This is the peer reviewd version of the followng article:

Calibration and Enhancement of Analytical IR-Drop Model for Resistive Crossbar Arrays / Rizzi, T., Lambertini, A., Zanotti, T., Lepri, N., Pavan, P., Puglisi, F.M., Ielmini, D., Zambelli, C.. - In: IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. - ISSN 0278-0070. - (2026), pp. 1-1. [10.1109/TCAD.2026.3685531]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

01/11/2026 15:06

(Article begins on next page)

Calibration and Enhancement of Analytical IR-Drop Model for Resistive Crossbar Arrays

Tommaso Rizzi, Alessandro Lambertini, Tommaso Zanotti, *Member, IEEE*, Nicola Lepri, Paolo Pavan, *Senior Member, IEEE,* Francesco Maria Puglisi, *Senior Member, IEEE,* Daniele Ielmini, *Fellow, IEEE,* Cristian Zambelli, *Member, IEEE*

***Abstract*—Resistive In-Memory Computing (IMC) architectures emerged as a promising solution for efficiently accelerating Neural Network (NN) workloads on-chip, particularly in edge Artificial Intelligence (AI) applications where energy and area constraints are critical. Among the various challenges limiting IMC adoption, voltage drops caused by parasitic resistances significantly degrade the computational accuracy. While high-fidelity nodal analyses can model these effects accurately, their computational cost makes them unsuitable for system-level AI accelerators design. Analytical models offer a compelling alternative thanks to their speed and simplicity. However, the absence of a systematic method to fit them to different resistive memory devices and crossbar configurations might limit their applicability. In this work, we overcome this limitation by introducing a calibration procedure across a variety of input patterns and weight matrices. Furthermore, we evaluate the models under non-linear I–V characteristics of Resistive Random Access Memory (RRAM) devices fabricated at IHP Microelectronics. Finally, we propose an enhanced analytical model for the better estimation of the voltage drops across the memory cells. Our modeling strategy is benchmarked against state-of-the-art approaches, achieving best-in-class performance among the analytical models in terms of accuracy and runtime. With more than two orders of magnitude improvement in simulation speed and a 6.6% average misalignment in the output currents approximation compared to SPICE-based nodal analysis, the presented solution is a compelling approach for evaluating large-scale IMC systems.**

***Index Terms*—Parasitic, Wire Resistance, IR-drop, Crossbar, Analytical Model**

I. Introduction

RECENT advances in both hardware and software expanded the scope of Artificial Intelligence (AI), extending its reach beyond large-scale data centers to edge devices across a wide range of domains. Edge AI systems offer advantages such as reduced power consumption and enhanced data privacy by enabling local data processing. However, meeting the stringent area and energy constraints at the edge remains a critical challenge for conventional computing architectures [1]. To address these limitations, both academia and industry are exploring a design paradigm known as In-Memory Computing (IMC). IMC integrates data storage and computation within the same physical units, thereby drastically reducing the energy overhead associated with data movement. A promising approach for implementing IMC architectures leverages crossbar arrays of non-volatile resistive switching memory devices, such as Resistive Random Access Memory (RRAM) [2] and Phase Change Memory (PCM) [3]. These devices can store information in their resistance states. They also efficiently perform Vector-Matrix Multiplications (VMMs) directly within their crossbar structure, leveraging Ohm and Kirchhoff laws (Fig. 1). The great potential of resistive crossbars stems from the fact that modern Neural Networks (NNs) require the execution of a large number of VMM operations, making IMC solutions a promising implementation for edge AI applications.

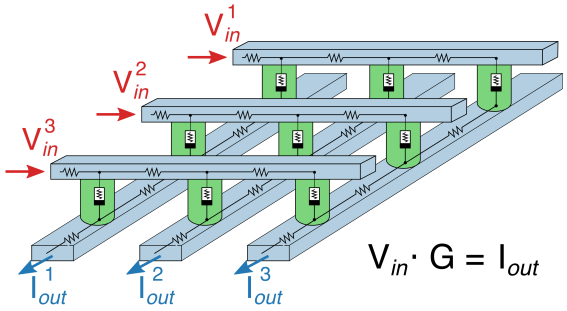


Fig. 1. 3×3 crossbar structure performing a VMM operation in the electrical domain via resistive memory cells highlighted in green. We illustrate the parasitic wire resistances as lumped components.

Despite ongoing research on IMC accelerator chips [4], [5], the absence of standardized fabricated architectures coupled with the plethora of alternative implementation technologies has driven the development of several simulation frameworks. These tools allow design-space exploration of AI models for IMC systems without committing to costly and still-maturing hardware solutions [6]–[8]. As comprehensive tools designed to span multiple design abstraction layers, such simulators often prioritize computational efficiency over high-fidelity

T. Rizzi is with IHP – Leibniz Institut fur innovative Mikroelektronik, 15236, Frankfurt (Oder), Germany, E-mail: rizzi@ihp-microelectronics.com.

A. Lamberti, T. Zanotti, P. Pavan, and F. M. Puglisi are with the Dipartimento di Ingegneria "Enzo Ferrari," Universitá di Modena e Reggio Emilia, 41125, Modena, Italy.

N. Lepri and D. Ielmini are with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano and IU.NET, 20133, Milano, Italy.

C. Zambelli is with the Dipartimento di Ingegneria, Università degli Studi di Ferrara, 44122, Ferrara, Italy.

This research was partially funded by the Brandenburg Ministry of Science, Research, and Cultural Affairs for the project within the StaF project INSEKT, by Deutsche Forschungsgemeinschaft (DFG) under the grant No. WE 3594/30-1, and from the Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU). The work of Tommaso Zanotti and Francesco Maria Puglisi was supported by PNRR MUR Project under the grant ECS_00000033_ECOSISTER.

Manuscript received MONTH DAY, YEAR.

performance estimation. Consequently, they typically abstract away hardware non-idealities and avoid computationally intensive nodal analysis performed by SPICE engines. A clear example of this abstraction strategy is the modeling of the IR drop effect [9], [10], which degrades signal integrity along array interconnects. Specifically, the effective voltage applied to resistive switching devices is reduced by line resistances, thereby altering the resulting output currents and ultimately affecting the computation outcome. Although incorporating these parasitic effects into SPICE netlists is feasible, the resulting simulation overhead is generally incompatible with the requirements of edge AI system.

