A Neural Network Approach for Calibrating Memristor Crossbars

Mohammad Rezaeifar*
He Chaoyi*
mrezaeifar@tamu.edu
chaoyi_he@tamu.edu
Departments of Electrical & Computer Engineering at Texas A&M University
College Station, Texas, USA

Kamran Entesari
Departments of Electrical & Computer Engineering at Texas A&M University
College Station, USA
kentesar@tamu.edu

Linda Katehi
Departments of Electrical & Computer Engineering and Materials Science & Engineering at Texas A&M University
College Station, USA
katehi@tamu.edu

Abstract

Memristor technology holds significant promise for both non-volatile memory and neuromorphic computing, owing to its compact structure and ability to emulate synaptic behavior. However, the performance of memristor-based crossbar arrays is often degraded by intrinsic non-idealities, including wire resistance and parasitic inductive and capacitive effects, particularly in 1T1R (one transistor–one memristor) architectures. These distortions compromise computational accuracy and are insufficiently addressed by current design methodologies.This paper introduces a neural network-based post-calibration framework that effectively compensates for these crossbar distortions. The proposed method maps the desired conductance matrix into a latent vector space and utilizes this representation alongside the measured output currents to reconstruct the intended outputs with high fidelity. By learning and correcting the systematic errors induced by parasitic and resistive effects, the framework substantially improves the accuracy and reliability of memristor crossbars.Experimental results demonstrate that the proposed approach robustly mitigates crossbar-induced errors and restores system performance, marking a substantial advancement in the practical deployment of memristor-based neuromorphic systems. This work contributes to the broader goal of enabling high-precision analog computing in emerging memory technologies.

CCS Concepts

• **In-Memory Computing** → **Memristive Devices as Emerging Technologies in Electronics**; • **Computing Error** → Machine learning.

Keywords

Memristor array, in-memory computing, memristor parasitics, memristor model, neural network.

1 Introduction

In the evolving landscape of computational hardware, memristor technology has emerged as a focal point of interest due to its unique ability to unify memory and logic in a compact, energy-efficient device. Its promise lies in both non-volatile memory applications and in emulating synaptic functions essential for neuromorphic computing systems. Memristors exhibit distinctive characteristics such as non-volatility, high integration density, and inherent non-linearity, making them strong candidates for next-generation computing architectures [1, 2].

However, transitioning memristor-based systems from theoretical models to practical implementations introduces a host of engineering challenges. One of the most critical among these is the degradation of performance in memristor crossbar arrays caused by parasitic effects, wire resistance, and voltage drops—particularly in 1T1R (one transistor–one memristor) structures. These non-idealities manifest as distortions that undermine the reliability and accuracy of analog matrix-vector multiplication, a core operation in neuromorphic and in-memory computing [3].

Although various hardware-level and circuit-level calibration strategies have been proposed—such as tuning the input voltages, redesigning peripheral circuitry, and introducing compensation mechanisms [4]—these methods often fall short in comprehensively addressing the underlying physical imperfections. As a result, system designers are frequently forced to accept trade-offs between accuracy, complexity, and energy efficiency.

Recent advancements suggest that data-driven methods, particularly those based on neural networks, hold significant promise in overcoming these limitations. Neural networks excel in learning complex, nonlinear mappings and can adaptively model the distortions introduced by parasitics and resistive losses in large-scale crossbar arrays [5]. This adaptability makes them ideal for post-fabrication calibration and error correction in analog hardware systems.

In this work, we propose a novel post-calibration methodology leveraging neural network-based modeling to restore computational accuracy in memristor crossbars. The core idea is to transform the desired conductance matrix into a compact latent vector representation. This latent vector, when used alongside the actual output currents measured from the crossbar, allows the neural network to infer and correct for systematic errors due to parasitic coupling and wire loss. Our approach not only addresses the shortcomings of traditional calibration techniques but also enhances the precision and robustness of memristor-based neuromorphic computing platforms.

The remainder of this paper is organized as follows: Section 2 provides a detailed overview of memristor crossbar architecture and its operational challenges. Section 3 outlines the data acquisition and memristor modeling process. Section 4 introduces the proposed neural network calibration framework and its integration with hardware simulations. Section 5 presents experimental evaluations demonstrating the efficacy of our approach. Finally, Section 6

*Both authors contributed equally to this research.

