Temperature-Resilient RRAM-Based In-Memory Computing for DNN Inference

Jian Meng, *Arizona State University, Tempe, AZ, 85287, USA*

Wonbo Shim, *Georgia Institute of Technology, Atlanta, GA, 30332, USA and also Seoul National University of Science and Technology, Nowon-gu, Seoul, 01811, South Korea*

Li Yang, Injune Yeo, and Deliang Fan, *Arizona State University, Tempe, AZ, 85287, USA*

Shimeng Yu, *Georgia Institute of Technology, Atlanta, GA, 30332, USA*

Jae-sun Seo, *Arizona State University, Tempe, AZ, 85287, USA*

Resistive random access memory (RRAM)-based in-memory computing (IMC) has emerged as a promising paradigm for efficient deep neural network (DNN) acceleration. However, the multibit RRAMs often suffer from nonideal characteristics such as drift and retention failure against temperature changes, leading to significant inference accuracy degradation. In this article, we present a new temperature-resilient RRAM-based IMC scheme for reliable DNN inference hardware. From a 90-nm RRAM prototype chip, we first measure the retention characteristics of multilevel HfO_2 RRAMs at various temperatures up to 120 °C, and then rigorously model the temperature-dependent RRAM retention behavior. We propose a novel and efficient DNN training/inference scheme along with the system-level hardware design to resolve the temperature-dependent retention issues with one-time DNN deployment. Employing the proposed scheme on a 256×256 RRAM array with the circuit-level benchmark simulator NeuroSim*, we demonstrate robust RRAM IMC-based DNN inference where $>$ 30% CIFAR-10 accuracy and $>$ 60% TinyImageNet accuracy are recovered against temperature variations.*

Deep neural networks (DNNs) have shown extraordinary performance in recent years for various applications,[1] including image classification, object detection, speech recognition, etc. Accuracy-driven DNN architectures tend to increase the model sizes and computations in a very fast pace, demanding a massive amount of hardware resources. Frequent communication between the processing engine and the on-/off-chip memory leads to high energy consumption, which becomes a bottleneck for the conventional DNN accelerator design.

To overcome such challenges, in-memory computing (IMC) has been proposed as a promising scheme for energy-efficient DNN acceleration. The weights are stored in the memory cells and the multiply-and-accumulate (MAC) operation is performed within the memory array by asserting multiple rows simultaneously. The weighted analog current is accumulated along the columns, which represents the MAC computation value and is subsequently digitized by analog-to-digital converter (ADC) circuits on the periphery.

Regarding the memory technologies for the IMC scheme, SRAM and DRAM are both volatile and suffer from the leakage power in the complementary metal–oxide–semiconductor (CMOS) devices. Such disadvantages promoted the nonvolatile memory (NVM) as an attractive solution for IMC-based DNN acceleration. Among different NVMs, RRAM devices can store

0272-1732 © 2021 IEEE
Digital Object Identifier 10.1109/MM.2021.3131114
Date of publication 13 December 2021; date of current version 1 February 2022.

multiple levels in one cell, resulting in dense storage as well as high MAC throughput.

On the other hand, having a high amount of computation in a small area can increase the power density, which can, in turn, elevate the temperature. Also, the temperature can be affected by the digital/analog modules adjacent to the RRAM-based IMC macros.

When the temperature increases, the ability to hold the programed values becomes weaker (the detailed information of the retention characteristics will be provided later), and the RRAM conductance starts to drift away. Such variation will affect the macro-level IMC results, layer-by-layer computations, and eventually the final output of the DNN, leading to incorrect inference predictions. Therefore, the RRAM-based DNN accelerator should have a more stringent retention requirement compared with the NVM memory storage.

To avoid the DNN accuracy loss caused by the conductance drifting, very frequent refresh operations will be required, but this introduces a large amount of additional energy consumption to the accelerator system. Therefore, alleviating the thermal retention issue becomes critical for energy-efficient RRAM-based IMC accelerator design.

Several prior works tackled such thermal issues from both algorithm and hardware perspectives. Temperature-aware refreshing techniques[2] were designed to adjust the refreshing frequency based on the operating temperature. Nevertheless, the proposed refresh operation requires the special RRAM architecture design. The on-device tuning algorithm[3] reduced the refreshing frequency by updating the conductance based on the saturation boundary of the RRAM device. However, acquiring the extra information of the device itself could be a burden for the control scheme and peripheral circuit design. In addition to the refreshing techniques, array-level column swapping techniques[4,5] change the RRAM mapping scheme by swapping the important weight from a higher temperature area with the cells from a lower temperature area. Nevertheless, manipulating the position of the deployed weight values is expensive for the hardware design. Furthermore, the reallocation cannot guarantee the reduction of the retention variations, and frequent refreshing may still be required after the swapping.

In Shin *et al.*'s work,[6] to cope with general noise in analog neural networks, the batch normalization (BN) parameters are recalibrated by using the exponential moving average (EMA) while injecting synthetic Gaussian noise to the weight. However, all DNN models in Shin *et al.*'s work[6] employ 32-bit floating-point precision, and the proposed algorithm/noise is not pertinent to any specific hardware. Furthermore, continuously calibrating the BN parameters during inference is expensive to implement in hardware.