To address this issue, several parasitic models have been proposed [7], [11]–[15]. Lammie *et al.* incorporated IR drop effects in their simulation framework *MemTorch* [6] using the model proposed in [14]. Their implementation devises an optimized solver to handle the system of equations governing the crossbar array, achieving accuracy comparable to SPICE simulations. In contrast, the *CrossSim* tool [7] accounts for signal degradation due to wire resistance through an iterative approach that progressively refines the voltage drop starting from ideal conditions. While both MemTorch and CrossSim significantly reduce the computational burden compared to SPICE simulations, training large-scale NNs can still be prohibitively time-consuming. In such scenarios, designers actively seek analytical models based on circuit-level simplifications [11]–[13] due to their reduced computational overhead. Among these, the model proposed by Jeong *et al.* [11] is one of the least computationally demanding approaches. Their model estimates parasitic degradation efficiently by applying correction factors to the output currents with minimal overhead. Since a single crossbar-specific parameter, denoted as p, constrains the dependency on the input pattern and weights, different VMM operations exploit a computation-reuse paradigm. This reusability, combined with an appropriate choice of p, enables accurate performance predictions at a fraction of the computational cost required by traditional nodal analysis. However, to the best of our knowledge, this analytical model has only been validated under the assumption of ideal linear I–V characteristics for the resistive switching devices. Additionally, the lack of a structured approach for determining the parameter p could hamper the adoption of the model.

In this article, we address these limitations by introducing a calibration procedure for selecting optimal values of the parameter p, and by proposing an analytical model achieving accurate voltage drop estimation along the metallization lines without compromising computational efficiency. Our work extends the preliminary analysis of [16], where emphasis was on the open-loop programming technique. The main contributions of this work are summarized as follows:

1) Calibration procedure for the parameter p in the model from Jeong. Given a target resistive switching technology and design specification, our procedure evaluates a representative set of weight matrices and input patterns to determine optimal p values. These values can be back-annotated and reused during simulation of arbitrary VMM operations, significantly improving accuracy over the previously adopted empirical values.
2) Evaluation of the model [11] under non-linear device characteristics. We assess the model performance by considering the non-linear I–V behavior for the case of resistive switching devices. Without loss of generality, we adopt non-linear characteristics fitted to RRAM devices fabricated in $130\,nm$ BiCMOS technology at IHP Microelectronics [17].
3) Proposal of an improved analytical model for parasitic voltage degradation. Our model accurately captures the voltage drop along crossbar interconnects, enabling good estimation of the voltages across each resistive cell. The model shows enhanced simulation accuracy when non-linear devices are present. We benchmark the proposed model against [11] and several state-of-the-art parasitic models for crossbar arrays [7], [12]–[14], evaluating both accuracy and computational performance.

The remainder of this article is as follows: Section II describes the resistive crossbar array structure and the existing modeling strategies to capture the parasitic wire resistances. Section III presents the calibration procedure to fit the p parameter to various scenarios, the adopted non-linear device representation, and the improved analytical model. Section IV benchmarks the proposed modeling strategies against well-known parasitic models for crossbar arrays. Section V concludes this study.

II. Backgrounds

A. Resistive Crossbar Arrays

With the growing interest in IMC architectures, the investigation of resistive switching devices is the key matter for cross-point elements in crossbar arrays. In this configuration, the input and output wires are perpendicular, enabling both data storage through the programmable resistance of the devices and analog computation within the electrical domain. Specifically, VMM operations can be executed by encoding matrix coefficients into the conductance values of the crossbar cells G and applying the input vector as voltage levels V. The result is then obtained by sensing the currents at the output lines I as shown in Fig. 1.

Although an ideal crossbar implementation can theoretically perform VMM operations without error, practical hardware realizations are subject to various non-idealities that significantly affect computational accuracy. One key source of degradation arises from parasitic effects along the metallization lines of the crossbar structure, which can distort the signal integrity before reaching the resistive elements.

While parasitic interconnects have both resistive and capacitive components [18], [19], the relatively low operating frequencies of most IMC architectures often allow for neglecting the capacitive effects in favor of the more dominant wire resistance [20]. Lumped linear resistors inserted between adjacent cross-points in the array are the modeling standard for resistive parasitics. However, this detailed representation substantially increases the number of nodes in the circuit netlist, thus enlarging the system of equations to be solved during nodal analysis. As a result, accurate SPICE simulations become computationally prohibitive for large-scale AI applications. The literature explores various simplified models of the parasitic resistance to mitigate this issue.

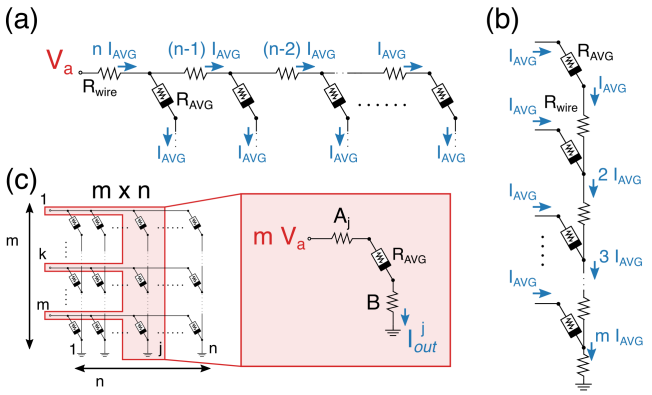


Fig. 2. Current redistribution along (a) input and (b) output metallization lines under assumptions in [11]. (c) Simplified circuit to compute correction factor for the resulting current on the jth output metal line based on [11].

B. Parasitic Resistive Models

As previously discussed, the absence of standardized hardware platforms has led to the development of numerous frameworks for simulating IMC accelerators. These tools implement various parasitic resistance models for crossbar arrays, each balancing the trade-off between simulation accuracy and computational efficiency by employing different simplifying assumptions on circuit topology and solving strategies [7], [11]–[15]. We identify three categories of parasitic models in the literature: i) *Kirchhoff-based models*, ii) *iterative models*, and iii) *analytical models*. The model in [14] is a representative *Kirchhoff-based model*, currently adopted in the MemTorch framework [6]. This approach employs Kirchhoff's laws to construct a system of equations based on the known topology of the crossbar array. The MemTorch framework solves this system using optimized sparse matrix operations. The result is a significant reduction in simulation time compared to traditional SPICE nodal analysis. *Iterative models*, such as the one used in CrossSim [7], estimate signal degradation caused by wire resistance through an iterative refinement process. Starting from ideal voltage conditions, the model updates the voltage drop along the metallization lines in successive steps until the difference between two consecutive iterations falls below an error threshold ϵ. The selected threshold directly influences the number of iterations. To this end, convergence speed and model accuracy expose a trade-off. In contrast, analytical models simplify the crossbar circuit to approximate parasitic effects through correction factors. These factors can be applied either to the output currents, as in Jeong et al. [11], or to the conductance matrix, as in the Diagonal Matrix Regression (DMR) model [12] and in the $\alpha\beta$ model [13]. These models introduce a set of approximations to enable the computation of an analytical solution. Examining [11] more in detail, as it represents the starting point for the proposed calibration procedure and improved analytical model, the following three simplifying assumptions are considered: i) all input voltages are at a constant level V_a, ii) all memory cells have the same resistance value R_{AVG}, and iii) the wire resistance R_{wire} is significantly smaller than R_{AVG}. Under these assumptions, the current flowing through each resistive cell I_{AVG} is approximated as constant, enabling a simplified estimation of parasitic effects. Fig. 2(a) illustrates the redistribution of current from a generic input terminal to the output metallization lines. Similarly, if each node along an output line receives I_{AVG} from its corresponding memory cell, the total output current becomes $m \cdot I_{AVG}$, where m is the number of input lines, as shown in Fig. 2(b). Assuming an $m \times n$ crossbar array, the contributions of wire resistances on the input and output metallization lines can be computed with the coefficients A_j and B using (1), where index j denotes the specific output line under consideration.

