PAWN: Programmed Analog Weights for Non-linearity Optimization in Memristor-based Neuromorphic Computing System

Saleh Ahmad Khan, *Student Member, IEEE,* Md. Oli-Uz-Zaman, *Student Member, IEEE,* and Jinhui Wang, *Senior Member, IEEE*

***Abstract*—Memristors offer advantages as a hardware solution for neuromorphic systems. However, their nonlinear device property makes the weight update inaccurately and reduces the inference accuracy of a neural network. A Programmed Analog Weights for Nonlinearity (PAWN) method is proposed in this paper to update the conductance of a memristor by following the nonlinear curve during the training in a neuromorphic system. The experiment results indicates the PAWN method is effective to alleviate the nonlinearity influence to memristors in all different LTP/LTD conditions. Especially in extreme nonlinearity (LTP=6, LTD=-6), the memristor-based neuromorphic system has significantly low accuracy (51.77%), but the PAWN method enables large accuracy improvement (9.87%) without the inference energy and latency overhead. In addition, overall performance of the neuromorphic system is also evaluated for further verification. Finally, comprehensive experiments show that the PAWN method is still greatly valid even considering device-to-device variations, cycle-to-cycle variations, various technology nodes, and different architectures.**

***Index Terms*—Memristor, neuromorphic system, artificial intelligence (AI), inference accuracy, nonlinearity.**

I. Introduction

Although the transistor scaling of traditional Complementary Metal-Oxide Semiconductor (CMOS) technology has supported the growing computational demand from customers in last decades, physical limitations such as quantum tunneling [1] would suppress its further development. Therefore, a new platform with high speed and low power enabling computationally intensive applications, such as large datacenters and IoT systems, is needed and necessary. Memristor-based neuromorphic system brings such promising technology which has a potential of making smooth transition from CMOS-based system to memristive devices by unlocking computing in memory (CIM) capability. Memristor uses multi-level conductance sates to regulate current flow as well as storing the amount of charge that has previously been flowed through it. If powering off, such programmed state and charge are not lost. Besides non-volatility and multilevel resistive state property, memristor exhibits characteristics like low computational complexity [2], sub-nanosecond switching speed [3]–[5], sub-10-nm scalability [6], low energy dissipation of few pJ per bit [3], [7]–[10], long write-erase endurance [11], and CMOS-compatibility [12], [13]. As a result, it can efficiently implement high performance neural networks in hardware for CIM [14]. In a neuromorphic computing system, the conductance switching property of the memristor is exceptionally suitable to represent the weight update of bio-inspired neural connection. Conductance of memristor in the neuromorphic system depends on the external stimulation, such as voltage pulses. Based on the input pulses, the conductance of memristor positively and negatively changes which enables the weight increase and decrease respectively as Long-Term Potentiation (LTP) and Long-Term Depression (LTP). However, the non-ideal property [15]–[18] of such conductance modulations including nonlinearity, device-to-device and cycle-to-cycle variations, and Stuck-at-Fault (SAF) defects have significantly negatively impacts the inference accuracy of such a neuromorphic system. Especially, the nonlinearity makes it challenging to determine a proper width or amplitude of input signals for achieving the desired conductance of memristors. It is reported that the linear conductance change is the major requirement of a memristor-based neuromorphic system to realize high accuracy for the online learning [19]. For example, four state-of-the-art memristors in literature, Ag:a-Si [20], TaO_x/TiO_2 [21], $PCMO$ [22], and AlO_x/HfO_2 [23] are all characterized by device nonlinearity. The inference accuracy based on Ag:a-Si with nonlinearity decreases over 20% than that without nonlinearity [19], [20].

Regarding the non-ideal properties, for overall performance improvement, researchers come up with diverse techniques including level scaling and pulse regulating method [24], pulse compression method [8], mapping transformation method [3] for optimizing conductance level, reforming pulse distribution, improving inference accuracy, and saving energy consumption in a neuromorphic hardware.

As for nonlinearity optimizations, researchers propose different techniques as three categories: new device structure, new control method, and new programming method. **New device structure:** A thermal enhanced layer is added to confine heat in the switching layer to the HfO_X memristor for the nonlinearity optimization in [25]; an ion-diffusion limiting layer is built for the TiN/TaO_X memristor for linearity enhancement in [26]; a charge trap layer is utilized to a gated Schottky diode in order to cancel the nonlinearity

Manuscript received Oct. 15, 2022. This work was supported in part by the National Science Foundation under Grant 2218046, 1953544, and 1855646. *(Corresponding author: Jinhui Wang.)*

Saleh Ahmad Khan, Md. Oli-Uz-Zaman, and Jinhui Wang are with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL 36688 USA (E-mail: jwang@southalabama.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

factor in [27]. Although these devices can achieve relatively better linear property, they may ignore the other important features designated for neuromorphic computing, such as the ON/OFF ratio and endurance characteristics. For instance, the reconfigurable gated Schottky diode has better linearity, but with low ON/OFF ratio [27], which limits its application as an analog synapse. Moreover, nonlinearity is widespread and intrinsic in almost all memristors. Due to the high process and implementation cost, it is impossible to always create a new device structure for nonlinearity optimizations. **New control method:** Some improved methods are proposed to control the conductance change for memristors with current, time-, flux-, and charge-domain. The current and time-domain control mechanisms are based on transistor gate voltage and time duration in the configuration of one transistor and one memristor [28], [29]. The flux- and charge-domain control method describes a device state as a function of flux or charge and change the conductance of a memristor according to quantization of the flux or charge [30]. In theory, these methods can accurately control the conductance of the memristor. However, the voltage or current inputs required by these methods are too complex to be implemented, because they need many irregular pulses which are difficult to generate. For example, for the time-domain control [31], 3rd, 4th, 7th, or 9th order function for the input voltage curve are required, but generating that voltage consumes too much time and power, it is often impossible to be realized at circuit level. **New programming method:** Programming methods are another solutions to achieve controllable conductance modulation. In [32], [33], the bipolar-pulse scheme applies a pair of positive and negative pulses with different amplitudes and durations. It partly mitigates nonlinearity at the low conductance stages where usually have large overshoots. However, the nonlinearity at the high conductance stages still exists. Also, in order to obtain precise conductance tuning, in [34]–[36], write-and-verify tuning with feedback circuits are used to adjust the device reliably. A linear and symmetric relation is demonstrated but using a much larger digital memory and multiple types of pulses in [35] and requiring to identify and verify the precise conductance of the device for each weight update. Consequently, extra processing circuits and a specific pulse generator are added, which increases the complexity of circuits and leads to area overhead and performance penalty.

