2018, pp. 316-331. doi: 10.1145/3173162.3173177

[6] A. Shafiee et al., ”ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars,” *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, Seoul, Korea (South), 2016, pp. 14-26, doi: 10.1109/ISCA.2016.12.

[7] F. Aguirre et al., “Hardware implementation of memristor-based Artificial Neural Networks,” in *Nature Communications*, vol. 15, no. 1, Mar. 2024. doi: 10.1038/s41467-024-45670-9

[8] A. Amirsoleimani et al., “In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor memristor integrated circuits: Design choices, challenges, and Perspectives,” in *Advanced Intelligent Systems*, vol. 2, no. 11, Aug. 2020. doi: 10.1002/aisy.202000115.

[9] E. P. -B. Quesada et al., ”Experimental Assessment of Multilevel RRAM-Based Vector-Matrix Multiplication Operations for In-Memory Computing,” in *IEEE Transactions on Electron Devices*, vol. 70, no. 4, pp. 2009-2014, April 2023, doi: 10.1109/TED.2023.3244509.

[10] D. Soudry, D. Di Castro, A. Gal, A. Kolodny and S. Kvatinsky, ”Memristor-Based Multilayer Neural Networks With Online Gradient Descent Training,” in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2408-2421, Oct. 2015, doi: 10.1109/TNNLS.2014.2383395.

[11] Y. Liao et al., ”Novel In-Memory Matrix-Matrix Multiplication with Resistive Cross-Point Arrays,” *2018 IEEE Symposium on VLSI Technology*, Honolulu, HI, USA, 2018, pp. 31-32, doi: 10.1109/VLSIT.2018.8510634.

[12] S. K. Kingra et al., ”Methodology for Realizing VMM with Binary RRAM Arrays: Experimental Demonstration of Binarized-ADALINE using OxRAM Crossbar,” *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180915.

[13] Y. Kim, Y. Zhang, and P. Li, ”A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning,” *2012 IEEE International SOC Conference*, Niagara Falls, NY, USA, 2012, pp. 328-333, doi: 10.1109/SOCC.2012.6398336.

[14] S. Zhang, H. H. Li, and U. Schlichtmann, “Connection-based processing-in-memory engine design based on resistive crossbars,” *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, Jan. 2021. doi:10.1145/3394885.3431523.

[15] S. Yu, W. Shim, X. Peng and Y. Luo, ”RRAM for Compute-in-Memory: From Inference to Training,” in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2753-2765, July 2021, doi: 10.1109/TCSI.2021.3072200.

[16] S. Tang et al., ”AEPE: An area and power efficient RRAM crossbar-based accelerator for deep CNNs,” *2017 IEEE 6th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*, Hsinchu, Taiwan, 2017, pp. 1-6, doi: 10.1109/NVMSA.2017.8064475.

[17] Liu, Sijia, et al. ”A memristor-based optimization framework for artificial intelligence applications.” *IEEE Circuits and Systems Magazine* 2018, vol.18.1, pp.29-44. doi: 10.1109/mcas.2017.2785421

[18] T. Dalgaty et al., ”In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling.” *Nature Electronics* 2021, vol.4, no.2, pp.151-161. doi: 10.1038/s41928-020-00523-3

[19] X. Liu et al., ”RENO: A high-efficient reconfigurable neuromorphic computing accelerator design,” *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, 2015, pp. 1-6, doi: 10.1145/2744769.2744900.

[20] S. Qu, B. Li, Y. Wang, D. Xu, X. Zhao, and L. Zhang, ”RaQu: An automatic high-utilization CNN quantization and mapping framework

for general-purpose RRAM Accelerator," *2020 57th ACM/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, 2020, pp. 1-6, doi: 10.1109/DAC18072.2020.9218724.

[21] Z. Li et al., "RRAM-DNN: An RRAM and Model-Compression Empowered All-Weights-On-Chip DNN Accelerator," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1105-1115, April 2021, doi: 10.1109/JSSC.2020.3045369.

[22] N. Uysal, B. Zhang, S. K. Jha, and R. Ewetz, "DP-map," Proceedings of the 39th International Conference on Computer-Aided Design, Nov. 2020. doi:10.1145/3400302.3415683.

[23] L. Chen et al., "Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, Lausanne, Switzerland, 2017, pp. 19-24, doi: 10.23919/DATE.2017.7926952.

[24] B. Liu, Hai Li, Yiran Chen, Xin Li, Qing Wu and Tingwen Huang, "Vortex: Variation-aware training for memristor X-bar," 2015 *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, 2015, pp. 1-6, doi: 10.1145/2744769.2744930.

[25] P.-Y. Chen, X. Peng, S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures", *IEEE International Electron Devices Meeting (IEDM)*, 2017, San Francisco, USA. doi: 10.1109/IEDM.2017.8268337

[26] Y. Luo, X. Peng, and S. Yu, "MLP+NeuroSimV3.0: Improving On-chip Learning Performance with Device to Algorithm Optimizations," *2019 ACM the International Conference on Neuromorphic Systems (ICONS)*, New York, USA. doi: 10.1145/3354265.3354266

[27] S. H. Jo et al., "Nanoscale memristor device as Synapse in Neuromorphic Systems," in *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, Mar. 2010. doi:10.1021/nl904092h.

[28] M. Rao et al., "Thousands of conductance levels in memristors integrated on CMOS," in *Nature*, vol. 615, no. 7954, pp. 823–829, Mar. 2023. doi:10.1038/s41586-023-05759-5.

[29] P. -Y. Chen et al., "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Austin, TX, USA, 2015, pp. 194-199, doi: 10.1109/ICCAD.2015.7372570.

[30] P. -Y. Chen, X. Peng and S. Yu, "NeuroSim: A Circuit-Level Macro Model for Benchmarking Neuro-Inspired Architectures in Online Learning," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*s, vol. 37, no. 12, pp. 3067-3080, Dec. 2018, doi: 10.1109/TCAD.2018.2789723.

[31] G. Tutuncuoglu and A. Mannodi-Kanakkithodi, "Role of defects in resistive switching dynamics of Memristors," in *MRS Communications*, vol. 12, no. 5, pp. 531–542, Sep. 2022. doi:10.1557/s43579-022-00243-z.

[32] L. Gao, P. -Y. Chen and S. Yu, "Programming Protocol Optimization for Analog Weight Tuning in Resistive Memories," in *IEEE Electron Device Letters*, vol. 36, no. 11, pp. 1157-1159, Nov. 2015, doi: 10.1109/LED.2015.2481819.

[33] J. B. Roldán et al., "Variability in resistive memories," in *Advanced Intelligent Systems*, vol. 5, no. 6, Mar. 2023. doi:10.1002/aisy.202200338.

[34] W. Wu et al., "A Methodology to Improve Linearity of Analog RRAM for Neuromorphic Computing," *2018 IEEE Symposium on VLSI Technology*, Honolulu, HI, USA, 2018, pp. 103-104, doi: 10.1109/VLSIT.2018.8510690.

[35] J. Woo et al., "Improved Synaptic Behavior Under Identical Pulses Using AlOx/HfO2 Bilayer RRAM Array for Neuromorphic Systems," in *IEEE Electron Device Letters*, vol. 37, no. 8, pp. 994-997, Aug. 2016, doi: 10.1109/LED.2016.2582859.

[36] S. Choi et al., "Sige epitaxial memory for neuromorphic computing with reproducible high performance based on engineered dislocations," in *Nature Materials*, vol. 17, no. 4, pp. 335–340, Jan. 2018. doi:10.1038/s41563-017-0001-5.