Cycle-to-Cycle Variation Suppression in ReRAM-Based AI Accelerators

Jingyan Fu*
Electrical and Computer Engineering
North Dakota State University
Fargo, ND, USA
jingyan.fu@ndsu.edu

Zhiheng Liao*
Electrical and Computer Engineering
North Dakota State University
Fargo, ND, USA
zhiheng.liao@ndsu.edu

Jinhui Wang
Electrical and Computer Engineering
University of South Alabama
Mobile, AL, USA
jwang@southalabama.edu

2023 IEEE Physical Assurance and Inspection of Electronics (PAINE) | 979-8-3503-3931-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/PAINE58317.2023.10317995

***Abstract*—As a non-volatile memory, currently ReRAM (Resistive Random Access Memory) is emerging for the low power and high performance AI accelerator design. However, ReRAM always suffer from significant cycle-to-cycle variations, which significantly degrades the inference accuracy. In this study, we firstly fabricate ReRAM wafers and test them. Then we propose both level optimization and pulse regulation methods to mitigate the adverse impact of cycle-to-cycle variations of ReRAM, improve the inference accuracy, lower the energy consumption, and decrease the latency of the AI accelerators.**

***Index Terms*—cycle-to-cycle variation, ReRAM (Resistive Random Access Memory), artificial intelligence, level, pulse, accuracy, energy, latency**

I. Introduction

CMOS (complementary metal-oxide-semiconductor transistor) technology based computing systems plateaus the process scaling [1]–[7], which cannot provide enough computational support for accelerators in AI applications, such as DNN (Deep Neural Networks) [8], [9]. ReRAM (Resistive Random Access Memory) is theoretically proposed in 1971 [10] and later are successfully physically fabricated in 2008 [11]. The ReRAM-based AI (artificial intelligence) accelerators show great potential to enable machine-learning applications in many critical areas [12]–[14], and characterized with high speed and endurance, low power and complexity, and great CMOS-compatibility. Despite extensive research being directed towards ReRAM, a large-scale commercialization of ReRAM-based AI accelerators have not yet been achieved. This is due to the unreliability of ReRAM [15], [16], which is caused by non-ideal device properties such as cycle-to-cycle variations. The physical mechanism of the conductance modulation in ReRAM is typically an ionic reconfiguration process based on electro/thermo-dynamics. Such atomic-level random process would result in unavoidable large variations in ReRAM. Researchers have many previous work to suppress such variations in ReRAM-based AI hardware design. Algorithm Level: In [2], algorithms of the mutual decision between conductance of ReRAM and Boolean functions are used to tolerate a maximum variation. In [9], a new algorithm is proposed to map arbitrary matrix values appropriately to ReRAM conductance to reduce computational errors. However, because variations usually come from ReRAM devices - hardware of AI accelerators, the algorithm level optimizations are usually resources-consuming. Circuit Level: In [17], the smart programming scheme and dummy column technologies are proposed to eliminate the off-state current and improve immunity to cycle-to-cycle variations. The experimental result shows the accuracy is improved to 95% from 70%. However, circuit level technologies need large additional peripheral circuits and increase silicon areas. Device Level: In [18], multiple ReRAM cells laid out in parallel are applied to improve the variation tolerance. But, it unavoidably induces area overhead of hardware. Therefore, optimization for the cycle-to-cycle variation in ReRAM-based AI accelerators is urgent. In this study, the ReRAM with the TiO_2/TiO_{2-x} architecture is fabricated. Then, we propose level optimization and pulse regulation methods to maximumly avoid the impact of cycle-to-cycle variations, improve the inference accuracy, lower the energy consumption, and decrease the latency of the AI accelerators, most important, without silicon area penalty.

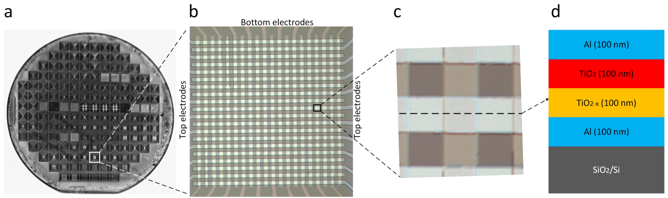

Fig. 1: a: A ReRAM wafer. b: A ReRAM chip. c: A ReRAM device. d: Cross-sectional schematic of a ReRAM with TiO_2/TiO_{2-x} structure.