$$\begin{aligned} A_j &= \sum_{i=1}^{j}(n+1-i)R_{\text{wire}} \qquad \text{(for } j\text{th output line)} \\ B &= \sum_{i=1}^{m} iR_{\text{wire}} \end{aligned} \tag{1}$$

For a generic jth output line, the equivalent circuit is illustrated in Fig. 2(c). Eq. (2), in this configuration, accounts for parasitic resistance in the output current, where I^j_{ideal} is the output current of the jth line without parasitic effects considered.

$$I^j = I^j_{ideal}\frac{R_{AVG}}{A_j + R_{AVG} + B} \tag{2}$$

It is important to note that the influence of the resistive switching devices is captured solely through the term R_{AVG} in the above equations. Consequently, the computation of the coefficients A_j and B is independent of the specific weight pattern mapped onto the crossbar array. In particular, when the resistive switching devices are between a Low Resistive State (R_{LRS}) and a High Resistive State (R_{HRS}), the original formulation of the model [11] suggests to calculate R_{AVG} as weighted average by applying (3). The crossbar-specific parameter p enables tuning the correction factors while improving model accuracy. However, in [11] the value of p is determined empirically, leaving the user without clear guidance on the optimization procedure. Overcoming such limitations is one of the research objectives of this study.

$$\begin{aligned} R_{AVG} &= \frac{a \cdot R_{LRS} + b \cdot R_{HRS}}{a+b} \\ a = (R_{LRS})^{-p}\ &,\quad b = (R_{HRS})^{-p} \end{aligned} \tag{3}$$

Fig. 3 compares the performance of different modeling strategies in terms of average (a) CPU time and (b) output currents estimation error over 400 VMM operations for a 64×64 crossbar array. A voltage level of $0.5V$ applies to all input terminals. The weight patterns generation is random. We model the resistive switching devices as linear with two discrete states: R_{LRS}=1 kΩ and R_{HRS}=30 kΩ. Typical values of wire resistance between adjacent nodes in such arrays are on the order of a few ohms [15], [18], [21]. Accordingly, and without loss of generality, we assume $R_{wire} = 1\,\Omega$ throughout this study. The [11] model implementation adopts for the parameter p the empirical value 0.17, as reported in the original work. Differently, the CrossSim model [7] fixes the default error threshold ϵ to $200\,\mu V$. The baseline for

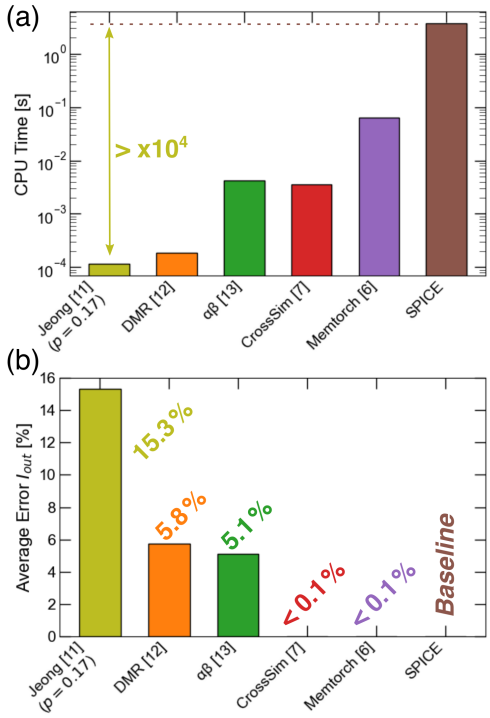


Fig. 3. (a) Average CPU time and (b) output current estimation error for 400 VMM operations on a 64×64 array considering linear cross-points elements.

the error estimation is the nodal analysis performed using Ngspice [22] with wire resistances modeled as lumped resistors. The work in [23] provides Python implementations of the considered parasitic models for the linear resistive devices scenarios. All simulations are performed on a Linux machine with an Intel(R) i7-8665U CPU at 1.90 GHz and 8 GB of RAM. As shown in Fig. 3, there is a clear trade-off between computational cost (CPU time) and estimation accuracy, reflecting the increasing simplifications adopted in each modeling strategy. Complex models such as those in CrossSim and Memtorch closely match SPICE results but offer only modest speedups. In contrast, analytical models provide significantly faster simulations at the expense of higher estimation error in the output current. Notably, the model in [11] is the least computationally demanding among all considered approaches. It enables a reduction of simulation time of over four orders of magnitude compared to traditional nodal analysis, thanks to its lightweight computation of correction factors. However, using the original empirical value of p results in a $3\times$ higher estimation error compared to other analytical models. The results highlight the power of this accelerated modeling method for rapid simulation. Nevertheless, accurately replicating parasitic resistance effects depends on the precise selection of p.

III. Proposed Solutions

A. Calibration Procedure

Relying on empirical tuning of the parameter p without a well-defined selection procedure has hindered the broader adoption of the model in [11]. To address this limitation,

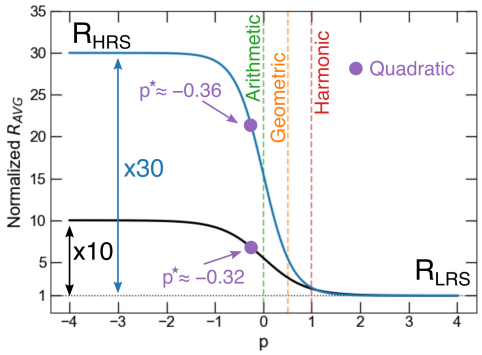


Fig. 4. Normalized resistance R_{AVG} as a function of p, shown for two memory windows ($R_{HRS}/R_{LRS} = 10$, black line, and 30, blue line). We highlight the Pythagorean means.

we define and evaluate a calibration procedure capable of identifying the optimal p for arbitrary VMM operations. This procedure involves testing a set of weight matrices and input patterns, thus enabling a data-driven selection of p.

