Edge Learning Using a Fully Integrated Neuro-Inspired Memristor Chip

Wenbin Zhang†, Peng Yao†, Bin Gao*, Qi Liu, Dong Wu, Qingtian Zhang, Yuankun Li, Qi Qin, Jiaming Li, Zhenhua Zhu, Yi Cai, Dabin Wu, Jianshi Tang, He Qian, Yu Wang, Huaqiang Wu*

1. School of Integrated Circuits, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China.

2. Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China.

*Corresponding author. Email: gaob1@tsinghua.edu.cn (B.G.); wuhq@tsinghua.edu.cn (H.W.)

† These authors contributed equally to this work.

ABSTRACT Learning is highly important for edge intelligence devices to adapt to different application scenes and owners. Current technologies for training neural networks require moving massive amounts of data between computing and memory units, which hinders the implementation of learning on edge devices. We developed a fully integrated memristor chip with the improvement learning ability and low energy cost. The schemes in the STELLAR architecture, including its learning algorithm, hardware realization, and parallel conductance tuning scheme, are general approaches that facilitate on-chip learning by using a memristor crossbar array, regardless of the type of memristor device. Tasks executed in this study included motion control, image classification, and speech recognition.

INDEX TERMS Memristor, edge computing, on-chip learning, STELLAR architecture, neuromorphic computing, neural networks.

I. INTRODUCTION

Humans' ability to learn plays a vital role in the growth of intelligence and fast adaptation to unseen scenes or dynamically changing environments. Edge artificial intelligence (AI) applications also require hardware with such learning abilities to enable the associated devices to adapt to new scenes or user habits [1]. However, deep neural network (DNN) training [2, 3] is typically implemented with conventional hardware based on the von Neumann computing architecture and high-precision digital computing paradigm [4]. The extensive data movement between the processor chip and off-chip main memory incurs massive energy consumption and accounts for most of the latency of the whole training process [5, 6]. Therefore, although cloud computing platforms can handle such energy-intensive training [4, 7], their high energy consumption hinders the implementation of learning on power-limited edge computing platforms [1]. By contrast, memristor-based neuro-inspired computing eliminates this extensive data movement through its disruptive computation-in-memory architecture and analog computing paradigm [6, 8-10]. A memristor crossbar array can store an analog synaptic weight and perform in situ vector-matrix multiplication (VMM) operations in parallel in a single time step by exploiting Ohm's law and Kirchhoff's law. A neuro-inspired computing chip that integrates multiple memristor crossbar arrays and complementary metal-oxide semiconductor (CMOS) circuits can easily implement DNN inference [11-14] and has great potential to handle fully on-chip learning without any assistance from off-chip memory [15-17].

The substantial energy efficiency enhancement provided by memristor-based neuro-inspired computing makes this paradigm promising for developing future chips that can enable low-power learning devices.

Several studies [18-25] have experimentally demonstrated learning using memristor crossbar arrays for in situ weight tuning although using software or external digital processors to implement the backpropagation (BP) algorithm [2]. However, realizing a complete fully integrated memristor chip with strong learning ability and low energy costs remains challenging. The key challenge lies in the inefficiency of mapping the BP algorithm to on-chip hardware. First, an in-memory implementation of the BP algorithm requires costly conductance tuning operations with write verification due to device nonidealities, such as device variability and nonlinear conductance modulation [15, 26-29]. Second, it is difficult to achieve efficient parallel conductance tuning with write verification [19-21, 23], which makes on-chip learning more time- and energy-consuming. Third, the high-precision data processing operations required during weight update

calculations require a large circuit area and high energy consumption, leading to unacceptable overhead [26, 30].

In this work, we demonstrated a memristor-based neuro-inspired computing chip that enabled fully on-chip learning, for which a memristor-featured sign-and-threshold-based learning (STELLAR) architecture was proposed. In this architecture, the on-chip updating scheme was first proposed to tune the memristor without verification. This scheme saved excessive write-and-read costs in the conductance tuning operations when compared to the write verification scheme, and moreover, it could accommodate device tuning issues of nonlinearity and asymmetry to realize software-comparable accuracies. Second, the on-chip calculation module was designed to determine the weight update direction, and this process solely involved the signs of the inputs, outputs, and errors instead of their high-precision formats. This design reduced the circuit design burden and avoided massive overhead during on-chip learning. Third, a cycle-parallel conductance tuning scheme was proposed, wherein conductance tuning was performed in a row-by-row parallel fashion. This scheme further reduced the induced energy consumption and latency and accommodated the limited endurance of memristors.