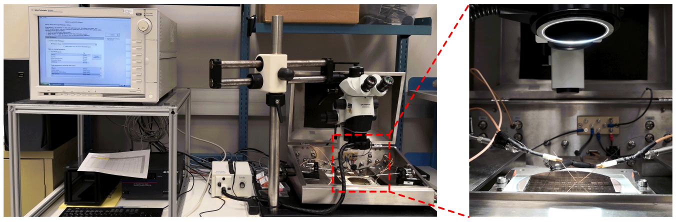

Fig. 2: Testing platform.

Both authors contributed equally to this research. This work was supported in part by the National Science Foundation under grants 2247343, 2218046, and 1953544.

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.

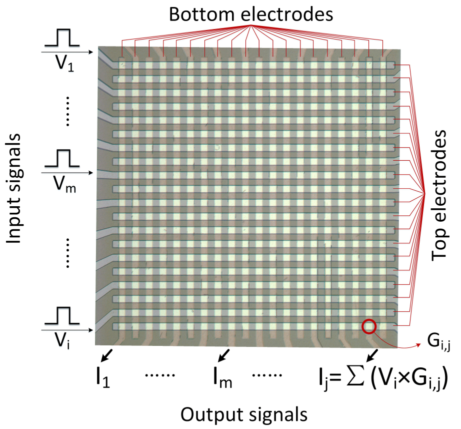

Fig. 3: Hardware implementation of the vector-matrix-multiplication using ReRAM. V_i, G_{ij}, and I_j represent the input signal in ith row, the conductance of the ReRAM in j_{th} column and i_{th} row, and the output current that represent the dot product result of V and G, respectively.

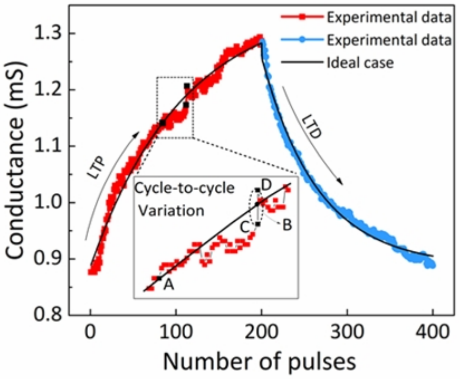

Fig. 4: Cycle-to-Cycle variation of ReRAM. Cycle-to-cycle variation is the deviation between target conductance and realistic updated conductance by pulses. LTP: long-term potentiation, LTD: long-term depression.

II. ReRAM and Related Background

A. ReRAM Device

ReRAM wafers based on in-house technology are fabricated and tested. The detailed image and geometry of ReRAM with TiO_2/TiO_{2-x} structure in this paper is schematically shown in Fig. 1. One ReRAM chip includes of 20 x 20 cells. Physically, a ReRAM cell is a 40 μm x 40 μm two-terminal device connecting two aluminous electrodes and sandwiching TiO_2/TiO_{2-x} layers to achieve stably tunable behavior. I-V characteristics from positive and negative voltage sweeping are carried out using a Keysight B1500a semi-conductor parameter analyzer in a voltage-sweep and volt-age-pulse mode. The wafer is set on the Micro-manipulator probe station and the pads are contacted by probe tips as shown in Fig. 2.

ReRAM arrays carry out the vector-matrix multiplication as shown in Fig. 3. Every row of the array gets input voltage pulses that are the vector. Each conductance of the ReRAM in every cross point composes the matrix. Every column of the array transmits an output current that is the sum of multiplication by the input signal and conductance in each cross point. To update the conductance of a ReRAM that has multilevel conductance from the minimum to the maximum, a positive pulse signal is applied to increase the conductance, which is called long-term potentiation (LTP) [19]. Conversely, long-term depression (LTD) is the process of decreasing the conductance by supplying a negative pulse signal until the conductance gets to the minimum [20]. Multilevel ReRAM effectively utilize such multi-value conductance to learn the features of data and realize an edge AI system [21], [22].