Some prior works modeled the conductance variations as an additive Gaussian noise with zero mean and temperature-related standard deviations.[7,8] However, assuming the noise to be zero mean cannot precisely reflect the on-chip thermal variations. Our measurements show that both the mean and standard deviations of the conductance are changing with different thermal conditions.

CONSIDERING CRITICAL RETENTION FAILURE ISSUES AGAINST TEMPERATURE VARIATIONS AND THE LIMITATIONS OF THE PREVIOUS TECHNIQUES, WE PROPOSE A TEMPERATURE-RESILIENT SOLUTION, INCLUDING BOTH NEW DNN TRAINING ALGORITHMS AND SYSTEM-LEVEL HARDWARE DESIGN, FOR RRAM-BASED IN-MEMORY COMPUTING.

Considering critical retention failure issues against temperature variations and the limitations of the previous techniques, we propose a temperature-resilient solution, including both new DNN training algorithms and system-level hardware design, for RRAM-based in-memory computing. Against temperature-dependent RRAM variations over time, we propose a novel and simple training algorithm that consists of progressive knowledge distillation (PKD) training and thermal-aware batch normalization adaptation (BNA) that achieves high robustness with largely improved accuracy without introducing any complex refreshing or deployment schemes. We use the circuit-level simulator NeuroSim[9] to evaluate the system-level performance under the retention variations. The proposed design has been evaluated on a number of CNNs with different model sizes and activation/weight precision values for CIFAR-10 and TinyImageNet data sets, demonstrating significant accuracy improvements with elevated model robustness. Overall, the main contributions of this work are as follows:

- We conduct a practical and comprehensive analysis along with the rigorous modeling to investigate the retention failure based on the actual RRAM chip measurement.
- We provide a new DNN training algorithm considering both thermal-changes and time-variations of

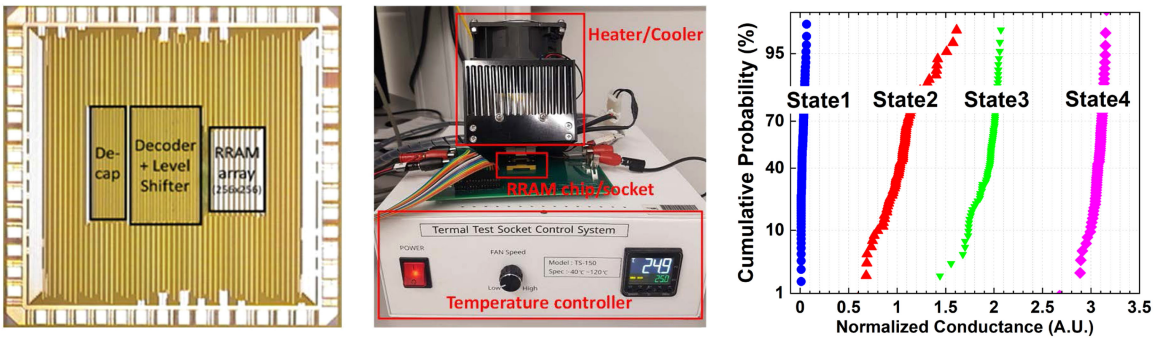

FIGURE 1. (a) Die photo of the 64-Kb HfO_2 1T1R RRAM prototype chip. (b) Temperature-controlled equipment connected to the RRAM chip. (c) Initial 2-bit cell conductance distribution after write–verify iterations.[11]

the conductance drifting, leading to the highly robust DNN models.

- We present a thermal-aware RRAM-based inference engine design.
- Performance analysis based on 2–4-bit DNNs with CIFAR-10 and TinyImageNet data sets is done.

TEMPERATURE-DEPENDENT RRAM CHARACTERISTICS AND MODELING

In general, the RRAM retention failure is caused by both conductance drifting and dispersion. We obtained actual RRAM conductance variation across different temperatures from our RRAM prototype chip. Figure 1(a) shows the die photo of the 256×256 1T1R HfO_2-based 2-bit-per-cell 90-nm RRAM prototype chip along with the peripheral circuits.[11] The conductance of the RRAM cells was measured through the National Instrument PXIe system with different operating temperatures over time. The temperature of the RRAM chip/socket was controlled by TS-150 equipment from Semicon Advance Technology, as shown in Figure 1(b). The chip measurements not only include the thermal characteristics but also contain other device-dependent nonideal effects, such as random telegraph noise.[10]

Static Retention Variations

Figure 1(c) depicts the cumulative probability distribution of the normalized conductance after initial programming at room temperature (25 °C).[11] State 1 represents the high-resistance state (HRS) while state 4 represents the low-resistance state (LRS). The intermediate states of states 2 and 3 between LRS and HRS are linearly spaced with respect to conductance. The conductance is initialized with the two-step write–verify scheme[12] under room temperature. The conductance of each state was controlled by SET and RESET current during the iterative SET and RESET loops; the bias conditions (V_G and V_D) are optimized, respectively, for each state. Once the conductance distributions of the RRAM cells meet the targeted range, the baking temperature starts ramping up (55 °C–120 °C).[11] When the targeted temperature is reached, the stress time counting begins and the conductance of the RRAM cells is measured intermittently from 20 to 80,000 s.