The fabricated neuro-inspired computing chip integrated two memristor crossbar arrays (~160,000 cells in total) and all the necessary circuit modules, including controllers for configuration, drivers for computing and programming, low-cost data converters, and memristor-featured learning modules. On the basis of the obtained hardware-measured results, the energy consumption of the memristor chip was a factor of 35 lower than that of a digital accelerator-based system. We demonstrated several improvement learning tasks, including motion control for a light-chasing car, image classification, and audio recognition. The scalability of the STELLAR schemes to large neural networks for improvement learning tasks was also verified with the simulation of a residual neural network on the CIFAR-100 dataset [31]. The memristor-based neuro-inspired computing chip could facilitate the development of edge AI devices that could adapt to new scenes and users.

II. MEMRISTOR-FEATURED ARCHITECTURE FOR ON-CHIP LEARNING

To support on-chip learning with appreciable energy efficiency, area efficiency, and accuracy, we proposed the STELLAR architecture. STELLAR architecture exploits the bidirectional analog switching behavior of the memristor device [32]. During the weight update stage, only the weight update direction must be calculated from the signs of the inputs, outputs, and errors. In addition, the architecture predefines a threshold, which filters out the small errors when calculating the error signs and plays a vital role in the convergence of the learning algorithm by avoiding updates that are exceedingly sensitive and unnecessary. By omitting these small updates, the memristor-based gradient vectors under the STELLAR update scheme could approximate conventional BP gradient vectors more closely to accommodate practical device nonideal factors (such as asymmetric tuning of device conductance). The detailed analysis and simulation can be found in materials and methods section 2 (STELLAR update scheme under device asymmetric switching). This threshold is hardware-reconfigurable to adapt to various learning tasks. The algorithmic details of the STELLAR architecture can be found in materials and methods section 1 (Algorithm for the STELLAR architecture).

Depending on the weight update direction, a corresponding identical SET or RESET pulse is applied to the memristor cells. With this scheme, we could realize energy-efficient hardware by avoiding the complex precise weight update calculation and write verification processes as well as the complex peripheral circuit design.

The learning performance of the STELLAR architecture was compared with that of the conventional methods through simulations on the Modified National Institute of Standards and Technology (MNIST) dataset [33]. Here, all memristors in the second layer were set to random conductance states before the learning process started. The learning accuracies of the conventional BP method without and with different write variations (1% and 3%, given as the percentages of the full conductance window of the memristor device) in comparison with those of the proposed method under various thresholds are presented. The simulation details can be found in the materials and methods section 3 (Comparison between STELLAR and the conventional BP algorithm). An appropriately selected threshold yielded improved convergence and learning accuracy. An extremely small threshold led to weight updates that were too frequent and an oscillating state of the network, and a threshold that was excessively large led to inadequate weight updates. Despite maintaining almost the same accuracy, the energy consumption of the STELLAR architecture was two orders of magnitude lower than that of the conventional BP method owing to the substantial reductions in the precise weight update calculations and the write verification overhead.

The differential weight representation in the STELLAR architecture is given by:

$$w = g^{+} - g^{-} \quad (1)$$

The STELLAR architecture realized positive and negative weights with differential pairs of memristor cells [20, 23, 34, 35]. In conventional crossbar arrays with one-transistor-one-resistor (1T1R) configurations, the two memristor cells in a differential pair are connected to different source lines (SLs), and subtraction is accomplished in a digital fashion. In crossbar arrays with two-transistor-two-resistor (2T2R) configurations, the two memristor cells in a differential pair are connected to the same SL, and subtraction is accomplished directly in the current domain. The 2T2R design greatly reduces the SL current and thus, the IR drop issues, hence enabling a larger array size. Here, we propose a cycle-parallel conductance tuning scheme for such a differential pair configuration.

In this scheme, the SET and RESET operations are performed alternately for the learning iterations of arriving

input samples (e.g., images). Taking the SET update mode as an example, the weight update conditions are:

$$\begin{cases} \Delta w > 0: \text{positive cell updated with SET to increase weight} \\ \Delta w < 0: \text{negative cell updated with SET to decrease weight} \\ \Delta w = 0: \text{neither cell updated} \end{cases} \quad (2)$$

The conductance tuning is performed in a row-by-row parallel fashion. The memristor devices in the same row were selected through their word line (WL) signals and tuned depending on the corresponding SL signals, and the bit line (BL) signals remained constant. The cycle-parallel conductance tuning scheme could be applied to either 1T1R or 2T2R memristor arrays. Because only one-half of the memristor devices were updated during each on-chip learning iteration, the cycle-parallel conductance tuning scheme reduced the induced energy consumption and alleviated the requirement regarding the memristor endurance. The detailed analysis of endurance requirements can be found in materials and methods section 4 (Endurance requirements for the cycle-parallel conductance tuning scheme).