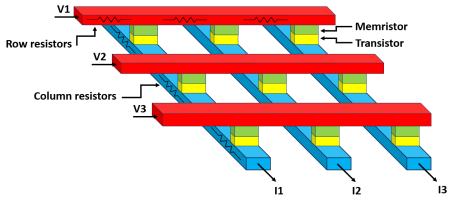

Figure 1: A memristor crossbar array structure.

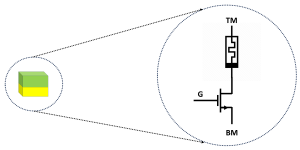

Figure 2: A memristor cell architecture integrating one transistor (1T) and one resistor (1R).

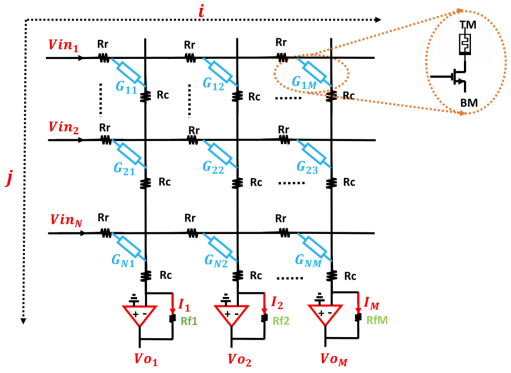

Figure 3: Memristor crossbar performing MVM. Conductance matrix G (size $N \times M$) is multiplied by voltage input vector V_{in} (size $1 \times N$).

concludes with a discussion of the broader implications and future directions of this research.

2 Crossbar Memristo

In-memory computing (IMC) represents a transformative paradigm shift in addressing the fundamental inefficiencies of traditional von Neumann architectures. Chief among these inefficiencies is the pronounced latency and energy overhead resulting from the physical separation of computation and memory units. This separation leads to significant data movement across the memory-compute interface, giving rise to the well-documented von Neumann bottleneck, which substantially limits system throughput and energy efficiency. IMC alleviates this bottleneck by integrating memory and logic into a unified platform, enabling data to be stored and processed within the same physical location. This co-location markedly enhances computational parallelism, reduces latency, and improves energy efficiency—especially in tasks involving parallel matrix–vector multiplication (MVM), a core operation in many signal processing, machine learning, and scientific computing applications [6–8].

At the heart of the IMC framework lies the memristor-based crossbar array, which serves as the fundamental computational engine (Fig. 1). These arrays consist of memristive elements positioned at the intersection of orthogonal metal lines, enabling dense, analog, and highly parallel MVM operations. Despite their architectural elegance and computational potential, memristor crossbars suffer from inherent analog non-idealities. Most notably, parasitic resistances from interconnect lines and undesired capacitive and inductive couplings introduce signal distortions that degrade the accuracy and stability of analog computations [7–9] .

To counter these non-idealities, many systems employ the 1T1R (one-transistor–one-resistor) architecture, wherein each memristive element is serially connected with a transistor (Fig. 2). This configuration enables precise control over individual cells, helps prevent sneak path currents, and enhances the overall robustness of the array. Additionally, the 1T1R architecture supports three-dimensional stacking, increasing the effective memory density and scalability of IMC platforms [10].

In analog-domain MVM execution, a voltage vector is applied across the crossbar rows (word lines), while memristive elements encode matrix coefficients as conductance values (Fig. 3). The resulting current on each column (bit line), governed by Ohm's and Kirchhoff's laws, corresponds to the dot product of the input voltage vector and the programmed conductance matrix. The output currents are typically processed via transimpedance amplifiers (TIAs) to convert them into usable voltage signals [11]:

$$I_j = \sum_{i=1}^{M} G_{ij} V_{\text{in},i} \tag{1}$$

Despite the efficiency of this approach, practical implementations face a variety of challenges due to parasitic effects. These include variations in memristive device behavior, resistance along metal interconnects, and capacitive/inductive couplings—each of which contributes to errors in the analog computation. Without appropriate mitigation, these effects can critically impair the accuracy of VMM operations in crossbar-based IMC systems [12, 13].

To address these issues, ongoing research explores several mitigation strategies. These include: (i) advanced fabrication techniques to reduce variability and interconnect resistance [4], (ii) circuit-level error-correction mechanisms [14], (iii) algorithm-level adaptations, such as modified training techniques that incorporate hardware-aware constraints [15], and (iv) dynamic compensation through control circuitry or post-processing. These multi-pronged efforts aim to enhance computational fidelity and accelerate the adoption of memristor-based IMC systems in real-world high-throughput applications [16, 17].