The measured retention characteristics of the RRAM cells in the prototype chip are characterized as the average conductance drifting μ and the standard deviation σ.[11] Overall, we varied the temperature from 25 °C to 120 °C, and measured the RRAM conductance for up to 80,000 s at each baking temperature. Based on the measurement results, the retention variation can be modeled based on the changes in μ and σ values for the corresponding retention temperature K and retention time t

$$\Delta\mu^K = \mu^K(t) - \mu^K_{\text{init}} = A^K_\mu \times \log t \tag{1}$$

$$\Delta\sigma^K = \sigma^K(t) - \sigma^K_{\text{init}} = B^K_\sigma \times \log t. \tag{2}$$

In this static variation scenario, the initial condition of the retention is defined as the measurement starting time (20 s) for each baking temperature. A^K_μ and B^K_σ are the temperature-dependent drifting rates, which can be modeled through linear regression. We formulate this as

$$A^K_\mu = m^K_\mu \times \frac{1}{K} + b^K_\mu \tag{3}$$

$$B^K_\sigma = \max\left(m^K_\sigma \times \frac{1}{K} + b^K_\sigma, 0\right). \tag{4}$$

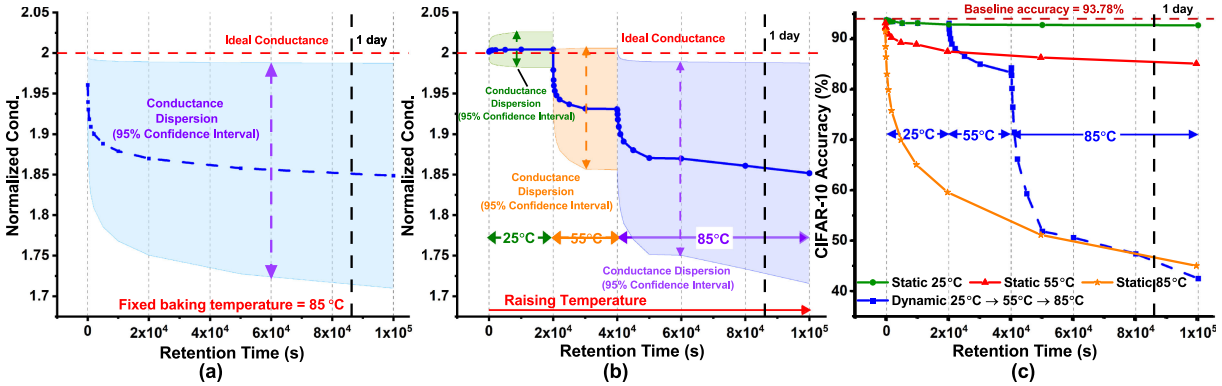

FIGURE 2. (a) Static and (b) dynamic retention variations of the normalized state 2 during the testing (10^5 s). (c) Inference accuracy of 2-bit ResNet-18 for CIFAR-10 is shown for static and dynamic thermal variations.

By combining (1)–(4), we can accurately model the retention variations with Gaussian noises for any given temperature and time. Figure 2(a) depicts the variation statistics at 85 °C baking temperature for 10^5 s of testing time.

Dynamic Retention Variations

In addition to the static retention variations observed over time at a fixed temperature, we also investigated the dynamic retention variations caused by the temporal temperature changes of the RRAM chip. In practice, if the temperature is increased from K_1 to K_2, it means that the initial condition at K_2 is the variation at the transition point from K_1. In the previous section, for static retention variations, we modeled the variation based on statistical changes from the initial conditions. Therefore, the dynamic retention variations can be modeled by accumulating multiple static variations with the updated initial conditions.

Let us assume that the temperature change from K_1 to K_2 happened at time T with the retention statistics of $R_1 = (\Delta\mu_T^{K_1}, \Delta\sigma_T^{K_1})$. Based on (1) and (2), the equivalent statistical changes with respect to K_2 can be computed as

$$\Delta\mu^{K_2} = \mu^{K_2}(t) - \mu_{\text{init}}^{K_2} \tag{5}$$

$$= A_\mu^{K_2}\log(t + T') - A_\mu^{K_2}\log T' \tag{6}$$

$$= A_\mu^{K_2}\log((t + T')/T') \tag{7}$$

where $T' = 10^{\Delta\mu_T^{K_1}/A_\mu^{K_2}}$. In other words, the initial condition of K_2 is the equivalent variation (with respect to K_1) at time T, which can be represented by the static time T'. $\Delta\sigma^{K_2}$ can also be computed in a similar way. Finally, given the total testing time, the dynamic retention variations can be modeled by accumulating the static variations at each temperature step. Figure 2(b) shows a particular dynamic retention scenario where the temperature changes from 25 °C to 55 °C to 85 °C within 10^5 s.

Impact of the Retention Variations

Deploying the pretrained quantized DNN model to the RRAM array involves decomposing the low-precision weights down to the bit-level representations and programming the corresponding conductance values, e.g., mapping one 4-bit weight onto two 2-bit RRAM cells. To understand the impact of the conductance distortions on network-level accuracy, we incorporate the static and dynamic retention variations of 2-bit RRAM cells into a 2-bit ResNet-18 model (i.e., both activation and weight precision values are 2-bit). Figure 2(c) shows the inference results obtained from the NeuroSim.[9] Compared to the static thermal variation, the inference accuracy degrades faster in the dynamic variation scenario because the nonideality is inherited and accumulated in both the temperature and time domains. The conductance variations are accumulated and propagated throughout the entire network, eventually leading to accuracy degradation. Employing very frequent refreshing techniques to recover such accuracy degradation in a short time period is expensive. Therefore, it is necessary to resolve this critical problem in an energy-efficient way.

