Adaptive Quantization as a Device-Algorithm Co-Design Approach to Improve the Performance of In-Memory Unsupervised Learning With SNNs

Yuhan Shi, Zhisheng Huang, Sangheon Oh, Nathan Kaslan, Jungwoo Song, and Duygu Kuzum

***Abstract*—Off-chip memory access is the primary bottleneck toward accelerating neural network operations and reducing energy consumption. In-memory training and computation using emerging nonvolatile memories (eNVMs) have been proposed to address this problem. However, a small number of conductance states limit in-memory online learning performance. Here, we introduce a device-algorithm co-design approach and its application to phase change memory (PCM) for improving learning accuracy. We present an adaptive quantization method, which compensates the accuracy loss due to limited conductance levels and enables high-accuracy unsupervised learning with low-precision eNVM devices. We develop a spiking neural network framework for NeuroSim platform to compare online learning performance of PCM arrays for analog and digital implementations and benchmark the tradeoffs in energy consumption, latency, and area.**

***Index Terms*—Emerging nonvolatile memory (eNVM), MNIST digit classification, phase change memory (PCM), quantization, unsupervised learning.**

I. Introduction

Neural networks (NNs) have revolutionized artificial intelligence (AI) and led to remarkable advances across diverse applications. However, a high level of parallelism required by NN operations necessitates continuous shuffling of a massive amount of NN parameters between memory and processor. This causes substantial computing power and time for conventional von Neumann-based computation systems such as CPUs/GPUs [1]. To eliminate delay and power consumption problems due to data transfer between CPU and memory, in-memory training and computing using emerging nonvolatile memory (eNVM) devices have been identified as a promising non-von Neumann approach [2]–[4]. Advances in new materials and eNVM devices offer new approaches to very low-energy computing with scalable devices [5]–[8]. Mapping NN training to eNVM arrays requires quantization of weight values into discrete conductance levels. Unfortunately, most of the synaptic devices demonstrated so far have limited conductance levels and cannot represent NN weights in ideal high precision (64-bit) as shown in Fig. 1.

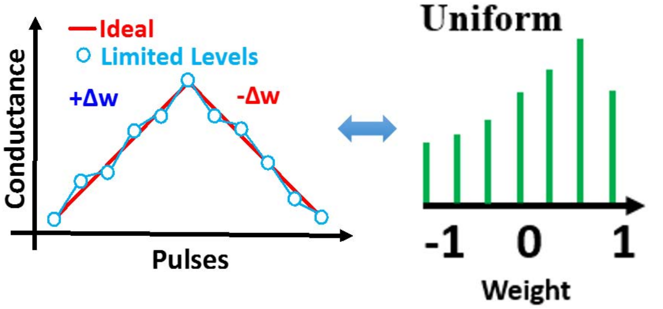

Fig. 1. Illustration of limited conductance levels of the device. Uniform quantization maps weights to conductance changes linearly.

All previous demonstrations of learning with synaptic arrays have adopted uniform quantization, which maps continuous NN weights into discrete device conductance values uniformly, leading to a steep decrease in accuracy for precisions less than 6-bits [5]–[8] for online learning. Various different quantization schemes have been proposed in [9]–[11]. However, this paper mainly focuses on the development of adaptive quantization to be applied to the eNVM devices. To overcome this accuracy degradation, we propose an adaptive quantization technique, which maps NN weights to the hardware conductances based on the distribution and importance of the weights. We apply this device-algorithm co-design approach to phase change memory (PCM) synapses for online unsupervised learning with a spiking neural network (SNN). SNNs allow sparse and event-driven parameter updates for energy-efficient

Manuscript received November 14, 2018; revised January 11, 2019 and February 4, 2019; accepted February 6, 2019. Date of publication February 26, 2019; date of current version March 22, 2019. This work was supported in part by the Office of Naval Research under Grant N000141612531, in part by the National Science Foundation under Grant ECCS-1752241 and Grant ECCS-1734940, and in part by Samsung Electronics. The review of this paper was arranged by Editor P.-Y. Du. *(Corresponding author: Duygu Kuzum.)*

Y. Shi, Z. Huang, S. Oh, N. Kaslan, and D. Kuzum are with the Electrical and Computer Engineering Department, University of California at San Diego, San Diego, CA 92093 USA (e-mail: dkuzum@eng.ucsd.edu).

J. Song is with DRAM PA Team, Memory Business, Samsung Electronics Company Ltd., Hwaseong 18448, South Korea.

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

Digital Object Identifier 10.1109/TED.2019.2898402

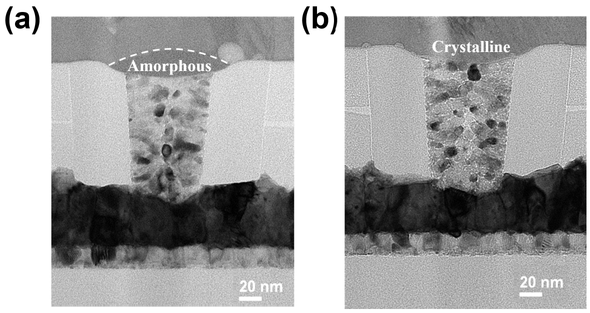

Fig. 2. Cross-sectional TEM image of an electronic synapse made of GST. (a) Low-conductance amorphous state. (b) High-conductance poly-crystalline state.

implementation of online learning in hardware. Therefore, SNNs have been widely explored for neuromorphic circuits in the past. In addition, SNNs are particularly suitable for unsupervised learning using unlabeled data, offering complementary skills to widely-adopted artificial NNs using supervised learning based on back-propagation.