Device/circuit-level schemes such as bit-slicing and multi-device cells (e.g., 2T2R) improve stability and linearity at the cost of additional area and design complexity. The method proposed here is complementary: a system-level, post-fabrication calibration that operates on measured outputs to suppress residual IR-drop and parasitic-induced distortions. Combining architectural redundancy with the learned inverse can yield additive benefits.

3 Data Collection and Memristor Modeling

Accurate computational modeling in memristor-based systems critically depends on the faithful representation of memristor current-voltage (I–V) behavior. High-fidelity modeling is essential to ensure reliable system-level predictions and to facilitate the integration of memristive devices into larger mixed-signal and neuromorphic architectures. To achieve this, we incorporate detailed I–V profiles into a Cadence Virtuoso simulation environment, which allows for comprehensive circuit-level analysis, including the interaction of memristors with CMOS circuitry and parasitic effects. The inclusion of interconnect parasitics, non-ideal current paths, and transistor behaviors in such simulations is crucial for realistically evaluating memristor-based vector-matrix multiplication (VMM) systems.

In this study, memristor behavior is modeled using a custom Verilog-A implementation, adapted from the validated model presented in [17]. The Verilog-A description is integrated with the 22 nm Fully Depleted Silicon-On-Insulator (FD-SOI) CMOS technology to co-simulate memristive devices alongside standard CMOS transistors in Cadence Virtuoso. This hybrid modeling enables precise emulation of real-world device characteristics and offers insight into the performance and non-idealities of the 1T1R architecture under varying electrical stimuli.

A critical component of this work is the generation of a comprehensive dataset for both training and validating the neural network-based post-calibration model. The dataset is composed of the output current responses of memristor crossbar arrays subjected to a wide range of input and conductance conditions. Specifically, we generate:

- 1,000 distinct weight matrices, each with conductance values randomly selected from a uniform distribution between 16.66 μS and 100 μS, representing the memristive states.
- 100 random input voltage vectors for each matrix, with each vector containing eight voltage values ranging from 0.1 V to 0.5 V, generated using MATLAB.

Each combination of input voltage and conductance matrix is applied to the crossbar circuit in Cadence Virtuoso, and the resulting column currents are recorded. These output currents serve as the analog VMM results and form the target dataset for our calibration framework.

Because training is performed offline on the 100,000 VMM outputs and inference reduces to a lightweight feedforward evaluation, operational overhead is minimal.

One of the primary challenges encountered during this process is the automation of large-scale data acquisition within the Cadence simulation environment. Manually modifying input vectors and weight matrices for each simulation is time-consuming and error-prone. To address this, we developed a parametric testbench leveraging Verilog-A behavioral modeling. This automation significantly accelerates the simulation process by dynamically updating voltage and conductance parameters for each iteration, thus enabling scalable and efficient dataset generation for learning-based calibration models.

The resulting dataset, comprising 100,000 VMM outputs, captures the nonlinearities and parasitic influences inherent to physical implementations and forms the foundation for the neural post-calibration process described in subsequent sections.

4 Machine learning Model and Hardware Modeling

The crossbar array system is prone to various distortions, including linearity and non-linearity issues, as well as white Gaussian noise and transistor thermal noise. These disturbances affect the accuracy of the output currents. The actual output can be mathematically represented by a model where $\mathcal{F}$ denotes a nonlinear operator.

$$I_{\text{real}} = \mathcal{F}(M, V_{\text{in}}) + \varepsilon \tag{2}$$

This operator accounts for wire resistance, memristor non-linearity, and other distortions, such as phase shifts induced by the 1T1R transistors. To mitigate these non-idealities, we employ a nonlinear inverse operator, parameterized by θ, designed to counteract the effects of both crossbar system imperfections and noise:

$$\begin{aligned} \hat{I}_{\text{out}} &= \mathcal{G}(I_{\text{real}}, M) \equiv \mathcal{G}_\theta(I_{\text{real}}, M) = \tilde{I}_{\text{out}} \\ &\rightarrow \max_\theta p_\theta(\hat{I}_{\text{out}} | I_{\text{real}}, M) \\ &= \max_\theta p_\theta(\hat{I}_{\text{out}} | \mathcal{F}(M, V_{\text{in}}) + \epsilon, M) \end{aligned} \tag{3}$$

Its function is to enhance the fidelity of $\hat{I}_{\text{out}}$ given the conductance matrix of the crossbar and the measured output current vector. The goal is to progressively align the output of the inverse operator with the theoretical output, absent any crossbar system non-idealities.

