

# An Efficient Weight Correction Method to Recover Non-ideal Errors in Pruned IRC Designs

Shih-Han Chang, Yi-Min Pan, Chong-En Hong, Chien-Nan Jimmy Liu

*Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu City, Taiwan, R.O.C.*

shchang.ee09@nycu.edu.tw; yiminpan0507@gmail.com; cehong.ee13@nycu.edu.tw; jimmyliu@nycu.edu.tw

**Abstract**—Resistive Random Access Memory (RRAM) has emerged as a leading candidate for In-Memory Computing (IMC) to accelerate Deep Neural Network (DNN) applications. Combined with proper pruning techniques, the cost and energy of DNN computation can be further reduced. However, besides the various non-ideal effects in analog computation, the pruned In-RRAM Computing (IRC) designs also suffer from extra quantization errors, which may significantly degrade the accuracy of DNN. Instead of recovering the errors after circuit implementation, this paper proposes a weight correction method to perform error compensation in advance. By adding proper margins to the stored weight values, the errors caused by IR drop, thermal effects, thermal crosstalk, and weight pruning in IRC designs can be considered simultaneously without additional circuit overhead. As shown in the experimental results, the proposed method effectively recovers the computing accuracy under various pruning rates, thus enhancing the performance and reliability of IRC designs.

**Index Terms**—RRAM thermal effect, IR-drop, Weight correction, In-RRAM Computing

## I. INTRODUCTION

Resistive Random Access Memory (RRAM) crossbar arrays have emerged as a promising candidate for In-Memory Computing (IMC) due to its non-volatility, high density, fast read/write speeds, and high energy efficiency. While the advantages of RRAM are significant, it also presents several design challenges, such as greater device variations, IR-drop, and thermal effects. These non-ideal effects may affect the resistance values, leading to incorrect storage of weights and consequently degrading the computation accuracy in In-RRAM Computing (IRC) designs.

To reduce power consumption and hardware cost, the weight pruning technique is commonly employed to filter the important features of input data for IRC design while preserving the similar model accuracy [1], [2]. However, besides the various non-ideal effects in analog RRAM crossbar arrays, the pruned IRC designs also suffer from extra quantization errors. As the pruning technique increases network sparsity, the remaining critical weights become more sensitive to these non-idealities, leading to severe degradation in the computational accuracy [3]. Therefore, how to compensate for possible errors during the design stage has become an important issue for IRC designs to keep good inference accuracy in real environment.

Among these non-idealities, IR drop caused by parasitic wire resistance is one of the dominant factors. Network re-training with extra error injection is often adopted to alleviate the accuracy degradation of IRC designs in the real environment. [4], [5] proposed a fast training framework that incorporates IR-drop

effects to improve RRAM cell endurance by reducing peak temperature. However, this approach introduces non-negligible overhead due to additional training time. Alternatively, [6], [7] design calibration peripherals to effectively mitigate deviations in storage values and computation results. However, these additional circuits incur substantial area overhead, leading to reduced memory density and diminishing the energy efficiency.

Thermal effect is another issue that will result in the degradation of IRC accuracy due to its conductance variation in RRAM cells. One popular approach is to adjust the data mapping of the weight matrix. [8] proposed a subarray-based weight remapping to avoid mapping important weights to high-temperature locations, thus reducing the impact of thermal effects. However, this method becomes less effective if the temperature is high in the entire crossbar array. Another method [9] introduced a current mirror circuit at the crossbar outputs to compensate for errors. While this approach avoids the need for additional training, it incurs extra circuit overhead. To further alleviate thermal crosstalk effect, [10] introduced a weight decomposition strategy that modifies the weight distribution between positive and negative crossbars, resulting in fewer low-resistance state (LRS) cells. Although this approach helps reduce the overall temperature in RRAM arrays, it still suffers from noticeable accuracy degradation.

## II. PROPOSED WEIGHT CORRECTION APPROACH

In this work, we propose a weight correction method that simultaneously addresses IR drop, thermal effects, and thermal crosstalk in IRC designs without retraining or additional circuit overhead. Instead of recovering errors after circuit implementation, the proposed method performs early-stage compensation for possible errors in analog RRAM crossbar arrays by changing the stored weights. This approach adds proper margins to the weight values so that the degraded values are very close to the desired values. It can mitigate the accuracy loss caused by the non-idealities of analog RRAM crossbar arrays through a time-efficient approach.