CHALLENGES OF DNN TRAINING WITH VARIATION/NOISE INJECTION

Challenge of Training With Variation Injection

Injecting the hardware noise during DNN training is an effective method to improve the robustness of the

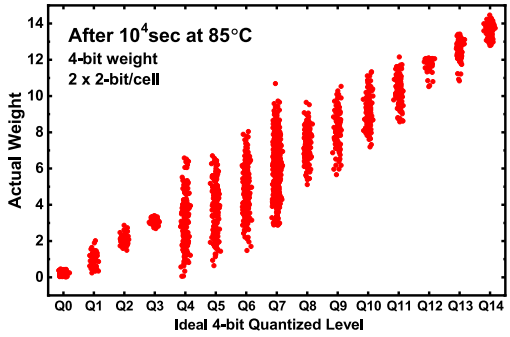

FIGURE 3. Distribution of the 4-bit distorted weights W_Q^* with the static thermal variations.[11]

model. As we described in the previous section, the DNN inference process performed by the RRAM hardware will be divided into the bit-level partial sum computations along the columns of the RRAM array.[9] However, injecting the bit-level noise during training requires the decomposition of the quantized weights. Such a decompose-and-reassemble process will largely slow down the training process and possibly lead to convergence failure. Therefore, to train the model with the injected retention variation, the first challenge is to learn how to inject the bit-level (conductance) noises efficiently without limiting the training process.

The decomposed computation of IMC is mathematically equivalent to the low-precision convolution computation performed by the software. Therefore, the nonideal cell levels (0–3) can be converted to the distorted low-precision weights via the shift-and-add procedure. Figure 3 shows the example of the nonideal distribution based on the 4-bit weight.[11] By subtracting the ideal weight levels from the distorted weights, the resultant noises consist of magnitude drifting and distribution dispersion. We inject the Gaussian noise based on the normalized drift and standard deviation to the corresponding weight levels after the ideal quantization. Given the full-precision weight W and quantization boundary a, the noise injected n-bit in-training quantization process can be formulated as

$$W_c = \min(\max(W, -a), a) \tag{8}$$

$$S = (2^{n-1} - 1)/a \tag{9}$$

$$W_Q = \text{round}(W_c \times S) \tag{10}$$

$$W_Q^* = \{W_q + \beta \times \mathcal{N}(\mu_q, \sigma_q)\}_{q=0}^{2^n-1} \tag{11}$$

$$W_{QF} = W_Q^*/S. \tag{12}$$

Equations (8)–(10) follow the same procedure as the ideal quantization. μ_q and σ_q represent the mean and standard deviation of the hardware variation noises with respect to each low-precision weight level. The tunable parameter β controls the intensity of the noise injection. W_{QF} represents the weights after dequantization from the distorted low-precision weight W_Q^*.[13]

To validate the effectiveness of such conversion, we perform the noise-injected training based on a pretrained 2-bit ResNet-18 model and the converted static variation at 55 °C for 5,000 s. As shown in Figure 4(a), at the selected time and temperature, the resultant model can successfully recover the accuracy even with the decomposed IMC inference.

We also use the noise-free clean low-precision model as the teacher to generate the soft labels and distill the knowledge[14] to the noise-injected student, which reduces the performance gap between the two models. As proved by the results in Figure 4(a), the model trained through the knowledge distillation achieved better inference accuracy.

Challenge of Training With Noise Injection

According to Figure 4(a), training the model while injecting the selected noise can only recover the inference performance at the corresponding temperature and time (55 °C, 5,000 s). The robustness of the trained model has a bad generality to the different scenarios (25 °C and 85 °C). Similarly, knowledge distillation[14] can improve the accuracy of the student model, but the improvements on generality are limited.

Such a generality issue is critical for the hardware inference because the devices usually start operating under the room temperature, and it is expensive to retrain the deployed DNN model based on the newly changed variations. Therefore, the second challenge is to learn how to improve the general model robustness across different temperature variations without retraining the DNN.

PROPOSED TEMPERATURE-RESILIENT RRAM IMC SCHEME

In this section, we present the proposed temperature-resilient solution for RRAM-based IMC inference. We propose a novel training algorithm that aims at improving the DNN robustness of the RRAM-based IMC hardware against the thermal variations.

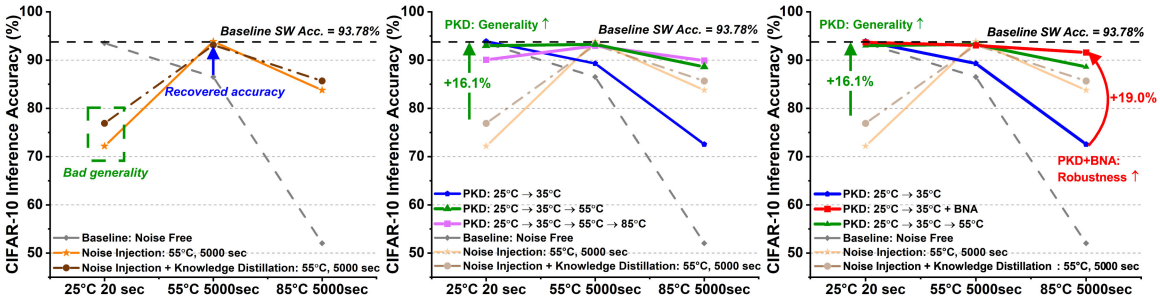

FIGURE 4. DNN hardware inference results after training the 2-bit ResNet-18 for the CIFAR-10 data set with (a) noise injection. (b) PKD at different temperatures and (c) PKD together with BNA, leading to improved robustness and generality.