While the experiments target primarily static parasitic distortions, the inverse operator $\mathcal{G}_\theta$ can be trained on temporally varying phenomena such as conductance drift, read-disturb, and device degradation. This is achieved by augmenting the dataset with measurements collected over stress/aging and by conditioning $\mathcal{G}_\theta$ on device state (e.g., time-since-program, read count, temperature), or by periodic fine-tuning on a small calibration set. As a post-fabrication calibration, this preserves hardware while adapting the learned correction to evolving device characteristics.

For the training of the inverse operator, we have constructed a machine learning model that serves as its proxy, as depicted in Fig. 4. Within this framework, a global convolution kernel is employed to extract the feature z_M from the ideal conductance matrix. This process confirms that a distinct ideal output vector can be deduced from the matrix and the actual (experimental) output vector. The objective of the model is not only to reproduce a series of instances but also to assimilate and condense their intrinsic and interrelated attributes into a streamlined, low-dimensional latent space. As a result, and as illustrated in Figure 4, the latent vector z_M is interpreted to embody critical, input-specific information through the encoding of the matrix instance. This encoding is then used as a supplemental conditional input for the neural network, which is fine-tuned to calibrate the crossbar output. The latent space vector is navigated through a series of concatenating layers in conjunction with the experimental current output vector I_{real}, as per the function:

$$\begin{aligned} O_{i+1} &= CS(O_i, z_m) \\ &= (W_1 O_i + b_1) \odot \sigma(W_2 z_m + b_2) + W_3 z_m, \\ i &= 0, 1, 2, \ldots, N-1, \end{aligned} \tag{4}$$

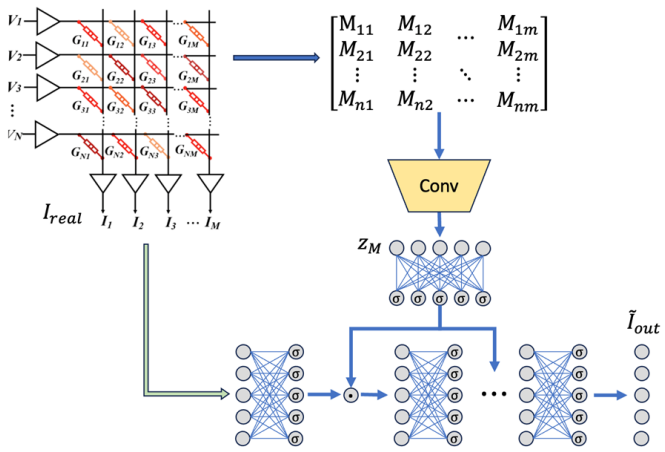

Figure 4: Model Structure

where N is the number of concatenating layers, initiating with $O_0 = I_{\text{real}}$ and culminating in $O_N = \tilde{I}_{\text{out}}$, σ denotes the sigmoid activation function. This methodical approach empowers the neural network to enhance the output iteratively, steering it towards an ideal condition by utilizing the latent vector for precise calibration.

where N is the number of concatsquash layers, initiating with $O_0 = I_{\text{real}}$ and culminating in $O_N = \tilde{I}_{\text{out}}$, σ, is the sigmoid activation function. This structured approach facilitates the neural network's capability to iteratively refine the output towards an ideal state, leveraging the latent vector to guide calibration accurately.

5 Experiment Results

The outcomes from the training and validation phases, illustrated in Figure 5, demonstrate that the model significantly enhances accuracy by reducing the mean squared error (MSE) between the actual crossbar currents and the ideal outputs by a factor of 1,000 ($\text{mean}(I_{\text{real}} - I_{\text{ideal}})^2$ vs $\text{mean}(\tilde{I}_{\text{out}} - I_{\text{ideal}})^2$) . This marked improvement is attributed to a key preprocessing step where all data values are rescaled by 10^5, addressing the issue that the original values were in the microscale (10^6) range. This rescaling is essential for several reasons: it prevents underflow and precision errors commonly associated with floating-point operations, aligns the data with the operational parameters of memristor crossbar arrays to boost sensitivity and efficiency, and facilitates better learning in neural networks by ensuring a uniform scale for features and a consistent flow of gradients. This preprocessing step significantly enhances the model's computational precision and overall efficacy by effectively adjusting the input data to meet the system's demands.

Given the lightweight inference and offline training, and the observed generalization to input types not seen during training, the approach does not require frequent retraining in practice.