Therefore, although above existing methods partly remove the influence of the nonlinearity on neuromorphic systems, in order to avoid cost for developing new structure, high order function in circuit level, and extra periphery circuits, a new solution is needed.

Accordingly, a new method - Programmed Analog Weights for Nonlinearity (PAWN) - is proposed in this paper to update the conductance of a memristor by following the nonlinear curve during the training of a neural network. The PAWN method avoids the deviation between the active device and learning algorithm. Thus, it will greatly enhance the inference accuracy. This paper will make the following contributions:

- The algorithm and working flow of the PAWN method is introduced in detail.
- Experiments are conducted to verify general nonlinearity mitigation of the PAWN method.
- Considering device-to-device variations, cycle-to-cycle variations, various technology nodes, different architectures, and overall performance, the PAWN method is further verified through comprehensive experiments.
- The PAWN method is compared with the state-of-the-art.

The rest of the paper is organized as follows. In Section II, a Programmed Analog Weights for Nonlinearity (PAWN) method is presented. The results and discussions regarding inference accuracy considering device-to-device variations, cycle-to-cycle variations, various technology nodes, and different architectures have been provided in Section III. Comparison with State-of-the-Art is described in section IV. Finally, the conclusion is drawn in V.

II. Methodology

A. *Nonlinearity in Memristor-based Neuromorphic Systems*

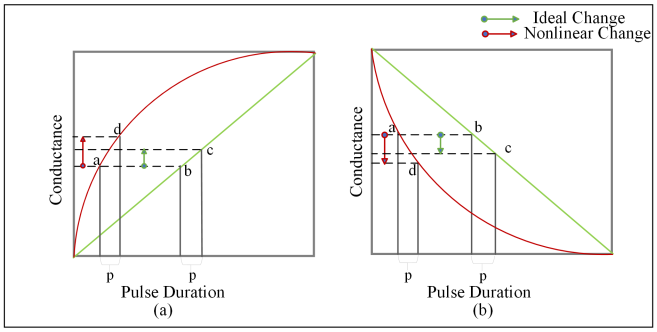

Fig. 1. Conductance Change with Nonlinearity in (a) LTP and (b) LTD

The nonlinearity of the memristor happens as the conductance of a memristor is not updated following the ideal curve. In a practical memristor, most of time, conductance does not change in proportion to the external stimulus. In case of multistate [37] or different models [38] of memristor presented in other research works, it also shows the same properties of conductance change. The conductance of the memristor represents the weight of the neural network and in the case of increasing the weight, it is termed as Long-Term Potentiation (LTP). Conductance can also be decreased and it is termed as Long-Term Depression (LTD). As demonstrated in Fig. 1, the red curve depicts the conductance of an actual memristor. The pulses have the same duration and the same amplitude, and the light green line represents the function of the ideal case. In LTP, as shown in Fig. 1 (a), according to the result obtained by the algorithm, the conductance of memristor theoretically needs to be changed from point b to c. Then, the corresponding number of pulses is calculated according to the ideal curve (green). However, when these pulses are applied to the actual memristor, instead of changing from point b to c, the device conductance changes from point a to d. Consequently, the actual change of conductance and the required change are not same. Similarly, Fig. 1 (b) shows the occurrence in the LTD, where the actual conductance changes to point d instead of point c. The nonlinearity of the memristor causes the weight

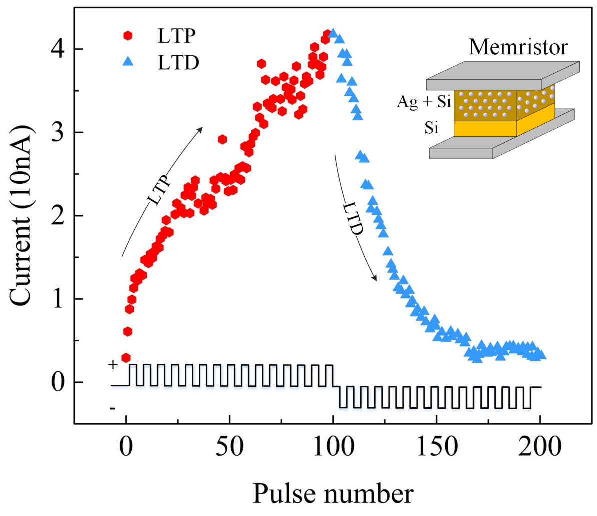

Fig. 2. Memristors Response to Pulse Stimulates.

change to be inconsistent with the change required by the learning algorithm, thereby reducing the inference accuracy.

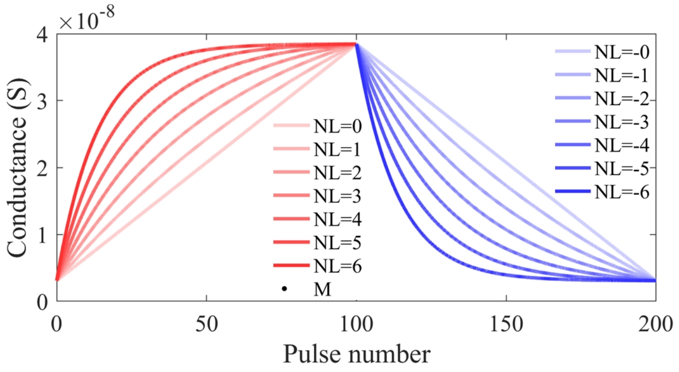

Fig. 3. Conductance Change Curves under Various Nonlinearity of LTP and LTD.