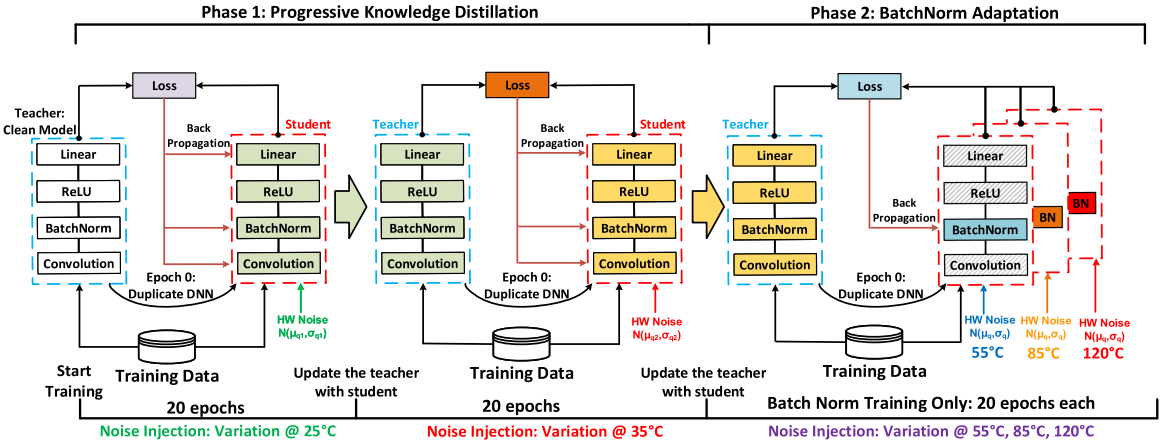

FIGURE 5. Overall DNN training process, including the proposed PKD in phase 1 and BNA algorithm during phase 2 of the training.

Progressive Knowledge Distillation (PKD)

We propose the PKD algorithm to resolve the generality and robustness challenges. The proposed PKD algorithm starts the training by injecting the low-temperature noises to the student model while the clean model is employed as the teacher. Subsequently, we change the injected noises to the higher temperature noises while using the previous student model as the new teacher. As shown in phase 1 of Figure 5, distilling the knowledge in a step-by-step fashion enables the student model to learn the high-temperature variations while matching up with the teacher that was trained with the low temperature. To improve the model's generality even further, the injected noises for each step (temperature) are generated based on the temporally averaged variation between 0 and 10,000 s.

As shown in Figure 4(b), the proposed PKD algorithm aided the DNN model to improve generality at the low temperature while learning the high-temperature variations for better robustness. The model trained by the PKD algorithm that performed noise injection with 55 °C variations can fully recover the accuracy under the 55 °C scenario while only having 0.8% accuracy degradation under the low-temperature 25 °C scenario. Compared to the conventional noise injection training method, the significant improvements achieved by the proposed PKD training algorithm demonstrate the potential of the model to maintain the high inference accuracy under different thermal variation scenarios without frequent refreshing. The results presented in Figure 4(b) are based on the static variations sampled from a relatively short operating time. Furthermore, even the improved performance under the high-temperature 85 °C scenario still exhibits ~5% accuracy loss. Considering the accuracy degradation with the low-temperature variation, naïvely applying the PKD training with the incremental noises will gradually make the resulting model performance irreversible to the previous training scenario resulting in degrading the low-temperature inference accuracy even further.

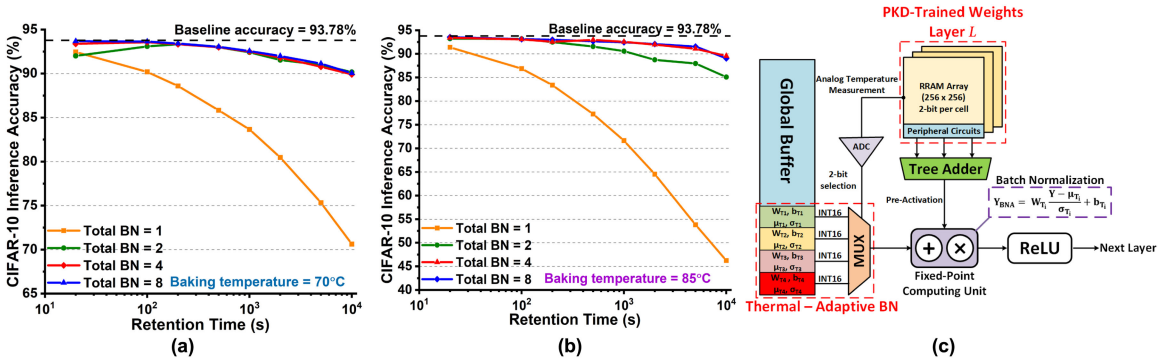

FIGURE 6. PKD+BNA: 2-bit ResNet-18 inference results under (a) 70 °C and (b) 85 °C with a different number of adaptive BN. (c) High-level hardware implementation of the proposed PKD-BNA algorithm.