The training phase uses random input voltage vectors and conductance matrices to validate the model's adaptability across different Vector Matrix Multiplication (VMM) contexts. In contrast, the validation phase employs sinusoidal input signals with conductance configured as a Finite Impulse Response (FIR) filter. Fig. 5 and Fig. 6 compare the network's outputs and the original crossbar responses,

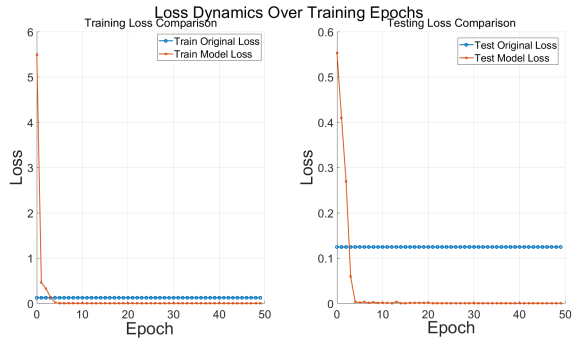

Figure 5: Training and Validating results

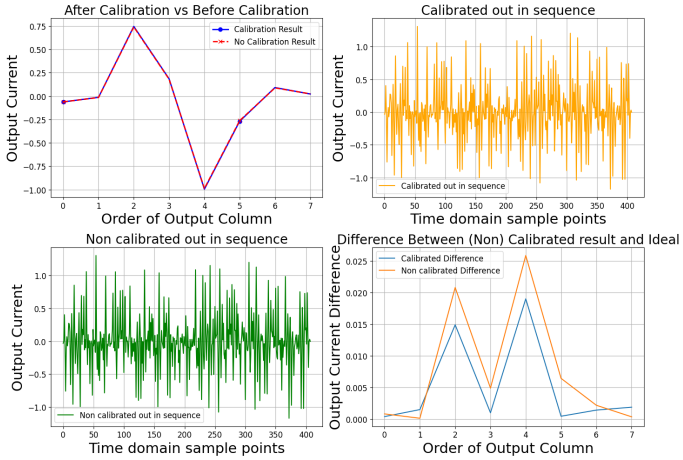

Figure 6: Visualization of the calibration results compared with crossbar original output.

demonstrating the model's accuracy. Table 1 comprehensively summarizes the performance metrics across the training, validation, and testing phases. The validation findings mainly highlight that the model's output quality is ten times superior to the original, even without being trained on conventional signals, showcasing its exceptional robustness and eliminating the necessity for frequent retraining.

Table 1: Training and Testing MSE Performance for Different Input Type

Input Data Type	$(\tilde{I}_{out} - I_{ideal})^2)$	$(I_{real} - I_{ideal})^2)$
Training with Random	0.000139	0.131878
Validating with Random	0.000149	0.122114
Testing with Sinusoidal	0.003398	0.032687
Testing with DC Input	0.003185	0.031794

6 Conclusion

The research presented in this paper introduces a cutting-edge, neural network-centric methodology for calibrating memristor crossbars, effectively overcoming the fundamental obstacles presented

by distortion and noise. The approach hinges on an innovative strategy of encoding the target conductance matrix into a latent space representation, enhancing memristor crossbars' precision and dependability for neuromorphic computing applications. The experimental outcomes corroborate the efficacy of this method, evidencing a substantial elevation in computational accuracy and the model's performance. Future endeavors will delve into the granularity of element-by-element memristor conductance tuning through neural networks. They will investigate the integration of this advanced approach into more sophisticated neuromorphic systems.

Future Work. Future directions will focus on several fronts. First, we plan to fabricate and test a silicon prototype to validate the calibration under real device noise, drift, temperature variations, and ageing effects. Second, the framework will be extended to explicitly handle time-dependent non-idealities such as conductance drift, read-disturb, and degradation, either by scheduled refresh of the training set or by lightweight conditional fine-tuning that adapts the learned inverse to evolving device states. Third, we will investigate the scalability of the repair network to larger or tiled crossbar arrays, where each tile can be encoded via a compact latent vector and calibrated in a batched or hierarchical manner. Finally, to facilitate reproducibility and adoption, we will release the Cadence simulation scripts, Verilog-A device models, and PyTorch training code.

Acknowledgments

The authors gratefully acknowledge the support of the Intelligent Electromagnetic Sensor Laboratories (IEMSL) at Texas A&M University for providing resources and guidance that contributed to the success of this work.

References

[1] Daniele Ielmini and H.-S. Philip Wong. In-memory computing with resistive switching devices. *Nature Electronics*, 1(6):333–343, 2018.