B. Cycle-to-Cycle Variation

A ReRAM cell can change its conductance from minimum to maximum when pulse voltage is larger than a threshold voltage [23]. At the same time, cycle-to-cycle variations generate different final conductance when the same number of pulses is applied in different updating cycles in a ReRAM, even when the ReRAM has the same beginning conductance, as indicated in Fig. 4. For example, if some given number of pulses are applied, a ReRAM starts at conductance A and aims to B, may reach between C and D due to variations, as shown in Fig. 4. ReRAM exhibit cycle-to-cycle variations because of the shape of the conductive filament, the oxygen vacancy distribution at and around the filament, and the changing location of the active filament between one cycle to the next. These three mechanisms originate from the coexistence of multiple sub-filaments and the active, current-carrying filament may change from cycle to cycle [24]. Thus, the cycle-to-cycle variation is a type of inherent randomness associated with the randomness in internal atomic configurations [25]. One of the major obstacles for the implementation of redox-based multilevel memristive memory or logic technology is the large cycle-to-cycle variation.

C. Level Optimization (LO)

Following the working flow of Fig. 5, the number of levels of conductance is set for a ReRAM. It means, between the maximum and the minimum conductance, ReRAM can change a certain number of levels. Usually, higher number of levels would achieve higher inference accuracy. Such a number of levels will map to the width of the pulse that is generated from the pulse generator in hardware implementation. The higher number of the levels corresponding to the narrower pulses and simultaneously, the system can achieve higher accuracy. But, a higher number of the levels also would bring larger cycle-to-cycle variations. This is because the pulse generator produces more pulses to tune the conductance when the algorithm calculates the same Δweight than that system has a lower number of the levels. Therefore, finding the optimized number of levels is necessary. In this work, we the number of the levels is a parameter and set from 10 to 200 and the step is 10.

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.