B. Device Specifics and Implementation in Image Recognition

In order to verify the proposed PAWN method, the fabricated memristor with Silver (Ag) and Silicon (Si) structure and tested current-pulse characteristics are utilized. As shown in Fig. 2, the curves indicates that the memristor is programmed by consecutive 100 identical positive (LTP, 3.2 V, 300 μs) pulses followed by consecutive 100 identical negative pulses (LTD, -2.8 V, 300 μs) [20]. We adopted MLP NeuroSIM [19] framework with above tested parameters and results to define device and verify the proposed PAWN method. The neural network uses iterative Stochastic Gradient Descent (SGD) algorithm on standard MNIST handwritten digits dataset. A three layer multilayer perceptron (MLP) neural network configuration is used in this work. The neural network consists of three layers with 400, 100 and 10 neuron topology (denoted as 400-100-10, neurons in hidden layer will change for the further architecture verification). The dataset includes 70000 images among which 60,000 are used to train the network and 10,000 images are used for testing. The input images from the training dataset are cropped and encoded into black and white data for simplification on the hardware implementation. The weights are mapped to the conductance of the memristor cells. Finally, the memristor cells are embedded in crossbar architecture where weights update with other hardware control logics (Adder, Mux, Registers) is performed.

C. Programmed Analog Weights for Nonlinearity

To eliminate the nonlinear property in the memristor-based neuromorphic system, a PAWN method is proposed in this work. This method is to ameliorate the error by updating weights following the nonlinear curve of memristor. Such a nonlinear curve is generated from a mathematical model (Equations (1) and (2)) [19], as shown in Fig. 3. The curve is labeled with a NL value (it is the normalized value of parameter A in Equations (1) and (2)) from +6 to -6, which represents the extent to the curve deviates from the ideal linear device. Here the positive (+) and negative (-) signs are merely to label LTP and LTD, respectively. Equation (3) is our proposed conductance update algorithm for LTP cases.

$$G_{LTP} = B(1 - e^{(-\frac{P}{A})}) + G_{min} \tag{1}$$

$$G_{LTD} = -B(1 - e^{(-\frac{P-P_{max}}{A})}) + G_{max} \tag{2}$$

$$G_{New} = (G_2 - G_1) \times (P - P_x) + G_1 \tag{3}$$

where G_{LTP} and G_{LTD} are the conductance for LTP and LTD cases, respectively. G_{max}, G_{min}, and P_{max} are extracted from the experimental and testing data, which represents the maximum conductance, minimum conductance, and maximum pulse number required to switch the device between the minimum and maximum conductance states. Parameter A controls the nonlinear behavior of weight update. B is a function of A that fits the functions within the range of G_{max}, G_{min}, and P_{max}. In the PAWN method, the number of pulses is calculated firstly from the initial conductance value of memristors (each memristor has a random initial value conductance). Then 2 new conductance values G_1 and G_2 are calculated (one from the current number of pulses calculated from the algorithm and another from the ceiling of the current number of pulses) based on Equation (1) for LTP cases and Equation (2) for LTD cases. For LTP cases in Equation (3), P is the number of current pulses and P_x is the integer value of P.

$$P = P_i + \frac{G_{old} - G_1}{G_2 - G_1} \tag{4}$$

For LTD cases, conductance values gradually decrease with the increased number of negative pulses. Therefore, to precisely get the pulse numbers, the conductance states in terms of the pulse number is calculated through Equation (4) where P_i is the initial number of pulses, G_{old} is the old conductance value (finalized in the last epoch), G_2 represents the conductance value after increasing the number of pulses and P is the pulse numbers needed to achieve the next conductance. Through this process, the weight update disturbance can be avoided and the inference accuracy of the neural network can be enhanced.

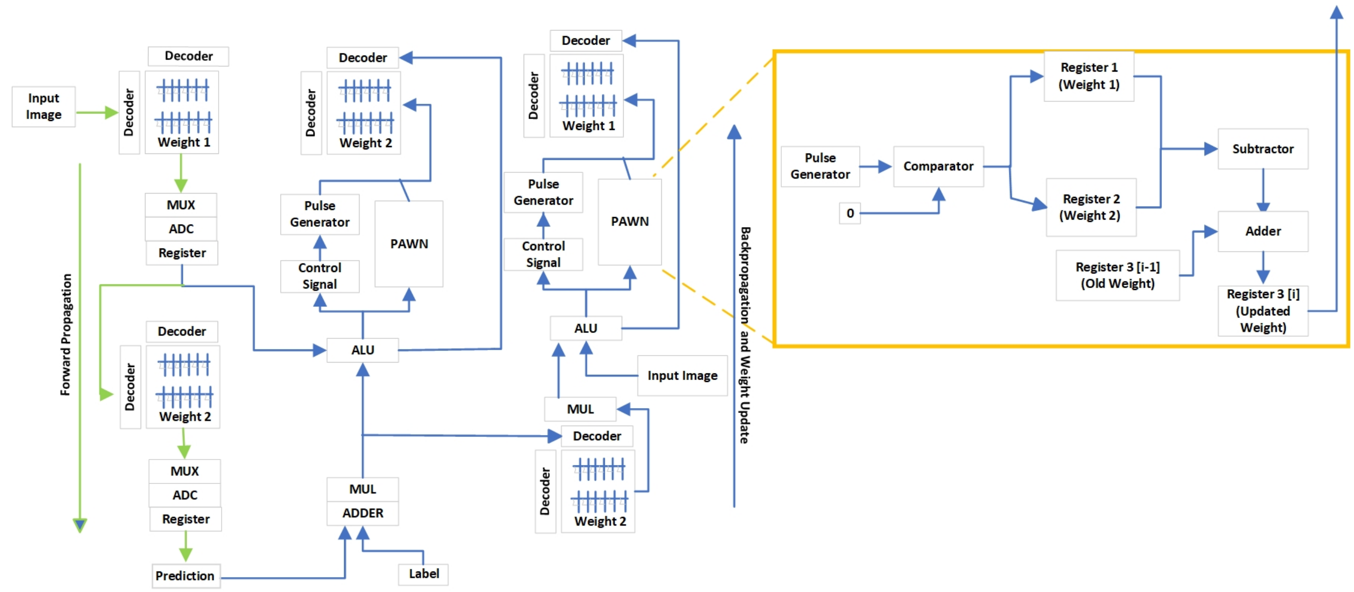

Fig. 4. Architecture of Memristor-based Neuromorphic System to Implement PAWN Method

D. Architecture Design

The architecture of memristor-based neuromorphic system to implement the PAWN method is shown in Fig. 4. It works as follows: In order to guarantee the conductance tuning is correctly finished, the weight change (Δweight) is firstly calculated in the ALU. In the PAWN block, control signals enables the pulse generator for producing positive/negative pulses for conductance tuning. A comparator is added as an indicator for distinguishing the polarity (positive/negative) of pulse. G_1 and G_2 in Equations (3) and (4) are stored in Register 1 and Register 2, respectively. A subtractor is used to take the difference between these two registers and the difference is fed into the adder. Finally, G_1 is taken as input to the same adder which generates the updated conductance. This process is iterated for all positive and negative pulses to make the weight update for LTP and LTD cases. It is also described in Algorithm 1 for the implementation of the PAWN method. Theoretically, higher NL (denoted in Fig. 3) for LTP and LTD means conductance deviates more from the ideal curve calculated from the algorithm (green line in Fig. 1), resulting in higher accuracy drops.