The parameter p controls the computation of the weighted average resistance R_{AVG}, used to calculate correction factors for the output currents. Using the definitions of a and b from (3), R_{AVG} can be reformulated as (4).

$$R_{AVG} = \frac{R_{LRS}^{1-p} + R_{HRS}^{1-p}}{R_{LRS}^{-p} + R_{HRS}^{-p}} \tag{4}$$

This formulation clarifies how the choice of p influences the calculation of R_{AVG}. As shown in (5), varying p among specific values yields the four classical Pythagorean means: harmonic, geometric, arithmetic, and quadratic. If $p = 1$, R_{AVG} corresponds to the harmonic mean, which places greater weight on smaller resistance values, particularly R_{LRS}. When $p = 0.5$, the expression reduces to the geometric mean, where R_{LRS} has less influence compared to the harmonic case. For $p = 0$, R_{AVG} becomes the arithmetic mean, representing a balanced average between R_{LRS} and R_{HRS}. Finally, depending on the resistance values of the states, there exists a $p^\star \in [-1, 0[$ such that R_{AVG} corresponds to the quadratic mean. For instance, $p^\star = -1$ when $R_{LRS} \simeq R_{HRS}$. Nevertheless, for p smaller than 0, the contribution of higher resistance states is amplified compared to the arithmetic mean.

$$\begin{aligned} p = 1 &\rightarrow R_{AVG} = \frac{2}{R_{LRS}^{-1} + R_{HRS}^{-1}} \\ p = 0.5 &\rightarrow R_{AVG} = \sqrt{R_{LRS} R_{HRS}} \\ p = 0 &\rightarrow R_{AVG} = \frac{R_{LRS} + R_{HRS}}{2} \\ p^\star \in [-1, 0[&\rightarrow R_{AVG} \simeq \sqrt{\frac{R_{LRS}^2 + R_{HRS}^2}{2}} \end{aligned} \tag{5}$$

Importantly, p does not restrict to these discrete values, and thus R_{AVG} varies continuously as a function of p, R_{HRS}, and R_{LRS}. Fig. 4 illustrates the normalized R_{AVG} curves when two different memory windows (R_{HRS}/R_{LRS}) are selected, respectively 10 and 30. Regardless of the chosen resistive states, the weighted average constrains between the lower

Input: Technology and Crossbar Specification
Output: Optimal p values for arbitrary VMM operation

Initialization:
Define discrete values for $V_{in}^{\%}$
Define discrete values for $R_{HRS}^{\%}$
Define discrete values for p
Calibration Loop:
for each $V_{in}^{\%}$ **do**
  for each $R_{HRS}^{\%}$ **do**
    for each of 50 random VMM **do**
      Generate input vector and weight matrix
      Run Nodal Analysis to compute baseline
      for each p value **do**
        Run model in [11]
        Compute average error vs. Nodal Analysis
      end for
      Select p minimizing average error
    end for
  end for
end for
return Look-up table of optimal p values

Fig. 5. Calibration procedure to determine optimal p values.

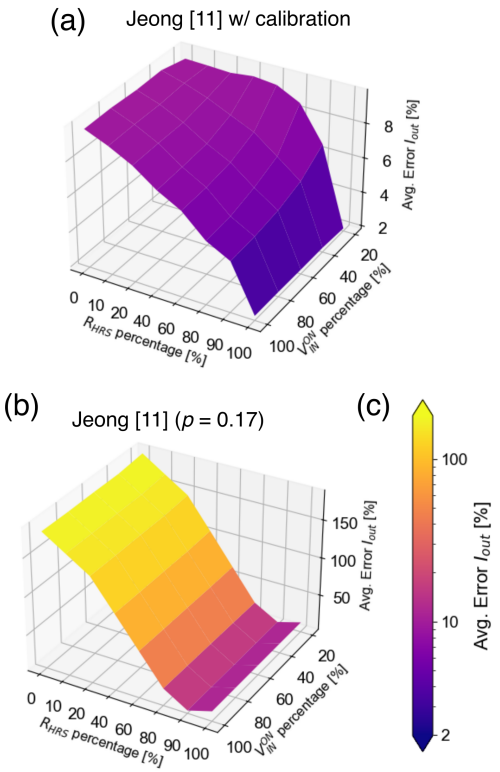


Fig. 6. Average output current estimation error for a 64×64 crossbar array when (a) model in [11] with the proposed calibration procedure and (b) same model with the empirical p value from [11] are used. Each VMM typology is defined by the descriptors $V_{in}^{\%}$ and $R_{HRS}^{\%}$. (c) reports the scale for the average error adopted in both plots.

(R_{LRS}) and upper (R_{HRS}) bounds. Increasing the memory windows leads to a steeper transition between these bounds without significantly altering the p values that make the weighted average converge to the boundaries. In particular, values of p outside the range $[-2, 3]$ have negligible effect on R_{AVG}. Therefore, we bound the search for optimal p values to this interval.

Fig. 5 shows the presented calibration procedure. It requires the specification of technology parameters and crossbar characteristics, including array size, resistive-switching device states, wire resistance, and input voltage level. Since the optimal value of the p parameter depends strongly on the specific weight matrix and input pattern, an exhaustive calibration over all possible combinations is computationally unfeasible. To manage this complexity, we categorize VMM operations using two descriptors: the percentage of active inputs under the assumption of binary voltage levels ($V_{in}^{\%}$) and the percentage of high-resistance cells ($R_{HRS}^{\%}$). By discretizing these descriptors, we reduce the calibration space to a representative subset of configurations. Specifically, $V_{in}^{\%}$ is sampled from the list $[100\%, 80\%, 60\%, 40\%, 20\%]$ and $R_{HRS}^{\%}$ from the list $[100\%, 90\%, 80\%, 60\%, 40\%, 20\%, 10\%, 0\%]$. For each resulting VMM category ($V_{in}^{\%}$, $R_{HRS}^{\%}$), 50 random VMMs are generated, yielding 2000 total operations for calibration.

Each VMM operation is first evaluated via nodal analysis to establish a baseline. While these simulations can take several minutes to complete, they are a one-time effort unless the technology or crossbar parameters are updated. Next, we define a discrete set of p values by sampling with a step size of 0.05 the range $[-2, 3]$. For each p value, we use the model from Jeong et al. [11] to estimate output currents and to compute the error relative to the nodal analysis baseline. Because the analytical model is computationally lightweight, this step introduces minimal overhead. Finally, for each VMM category ($V_{in}^{\%}$, $R_{HRS}^{\%}$), the p value that minimizes the average estimation error is selected as optimal. The fundamental idea behind this calibration procedure is that any parasitic model for a crossbar array can extract $V_{in}^{\%}$ and $R_{HRS}^{\%}$ from the given input vector and weight matrix. Therefore, when the identified p values are back-annotated, that model can efficiently select the optimal p parameter for the current VMM operation. If the percentage of active input and high-resistive cells is not in the pool of considered cases during calibration, we apply interpolation with the two nearest topologies. This approach enables the adoption of the calibrated model for an arbitrary VMM configuration.

To evaluate the effectiveness of the proposed calibration procedure, we compare the model from Jeong et al. [11] using the original empirical value ($p = 0.17$) against the version employing the calibrated p values, in terms of average estimation error on the output current. To ensure statistical independence and avoid artifacts favoring the calibrated model, we generated a new pool of 2000 VMM operations using the same configuration as in the calibration phase, but considering a different random seed for this analysis. Fig. 6 shows the average output current estimation error for a 64×64 crossbar array when the two modeling strategies are adopted. As can be seen, both models achieve better estimations for $R_{HRS}^{\%}$ close to 100 %. We attribute this result to the resistance of

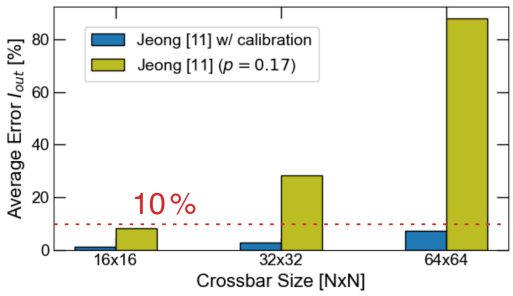


Fig. 7. Average output currents estimation error across 2000 VMMs for different crossbar sizes when using the model in [11] with and without the proposed calibration procedure.