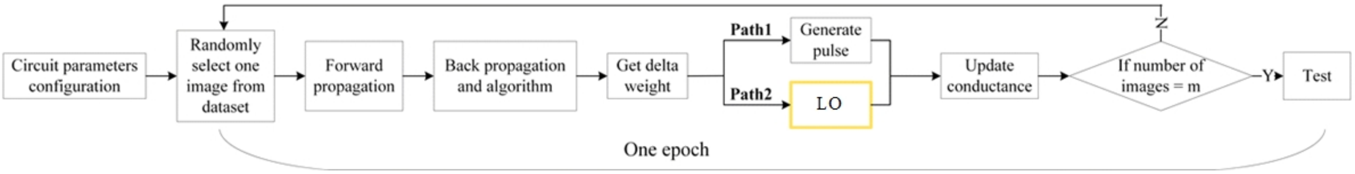

Fig. 5: Working flow of level optimization

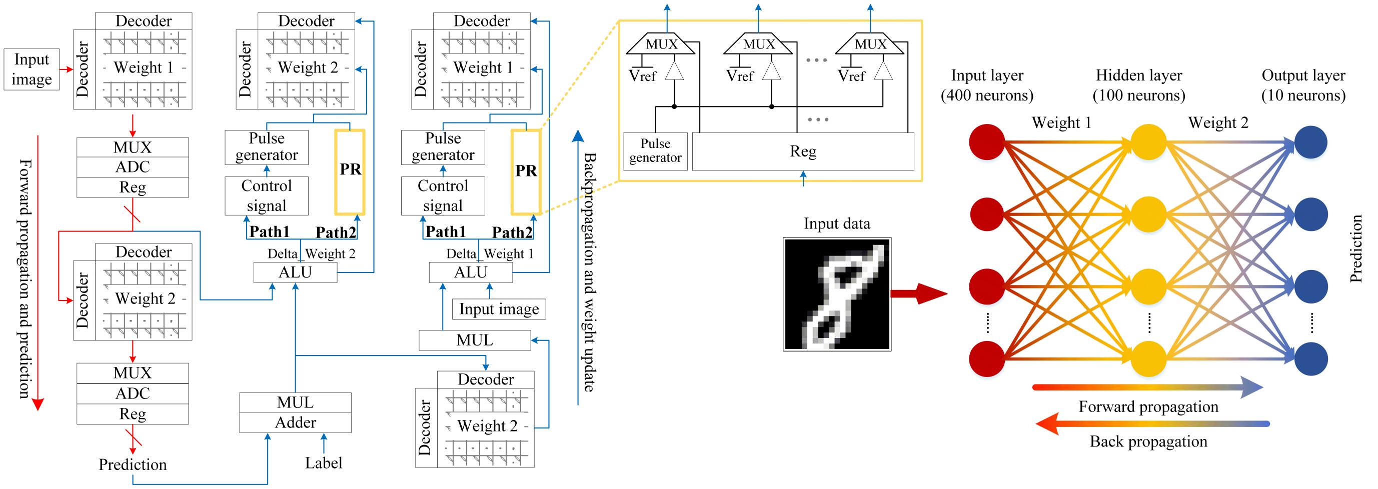

Fig. 6: Architecture of ReRAM-based AI accelerator

D. Pulse Regulation (RP)

A pulse regulation (RP) method is also proposed in our study. In PR method, only one pulse is applied to update conductance each time and keeps the original width of writing pulses. At circuit level, MUXs are added to generate one pulse whenever a conductance of a ReRAM cell needs to be tuned according to the algorithm, as shown in Fig. 6. The decoder is also used to pick a signal from an ALU for each row update. At the same time, registers store the values of Δweight that are calculated by an ALU. Then these values are transmitted to MUXs as control signals. MUXs select one writing pulse that comes from a pulse generator as output when control signals are enabled. The enabled signal means that the corresponding ReRAM cell needs to be updated and that corresponding Δweight value is greater than or equals weight change by one pulse. In this case, the PR method directly optimize each weight update, reduce the number of pulses, and mitigate the impact of cycle-to-cycle variations as expected.

III. Experiments Results

A. Experiment Environment

To verify the proposed level optimization and pulse regulation (PR) methods, the multi-layer perceptron platform (MLP platform) is utilized to emulate the learning classification scenario with Modified National Institute of Standards and Technology handwritten dataset. We adopt NeuroSim+ [26] with the fully connected networks structure and the parameters of devices are tested results of the maunfactured ReRAM mentioned in Section II. This platform contains a three-layer neural networks with 5 algorithms: SGD, Momentum, AdaGrad, RMSProp, and Adam. For the level scaling method, each evaluation trains 125 epochs. For the PR method, each evaluation trains 200 epochs. Note that, the networks will continually learn the feature of an input data after the last epoch since this platform is online learning networks. The conductance of a ReRAM with multilevel as shown in Fig. 4, is increased by supplying a positive pulse until the conductance gets to the maximum. This increasing process is long-term potentiation (LTP). Conversely, long-term depression (LTD) is the process of decreasing the conductance by supplying a negative pulse until the conductance gets to the minimum.

B. Experiments with Level Optimization (LO)

In order to investigate the effectiveness of LO method, the LTP and LTD with the different number of the levels are verified. The inference accuries with/without cycle-to-cycle variations, are shown in Fig. 7. It indicates the number of the levels from 10 to 200 and step is 10 for five different algorithms. When the experiment does not consider cycle-to-cycle variation, the accuracy increase to the bright area from the dark area with the increasing number of the level. Ignoring cycle-to-cycle variations, the highest accuracy locates at the number of the levels = 200 for both LTP and LTD in five algorithms. Considering cycle-to-cycle variations, It concludes that increasing the number of the levels does increase the inference accuracy. In bright areas of the figures, the inference accuracies are around 90% in the lower left corner and higher than 93% in the upper right corner. The optimized accuracy

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.

for five algorithms are respectively: number of the levels (LTP/LTD) for SGD: 50/40, Momentum: 60/50, AdaGrad: 60/50, RMSProp: 50/50, and Adam: 50/40. Therefore, LO method optimizes the number of the levels, so the system achieves higher inference accuracy by mitigating cycle-to-cycle variations.

C. Experiments with Pulse Regulation (PR)

The PR method improves all accuries for five algorithms, as shown in Fig. 8. Note that, for evaluating the effect of the PR method, 100 epochs with 500 images for each epoch are set. The only negative impact of the PR method is to slow down the learning speed, which, however, only exits at the several beginning learning epochs and is reflected by the red curves below the blue curves in Fig. 8. However, all inference accuracies of five algorithms have significant improvement with the PR method after 100 epochs. In addition, the PR method effectively produces a smoother convergence of the training process, which reduces the excessive fluctuation of the inference accuracy.