In this paper, we investigate adaptive quantization methodologies to train SNNs with low-bit precision synaptic devices for online learning in hardware. We also study the impact of adaptive quantization on abruptness and asymmetry of device conductance. Then, we develop an SNN framework for NeuroSim, which is an integrated device-to-algorithm simulator (SNN + NeuroSim). Using SNN + NeuroSim, we explore the system-level performance of the implementation of unsupervised learning with PCM arrays for analog and digital architectures in various technology nodes.

II. PCM Characterization

In this paper, we use $Ge_2Sb_2Te_5$ (GST), a phase change material, to implement PCM-based synaptic devices. A 200-nm-thick GST is deposited between a bottom electrode (75 nm diameter) and a top electrode. Cross-sectional TEM images of a PCM synaptic device programmed into low-conductance (7 V, 50 ns) and high-conductance (1.2 V, 50 ns) states are shown in Fig. 2(a) and (b). At high-conductance state, the GST is poly-crystalline. At low-conductance state, an amorphous cap starts to form at the bottom electrode interface, determining the resistance of the PCM device.

We investigate the gradual programing in PCM synapses. When identical amplitude pulses (2 V, 50 ns) are used, PCM synapses exhibit gradual programing only for conductance increase [Fig. 3(a)]. Fig. 3(b) and (c) show the gradual conductance change of our PCM device in both high- and low-conductance (G) regimes. To achieve both gradual set and reset in our PCM device, we need to apply pulses with increasing amplitude. In high-G regime [Fig. 3(b)], the gradual set (increasing conductance) programing of the PCM devices is achieved by using staircase pulses (20 pulses per each voltage step of 0.1 V starting from 0.5 to 0.9 V), and gradual reset (decreasing conductance) is achieved using pulses with increasing amplitude from 2 to 4 V with 20-mV-voltage steps. In low-G regime [Fig. 3(c)], the gradual set is performed by staircase pulses with an increasing step of 50 mV in the range of 1–1.7 V (four pulses for each step), and gradual reset is performed by pulses with increasing amplitude from 5.7 to 7.3 V with 25-mV-voltage steps. The current for gradual set ranges from 0.04 to 0.25 mA, and the current for gradual reset ranges from 2 to 2.4 mA in low-G regime. The 0.1-V and 40-ns pulse is used to read the device conductance. We plot the callout window in Fig. 3(c) to clearly show the abruptness during gradual reset. If we implement large synaptic core array, IR drop across metal lines can affect the accuracy. To avoid the accuracy drop due to the IR drop, ON-resistance of memory cell needs to be higher than 10 kΩ for online learning case [12]. Since ON-state resistance of high-G regime is 5 kΩ, we use low-G PCM data (ON-state: 200 kΩ) for in-memory NN training in this paper. Our PCM device exhibits ~55 levels for gradual conductance increase and decrease, corresponding to ~6-bit precision.

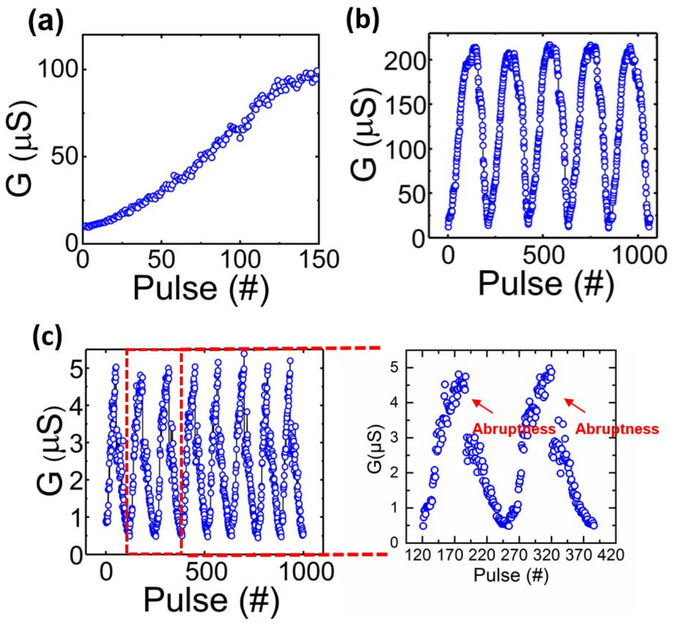

Fig. 3. (a) Measured conductance increase programmed by the same amplitude pulses. We use pulses with 50 ns of pulsewidth, 5 ns of rise time, and 5 ns of fall time. (b) and (c) Gradual switching characteristics of the device in high-G and low-G regimes, respectively. For both high-G and low-G regimes, we use pulses with 10 ns of pulsewidth, 5 ns of rise time, and 5 ns of fall time for gradual set and 20 ns of pulsewidth, 5 ns of rise time, and 5 ns of fall time for gradual reset. The callout window in (c) shows the abrupt conductance change during gradual reset.