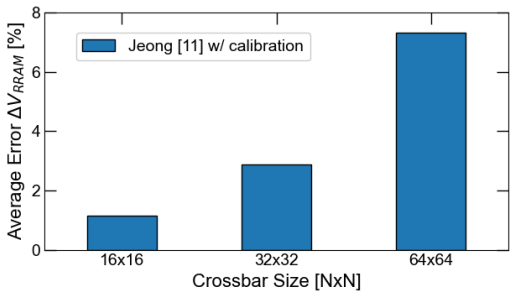


Fig. 8. Average voltage drop estimation errors across 50 VMM operations for various crossbar sizes using the model in [11] with the proposed calibration procedure. We bias input lines at 0.5 V and sample the weight matrices from a uniform distribution.

the memory cells being significantly higher than R_{wire} which makes the parasitic degradation less critical. Nevertheless, regardless of the VMM typology, the calibration procedure reduces the misalignment from the nodal analysis by around one order of magnitude compared to the case in which the empirical value $p = 0.17$ is used. For such 64×64 crossbar, the calibration procedure required approximately 110 minutes CPU time to complete. We evidence that the time dedicated to performing exclusively nodal analysis simulations for baseline results is up to $98\,\%$. The computational cost disparity between classical nodal analysis and the analytical approach is already a clear indicator of the significant advantage in simulation time enabled by the parasitic model after the calibration procedure concludes. Although such a requirement is not negligible, parallel executions of the SPICE simulations enabled by the independent nature of each VMM can reduce the elapsed time for the user. Additionally, this time-intensive step is a one-time procedure, making it acceptable for most design workflows.

Fig. 7 reports the average output current estimation errors across 2000 VMM operations for three crossbar sizes. The observed trends indicate that not calibrating the model degrades the estimation of the output current more when the crossbar size increases. Notably, the model in [11] can maintain an average error below 10% with a calibrated p. This outcome, together with the acceptable computational cost for a one-time procedure, makes the proposed calibration a recommended initial step before applying the analytical model in any IMC simulator. Coherently, initial calibration is applied throughout the remainder of this study to both the original model in [11] and the proposed enhanced analytical model.

B. Non-linear Devices

Noticeably, the coefficients A_j and B obtained from simplifications in [11] are an estimate of the voltage drop $\Delta V_{j,k}$ across the arbitrary cell at the intersection of the jth output line and the kth input line. Considering the circuit in Fig. 2(c) and applying an input voltage V_a, it is possible to approximate the voltage drop as (6).

$$\Delta V_{j,k} = V_a\left(\frac{R_{AVG}}{A_j + R_{AVG} + B}\right) \tag{6}$$

Fig. 8 reports the average voltage drop estimation error for various crossbar sizes. We compute the error as the difference between the value estimated by (6) and the results obtained from SPICE simulations for every cell in the array. The analysis is conducted on 50 VMM operations, with all input lines driven at 0.5 V, and conductance values sampled between LRS and HRS, considering an uniform distribution. As shown, the error increases with the array size, reaching nearly 8% for a 64×64 configuration. Inaccurate voltage drop estimations can significantly distort the estimated output currents by the parasitic model, particularly when non-linear I-V characteristics are involved. To the best of the authors' knowledge, the model in [11] has been validated only for linear resistive elements, leaving such an issue unexplored.

As a preliminary investigation, arbitrary non-linearity is considered for each memory cell following a similar approach to that in [15]. Specifically, the cross-point devices are modelled as voltage-controlled current sources described by (7).

$$I = \frac{V}{R_{cell}}\left(1 + \frac{V}{\lambda}\right) \tag{7}$$

Fig. 9(a) shows the resulting I-V curves for $R_{cell} = 1\,k\Omega$ and different values of the linearity factor λ. Smaller λ values yield stronger non-linear behavior, while $\lambda = \infty$ leads to the linear response. Fig. 9(b) reports the average output current estimation error of [11], relative to SPICE-based nodal analysis, under different device non-linear behaviors. The computation of the average values is performed across 50 VMMs generated similarly to Fig. 8. Consistent with the voltage drop analysis, larger crossbars result in greater estimation errors in the output currents. Furthermore, the accuracy degradation becomes more pronounced as device non-linearity increases, highlighting the critical role of non-linearity when assessing the accuracy of parasitic models.

To ensure realistic assessment, the non-linear I–V characteristics of RRAM cells fabricated by IHP Microelectronics in $130\,nm$ BiCMOS technology [17] are adopted throughout the remainder of this work. The description of the full resistive switching behavior requires the adoption of differential equations [24]–[26]. However, this study focuses exclusively on VMM operations, where input voltages are deliberately limited to avoid unintentional programming of the memory cells. Considering the widely used Stanford-PKU RRAM model [24], under such conditions, the dynamic evolution of the state

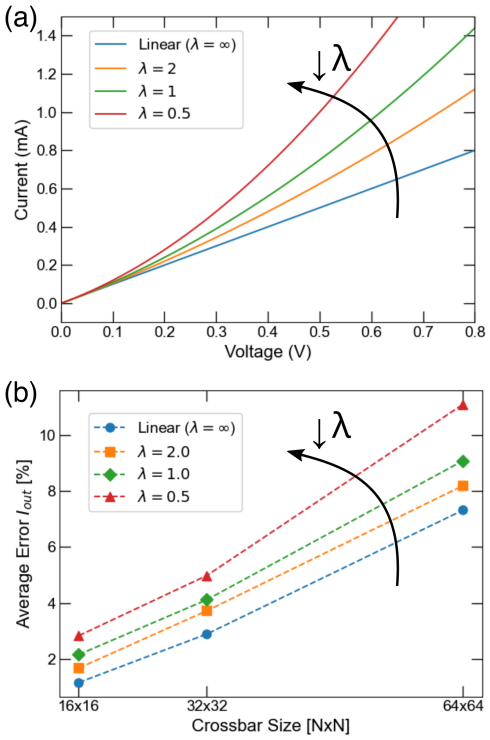


Fig. 9. (a) I-V characteristics for various λ values. (b) Average output currents estimation error by the model in [11] under different array sizes when the resistive memory cells' non-linearity is changed.