What is more, because the PAWN block is added to the architecture, it will inevitably generate hardware overhead including a pulse generator, comparator, registers, subtractor, and adder. However, the neuromorphic system consists of memristor arrays, analog-to-digital converter (ADC), accumulation circuits on chip (adders and accumulation units), and other peripheral circuits (decoders, MUX, switch matrix, buffers, and activation units) [39]. As compared with the entire neuromorphic system, the PAWN block is tiny and can be negligible.

III. Result And Discussion

Comprehensive experiments have been performed to validate the proposed PAWN method that seeks to mitigate the

Algorithm 1 Pseudocode for PAWN

```
Weight difference $deltaweight$, conductance numbers index $i$, Maximum conductance state $N_{max}$, Number of Pulse $numPulse$
if $deltaweight > 0$ :
    Initialize current pulse;
if $numPulse > 0$ :
    $G_2$ = Conductance calculated from the ceiling of current pulse;
    $G_1$ = Conductance calculated from the current pulse;
    Updated Conductance = $G_{old}$ + ($G_2$ - $G_1$)
else:
    Updated Conductance = $G_{old}$
else if: $deltaweight < 0$ :
    Initial pulse position, $pulse\_position$ = -1
for $i = 0, 1, 2, ..., N_{max}$
    $Weight\_i_1$ = Conductance for current pulse
    $Weight\_i_2$ = Conductance for (current pulse + 1)
    $Temp$ = ($Weight\_i_1$ - cond.) * ($Weight\_i_2$ - cond.)
    if $Temp <= 0$ :
        $pulse\_position$ = $i$
if $numPulse < 0$ :
    $G_1$ = Conductance with $pulse\_position$;
    $G_2$ = Conductance with ($pulse\_position$ +1);
    $new\_pulse$ = $pulse\_position$ + $\frac{G_{old}-G_1}{G_2-G_1}$
    Updated Conductance = conductance with $new\_pulse$;
else:
    Updated Conductance= $G_{old}$; //no conductance update
```

TABLE I
ACCURACY FOR DIFFERENT NONLINEARITIES WITH 32 NM TECHNOLOGY AND (400-100-10) ARCHITECTURE

LTP	LTD	Before PAWN				After PAWN				Accuracy
		Accuracy	Energy (J)	Latency (S)	Overall	Accuracy	Energy (J)	Latency (S)	Overall	Improvement
0	0	93.12%	4.1943e-04	4.0590	546.9712	93.12%	4.1943e-04	4.0590	546.9712	0%
1	-1	87.81%	4.1948e-04	4.0590	515.7196	89.37%	4.1943e-04	4.0590	524.9443	1.56%
2	-2	77.37%	4.1950e-04	4.0590	454.3824	83.81%	4.1947e-04	4.0590	492.2388	6.44%
3	-3	66.03%	4.1951e-04	4.0590	387.7751	78.81%	4.1947e-04	4.0590	462.8724	6.44%
4	-4	64.35%	4.1950e-04	4.0590	378.9751	73.39%	4.1947e-04	4.0590	431.0394	9.04%
5	-5	62.20%	4.1949e-04	4.0590	365.3000	71.24%	4.1946e-04	4.0590	418.4218	9.04%
6	-6	51.77%	4.1947e-04	4.0590	304.0592	61.64%	4.1944e-04	4.0590	362.0543	9.87%

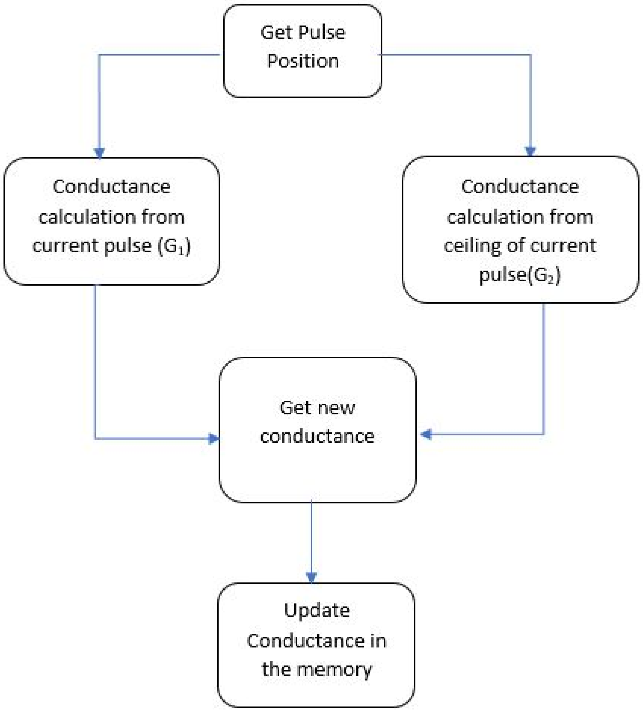
Get Pulse Position
Conductance calculation from current pulse (G_1)
Conductance calculation from ceiling of current pulse(G_2)
Get new conductance
Update Conductance in the memory

Fig. 5. Working Flow of Weight Programming in Memristor-based Neuromorphic Hardware

nonlinearity of memristors in a hardware implementation for a neuromorphic system. For various LTP (1 to 6) and LTD (-1 to -6), experiments are conducted following the working flow of the PAWN method in Fig. 5. At the beginning of the training process, the accuracy is very low for default conditions since the weights are randomized, but after several epochs, the weights show more stability. The experiment runs for 125 epochs to achieve optimum performance.