Batch Normalization Adaptation (BNA)

To further improve the PKD algorithm, we propose the BNA algorithm to elevate the robustness with high hardware compatibility.

After the PKD training, we freeze the weight update process of all the convolutional and fully connected layers, and then continue the noise-injection training with a high-temperature variation. By doing so, the BN parameters will be individually trained with respect to the different thermal variations while all weights and learnable parameters (e.g., trainable activation quantization range[15]) remain the same. The output preactivation of each layer will be normalized by the corresponding BN with the measured temperature T. Mathematically, given the current temperature T and the preactivation Y, the normalization can be simply expressed as

$$Y_{\text{BNA}} = W_T \frac{Y - \mu_T}{\sigma_T} + b_T. \tag{13}$$

Phase 2 in Figure 5 shows the training process of the proposed BNA algorithm. BNA trains the BN individually to adapt to the changed activation distribution caused by the thermal variations. Consequently, the robustness of the model can be improved even further without changing the values of the DNN weights. As shown in Figure 4(c), normalizing the preactivation by the separately trained BN with 55 °C and 85 °C variations significantly improves the inference accuracy under the high-temperature variations. The combination of BNA and the model trained under the 35 °C variations achieved the best performance with high generality and robustness.

The only overhead introduced by the proposed BNA algorithm is the extra BN parameters with respect to the different temperature ranges. Given the operating temperature range from 25 °C to 120 °C, Figure 6(a) and (b) shows the impact of dividing the total temperature range on different number of subsets for BNA training. Considering the minimum accuracy and generality difference between the 4-step training (BN $= 4$) and 8-step training (BN $= 8$), we choose to use four adapted sets of BN parameters to cover the temperature ranges of [25 °C, 50 °C), [50 °C, 70 °C), [70 °C, 90 °C), and [90 °C, 120 °C].

System-Level Inference Hardware Design

After training the model with both PKD and BNA algorithms, PKD-trained low-precision weights will be mapped to the RRAM array. To implement the BNA in hardware, the additional circuits for on-chip temperature measurement and BN multiplexing are necessary.

For the on-chip temperature sensor, we adopted the compact temperature sensor circuits from Yang *et al.*'s work,[16] where the proportional-to-absolute-temperature and complementary-to-absolute-temperature voltages are digitized using a 16-bit off-chip ADC for accurate temperature digitization.

In our work, as discussed in the previous section, we coarsely divide the temperature into four ranges and have a corresponding set of BN parameters for each range. Therefore, we only need a 2-bit ADC to quantize the analog temperature sensor voltages, for which we employed a flash ADC with three sense amplifiers. The 2-bit ADC output is connected to the select signal of the 4-to-1 multiplexer, which chooses the corresponding pretrained BN parameters from the on-chip buffer.

We use 16-bit fixed-point representation for all BN parameters for better hardware compatibility. The BN operation for DNN inference will be performed inside the fixed-point computing unit. Figure 6(c) depicts the high-level hardware implementation of the proposed RRAM-based IMC scheme using the PKD-BNA

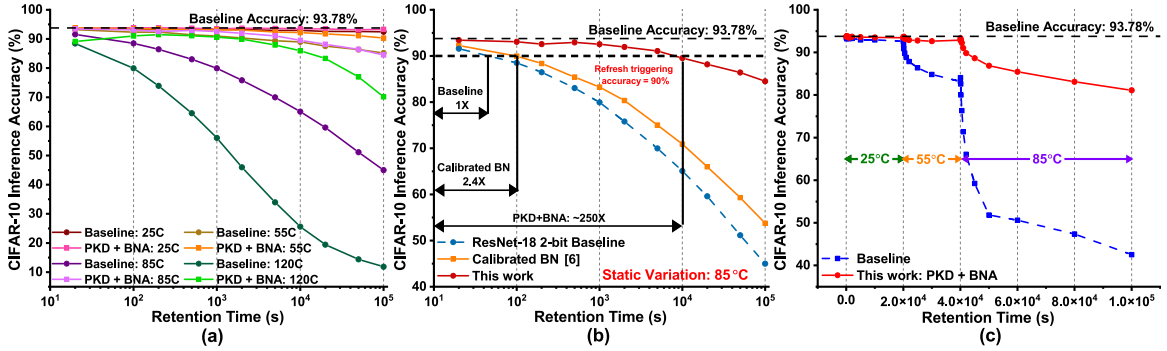

FIGURE 7. Experiments of 2-bit ResNet-18 on the CIFAR-10 data set. (a) Inference accuracy with static variations. (b) Accuracy and refresh frequency comparison among the baseline model, prior work,[6] and the proposed work. (c) Inference accuracy with dynamic thermal variations.

algorithm. Compared with other solutions that require either very frequent refreshing or continuous BN calibration, our proposed design is simple and the hardware overhead is minimal.