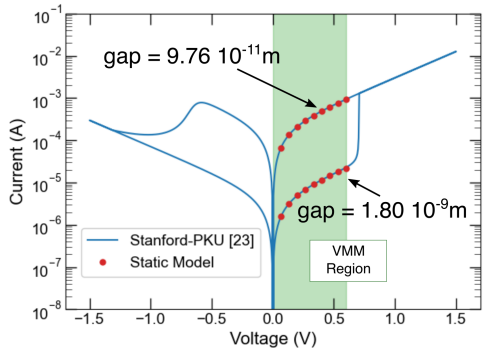


Fig. 10. I-V curve obtained with Stanford-PKU model fitted on IHP technology. The dotted lines represent the simplified modeling description restricted to the VMM operating region.

variable gap is negligible and, therefore, the equations set can be reduced to the static formulation (8).

$$I = I_0 e^{-\frac{gap}{g_0}} sinh(\frac{V}{V_0}) \tag{8}$$

The I_0, g_0, and V_0 are fitting parameters extracted from experimental data. Two states are assumed and distinguished by the corresponding gap values: $gap_{LRS} = 9.76\,10^{-11}\,m$ and $gap_{HRS} = 1.80\,10^{-9}\,m$. With a voltage level of $0.2V$ applied to the RRAM cell, the two resistive states are approximately $1k\Omega$ and $30k\Omega$. As shown in Fig. 10, the static model exactly matches the complete Stanford-PKU one in the VMM operating range ($V_{in} \leq 0.6V$), as expected. This

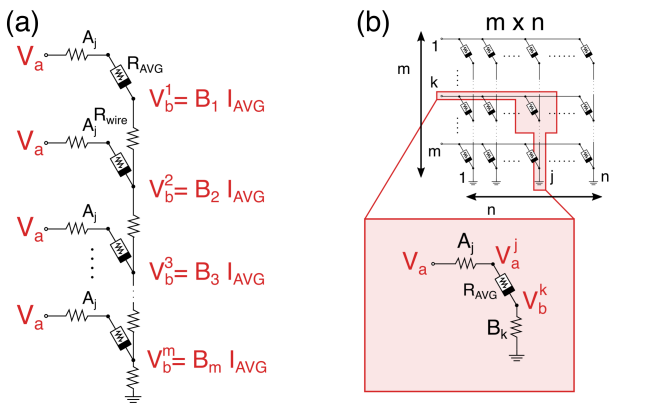


Fig. 11. (a) Proposed nodes level voltage drop modeling for a generic jth output line. (b) Equivalent circuit for an arbitrary (j, k) memory cell including the resistive parasitic elements.

justifies the adoption of the static model for the remainder of this study. Additionally, avoiding differential equations when modeling the resistive cells allows for describing the entire netlist with SPICE native components, without incurring in computationally demanding Verilog-A modules.

C. Parasitic Model

A key limitation of the model in [11] lies in its inability to capture voltage degradation along individual nodes of the output metal lines, as expressed in (6). This shortcoming originates from the fact that such granularity is unnecessary when estimating correction factors for output currents in crossbars using linear elements. However, the presence of non-linear devices necessitates a more accurate evaluation of the voltage applied to each memory cell, implying a detailed estimation of the voltage drop across the output metal lines. To address this, we propose an enhanced version of the model presented in [11], capable of achieving voltage estimation at the node level without introducing additional simplifying assumptions. Fig. 11(a) illustrates the proposed method for calculating node voltages along a generic jth output line. This finer-grained approximation introduces distinct B_k coefficients, where the index k identifies the specific input line. These newly defined coefficients are given by (9).

$$B_k = \sum_{i=k}^{m} i R_{\text{wire}} \tag{9}$$

The equivalent circuit for an arbitrary memory cell located at the intersection of the jth output and kth input lines is shown in Fig. 11(b). Unlike the original model, this circuit no longer represents the entire output line. V_a is the limit for the applied input voltage. Based on this refined representation, the relations reported in (10) hold.

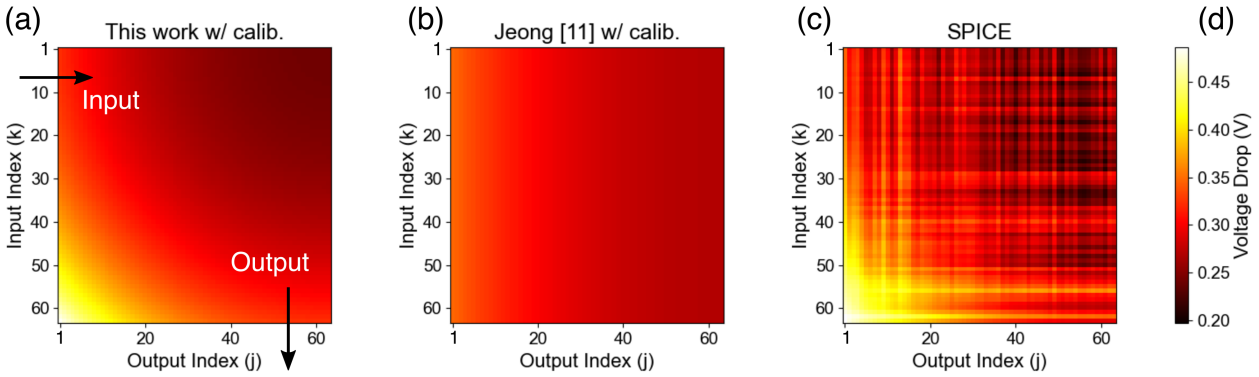


Fig. 12. Estimated voltage drop across each memory cell for a 64×64 VMM operation with (a) proposed model with calibration, (b) model in [11] with calibration, and (c) nodal analysis in SPICE. (d) Reports the scale for the voltage drop adopted in all plots.

$$V_a - V_a^j = \sum_{i=1}^{j}(n+1-i)R_{\text{wire}}I_{AVG} = A_j I_{AVG}$$
$$V_b^k - 0 = \sum_{i=k}^{m} iR_{\text{wire}}I_{AVG} = B_k I_{AVG} \quad (10)$$
$$I_{AVG} = \frac{V_a}{A_j + R_{AVG} + B_k}$$

Consequently, we approximate the voltage drop across the resistive element at the intersection between the jth output line and the kth input line as in (11).

$$\Delta V_{j,k} = V_a^j - V_b^k = V_a\left(\frac{R_{AVG}}{A_j + R_{AVG} + B_k}\right) \quad (11)$$

IV. Comparison with State-of-the-Art

To benchmark the effectiveness of the proposed modeling approach, we compare it against several well-known parasitic models for crossbar arrays. All evaluations consider non-linear resistive devices described by (8), fitted to IHP RRAM technology. In the analytical models, as well as in MemTorch, the non-linear behavior is applied after estimating the voltage drop across each cell, whereas in CrossSim the non-linearity is incorporated directly within the iterative process. After applying the non-linearity, both the model from Jeong et al. [11] and the proposed analytical model are calibrated using the procedure detailed in the previous section.