A. General Nonlinearity Mitigation

As listed in Table I, using 32 nm technology and (400-100-10) architecture mentioned in Section II.B, if no nonlinearity or other fault is considered, the memristor-based neuromorphic system can achieve 93.12% accuracy. When nonlinearity (LTP, LTD) are incorporated into the model, the accuracy starts to decline. And with the increased LTP/LTD, the accuracy decreases more. Also, the Table I indicates that the PAWN method is effective to alleviate the nonlinearity influence to memristors in all different LTP/LTD conditions. Especially in extreme nonlinearity (LTP=6 and LTD=-6), the memristor-based neuromorphic system has significantly low accuracy (51.77%), but has the great accuracy improvement (9.87%) after the PAWN method is applied.

B. Energy, Latency, and Overall Performance

The inference energy and latency are also listed in Table I. Because the proposed PAWN is only used in the training stage for the weight mapping from learning algorithm to memristors, it will not generate the inference energy and latency overhead. Furthermore, considering the trade-off among inference accuracy, energy, and latency, the overall performance is calculated using following equation, and the PAWN method is still effective to improve the overall performance of neuropmorphic systems.

$$P_{overall} = \frac{Accuracy}{Energy \times Latency} \tag{5}$$

In addition, as compared with very high accuracy (over 98%) from pure learning algorithm, lower than 90% accuracy is normal conditions for memristor based neuromorphic systems [40]. This is because memristors have non-ideal properties such as large nonlinearity and device variations (device variations will be discussed in following sections), and more importantly, quantization effects from Analog-to-Digital Converter (ADC) and from the mapping technique are inevitable. Our proposed PAWN method provide a solutions to improve the overall performance of neuromorphic systems with such non-ideal properties. Although it cannot recover the accuracy to 90%, it is not limited to a specific memristor, but universal for all memristor based neuromorphic systems, and avoid the inference energy and latency. It provides a good reference for following designer and different applications.

C. Impact of Device-to-Device Variations

Since the switching mechanism of the memristor conductance is prompted by the applied voltage, a memristor switches its conductance levels from one to another when pulse is larger than a threshold voltage for at least the minimum required time [41]. However, variations are always concerns in memristors, as shown in Fig. 6. From weight update perspective, one of the synaptic device variations is the spatial variation from device-to-device, which results in different conductance changes when the same pulse is applied in different memristors. The stochastic nature of the formation and rupture of conductive

TABLE II
ACCURACY COMPARISON OF NONLINEARITIES WITH DEVICE-TO-DEVICE VARIATIONS USING 32 NM TECHNOLOGY AND (400-100-10) ARCHITECTURE

σ	LTP	LTD	Accuracy Before PAWN	Accuracy After PAWN	Accuracy Improvement
5%	1	-1	86.34%	89.89%	3.55%
	2	-2	80.06%	84.00%	3.94%
	3	-3	74.99%	80.78%	5.79%
	4	-4	71.23%	72.43%	1.20%
	5	-5	63.23%	70.09%	6.86%
	6	-6	53.13%	69.27%	16.14%
10%	1	-1	86.80%	89.62%	2.82%
	2	-2	77.16%	84.10%	6.94%
	3	-3	68.93%	78.63%	9.70%
	4	-4	69.80%	75.54%	5.74%
	5	-5	61.96%	67.69%	5.73%
	6	-6	32.22%	64.82%	32.60%
20%	1	-1	87.42%	90.15%	2.73%
	2	-2	79.55%	83.05%	3.50%
	3	-3	74.90%	79.65%	4.75%
	4	-4	68.77%	74.17%	5.40%
	5	-5	62.28%	72.97%	10.69%
	6	-6	42.04%	65.52%	23.48%

TABLE III
ACCURACY COMPARISON OF NONLINEARITIES WITH CYCLE-TO-CYCLE VARIATIONS USING 32 NM TECHNOLOGY AND (400-100-10) ARCHITECTURE

σ	LTP	LTD	Accuracy Before PAWN	Accuracy After PAWN	Accuracy Improvement
0.5%	1	-1	86.23%	87.48%	1.25%
	2	-2	78.01%	69.50%	-8.51%
	3	-3	67.77%	65.65%	-2.12%
	4	-4	68.91%	62.46%	-6.45%
	5	-5	61.57%	63.47%	1.90%
	6	-6	53.66%	64.53%	10.87%

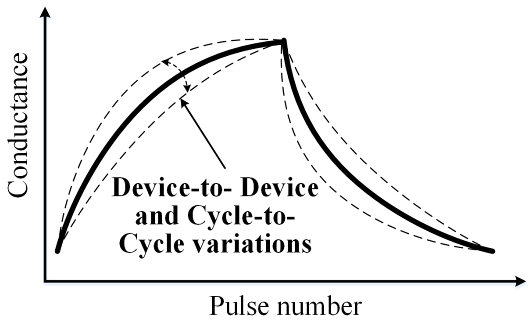

Fig. 6. Device-to-Device and Cycle-to-Cycle Variations

filament is believed to be the main reason for this variations. It is a major hindrance for information storage in memristors [42]. Due to this random nature of the conductive filament, the prediction and the precise control of the shape of the conductive filament becomes extremely challenging [43].

Device-to-device variation represents nonlinearity variations of memristors in crossbar array, which subjects to $N(NL(LTP), \sigma)$ and $N(NL(LTD), \sigma)$ distribution [44]. In our experiments, the parameter (σ) is varied from 5% to 20%. From results listed in Table II, using 32 nm technology and (400-100-10) architecture, generally, in each case, the accuracy with PAWN method shows great improvement, verifying the efficiency of the PAWN method. With the increase of (σ), LTD, and LTP, the accuracy improvement increases. For each specific σ, the condition with LTD=6 and LTP=-6 always have the highest accuracy improvement. They are respectively 16.14%, 32.60%, and 23.48%.

Since different devices have different weight update curve (depending on A in Equations (1) and (2)), the accuracy for same LTP and LTD in different devices varies. Here, because the PAWN method updates the weight with SGD algorithm which is fundamentally stochastic in nature, the accuracy improvement with the PAWN method does not follow any specific pattern. The same interpretation of accuracy applies for Table III, IV, and V where the impact of different technology nodes, cycle to cycle variations, and architectures are investigated.