EXPERIMENTAL RESULTS

In this section, we present the experimental results on CIFAR-10 and TinyImageNet data sets. The PKD algorithm fine-tuned the pretrained low-precision DNN model using stochastic gradient descent for optimization and the straight-through estimator[17] for gradient approximation. The baseline 2-bit and 4-bit DNN models are fully quantized for all layers using the PACT quantizer.[15] For both PKD and BNA training, we incorporated the EMA[18] technique with a momentum of 0.9997 to improve the knowledge distillation. We use the circuit-level simulator NeuroSim[9] to evaluate the hardware performance of the proposed design. The RRAM array size is 256×256 and 6-bit ADCs are employed at the column periphery to digitize the IMC partial sum.

Static Retention Variation Results

We first evaluated the proposed scheme based on static retention variations, with the baking temperature varying from 25 °C to 120 °C. The injected variation of each temperature is Gaussian noise, where the mean and standard deviation are averaged across the operating time from 20 to 10^4 s. After implementing the proposed PKD training from 25 °C to 35 °C with 20 epochs fine-tuning, we subsequently apply the BNA algorithm for training with 55 °C, 85 °C and 120 °C noises throughout 20 epochs for each temperature.

Figure 7(a) shows the RRAM IMC hardware inference results with the 2-bit ResNet-18 model for static retention variations at different temperatures. For each of the four temperature ranges we employ for BNA, we used one set of fixed-point BN parameters (trained by BNA) to cover the entire time period of the experiment (20–10^5 s). As shown in Figure 7(b), calibrating BN with EMA[6] has a limited improvement to the DNN model robustness. Compared to the IMC inference results when we use the baseline quantized DNN model without any noise injection, the proposed method improved the inference accuracy by a significant margin. When the temperature changes, the corresponding set of BN parameters will be selected, and none of the RRAM weights will be updated or retrained. Even though the accuracy cannot be fully recovered when running the inference with a long operating time and high temperature (e.g., 120 °C), the high degree of robustness in our scheme will significantly reduce the energy consumption of the periodic RRAM refreshing. If we assume that refresh will be triggered when the inference accuracy is lower than 90%, Figure 7(b) shows that the baseline model requires periodic refreshing after ∼30 s of operation. On the other hand, the proposed PKD-BNA method can maintain >90% accuracy until 10^4 s. Compared to the ideally quantized baseline model and Calibrated BatchNorm,[6] the proposed scheme can reduce the refreshing frequency by ∼250× and ∼100×, respectively.

WITHOUT RETRAINING OR UPDATING ANY DNN WEIGHTS AFTER THE INITIAL RRAM PROGRAMMING, OUR PROPOSED SCHEME LARGELY IMPROVES THE INFERENCE ACCURACY ACROSS ALL EXPERIMENTS AND ENHANCES THE ROBUSTNESS OF RRAM HARDWARE AGAINST A WIDE RANGE OF TEMPERATURE VARIATIONS.

TABLE 1. RRAM hardware inference accuracy results for 4-bit ResNet-20/ResNet-18 on CIFAR-10/tinyimagenet data set.

Dataset	DNN model	Scheme	Accuracy for 25°C (static) at 20 seconds	Accuracy for 55°C (static) at 1,000 seconds
CIFAR-10	4-bit ResNet-20	Baseline	91.61 ± 0.46	67.15 ± 1.04
		This work	**91.69 ± 0.31**	**91.39 ± 0.42**
TinyImageNet	4-bit ResNet-18	Baseline	70.51 ± 0.51	0.63 ± 0.25
		This work	**71.23 ± 0.44**	**67.56 ± 0.52**

We use the bold font to highlight the accuracy improvements of the proposed algorithm.

It has been shown in Gao *et al.*'s work[19] that wider DNNs usually have a relatively higher robustness. In this work, we applied the proposed algorithm to both a large model (e.g., ResNet-18 with 11.17 million parameters) and a compact model (e.g., ResNet-20 with 0.27 million parameters). Following the training scheme of Figure 5, Table 1 shows successful inference accuracy recovery with the compact 4-bit ResNet-20 model for the CIFAR-10 data set. Table 1 also shows the performance of the proposed scheme with the large 4-bit ResNet-18 model for the TinyImageNet data set. The model trained by the complex data set (e.g., TinyImageNet) is more sensitive to the variations. Fully recovering the model accuracy might require additional sets of adaptive BN parameters within a single operating temperature.

Dynamic Retention Variation Results

Assuming that the temperature of the RRAM hardware increases over time as in Figure 2, the proposed scheme is applied to the 2-bit ResNet-18 for the CIFAR-10 data set to evaluate the dynamic retention variation scenario. As shown in Figure 7(c), the operating temperature changes occurred at the times of 2×10^4 (25°C → 55°C) and 4×10^4 (55°C → 85°C). While the baseline DNN suffered ∼48% accuracy degradation (at 1×10^5 s), our proposed scheme showed only ∼8% accuracy drop against the large temperature variation in the same period.

CONCLUSION

In this article, we first analyzed and modeled the retention failure caused by the thermal variations from prototype RRAM chip measurements. To resolve the RRAM retention issue, we presented the PKD and BNA algorithms that can efficiently recover the hardware inference accuracy against temperature variations. Considering the crossbar-based IMC RRAM hardware, we also proposed the high-level hardware system design and evaluated the hardware inference accuracy with different model precisions and architectures for the CIFAR-10 and TinyImageNet data sets. Using the proposed algorithm/hardware scheme, the 2-bit ResNet for the CIFAR-10 data set can recover over 30% inference accuracy, and the 4-bit ResNet-18 for the TinyImageNet data set can recover over 60% accuracy. Without retraining or updating any DNN weights after the initial RRAM programming, our proposed scheme largely improves the inference accuracy across all experiments and enhances the robustness of RRAM hardware against a wide range of temperature variations.