Fig. 1 shows the overall flow of the proposed weight correction method. To enhance the weight compression ratio under the target sparsity requirement, we apply a column-vector pruning technique [11] with the granularity of OU-sized column vectors. Then, based on the drop rate for each RRAM cell, which is determined by the given weight map and the ambient temperature map of the crossbar array, the proposed weight correction method adjusts the weights to approximate their original values under non-ideal effects. Finally, to further

improve accuracy, the corrected weight map is used to update the drop rates and refine the weight correction.

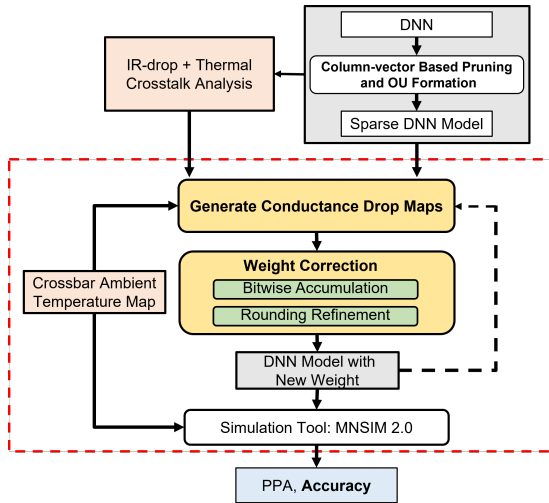


Fig. 1: Overall weight correction algorithm flow.

### A. Conductance Drop Map

To accurately model the impact of non-idealities in RRAM crossbar arrays, we construct a conductance drop map to support the calculation of weight correction. As illustrated in Fig. 2, the conductance drop consists of two components: the IR-drop rate and the thermal drop rate. The IR-drop rate is defined as the ratio between the sum of the ideal current and the current deviation to the ideal current [12]. For the thermal drop rate, we first generate a thermal crosstalk map by analyzing the spatial conductance distribution, which captures the temperature rise induced by LRS cells, based on the given steady-state thermal distribution [13]. This map is then combined with the ambient temperature distribution of the crossbar, which reflects external environmental heating, to estimate the effective temperature at each RRAM cell. Considering the two effects together, the total conductance drop for each RRAM cell is computed as the product of the thermal drop rate and the IR-drop rate.

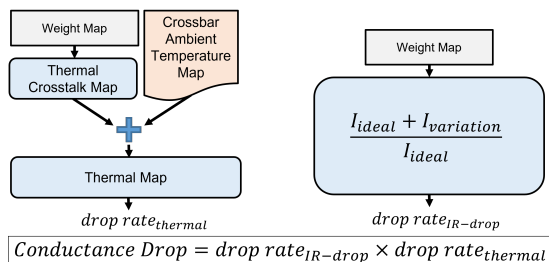


Fig. 2: Construction of conductance drop map.

### B. Weight Correction Algorithm

Fig. 3 is an example of an 8-bit weight with an original value of 96. Due to the non-ideal effects, the effective conductance after mapping is dropped, resulting in a significantly reduced actual weight value of 83. To compensate for this mismatch, we

apply a two-stage correction procedure. First, we perform bitwise accumulation by initializing an all-zero vector and scanning bits from the most significant bit to the least significant bit. If adding its effective value results in a sum smaller than the target value, the corresponding bit is set to 1. Second, we apply a rounding refinement by adding 1 to the final vector if its actual value is closer to the original weight. This approach identifies a new weight vector (e.g., 110), which closely approximates the original weight of 96 after being subjected to the same non-ideal effects. Because the distribution of neural network weights is often centered around the mean value in the allowable range, most of the weights can be effectively corrected without value range constraints.

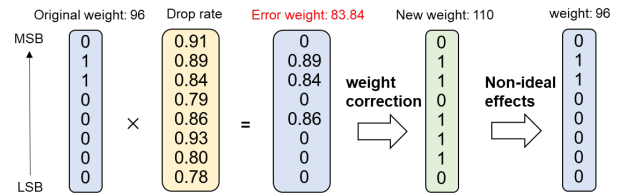


Fig. 3: Example of weight correction.