D. Impact of Cycle-to-Cycle Variations

The fluctuation in conductance change at each programming pulse is referred to as the cycle-to-cycle weight update variation. This fluctuation (σ) is represented as a percentage of the entire conductance range. Because of the form of the conductive filament, the oxygen vacancy distribution at and around the filament, and the shifting position of the active filament from one cycle to the next, memristors display significant cycle-to-cycle variations [45], as shown in Fig. 6. In Table III, 0.5% cycle to cycle variation is considered for different LTP and LTD cases which shows 10.87% accuracy improvement in extreme nonlinearity (LTD=6 and LTP=-6). In some cases, the accuracy tends to fall after the PAWN method is applied. This is because the proposed PAWN method follows onlinear curves but cycle-to-cycle variation introduces

TABLE IV
ACCURACY COMPARISON OF NONLINEARITIES WITH DIFFERENT TECHNOLOGY NODES IN (400-100-10) ARCHITECTURE

Technology Node	LTP	LTD	Accuracy Before PAWN	Accuracy After PAWN	Accuracy Improvement
32 nm	1	-1	87.81%	89.37%	1.56%
	2	-2	77.37%	83.81%	6.44%
	3	-3	66.03%	78.81%	12.78%
	4	-4	64.35%	73.39%	9.04%
	5	-5	62.20%	71.24%	9.04%
	6	-6	51.77%	61.64%	9.87%
14 nm	1	-1	88.00%	89.40%	1.40%
	2	-2	75.54%	82.56%	7.02%
	3	-3	67.34%	78.94%	11.60%
	4	-4	64.35%	74.19%	1.78%
	5	-5	58.80%	68.71%	9.91%
	6	-6	44.26%	67.88%	23.62%
10 nm	1	-1	86.79%	89.73%	2.94%
	2	-2	74.21%	84.12%	9.91%
	3	-3	74.05%	78.52%	3.63%
	4	-4	71.68%	77.76%	6.08%
	5	-5	64.53%	71.66%	7.13%
	6	-6	50.81%	69.20%	18.39%

the erratic change in weight updates and makes the weight update hard to follow such nonlinear curves. Therefore, for some cycle to cycle variations (LTP/LTD= 2/-2, 3/-3, 4/-4) accuracy even drops as compared with the accuracy before the PAWN method is applied.

E. Impact of Technology Nodes

Following Moore's law, a new technology node is released every 2 years to make feature size of transistors smaller in every 2 years. Starting in 1971 with 10 μm technology node, the semiconductor industry have invested heavily to develop those technology nodes that enable essentially faster, cheaper, and smaller chips. It is pertinent to verify the proposed PAWN method for different technology nodes to make sure it can be similarly effective for smaller and faster memristor-based neuromorphic systems. In this verification process, technology nodes from 32 nm to 10 nm are considered for different LTP and LTD configurations.

From Table IV, using (400-100-10) architecture, it can be observed that the PAWN method enable all accuracy improvement in different technology nodes. In each technology node, with the increased LTP and LTD, the accuracy decrease. Also in each technology node, the condition with LTD=6 and LTP=-6 always has the high accuracy improvement. They are respectively 9.87% for 32 nm, 23.62% for 14 nm, and 18.39% for 10 nm.

F. Impact of Different Architectures

To further investigate the impact of different architectures for nonlinearity, different numbers of neurons in the hidden layer are considered. In the previous experiments, the structure of the neural network is (400-100-10) for three-layer network. Now, (400-150-10), (400-250-10), and (400-350-10) architectures are included in the experiments. As accurate weight/conductance update is related with the number of neurons, this set of experiments verifies the effectiveness of the PAWN method in different architectures.

From Table V, using 32 nm technology, after the PAWN method is applied, the more neurons in the hidden layers, the higher accuracy is obtained. That is, from LTP/LTD=1/-1 to LTP/LTD=6/-6, with the PAWN method, (400-150-10) architecture has accuracy as 65.98%-90.98%, while (400-250-10) and (400-350-10) architectures have accuracies as high as 75.23%-92.16% and 75.53%-92.42%. However, the accuracy improvement is stochastic. The highest accuracy improvement for (400-150-10), (400-250-10), and (400-350-10) architectures are respectively 42.87% at LTD/LTP=6/-6, 69.57% at LTD/LTP=4/-4, and 16.60% at LTD/LTP=6/-6.

IV. COMPARISON WITH STATE-OF-THE-ART

The PAWN technique offers a simple and feasible method to address nonlinearity issue in memristor-based neuromorphic systems. As shown in Table VI, nine other research works which addresses to mitigate nonlinearity in memristors are compared with this work. [32] considers device variations during weight update, but excludes other 4 certeria. In [46], [33] [47], [39], and [3], the nonlinearity is optimized, but they do not explore criteria mentioned in Table VI. Although [48], [49] uses linear optimization method to mitigate extreme nonlinearities, but fails to validate in different variations, technology nodes, and architectures. [44] has verified similar criteria with our work, but ignores different technology nodes. The PAWN method stands out in terms of many criteria including device variations, cycle to cycle variations, technology nodes, and network architectures, finally mitigating effects in extreme nonlinearity conditions. The proposed method not only provides accuracy enhancement in different configurations but also meets the requirements of device characteristics which makes it an efficient solution for different applications.