ACKNOWLEDGMENTS

The authors would like to thank Winbond Electronics for RRAM chip fabrication support. This work was supported in part by JUMP CBRIC, in part by JUMP ASCENT, in part by the SRC AIHW Program, and in part by the National Science Foundation under Grant 1652866/1715443/1740225.

REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
2. Y. Xiang *et al.*, "Impacts of state instability and retention failure of filamentary analog RRAM on the performance of deep neural network," *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4517–4522, Nov. 2019.
3. M. Cheng *et al.*, "TIME: A training-in-memory architecture for RRAM-based deep neural networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 5, pp. 834–847, May 2019.
4. M. V. Beigi and G. Memik, "Thermal-aware optimizations of ReRAM-based neuromorphic computing systems," in *Proc. IEEE/ACM Des. Autom. Conf.*, 2018, pp. 1–6.
5. H. Shin, M. Kang, and L.-S. Kim, "A thermal-aware optimization framework for ReRAM-based deep neural network acceleration," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2020, pp. 1–9.
6. L. H. Tsai *et al.*, "Robust processing-in-memory neural networks via noise-aware normalization," 2020, *arXiv:2007.03230*. [Online]. Available: https://arxiv.org/abs/2007.03230

7. Z. He, J. Lin, R. Ewetz, J. Yuan, and D. Fan, "Noise injection adaption: End-to-end ReRAM crossbar non-ideal effect adaption for neural network mapping," in *Proc. 56th ACM/IEEE Des. Autom. Conf.*, 2019, pp. 1–6.
8. B. Feinberg, S. Wang, and E. Ipek, "Making memristive neural network accelerators reliable," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit.*, 2018, pp. 52–65.
9. X. Peng, S. Huang, Y. Luo, X. Sun, and S. Yu, "DNN NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," in *Proc. IEEE Int. Electron Devices Meeting*, 2019, pp. 32.5.1–32.5.4.
10. F. M. Puglisi, L. Larcher, A. Padovani, and P. Pavan, "A complete statistical investigation of RTN in HfO_2-based RRAM in high resistive state," *IEEE Trans. Electron Devices*, vol. 62, no. 8, pp. 2606–2613, Aug. 2015.
11. W. Shim, J. Meng, X. Peng, J.-S. Seo, and S. Yu, "Impact of multilevel retention characteristics on RRAM based DNN inference engine," in *Proc. IEEE Int. Rel. Phys. Symp.*, 2021, pp. 1–4.
12. W. Shim *et al.*, "Two-step write-verify scheme and impact of the read noise in multilevel RRAM-based inference engine," *Semicond. Sci. Technol.*, vol. 35, no. 11, 2020, Art. no. 115026.
13. R. Krishnamoorth, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018, *arXiv:1806.08342*. [Online]. Available: https://arxiv.org/abs/1806.08342
14. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: https://arxiv.org/abs/1503.02531
15. J. Choi *et al.*, "Accurate and efficient 2-bit quantized neural networks," in *Proc. Conf. Mach. Learn. Syst.*, 2019, pp. 348–359.
16. T. Yang, S. Kim, P. R. Kinget, and M. Seok, "Compact and supply-voltage-scalable temperature sensors for dense on-chip thermal monitoring," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2773–2785, Nov. 2015.
17. Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*. [Online]. Available: https://arxiv.org/abs/1308.3432
18. P. Izmailov *et al.*, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 876–885.
19. R. Gao *et al.*, "Convergence of adversarial training in overparametrized neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 13029–13040, 2019.

JIAN MENG is working toward a Ph.D. degree with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA. He is a Graduate Student Member of IEEE. Contact him at jmeng15@asu.edu.

WONBO SHIM is an Assistant Professor with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, South Korea. He is a Member of IEEE. Contact him at wshim30@gatech.edu.

LI YANG is working toward a Ph.D. degree with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA. He is a Graduate Student Member of IEEE. Contact him at lyang166@asu.edu.

INJUNE YEO is Postdoctoral Researcher with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA. He is a Member of IEEE. Contact him iyeo3@asu.edu.

DELIANG FAN is an Assistant Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA. His research interests include processing-in-memory circuit, architecture and application cross-layer co-design, and hardware-aware deep learning optimization. He was the recipient of three Best Paper Awards from GLSVLSI 2019, ISVLSI 2018, and ISVLSI 2017. He is a Member of IEEE. Contact him at dfan@asu.edu.

SHIMENG YU is an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include nanoelectronic devices and circuits for energy-efficient computing systems. He was the recipient of the NSF CAREER Award in 2016, the IEEE Electron Devices Society (EDS) Early Career Award in 2017, and the Semiconductor Research Corporation (SRC) Young Faculty Award in 2019. He is a Senior Member of IEEE. Contact him at shimeng.yu@ece.gatech.edu.

JAE-SUN SEO is an Associate Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, USA. His research focuses on energy-efficient hardware design for machine learning and neuromorphic computing. He was the recipient of the NSF CAREER Award in 2017 and the Intel Outstanding Researcher Award in 2020. He is a Senior Member of IEEE. Contact him at jaesun.seo@asu.edu.