III. Neural Network Model

SNNs have been extensively investigated by the neuromorphic circuits community since they offer sparse and event-driven parameter updates for energy-efficient implementation of online learning in hardware [13]. In this paper, we use an SNN model to investigate unsupervised online learning with PCM arrays. Our SNN model for unsupervised learning is summarized in Fig. 4(a) and (b). For the training, synaptic weights are updated using a timing- and weight-based learning rule [Fig. 4(c) and (d)]. The iterative training cycle consists of first converting all input digits to Poisson prespike trains, computing the membrane potentials for the output layer, generating a postspike using a probabilistic firing mechanism, and finally updating the synaptic weights using

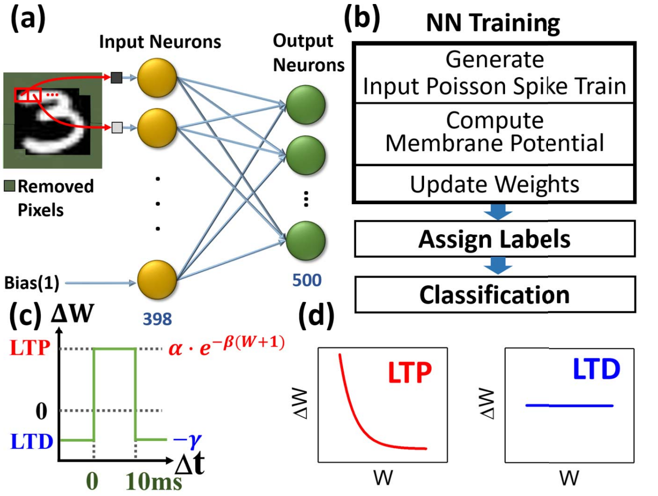

Fig. 4. (a) SNN architecture with fully connected structure. Each pixel of an MNIST image is corresponding to one of the input neurons. The number of output layer neurons ranges from 100 to 500. (b) Algorithm used for training of the SNN. (c) Simplified STDP rule used for SNN. (d) LTP update is an exponential decaying function that depends on the current weight, and the LTD update is a constant.

the simplified spike-timing-dependent plasticity (STDP) rule shown in Fig. 4(c) and (d).

According to this rule, if the time difference between the postspike and prespike is within a 10-ms window, the synaptic weight is increased by ΔW_{LTP} according to the long-term potentiation (LTP) rule in (1). Otherwise, the synaptic weight is decreased by ΔW_{LTD} using the long-term depression (LTD) rule in (2)

$$\Delta W_{\text{LTP}} = \alpha \times \exp(-\beta(W+1)) \tag{1}$$
$$\Delta W_{\text{LTD}} = -\gamma. \tag{2}$$

The parameters α and β control the LTP strength. W is the current weight value. The parameter γ determines the depression scale. The network is trained in an unsupervised fashion with 60000 MNIST digits. After training is done, we assign labels to the output neurons and perform inference with the MNIST test set of 10000 handwritten digits. The classification accuracy for our SNN is 94.05% for an ideal 64-bit floating point. This accuracy is already high for unsupervised learning and can be further increased up to 98.17% if supervision is introduced into the SNN [13].

There are two ways to use our PCM devices for implementing online training of our SNN model. First, since our device can only achieve the gradual SET using identical pulses [Fig. 3(a)], we can use 2-PCM configuration [14] for online training. An alternative way is to use the device characteristics shown in Fig. 3(c). Since we use nonidentical pulses for gradual switching of the devices, an additional read step is required before updating the weights in hardware as suggested by Chen *et al.* [15]. Fig. 5 shows how this test scheme can be applied to online training. Before the weight update, we read the device conductance from the PCM array. The peripheral neuron circuit then calculates the weight update (ΔW), which is converted into ΔG to calculate the number of programing pulses (ΔP) and amplitudes based on ΔG and the current conductance state. Finally, the programing circuitry will apply the pulses to update the conductance of the PCM devices in the array.

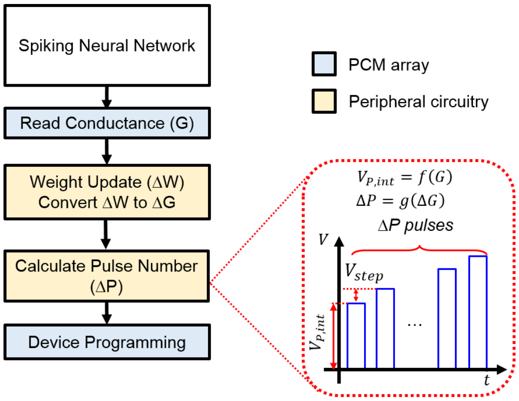

Fig. 5. Schematic of online training using nonidentical pulse scheme for SNN.