V. CONCLUSION

In this paper, a Programmed Analog Weights for Nonlinearity (PAWN) method is presented to mitigate nonlinearity impact on the memristor-based neuromorphic systems. The detailed experiments are conducted in all different LTP/LTD

TABLE V
ACCURACY COMPARISON OF NONLINEARITIES WITH VARIABLE NEURONS IN HIDDEN LAYER USING 32 NM TECHNOLOGY

Neurons in Hidden Layer	LTP	LTD	Accuracy Before PAWN	Accuracy After PAWN	Accuracy Improvement
150	1	-1	87.35%	90.98%	3.63%
	2	-2	79.25%	86.78%	7.53%
	3	-3	74.76%	82.77%	8.01%
	4	-4	72.46%	81.13%	8.67%
	5	-5	38.66%	72.42%	33.76%
	6	-6	23.11%	65.98%	42.87%
250	1	-1	89.98%	92.16%	2.18%
	2	-2	60.80%	88.36%	27.56%
	3	-3	56.63%	83.85%	27.22%
	4	-4	31.60%	79.67%	69.57%
	5	-5	26.07%	76.66%	50.59%
	6	-6	33.53%	75.23%	41.70%
350	1	-1	88.87%	92.42%	3.55%
	2	-2	85.77%	88.56%	2.79%
	3	-3	70.00%	82.08%	12.08%
	4	-4	77.55%	81.24%	3.69%
	5	-5	76.74%	79.87%	3.13%
	6	-6	59.93%	75.53%	15.60%

TABLE VI
COMPARISON WITH STATE-OF-THE-ART

Criteria	[46]	[33]	[44]	[47]	[32]	[39]	[3]	[48]	[49]	This Work
Device to Device Variations	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓✓
Cycle to Cycle Variation	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Different Architectures	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Technology Nodes	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Nonlinearity Mitigation in Extreme Conditions (LTP/LTD=+/-6)	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓✓
✓(Work is done) ✓✓(More data than previous work)										

conditions from 1/-1 to 6/-6. Especially, in extreme nonlinearity (LTP=6, LTD=-6), the PAWN method can always has the high improvement for the inference accuracy. Also, the PAWN method has significant immunity to device-to-device variations. For example, in specific σ=10%, the condition with LTD=6 and LTP=-6 has the accuracy improvement as high as 32.60%. What is more, even with different technology nodes and architectures, the PAWN method still effectively improves accuracy up to 23.62% and 69.57%, respectively. Actually, the PAWN method is for the algorithm of conductance updating, but not depending on the specific device, so it is a universal method for any memristor with different materials and mechanisms. In future work, we will consider larger nonlinearity such as -7/+7, -8/+8, and -9/+9 to adapt to more devices or extreme conditions. And some other non-ideal properties, such as Stuck-At-Fault (SAF) may be considered for further verification.

ACKNOWLEDGMENTS

The authors would like to thank Jingyan Fu, who provided great help at the beginning of this work.

REFERENCES

[1] L. B. Kish, “End of moore’s law: thermal (noise) death of integration in micro and nano electronics,” *Physics Letters A*, vol. 305, no. 3, pp. 144–149, 2002.

[2] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Team: Threshold adaptive memristor model,” *IEEE transactions on circuits and systems I: regular papers*, vol. 60, no. 1, pp. 211–221, 2012.

[3] 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, pp. 10–24, 2022.

[4] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams, “Sub-nanosecond switching of a tantalum oxide memristor,” *Nanotechnology*, vol. 22, no. 48, p. 485203, 2011.

[5] B. J. Choi, A. C. Torrezan, J. P. Strachan, P. Kotula, A. Lohn, M. J. Marinella, Z. Li, R. S. Williams, and J. J. Yang, “High-speed and low-energy nitride memristors,” *Advanced Functional Materials*, vol. 26, no. 29, pp. 5290–5296, 2016.

[6] B. Govoreanu, G. S. Kar, Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve *et al.*, “10×10nm 2 Hf/HfO_x crossbar resistive ram with excellent performance, reliability and low-energy operation,” in *International Electron Devices Meeting*, 2011, pp. 31.6.1–31.6.4.

[7] S. C. Bartling, S. Khanna, M. P. Clinton, S. R. Summerfelt, J. A. Rodriguez, and H. P. McAdams, “An 8mhz 75μa/mhz zero-leakage non-volatile logic-based cortex-m0 mcu soc exhibiting 100% digital state retention at vdd = 0v with <400ns wakeup and sleep transitions,” in *International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 432–433.

[8] 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.

[9] N. Sakimura, Y. Tsuji, R. Nebashi, H. Honjo, A. Morioka, K. Ishihara, K. Kinoshita, S. Fukami, S. Miura, N. Kasai *et al.*, “10.5 a 90nm 20mhz fully nonvolatile microcontroller for standby-power-critical applications,” in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 184–185.

[10] P.-F. Chiu, M.-F. Chang, C.-W. Wu, C.-H. Chuang, S.-S. Sheu, Y.-S. Chen, and M.-J. Tsai, “Low store energy, low vddmin, 8t2r nonvolatile latch and sram with vertical-stacked resistive memory (memristor) devices for low power mobile applications,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 6, pp. 1483–1496, 2012.

[11] K.-H. Kim, S. Hyun Jo, S. Gaba, and W. Lu, “Nanoscale resistive memory with intrinsic diode characteristics and long endurance,” *Applied Physics Letters*, vol. 96, no. 5, pp. 05 310.1–53 106.3, 2010.

[12] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov *et al.*, "Memristor-cmos hybrid integrated circuits for reconfigurable logic," *Nano letters*, vol. 9, no. 10, pp. 3640–3645, 2009.

[13] 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, 2020.

[14] E. C. Apollos, S. A. Adeshina, and N. A. Nnanna, "Memristor-based cim architecture for big data era," in *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, 2019, pp. 1–6.

[15] G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015.

[16] G. W. Burr, P. Narayanan, R. M. Shelby, S. Sidler, I. Boybat, C. di Nolfo, and Y. Leblebici, "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Comparative performance analysis (accuracy, speed, and power)," in *2015 IEEE International Electron Devices Meeting (IEDM)*, 2015, pp. 4.4.1–4.4.4.

[17] S. Sidler, I. Boybat, R. M. Shelby, P. Narayanan, J. Jang, A. Fumarola, K. Moon, Y. Leblebici, H. Hwang, and G. W. Burr, "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Impact of conductance response," in *2016 46th European Solid-State Device Research Conference (ESSDERC)*, 2016, pp. 440–443.

[18] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *2011 International Electron Devices Meeting*, 2011, pp. 4.4.1–4.4.4.

[19] 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.

[20] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[21] 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.

[22] S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. Lee, B. R. Lee, B. H. Lee, and H. Hwang, "Neuromorphic speech systems using advanced reram-based synapse," in *2013 IEEE International Electron Devices Meeting*, 2013, pp. 25.6.1–25.6.4.

[23] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang, "Improved synaptic behavior under identical pulses using alox/hfo2 bilayer rram array for neuromorphic systems," *IEEE Electron Device Letters*, vol. 37, no. 8, pp. 994–997, 2016.