To visually compare node voltage estimations along the output metal lines, we simulate a 64×64 VMM operation. We bias all input terminals at 0.5 V, and we assume that 80% of the memory cells are in the HRS state when randomly drawing the weight values. Fig. 12 presents heatmaps of the estimated voltage drop across each cell using (a) the proposed model, (b) the model in [11], and (c) SPICE-based nodal analysis. Notably, the [11] model fails to capture voltage variations along the vertical interconnects, estimating instead a constant voltage drop for all positions on the output metal line. The similarities between the results of the proposed model and the reference nodal analysis, especially in contrast to the approach

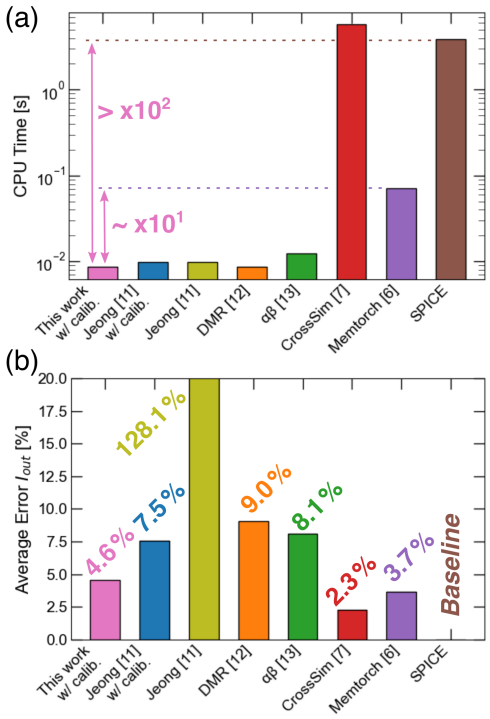


Fig. 13. (a) Average CPU time and (b) output current estimation error for 400 VMM operations on a 64×64 array considering non-linear resistive devices.

in [11], demonstrate the improved accuracy enabled by the refined B_k coefficients.

Since a key requirement of analytical parasitic models is their lightweight computational cost, any improvement in voltage estimation must not come at the expense of significant penalty in runtime efficiency. Fortunately, the proposed model retains the advantage of decoupling the computation of the coefficients from the specific weights and inputs, similar to the approach in [11], thus maintaining low computational complexity even across varying VMM operations. Fig. 13(a) shows the average CPU time required to simulate 400 VMM operations on a 64×64 crossbar array with all input terminals

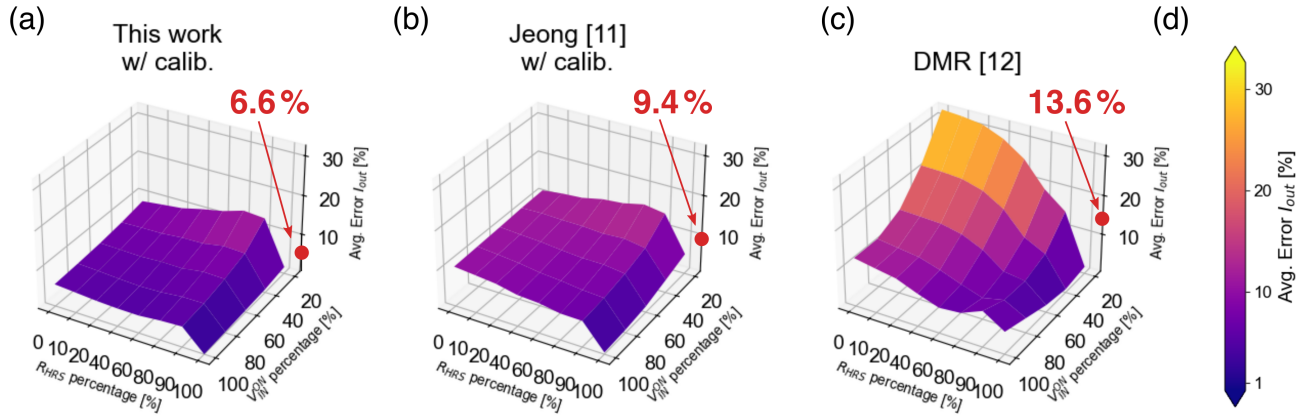


Fig. 14. Average output currents error estimation for random VMMs of different typologies ($V_{in}^{\%}$, $R_{HRS}^{\%}$) with (a) the proposed model with calibration, (b) model in [11] with calibration, and (c) DMR model. The red dot indicates the mean error across all VMM typologies. (d) Reports the scale for the average error adopted in all plots.

set to $0.5V$. The weights patterns are drawn selecting $R_{HRS}^{\%}$ from the list $[100\%, 90\%, 80\%, 60\%, 40\%, 20\%, 10\%, 0\%]$. In particular, we generated 50 weight patterns for each different value of $R_{HRS}^{\%}$. When comparing the achieved results with the linear case from Fig. 3(a), we note that the inclusion of non-linear devices impacts the performance of the analytical models more than the Kirchhoff-based model and the nodal analysis. Nevertheless, the proposed approach significantly speeds up computations, running 100 times faster than SPICE and 10 times faster than MemTorch. A notable difference from the linear case that can be seen in Fig. 3(a) is the degraded performance of the CrossSim parasitic model, which is now slower than nodal analysis. We attribute this result to the implemented iterative algorithm, which is not suited for non-linear components and requires many iterations to converge to solutions within the selected error threshold ($\epsilon = 200\,\mu$V).

Fig. 13(b) reports the average percentage error in the estimated output currents for the same VMM operations. While the Kirchhoff-based model [6] and the iterative model [7] maintain good accuracy even with non-linear resistive devices, analytical models such as DMR [12] and $\alpha\beta$ [13] exhibit higher deviations compared to the linear case, with average percentage errors around 10%. Importantly, the original version of the model in [11] performs the worst, with an average error of 128%. The significant improvement after calibration highlights the importance of selecting optimal values for the p parameter. Meanwhile, the proposed model achieves the best-in-class average accuracy among the analytical approaches, with less than 5% deviation from the nodal analysis. This accuracy, paired with minimal computational cost, offers a compelling balance over more complex models [6], [7].

To further assess the analytical models' accuracy under diverse input patterns, we generate 2000 random VMM operations, varying the descriptors $V_{in}^{\%}$ and $R_{HRS}^{\%}$ similarly to the methodology used for calibration but with a different random seed. Fig. 14 presents the average output current error as a function of these two descriptors. For the sake of clarity, we limited the reported results to the three fastest and most accurate analytical models: the proposed approach, the model in [11], and the DMR model. The DMR model performs reasonably well when all inputs are active, but its accuracy significantly degrades at lower $V_{in}^{\%}$. Unfortunately, this is a common scenario for IMC architectures that employ input bit-slicing design strategies [20], [27]. Conversely, the calibration procedure maintains Jeong et al. model [11] errors below 15% even when a vast portion of the input terminals are not biased, with an average error of 9.4%. Finally, the proposed model consistently achieves the highest accuracy across all VMM configurations with an overall average error of 6.6%. The model maintains strong agreement with SPICE-based nodal analysis regardless of weight matrices and input patterns.