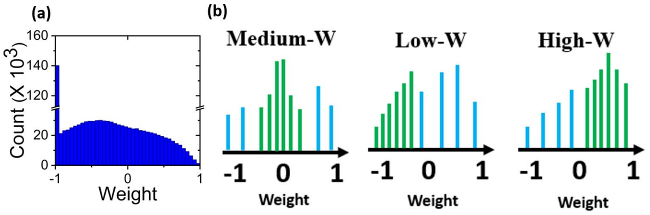

Fig. 6. (a) Weight distribution during the training of the first 5000 samples. (b) Illustrations of medium-*W*, low-*W*, and high-*W* quantization.

IV. Adaptive Quantization for Low-Precision Synapses

Although low-precision weights can be used for inference, online learning requires a high-precision representation of weights to achieve high accuracy [12]. Therefore, mapping network training to eNVM arrays requires the quantization of weight values with high precision. Uniform quantization ignores the distribution and evolution of weights during training and treats all the weights with equal importance. However, every weight does not equally contribute to learning outcome and hence, unimportant weights do not require high precision. To address that, we develop adaptive quantization for quantizing weights based on their distribution during training using Lloyd maximum quantization [16].

To train an adaptive quantizer, we use the evolution of weights in the first 5000 training samples [Fig. 6(a)]. We investigate medium-W, low-W, and high-W quantizers [Fig. 6(b)], allocating more levels to intermediate, negative, and positive weights, respectively. Fig. 7(a) and (b) show the weight visualizations of output neurons for ideal 64-bit software simulation and 4-bit low-W quantizer, respectively, as representative examples.

We also explore the performance of different quantizers for training SNN using PCM data [Fig. 3(c)] with lower precision. To implement adaptive quantization with the PCM data, we first choose the number of quantized levels to distribute in positive ([0, 1]) and negative ([−1, 0]) regions according to the bit precision and the type of quantizer. Then, we use

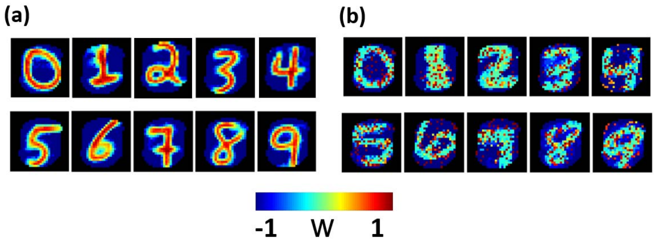

Fig. 7. Weight visualizations of ten representative output neurons at the end of training. (a) Ideal software 64-bit. (b) Low-*W* 4-bit adaptive quantization.

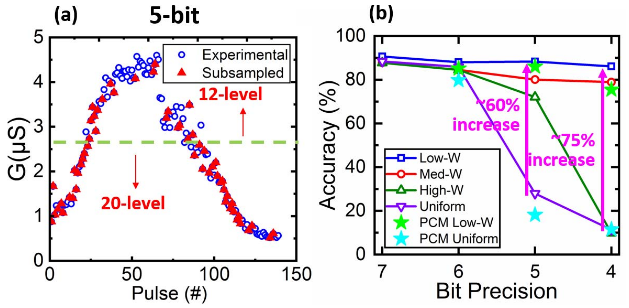

Fig. 8. (a) PCM gradual programing data (blue circle) is subsampled (red triangle) to 32 levels (5-bit) to perform adaptive quantization. (b) Comparison of quantization methods. Low-*W* achieves ~60% and ~75% increase in 5-bit and 4-bit over uniform. Lines represent accuracies of different quantization techniques without using device data. Stars represent accuracies of different quantization techniques using PCM device data from Fig. 3(c).

Lloyd–Max quantization [16] to obtain quantization intervals and their corresponding quantization values for the quantizers. Since our NN weights range from -1 to 1 while the device conductance data range from 0.4 to 5.5 μS, we use a linear transformation to map the quantized weight (W_{quan}) to the closest PCM conductance values in the following equation:

$$G = W_{\text{quan}} \frac{(G_{\max} - G_{\min})}{2} + \frac{(G_{\max} + G_{\min})}{2}. \tag{3}$$

Fig. 8(a) shows that quantized weight levels are mapped to PCM conductance values for low-*W* quantization as a representative example. PCM gradual programing data from Fig. 3(c) are subsampled to 32 levels (5-bit quantization). For low-*W* quantization, a larger number of levels were allocated to low-conductance values. Similar mappings are performed for uniform, medium-*W*, and high-*W* quantizers. Fig. 8(b) shows theoretically simulated classification accuracies of different quantization techniques without using device data, shown by solid lines. Fig. 8(b) also includes adaptive quantization applied directly to PCM data, shown by star symbols. Low-*W* adaptive quantization boosts classification accuracy by ~60% for 5-bit and ~75% for 4-bit precisions.