[24] 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.

[25] W. Wu, H. Wu, B. Gao, N. Deng, S. Yu, and H. Qian, "Improving analog switching in hfo x-based resistive memory with a thermal enhanced layer," *IEEE Electron Device Letters*, vol. 38, no. 8, pp. 1019–1022, 2017.

[26] Z. Wang, M. Yin, T. Zhang, Y. Cai, Y. Wang, Y. Yang, and R. Huang, "Engineering incremental resistive switching in tao x based memristors for brain-inspired computing," *Nanoscale*, vol. 8, no. 29, pp. 14 015–14 022, 2016.

[27] J.-H. Bae, S. Lim, B.-G. Park, and J.-H. Lee, "High-density and near-linear synaptic device based on a reconfigurable gated schottky diode," *IEEE Electron Device Letters*, vol. 38, no. 8, pp. 1153–1156, 2017.

[28] B. Chen, J. Kang, P. Huang, Y. Deng, B. Gao, R. Liu, F. Zhang, L. Liu, X. Liu, X. Tran *et al.*, "Multi-level resistive switching characteristics correlated with microscopic filament geometry in tmo-rram," in *2013 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*. IEEE, 2013, pp. 1–2.

[29] X. Xu, H. Lv, Y. Li, H. Liu, M. Wang, Q. Liu, S. Long, and M. Liu, "Degradation of gate voltage controlled multilevel storage in one transistor one resistor electrochemical metallization cell," *IEEE Electron Device Letters*, vol. 36, no. 6, pp. 555–557, 2015.

[30] H. Kim, M. P. Sah, C. Yang, and L. O. Chua, "Memristor-based multilevel memory," in *2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010)*, 2010, pp. 1–6.

[31] A. Bagheri-Soulla and M. Ghaznavi-Ghoushchi, "A high-precision time-domain rram state control approach," *Microelectronics journal*, vol. 74, pp. 94–105, 2018.

[32] 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.

[33] I.-T. Wang, C.-C. Chang, L.-W. Chiu, T. Chou, and T.-H. Hou, "3d ta/taox/tio2/ti synaptic array and linearity tuning of weight update for hardware neural network applications," *Nanotechnology*, vol. 27, no. 36, p. 365204, 2016.

[34] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature electronics*, vol. 1, no. 1, pp. 52–59, 2018.

[35] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[36] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang *et al.*, "Memristor-based analog computation and neural network classification with a dot product engine," *Advanced Materials*, vol. 30, no. 9, p. 1705914, 2018.

[37] C. Wang, Z. Si, X. Jiang, A. Malik, Y. Pan, S. Stathopoulos, A. Serb, S. Wang, T. Prodromakis, and C. Papavassiliou, "Multi-state memristors and their applications: An overview," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 1–1, 2022.

[38] M. M. A. Chawa, R. Picos, and R. Tetzlaff, "A compact memristor model for neuromorphic reram devices in flux-charge space," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 9, pp. 3631–3641, 2021.

[39] 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*, 2022.

[40] S. Yu, W. Shim, X. Peng, and Y. Luo, "Rram for compute-in-memory: From inference to training," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2753–2765, 2021.

[41] 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.

[42] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[43] F. Zahoor, T. Z. Azni Zulkifli, and F. A. Khanday, "Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications," *Nanoscale research letters*, vol. 15, no. 1, pp. 1–26, 2020.

[44] 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.

[45] C. Baeumer, R. Valenta, C. Schmitz, A. Locatelli, T. O. Menteş, S. P. Rogers, A. Sala, N. Raab, S. Nemsak, M. Shim, C. M. Schneider, S. Menzel, R. Waser, and R. Dittmann, "Subfilamentary networks cause cycle-to-cycle variability in memristive devices," *ACS Nano*, vol. 11, no. 7, pp. 6921–6929, 2017, pMID: 28661649.

[46] L. Xia, M. Liu, X. Ning, K. Chakrabarty, and Y. Wang, "Fault-tolerant training with on-line fault detection for rram-based neural computing systems," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1–6.

[47] 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.

[48] Y. Cassuto, S. Kvatinsky, and E. Yaakobi, "Information-theoretic sneak-path mitigation in memristor crossbar arrays," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 4801–4813, 2016.

[49] J. H. Nickel, J. P. Strachan, M. D. Pickett, C. T. Schamp, J. J. Yang, J. A. Graham, and R. S. Williams, "Memristor structures for high scalability: Non-linear and symmetric devices utilizing fabrication friendly materials and processes," *Microelectronic engineering*, vol. 103, pp. 66–69, 2013.

Saleh Ahmad Khan (Student Member, IEEE) received the BS degree in Electrical and Electronics Engineering from American International University Bangladesh, Bangladesh in 2019. He is currently pursuing his MS degree at University of South Alabama, Mobile, AL, USA. He received Dean's Award from American International University, Bangladesh for best undergraduate capstone project. His research focuses on AI Hardware Design and Neuromrophic Computing.

Md Oli-Uz-Zaman (Student Member, IEEE) received the BS degree in Electronics and Telecommunication Engineering from Rajshahi University of Engineering and Technology (RUET), Bangledesh, in 2016. He is currently pursuing his PhD degree at University of South Alabama, Mobile, AL, USA. His research focuses on AI Hardware design and Neuromrophic Computing.

Jinhui Wang (Senior Member, IEEE) received the BE degree from Hebei University, Hebei, China, and the PhD degree from Beijing University of Technology, Beijing, China, all in Electrical Engineering. He was a Postdoc Fellow at University of Rochester, NY, USA, a Visiting Professor at the State University of New York at Buffalo, NY, USA, and a Visiting Scholar at IMEC, Leuven, Belgium. He is currently an Associate Professor with the Department of Electrical and Computer Engineering at the University of South Alabama, Mobile, AL, USA. His research interests include neuromorphic computing, AI technology, low-power, high-performance, and variation-tolerant IC design, 3D IC, and thermal solution in VLSI. He has published over 160 refereed journal/conference papers and 31 patents in the area of emerging semiconductor technologies. His previous work have received the Best Paper Award/Nomination at DATE 2021, ISVLSI 2019, ISLPED 2016, ISQED 2016, and EIT 2016.