## III. EXPERIMENT RESULTS

Fig. 4 presents the accuracy comparison of different methods under various pruning rates and linearly scaled temperature distributions (0,  $\pm 10K$ ,  $\pm 20K$ , respectively) adapted from [8]. In the traditional approach, error sources are not considered, and the inference accuracy drops sharply in realistic noisy environments, as shown by the gray bar labeled “baseline” in Fig. 4. The proposed weight correction method consistently outperforms all other methods across different settings. Specifically, under the most severe thermal condition and the highest pruning rate, our method still achieves 82.95% accuracy, whereas TARA [8] and TOPAR [10] drop drastically to 14.28% and 14.16%, respectively. This shows that the proposed method successfully recovers the error caused by various non-ideal effects, thus helping designers make proper adjustments at an early stage.

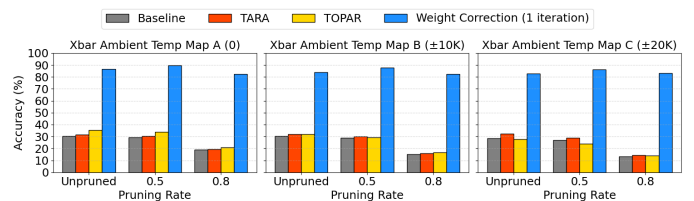


Fig. 4: Accuracy comparison under different temperature maps and pruning rates (evaluated on CIFAR-10 with VGG-8).

## REFERENCES

- [1] P. Wang et al., “Snrram: An efficient sparse neural network computation architecture based on resistive random-access memory,” in Proc. ACM/IEEE Design Autom. Conf., 2018.
- [2] J. Lin et al., “Learning the sparsity for reram: Mapping and pruning sparse neural network for reram based accelerator,” in Asia and South Pacific Design Autom. Conf., 2019.

- [3] A. Bhattacharjee et al., "Examining and Mitigating the Impact of Crossbar Non-idealities for Accurate Implementation of Sparse Deep Neural Networks," *Design, Autom. & Test in Europe Conf. & Exhib.*, 2022.
- [4] M. E. Fouda et al., "IR-QNN Framework: An IR Drop-Aware Offline Training of Quantized Crossbar Arrays," in *IEEE Access*, vol. 8, pp. 228392-228408, 2020.
- [5] W. Li et al., "RRAMedy: Protecting ReRAM-Based Neural Network from Permanent and Soft Faults During Its Lifetime," *IEEE Intl. Conf. on Computer Design*, 2019.
- [6] M. Lee et al., "Victor: A variation-resilient approach using cell-clustered charge-domain computing for high-density high-throughput MLC CiM," in *Proc. ACM/IEEE Design Autom. Conf.*, 2023.
- [7] O. Yousuf et al., "Robust Hardware-Aware Neural Networks for FeFET-Based Accelerators," in *IEEE Trans. on Nanotechnology*, vol. 24, pp. 189-200, 2025.
- [8] M. V. Beigi et al., "Thermal-aware Optimizations of ReRAM-based Neuromorphic Computing Systems," *55th ACM/IEEE Design Autom. Conf.*, 2018.
- [9] Y. Ling et al., "Temperature-Dependent Accuracy Analysis and Resistance Temperature Correction in RRAM-Based In-Memory Computing," in *IEEE Trans. on Electron Devices*, vol. 71, no. 1, pp. 294-300, Jan. 2024.
- [10] H. Shin et al., "A Thermal-aware Optimization Framework for ReRAM-based Deep Neural Network Acceleration," *2020 IEEE/ACM Intl. Conf. On Computer Aided Design*, 2020.
- [11] S. Yang et al. "AUTO-PRUNE: Automated DNN pruning and mapping for ReRAM-based accelerator." *Proceedings of the 35th ACM Intl. Confer. on Supercomputing*. 2021.
- [12] S. -H. Chang et al., "Mitigating Computation Errors of In-RRAM Computing Through Network Re-Training With IR-Drop and Data Allocation Effects," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2025.
- [13] S. Pande et al., "Thermal Crosstalk Analysis in ReRAM Passive Crossbar Arrays," *2024 37th Intl. Conf. on VLSI Design and 23rd Intl. Conf. on Embedded Systems*, 2024.
- [14] Z. Zhu et al., "MNSIM 2.0: A Behavior-Level Modeling Tool for Processing-In-Memory Architectures," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 11, pp. 4112-4125, 2023.