Table I summarizes the performance of 5-bit adaptive quantization against 5-bit uniform quantization and ideal 64-bit software simulation. A 5-bit (A.Q.) is the software simulation result based on the 5-bit adaptive quantization without using the device data, and PCM 5-bit (A.Q.) directly uses subsampled device conductance from Fig. 8(a) to perform adaptive quantization. Our results suggest that adaptive quantization can enable the use of eNVM devices with limited conductance levels.

TABLE I
CLASSIFICATION ACCURACY FOR DIFFERENT QUANTIZATION SCHEMES AND PCM DATA

Precision	Accuracy
64-bit	94.05 %
5-bit	27.96 %
5-bit (A.Q.)	88.31 %
PCM 5-bit (A.Q.)	86.05 %

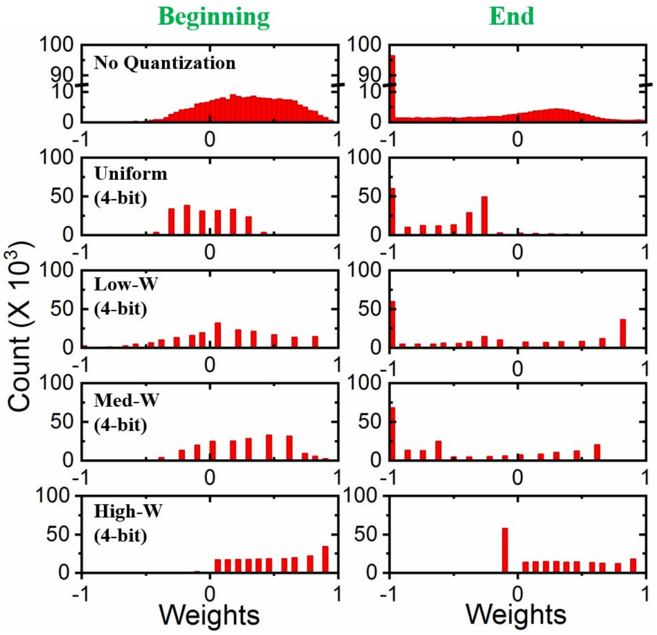

Fig. 9. Weights at the beginning and the end of training of no quantization (64-bit), 4-bit uniform, low-*W*, medium-*W*, and high-*W* quantization.

To better understand the effect of different adaptive quantizations on the weight development, we plot weight distribution at the beginning (after presenting 100th sample to the network) and the end of the training for no quantization (64-bit) along with all four quantizers (4-bit) in Fig. 9. To achieve high accuracy, the distribution of the trained weights should well represent the input features of MNIST digits. In the MNIST case, the distribution of the trained weights can be divided into two distinct parts, namely, the foreground pixels (green, yellow, and red; positive weights [0, 1]) and background pixels (blue; negative weights [-1, 0]) as shown in Fig. 7(a). Therefore, both positive and negative weights are important for creating a contrast between foreground and background pixels. As shown in Fig. 9, no quantization represents both foreground and background pixels very well by distributing the weights in [-1, 1]. Among the four quantizers, low-*W* and medium-*W* adaptive quantization have weights distributed in a similar range with no quantization case. Moreover, compared to medium-*W* quantization, low-*W* quantization has more

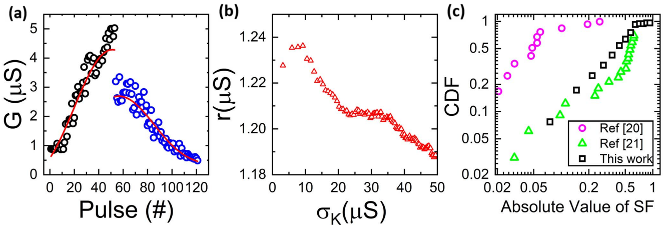

Fig. 10. (a) Noise-free curve of our device data [Fig. 3(c)] using GPR method. (b) *r* against varying σ_K values for our device. *r* represents the absolute difference between predicted and observed *G* values [17]. (c) Plot of cumulative distribution function of absolute values of SF for our data and device data from [20] and [21].

positive weights and its maximum weight value is closer to 1. This indicates that low-W provides a better contrast between foreground and background pixels than medium-W. Therefore, medium-W shows a slightly lower accuracy than low-W [Fig. 8(b)] while low-W achieves a more accurate representation of input features than other quantizers and shows the highest accuracy [Fig. 8(b) and Table I]. However, for uniform and high-W quantization, the weights get stuck at the negative range or positive range and do not develop properly during training. Hence, the accuracy with uniform or high-W quantizations is lower than low-W and medium-W. It is important to note that the choice of different quantizers depends on the network algorithm.