Furthermore, because the updating pulses truncate to one in each conductance update, the number of updating pulses and energy have been significantly decreased in 100 epochs. For example, in Momentum and RMSProp algorithms, energy consumption are saved up to 12.888% and 16.104% and latency are decreased up to 26.062% and 27.854% as shown in Table I. This is because every iteration has the designated reading latency since the process of a vector-matrix multiplication is executed using a parallel reading strategy. However, the system updates its weight row by row, which indicates a parallel writing strategy cannot be implemented for all rows at the same time, otherwise, the system will have unacceptable area overhead [27]. Each row's writing latency is determined by the maximum number of writing pulses as a critical path. Thereby, the main latency for ReRAM arrays is writing latency that strongly depends on the maximum update pulses of each row. With the PR method, the maximum number of the writing pulses decreases to one, which reduces the total latency of the system. Without the PR method, each row needs registers and counter to record and control the updating process since the time of updating process in the different training iterations is probably different [26], [28]–[30]. With the PR method, those two components (registers and counter) are not needed because the selected row only uses one pulse to update the conductance of ReRAM-based AI accelerators. Instead, one multiplexer is added to the system, as shown in Fig. 6. Therefore, the PR method optimizes the inference accuracy, improves energy efficiency, and reduces system latency and area.

IV. Discussions

As shown in Fig. 9, the number of the levels with the highest accuracy decreases through the increasing of the variations. In fact, the reason for this correspondence is that an excessively large number of levels will cause the value of the variation exceeds the conductance value of a single level. Simultaneously, a too small level number will cause that the weight value loses too much precision and reduces the final inference accuracy.

According to the mechanism of the cycle-to-cycle variation, the PR method efficiently reduces the cycle-to-cycle variation by compressing the number of up-date pulse to one. For every updating, the cycle-to-cycle variation is limited with one pulse's impact, which minimizes the cycle-to-cycle variation. Note that, the inference accuracies have significant improvement with the PR method. The reasons are two aspects. 1) The PR method minimizes the cycle-to-cycle variation. 2) Each update step uses at most one pulse to tune conductance. One pulse to tune conductance means smaller steps is achieved in the direction of convergence, while a big step will make the learning jump over minimum point of weight. What's more, energy consumption and system latency are correspondingly reduced when the PR method is adopted in the system by compressing the number of update pulse to one.

V. Conclusion

We propose the level scaling and the PR methods that are simple, feasible, and universal methods to effectively mitigate the impact of cycle-to-cycle variations in ReRAM-based AI accelerators. We prove that because of cycle-to-cycle variations, the inference accuracy in the maximum number of the levels is not optimal for the real device. As for different materials based multilevel ReRAM, the level scaling method can be used to optimize accelerators through selecting appropriately the number of the levels. Similarly, the PR method mitigates the impact of cycle-to-cycle variation by compressing the number of updating pulses to one as well as improves energy efficiency up to 16.104% and reduce system latency up to 27.854%. Besides accuracy improvement, the level scaling and the PR methods can also lower the energy consumption and decrease the latency of the ReRAM-based AI accelerators.

Acknowledgment

This work was supported in part by the National Science Foundation under Grants 2247343, 2218046, and 1953544. The author acknowledges financial support from the Alabama Graduate Research Scholars Program (GRSP) funded through the Alabama Commission for Higher Education and administered by the Alabama EPSCoR.

References

[1] M. M. Waldrop, "The chips are down for moore's law," *Nature News*, vol. 530, no. 7589, p. 144, 2016.

[2] J. Rajendran, H. Maenm, R. Karri, and G. S. Rose, "An approach to tolerate process related variations in memristor-based applications," in *2011 24th Internatioal Conference on VLSI Design*, 2011, pp. 18–23.

[3] G. Yuan, P. Behnam, Y. Cai, A. Shafiee, J. Fu, Z. Liao, Z. Li, X. Ma, J. Deng, J. Wang, M. Bojnordi, Y. Wang, and C. Ding, "Tinyadc: Peripheral circuit-aware weight pruning framework for mixed-signal dnn accelerators," in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2021, pp. 926–931.

[4] J. Edstrom, Y. Gong, A. A. Haidous, B. Humphrey, M. E. Mccourt, Y. Xu, J. Wang, and N. Gong, "Content-adaptive memory for viewer-aware energy-quality scalable mobile video systems," *IEEE Access*, vol. 7, pp. 47 479–47 493, 2019.

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.