[2] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7):529–544, 2020.

[3] Indranil Chakraborty, Mustafa Ali, Aayush Ankit, Shubham Jain, Sourjya Roy, Shrihari Sridharan, Amogh Agrawal, Anand Raghunathan, and Kaushik Roy. Resistive crossbars as approximate hardware building blocks for machine learning: Opportunities and challenges. *Proceedings of the IEEE*, 108(12):2276–2310, 2020.

[4] Pai-Yu Chen, Deepak Kadetotad, Zihan Xu, Abinash Mohanty, Binbin Lin, Jiepeng Ye, Sarma Vrudhula, Jae sun Seo, Yu Cao, and Shimeng Yu. Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 854–859. IEEE, 2015.

[5] Hritom Das, Rocco D. Febbo, Charles P. Rizzo, Nishith N. Chakraborty, James S. Plank, and Garrett S. Rose. Optimizations for a current-controlled memristor-based neuromorphic synapse design. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 13(4):889–900, 2023.

[6] Chenchen Liu, Qing Qian, Bonan Yan, Jianlei Yang, Xiaocong Du, Weijie Zhu, Hao Jiang, Qing Wu, Mark Barnell, and Hai Li. A memristor crossbar based computing engine optimized for high speed and accuracy. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 110–115. IEEE, 2016.

[7] Corey Lammie, Olga Krestinskaya, Alex James, and Mostafa Rahimi Azghadi. Variation-aware binarized memristive networks. In *26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 490–493. IEEE, 2019.

[8] Nicola Lepri, Matteo Baldo, Paolo Mannocci, Andrey Glukhov, Vladimir Milo, and Daniele Ielmini. Modeling and compensation of ir drop in crosspoint accelerators of neural networks. *IEEE Transactions on Electron Devices*, 69(3):1575–1581, 2022.

[9] Cong Xu, Dimin Niu, Naveen Muralimanohar, Rajeev Balasubramonian, Tao Zhang, Shimeng Yu, and Yuan Xie. Overcoming the challenges of crossbar resistive memory architectures. In *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 476–488. IEEE, 2015.

[10] Huihan Li, Shaocong Wang, Xumeng Zhang, Wei Wang, Rui Yang, Zhong Sun, Wanxiang Feng, Peng Lin, Zhongrui Wang, and Linfeng Sun. Memristive crossbar arrays for storage and computing applications. *Advanced Intelligent Systems*, 3(9):2100017, 2021.

[11] Daniele Ielmini and Giacomo Pedretti. Device and circuit architectures for in-memory computing. *Advanced Intelligent Systems*, 2(7):2000040, 2020.

[12] Amirali Amirsoleimani, Fabien Alibart, Victor Yon, Jianxiong Xu, M. Reza Pazhouhandeh, Serge Ecoffey, Yann Beilliard, Roman Genov, and Dominique Drouin. In-memory vector-matrix multiplication in monolithic cmos-memristor integrated circuits: Design choices, challenges, and perspectives. *Advanced Intelligent Systems*, 2(11):2000115, 2020.

[13] Carlos Silva, Jonas Deuermeier, Weidong Zhang, Emanuel Carlos, Pedro Barquinha, Rodrigo Martins, and Asal Kiazadeh. Perspective: Zinc-tin oxide based memristors for sustainable and flexible in-memory computing edge devices. *Advanced Electronic Materials*, 9(11):2300286, 2023.

[14] Woorham Bae, Jin-Woo Han, and Kyung Jean Yoon. In-memory hamming error-correcting code in memristor crossbar. *IEEE Transactions on Electron Devices*, 69(7):3700–3707, 2022.

[15] Chenchen Liu, Miao Hu, John Paul Strachan, and Hai Li. Rescuing memristor-based neuromorphic design with high defects. In *Proceedings of the 54th Annual Design Automation Conference (DAC)*, pages 1–6. ACM, 2017.

[16] Tiancheng Cao, Chen Liu, Yuan Gao, and Wang Ling Goh. Parasitic-aware modeling and neural network training scheme for energy-efficient processing-in-memory with resistive crossbar array. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(2):436–444, 2022.

[17] Nicola Lepri, Matteo Baldo, Paolo Mannocci, Andrey Glukhov, Vladimir Milo, and Daniele Ielmini. Modeling and compensation of ir drop in crosspoint accelerators of neural networks. *IEEE Transactions on Electron Devices*, 69(3):1575–1581, 2022.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009