We have shown that adaptive quantization can effectively boost the accuracy of low-precision devices. Furthermore, we investigate its effects on the abruptness of conductance change and asymmetry of weight update. Abruptness and asymmetry are the two nonideal effects, which could be impacted by different adaptive quantization schemes. Other nonideal characteristics such as nonlinearity and variation of PCM devices have been extensively studied in [12], [14], [15], [17], and [18], previously.

First, we investigate the effectiveness of adaptive quantization on the abruptness of conductance change. As shown in Fig. 8(b), PCM uniform quantization (cyan stars) performs slightly worse than theoretically simulated uniform quantization (purple line) because there are no conductance levels to represent weights in LTD part due to the abruptness [Fig. 3(c) callout window]. However, PCM low-W quantization (green stars) achieves reasonable accuracy in low bit precision and suffers less from the abruptness. This suggests that the low-W quantization could help to mitigate the effect of the abruptness in device conductance on accuracy.

In addition to abruptness, symmetry of the weight update is another important consideration of the online training [17], [19]. Here, we characterize the symmetry of our device using the Gaussian process regression (GPR) method presented in [17]. We extract the noise-free curve for our PCM data [Fig. 3(c)] as shown in Fig. 10(a). We vary the σ_K value in the range between 0 and 50 [Fig. 10(b)] to find the optimum value for GPR fitting ($\sigma_K = 31.6$). Based on the fitting, we then characterize the symmetry factor (SF) of our device. SF of our device is presented along with the SF of device from [20] and [21] in Fig. 10(c). The device data from [20] show good switching symmetry according to the symmetry requirement specified in [19]. Our device is less symmetric than the device from [20] but more symmetric than the device from [21]. Therefore, we use the most asymmetric device [21] to investigate the impact of adaptive quantization on accuracy. We incorporate this device data (~6-bit) directly into our simulation. In addition, we implement 6-bit low-W quantization based on this data. Table II shows that the low-W quantization improves the accuracy to 81.13% while the device data only show 64.39% accuracy due to the asymmetry of the weight update. These results suggest that adaptive quantization could be helpful to improve the accuracy for the devices that exhibit the asymmetric weight update.

TABLE II
Classification Accuracy for Asymmetric Device

Precision	Accuracy
Device [21] (~6-bit)	64.39 %
Device [21] 6-bit (A.Q.)	81.13 %

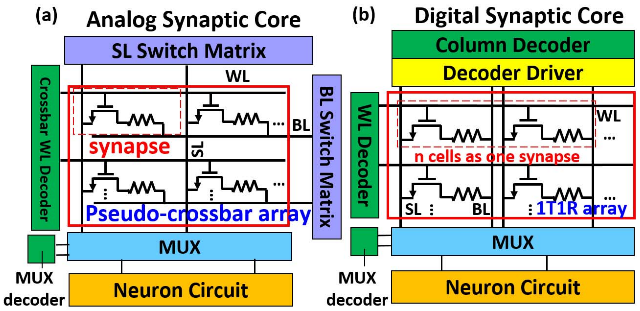

Fig. 11. (a) Analog synaptic core with pseudo-crossbar 1T1R array and peripheral circuitry. Each eNVM represents one synapse. (b) Digital synaptic core consists of 1T1R eNVM array with peripheral circuitry. *n* eNVM cells represent one synapse.

V. Circuit-Level Performance Benchmark

In order to investigate performance gains as a result of adaptive quantization, we develop an SNN framework for NeuroSim [12]. NeuroSim is a C++ based simulator with hierarchical organization starting from the experimental device data and extending to array architectures with peripheral circuit modules and algorithm level NN models. SNN + NeuroSim can simulate circuit-level performance metrics (energy, area, latency, and leakage power) at runtime of online learning, while providing instruction-accurate classification accuracy for the SNN using experimental PCM data. For the implementation of NN training with PCM arrays, synaptic weights can be represented in either analog formats or binary (digital). For an analog implementation, the cells can be arranged into a pseudo-crossbar array and synaptic weights are stored in the form of multilevel conductances [Fig. 11(a)]. For a digital implementation, *n* binary 1T1R cells are grouped to represent one synaptic weight [Fig. 11(b)] and each cell is programmed to high- or low-conductance states.

We apply adaptive quantization to both analog and digital approaches to reduce bit precision for in-memory online learning. With SNN + NeuroSim, we simulate analog synaptic

TABLE III
SUMMARY OF BENCHMARK RESULTS OF PCM DEVICE DATA AND DEVICE/ALGORITHIM CO-DESIGN FOR ANALOG AND DIGITAL ARCHITECTURES (14 nm)

	PCM Device Data (Fig. 3c)		**Device/Algorithm Co-design**	
	Analog	**Digital**	**Analog 4-bit (A.Q.#)**	**Digital 4-bit(A.Q.#)**
Bit precision	50 levels (~6)	6	16 levels (~4)	4
R_{on}	200kΩ	200kΩ	200kΩ	200kΩ
ON/OFF ratio	10	10	10	10
LTP pulse	1V-1.7V/10ns	1.2V/50ns	1V-1.7V/10ns	1.2V/50ns
LTD pulse	5.7V-7.3V/20ns	7V/50ns	5.7V-7.3V/20ns	7V/50ns
Accuracy*	85.12%	85.87%	86.11%	86.11%
Area(μm^2)	2990	9420.94	2990	6420
Latency* (s)	**3.29**	12.2	**0.22**	1.32
Energy* (mJ)	5.36	**2.97**	2.49	**1.89**
Leakage Power (μW)	53.8	54.1	53.8	49.3

*for 3 training epochs (60k images/epoch) # A.Q.: adaptive quantization.