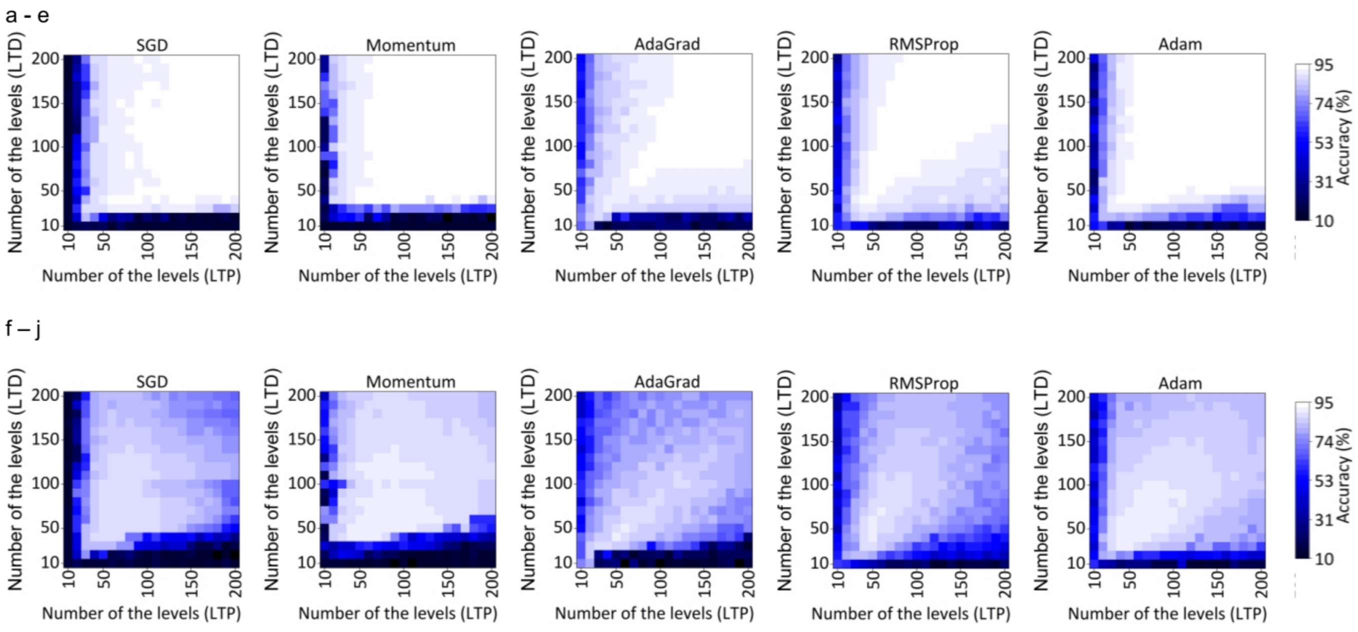

Fig. 7: a-e Distributions of inference accuracy without cycle-to-cycle variation (α=0) with different LTP and LTD number of the levels (from 10 to 200, step is 10) in 5 algorithms. f-l Distributions of inference accuracy with different LTP and LTD number of the level values (from 10 to 200, step is 10) in 5 algorithms under measured cycle-to-cycle variation.