V. Conclusions

This work has deepened the investigation around parasitic modeling for resistive crossbar arrays, overcoming several limitations from the well-known analytical model presented by Jeong et al. in [11]. The proposed calibration procedure significantly enhances the accuracy of the model by addressing its primary drawback, the adoption of an empirically defined parameter p. Furthermore, we investigated the behavior of the model by considering non-linear memory cells, specifically RRAM devices fabricated at IHP Microelectronics. After identifying the coarse-grain voltage estimation along output metal lines as a dominant factor of error, we introduced an enhanced modeling approach that accurately estimates node voltages with high computational efficiency. Achieving more than two orders of magnitude speedup and 6.6% average output currents estimation error compared to SPICE-based nodal analysis for 64×64 crossbars, the proposed model proved comparable precision and more efficient computation compared to state-of-the-art modeling strategies adopted in popular IMC frameworks. The compelling balance between accuracy and efficiency makes it a promising solution for large-scale IMC systems explorations.

REFERENCES

[1] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, 2018.

[2] V. Milo, A. Glukhov, E. Pérez, C. Zambelli, N. Lepri, M. K. Mahadevaiah, E. P.-B. Quesada, P. Olivo, C. Wenger, and D. Ielmini, "Accurate program/verify schemes of resistive switching memory (RRAM) for in-memory neural network circuits," *IEEE Transactions on Electron Devices*, vol. 68, no. 8, pp. 3832–3837, 2021.

[3] X. Sun, W. Khwa, Y. Chen, C. Lee, H. Lee, S. Yu, R. Naous, J. Wu, T. Chen, X. Bao *et al.*, "PCM-based analog compute-in-memory: Impact of device non-idealities on inference accuracy," *IEEE Transactions on Electron Devices*, vol. 68, no. 11, pp. 5585–5591, 2021.

[4] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao *et al.*, "A compute-in-memory chip based on resistive random-access memory," *Nature*, vol. 608, no. 7923, pp. 504–512, 2022.

[5] R. Khaddam-Aljameh, M. Stanisavljevic, J. F. Mas, G. Karunaratne, M. Braendli, F. Liu, A. Singh, S. M. Müller, U. Egger, A. Petropoulos *et al.*, "HERMES Core–A 14nm CMOS and PCM-based In-Memory Compute Core using an array of 300ps/LSB Linearized CCO-based ADCs and local digital processing," in *2021 Symposium on VLSI Circuits*. IEEE, 2021, pp. 1–2.

[6] C. Lammie, W. Xiang, B. Linares-Barranco, and M. R. Azghadi, "MemTorch: An open-source simulation framework for memristive deep learning systems," *Neurocomputing*, vol. 485, pp. 124–133, 2022.

[7] T. P. Xiao, C. H. Bennett, B. Feinberg, M. J. Marinella, and S. Agarwal, "CrossSim Inference Manual v2. 0," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia …, Tech. Rep., 2022.

[8] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+ NeuroSim V2. 0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 11, pp. 2306–2319, 2020.

[9] 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 *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2015, pp. 17–3.

[10] C. Huang, N. Xu, K. Qiu, Y. Zhu, D. Ma, and L. Fang, "Efficient and optimized methods for alleviating the impacts of IR-drop and fault in RRAM based neural computing systems," *IEEE Journal of the Electron Devices Society*, vol. 9, pp. 645–652, 2021.

[11] Y. Jeong, M. A. Zidan, and W. D. Lu, "Parasitic effect analysis in memristor-array-based neuromorphic systems," *IEEE Transactions on Nanotechnology*, vol. 17, no. 1, pp. 184–193, 2017.

[12] Y. Liao, B. Gao, P. Yao, W. Zhang, J. Tang, H. Wu, and H. Qian, "Diagonal matrix regression layer: Training neural networks on resistive crossbars with interconnect resistance effect," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 8, pp. 1662–1671, 2020.

[13] T. Cao, C. Liu, Y. Gao, and W. L. Goh, "Parasitic-aware modelling for neural networks implemented with memristor crossbar array," in *2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*. IEEE, 2021, pp. 122–126.

[14] A. Chen, "A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics," *IEEE Transactions on Electron Devices*, vol. 60, no. 4, pp. 1318–1326, 2013.

[15] N. Lepri, M. Baldo, P. Mannocci, A. Glukhov, V. Milo, and D. Ielmini, "Modeling and compensation of IR drop in crosspoint accelerators of neural networks," *IEEE Transactions on Electron Devices*, vol. 69, no. 3, pp. 1575–1581, 2022.

[16] T. Rizzi, T. Zanotti, N. Lepri, E. Pérez, F. M. Puglisi, D. Ielmini, and C. Zambelli, "Analytical Model for Parasitic Resistances of Crossbar Arrays Suitable for Open-loop Programming Schemes Reliability Analysis," in *2024 IEEE International Integrated Reliability Workshop (IIRW)*, 2024, pp. 1–6.

[17] K. Dorai Swamy Reddy, E. Pérez, A. Baroni, M. K. Mahadevaiah, S. Marschmeyer, M. Fraschke, M. Lisker, C. Wenger, and A. Mai, "Optimization of technology processes for enhanced CMOS-integrated 1T-1R RRAM device performance," *The European Physical Journal B*, vol. 97, no. 11, p. 181, 2024.

[18] M. E. Fouda, A. M. Eltawil, and F. Kurdahi, "Modeling and analysis of passive switching crossbar arrays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 270–282, 2017.

[19] T. Zanotti, F. M. Puglisi, and P. Pavan, "Circuit reliability analysis of rram-based logic-in-memory crossbar architectures including line parasitic effects, variability, and random telegraph noise," in *2020 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2020, pp. 1–5.

[20] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Applied Physics Reviews*, vol. 7, no. 3, 2020.

[21] L. Xia, P. Gu, B. Li, T. Tang, X. Yin, W. Huangfu, S. Yu, Y. Cao, Y. Wang, and H. Yang, "Technological exploration of RRAM crossbar array for matrix-vector multiplication," *Journal of Computer Science and Technology*, vol. 31, no. 1, pp. 3–19, 2016.

[22] Ngspice, "Ngspice, the open source Spice circuit simulator." [Online]. Available: https://ngspice.sourceforge.io/index.html(accessedJuly2025)

[23] A. Lambertini, T. Zanotti, P. Pavan, A. Padovani, and F. M. Puglisi, "Simulation and benchmarking of crossbar parasitic resistance models: Accuracy and performance comparison," *APL Machine Learning*, vol. 3, no. 2, 2025.

[24] Z. Jiang, Y. Wu, S. Yu, L. Yang, K. Song, Z. Karim, and H.-S. P. Wong, "A compact model for metal–oxide resistive random access memory with experiment verification," *IEEE Transactions on Electron Devices*, vol. 63, no. 5, pp. 1884–1892, 2016.

[25] T. Zanotti, P. Pavan, and F. Puglisi, "Comprehensive physics-based rram compact model including the effect of variability and multi-level random telegraph noise," *Microelectronic Engineering*, vol. 266, p. 111886, 2022.

[26] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "Vteam: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.

[27] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.