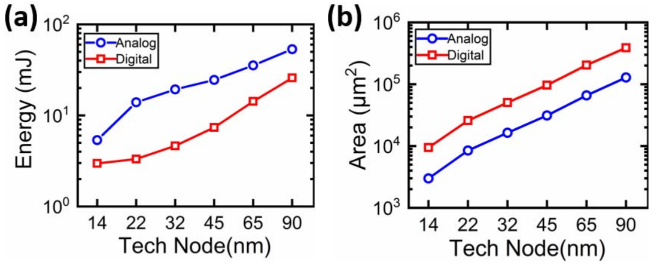

Fig. 12. (a) Energy consumption and (b) chip area versus technology node (nm) for analog and digital synaptic cores.

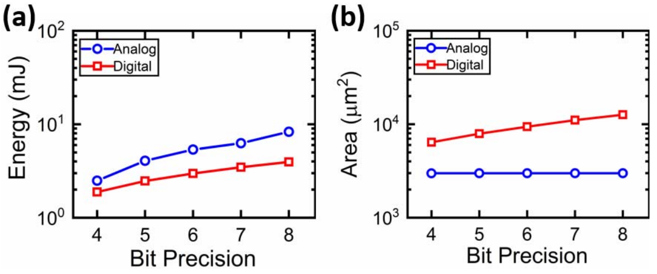

Fig. 13. (a) Energy consumption and (b) chip area versus bit precision for analog and digital synaptic cores.

core [Fig. 11(a)] mapping network weights into discrete conductance levels of the PCM device data [Fig. 3(c)] and digital synaptic core [Fig. 11(b)] using binary states of memory cells. Figs. 12 and 13 show total energy consumption and chip area for analog and digital architectures as a function of technology node and bit precision, respectively. Note that, the technology node used in our simulation refers to the transistors of the peripheral circuit.

The analog implementation consumes more energy than the digital [Figs. 12(a) and 13(a)] mainly due to the voltage levels used in the write operation of pseudo-crossbar array [Fig. 11(a)] [12]. On the other hand, analog implementation always occupies less chip area than digital implementation because a smaller number of devices are used [Figs. 12(b) and 13(b)]. To make a fair comparison between analog and digital synaptic core, we plot energy versus area (under different technology nodes) for both cases in Fig. 14. As can be seen in Fig. 14, analog occupies less area while consumes more energy than digital. As shown in Fig. 12(a) and (b), energy and area increase with the technology node since transistors for larger technology require higher V_{dd} and larger area. Fig. 13(a) shows that energy consumption continues increasing as the bit precision increases, indicating that it is critical to reduce bit precision to significantly improve the energy efficiency. Fig. 13(b) shows that the total neurosynaptic core area does not change for the analog implementation with different bit precisions since single devices are used for all cases. On the other hand, the use of higher bit precision for the digital case increases the chip area. Therefore, adaptive quantization can help to reduce bit precision while substantially decreasing energy consumption, chip area, and latency.

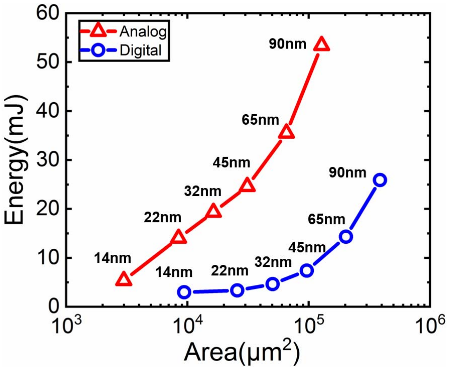

Fig. 14. Energy versus area (under different technology nodes: 14, 22, 32, 45, 65, and 90 nm) for analog and digital synaptic cores.

Table III summarizes the benchmarking results for online learning with SNN for analog and digital architectures using PCM device data (left two columns) and device-algorithm co-design approach using adaptive quantization (right two columns). The best performance metrics are highlighted in yellow and blue. Our device-algorithm co-design approach applies 4-bit low-W quantizers, which allocate more levels for negative weights. We use the simulation results for 14-nm technology node in Table III. As can be seen in analog (first column) and analog 4-bit (third column) cases in Table III, adaptive quantization allows the use of 16 conductance levels to reduce energy and latency while achieving better accuracy (86.11%). As shown in digital (second column) and digital 4-bit (fourth column) cases in Table III, 4-bit precision enabled by adaptive quantization achieves a ~tenfold decrease in latency (red dashed boxes), while also decreasing the energy consumption and chip area, and providing a higher classification accuracy. For both PCM device data and device/algorithm co-design cases, our benchmarking results suggest that analog implementation provides better latency than the digital while digital has lower energy consumption. However, it is important to note that the use of analog or digital implementation to achieve the best performance strongly depends on the device characteristics and programing pulse parameters. Adaptive quantization enables both lower energy and shorter latency. Particularly for digital implementation, adaptive quantization provides a substantial decrease in latency by enabling 4-bit precision.