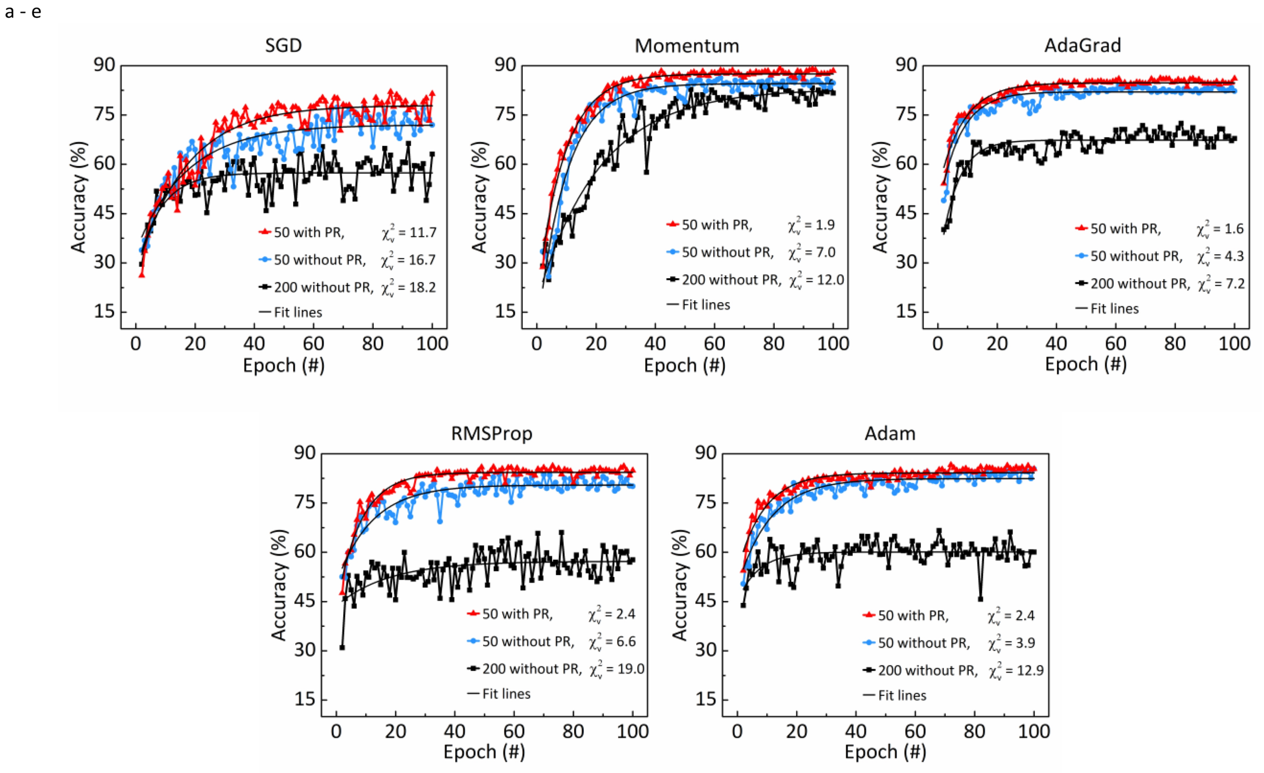

Fig. 8: Experimental results with/without pulse Regulation regarding inference accuracy. 50 and 200 are the number of the levels in different evaluations. χ_ν^2 is the Reduced Chi-Sqr values with analyzing data. Each curve includes 100 epochs and each epoch includes 500 images.

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Experimental Results with/without Pulse Regulation

Algorithm	Energy with PR	Energy without PR	Energy Saved (%)	Latency with PR	Latency without PR	Latency Reduced (%)
SGD	0.313	0.361	13.310	15.059	17.728	15.057
Momentum	0.501	0.575	12.888	23.912	32.340	26.062
AdaGrad	0.642	0.671	4.233	31.333	37.290	15.974
RMSProp	1.451	1.729	16.104	75.962	105.288	27.854
Adam	1.072	1.197	10.394	54.422	68.703	20.787

Energy: mJ, Latency: Min. Each evaluation includes 100 epochs and each epoch includes 500 images with 50 levels of conductance.

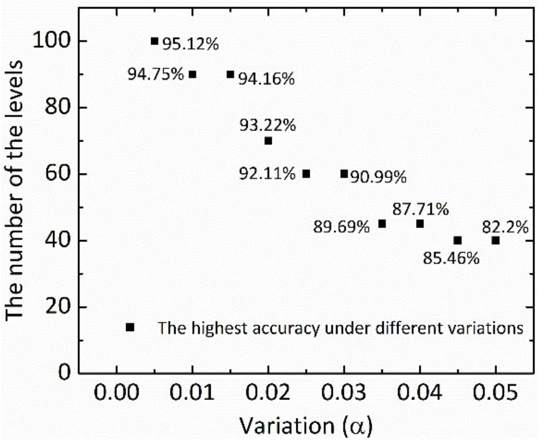

Fig. 9: Number of Levels with Highest Accuracy under Different Cycle-to-Cycle Variations.

[5] J. Fu, Z. Liao, N. Gong, and J. Wang, “Linear optimization for memristive device in neuromorphic hardware,” in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2019, pp. 453–458.

[6] J. Fu, Z. Liao, and J. Wang, “Cycle-to-cycle variation enabled energy efficient privacy preserving technology in ann,” in *2020 IEEE 33rd International System-on-Chip Conference (SOCC)*, 2020, pp. 66–71.

[7] Z. Liao, J. Fu, C. Ding, and J. Wang, “Pulse truncation enabled high performance and low energy memristor-based accelerator,” in *SoutheastCon 2022*, 2022, pp. 473–478.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[9] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, “Dot-product engine for neuromorphic computing: Programming 1t1m crossbar to accelerate matrix-vector multiplication,” in *2016 53nd acm/edac/ieee design automation conference (dac)*, 2016, pp. 1–6.

[10] L. Chua, “Memristor-the missing circuit element,” *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[11] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[12] J. Fu, Z. Liao, N. Gong, and J. Wang, “Mitigating nonlinear effect of memristive synaptic device for neuromorphic computing,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 377–387, 2019.

[13] J. Fu, Z. Liao, and J. Wang, “Memristor-based neuromorphic hardware improvement for privacy-preserving ann,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2745–2754, 2019.

[14] J. Fu, Z. Liao, J. Liu, S. C. Smith, and J. Wang, “Memristor-based variation-enabled differentially private learning systems for edge computing in iot,” *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9672–9682, 2021.

[15] M. Oli-Uz-Zaman, S. A. Khan, G. Yuan, Z. Liao, J. Fu, C. Ding, Y. Wang, and J. Wang, “Mapping transformation enabled high-performance and low-energy memristor-based dnns,” *Journal of Low Power Electronics and Applications*, vol. 12, no. 1, p. 10, 2022.

[16] Z. Liao, J. Fu, and J. Wang, “Ameliorate performance of memristor-based anns in edge computing,” *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1299–1310, 2021.

[17] P.-Y. Chen, B. Lin, I.-T. Wang, T.-H. Hou, J. Ye, S. Vrudhula, J.-s. Seo, Y. Cao, and S. Yu, “Mitigating effects of non-ideal synaptic device characteristics for on-chip learning,” in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2015, pp. 194–199.

[18] J. Rajendran, R. Karri, and G. S. Rose, “Improving tolerance to variations in memristor-based applications using parallel memristors,” *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 733–746, 2014.

[19] J. Fu, Z. Liao, and J. Wang, “Level scaling and pulse regulating to mitigate the impact of the cycle-to-cycle variation in memristor-based edge ai system,” *IEEE Transactions on Electron Devices*, vol. 69, no. 4, pp. 1752–1762, 2022.

[20] S. A. Khan, M. Oli-Uz-Zaman, and J. Wang, “Pawn: Programmed analog weights for non-linearity optimization in memristor-based neuromorphic computing system,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 436–444, 2023.

[21] M. Oli-Uz-Zaman, S. A. Khan, W. Oswald, Z. Liao, and J. Wang, “Reconfigurable mapping algorithm based stuck-at-fault mitigation in neuromorphic computing systems,” in *Proceedings of the Great Lakes Symposium on VLSI 2023*, 2023, pp. 261–266.

[22] M. Oli-Uz-Zaman, S. A. Khan, G. Yuan, Y. Wang, Z. Liao, J. Fu, C. Ding, and J. Wang, “Reliability improvement in rram-based dnn for edge computing,” in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 581–585.

[23] M. Uddin, M. S. Hasan, and G. S. Rose, “On the theoretical analysis of memristor based true random number generator,” in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 2019, pp. 21–26.

[24] C. Baeumer, R. Valenta, C. Schmitz, A. Locatelli, T. O. Mentes, S. P. Rogers, A. Sala, N. Raab, S. Nemsak, M. Shim *et al.*, “Subfilamentary networks cause cycle-to-cycle variability in memristive devices,” *ACS nano*, vol. 11, no. 7, pp. 6921–6929, 2017.

[25] J.-H. Lee, D.-H. Lim, H. Jeong, H. Ma, and L. Shi, “Exploring cycle-to-cycle and device-to-device variation tolerance in mlc storage-based neural network training,” *IEEE Transactions on Electron Devices*, vol. 66, no. 5, pp. 2172–2178, 2019.

[26] P.-Y. Chen, X. Peng, and S. Yu, “Neurosim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3067–3080, 2018.

[27] L. Gao, I.-T. Wang, P.-Y. Chen, S. Vrudhula, J.-s. Seo, Y. Cao, T.-H. Hou, and S. Yu, “Fully parallel write/read in resistive synaptic array for accelerating on-chip learning,” *Nanotechnology*, vol. 26, no. 45, p. 455204, 2015.

[28] P.-Y. Chen, X. Peng, and S. Yu, “Neurosim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures,” in *2017 IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 6–1.

[29] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, “Everything you wanted to know about deep learning for computer vision but were afraid to ask,” in *2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*, 2017, pp. 17–41.

[30] M. Oli-Uz-Zaman, S. A. Khan, W. Oswald, Z. Liao, and J. Wang, “Stuck-at-fault immunity enhancement of memristor-based edge ai systems,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 4, pp. 922–933, 2022.

Authorized licensed use limited to: UNIV OF SOUTH ALABAMA. Downloaded on December 25,2023 at 05:17:22 UTC from IEEE Xplore. Restrictions apply.