VI. Conclusion

This paper demonstrated that accuracy loss due to limited conductance levels can be compensated by adaptive quantization. We also showed that abruptness and asymmetry in device conductance can be mitigated by the adaptive quantization. Benchmarking results with our SNN + NeuroSim platform showed that digital PCM architecture achieves lower energy consumption than the analog one, while the analog PCM is preferred for smaller chip area and lower latency. Our device-algorithm co-design solutions suggested that energy consumption, chip area, and latency can be significantly reduced by lowering bit precision with adaptive quantization and engineering the eNVM characteristics.

References

[1] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[2] S. B. Eryilmaz *et al.*, “Experimental demonstration of array-level learning with phase change synaptic devices,” in *IEDM Tech. Dig.*, vol. 4, Dec. 2013, pp. 25.5.1–25.5.4.

[3] D. Kuzum, R. G. Jeyasingh, and H.-S. P. Wong, “Energy efficient programming of nanoelectronic synaptic devices for large-scale implementation of associative and temporal sequence learning,” in *IEDM Tech. Dig.*, vol. 4, Dec. 2011, pp. 30.3.1–30.3.4.

[4] Y. Jeong and W. Lu, “Neuromorphic computing using memristor crossbar networks: A focus on bio-inspired approaches,” *IEEE Nanotechnol. Mag.*, vol. 12, no. 3, pp. 6–18, Sep. 2018.

[5] R. Ge *et al.*, “Atomristor: Nonvolatile resistance switching in atomic sheets of transition metal dichalcogenides,” *Nano Lett.*, vol. 18, no. 1, pp. 434–441, Dec. 2017.

[6] M. Kim *et al.*, “Zero-static power radio-frequency switches based on $\mathrm{MoS_2}$ atomristors,” *Nature Commun.*, vol. 9, no. 1, Jun. 2018, Art. no. 2524.

[7] D.-H. Kang *et al.*, “Poly-4-vinylphenol (PVP) and poly(melamine-co-formaldehyde) (PMF)-based atomic switching device and its application to logic gate circuits with low operating voltage,” *ACS Appl. Mater. Interfaces*, vol. 9, no. 32, pp. 27073–27082, Aug. 2017.

[8] S. Lashkare, N. Panwar, P. Kumbhare, B. Das, and U. Ganguly, “PCMO-based RRAM and NPN bipolar selector as synapse for energy efficient STDP,” *IEEE Electron Device Lett.*, vol. 38, no. 9, pp. 1212–1215, Sep. 2017.

[9] C. Louizos, M. Reisser, T. Blankevoort, E. Gavves, and M. Welling, “Relaxed quantization for discretized neural networks,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, May 2019, pp. 1–15.

[10] E. Park, J. Ahn, and S. Yoo, “Weighted-entropy-based quantization for deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7197–7205.

[11] J. Wang, J. Lin, and Z. Wang, “Efficient hardware architectures for deep convolutional neural network,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 6, pp. 1941–1953, Nov. 2018.

[12] P.-Y. Chen, X. Peng, and S. Yu, “NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018.

[13] S. R. Kulkarni and B. Rajendran, “Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization,” *Neural Netw.*, vol. 103, pp. 118–127, Jul. 2018.

[14] G. W. Burr *et al.*, “Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element,” *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Nov. 2015.

[15] P.-Y. Chen *et al.*, “Mitigating effects of non-ideal synaptic device characteristics for on-chip learning,” in *Proc. IEEE/ACM IICCAD*, Nov. 2015, pp. 194–199.

[16] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[17] N. Gong *et al.*, “Signal and noise extraction from analog memory elements for neuromorphic computing,” *Nature Commun.*, vol. 9, no. 1, May 2018, Art. no. 2102.

[18] S. Oh, Y. Shi, X. Liu, J. Song, and D. Kuzum, “Drift-enhanced unsupervised learning of handwritten digits in spiking neural network with PCM synapses,” *IEEE Electron Device Lett.*, vol. 39, no. 11, pp. 1768–1771, Nov. 2018.

[19] T. Gokmen and Y. Vlasov, “Acceleration of deep neural network training with resistive cross-point devices: design considerations,” *Frontiers Neurosci.*, vol. 10, p. 333, Jul. 2016.

[20] J. Woo *et al.*, “Improved synaptic behavior under identical pulses using $\mathrm{AlO}_x/\mathrm{HfO_2}$ bilayer RRAM array for neuromorphic systems,” *IEEE Electron Device Lett.*, vol. 37, no. 8, pp. 994–997, Aug. 2016.

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