III. CHIP DESIGN, FABRICATION, AND MEASUREMENTS

The overall circuit implementation of the proposed STELLAR architecture was designed and fabricated. This memristor chip consisted of controllers for configuration; a 2T2R memristor array (1568×100), a 1T1R memristor array [100×20; see materials and methods section 5 (Weight configuration based on the 1T1R memristor array) for details]; BL, WL, and SL drivers for computing and programming; low-cost analog-to-digital converters (ADCs); modules for memristor-featured on-chip learning (i.e., error-circulating subtractors and weight update logic); and input and output buffers. The first-layer memristor array adopted a 2T2R configuration to reduce the IR drop issues occurring in such a large array, and the second-layer memristor array adopted a 1T1R configuration to support more flexible in situ weight tuning. The controllers decoded the input stage selection signals and provided the output configuration signals to other circuit modules to switch the chip to different working stages [see materials and methods section 6 (Circuit design of the memristor chip) and fig. S5]. Resolution-adjustable ADCs (RA-ADCs) feature configurable resolutions and support flexible threshold values [11]. The error calculation was accomplished with subtractors, which were realized with counters. The weight update logic determined the weight update direction and conductance tuning operations.

A micrograph of the fabricated chip is presented. The chip area breakdown is described in fig. S6B. The memristor device used a material stack of TiN/HfO_x/TaO_n/TiN, and the fabrication process was compatible with the standard CMOS process [see materials and methods section 7 (Fabrication of the memristor chip) and fig. S6A]. Consequently, the memristors could be conveniently integrated with complex CMOS circuits to produce an excellent yield (almost 100% of all 160,000 cells). The cross-section transmission electron microscopy (TEM) image showed the integration of memristor cells with CMOS circuitry. The fabricated memristors exhibited uniform and repeatable bidirectional analog switching with identical pulse trains (fig. S6D). The ~160,000 total on-chip memristor cells could be uniformly programmed to 32 conductance states, with maximum, minimum, and average success rates of 99.98, 99.69, and 99.90%, respectively [see materials and methods section 8 (Measurements of the memristor devices) and fig. S6C].

The on-chip inference was first demonstrated for MNIST handwritten digit classification. The weights were off-chip trained and then transferred to the chip as memristor conductance [see materials and methods section 9 (Off-chip training and on-chip inference)]. The measured classification accuracy of each class (0 to 9) is presented; the average accuracy was 95.8%. The effect of cell conductance fluctuations on the chip accuracy was also evaluated. The accuracy was monitored for 48 days, and no obvious accuracy degradation was observed. Real-time handwritten digit recognition with the memristor chip was also demonstrated.

An on-chip learning task, MNIST image classification, was further demonstrated to verify the on-chip learning ability based on a 784-100-10 multilayer perceptron (MLP). The weights in the first layer were trained off-chip and then transferred to the chip as memristor conductance. The memristors in the second layer were first programmed to the high-resistance state (HRS) and then updated using the STELLAR scheme. All data processing and signal control processes were executed on the chip. After three epochs of on-chip learning with the training set, the classification accuracies were increased from 8.6 and 8.4% to 94.9 and 92.3% on the training set and test set, respectively. The energy consumption of the on-chip learning was evaluated with hardware-measured results [see materials and methods section 11 (Energy consumption benchmark)]. The energy consumption of a digital accelerator-based system [36] was also evaluated for one training iteration with the same MLP network. The energy breakdown of the memristor chip during the on-chip learning process is presented. The energy consumption could be further reduced by optimizing the ADC design [37-39].

IV. ON-CHIP IMPROVEMENT LEARNING

The memristor chip was used to further demonstrate four improvement learning tasks, including the motion control task of learning new samples, audio recognition task of learning new samples, image classification task of learning a new class, and motion control task of learning a new class. The improvement learning featured the fast learning of new knowledge and maintaining preacquired knowledge.

As illustrated, the learning of new knowledge (e.g., new samples or classes) was quickly realized with only a few new inputs, and this learning was done without losing preacquired knowledge. The implementation of improvement learning mainly included two stages. First, a model was trained off-chip on the base dataset, and then the model was transferred to the chip. Next, under the STELLAR architecture, on-chip improvement learning with new data was performed on the

basis of the transferred memristor chip. Improvement learning is a specific learning format to implement lifelong learning. This learning scheme aims to rapidly learn knowledge from the new data without forgetting previous knowledge on the old data, making it different from transfer learning [40] that focuses on transferring knowledge from original data to new data, regardless of the accuracy drop on the original data. We demonstrated these improvement learning tasks because we consider that recognizing new classes or samples with a base model is a practical and promising edge learning task.

A. Motion Control Task of Learning New Samples

We first demonstrated the learning of new samples in a motion control task of a light-chasing car. The car was designed to pursue the location of a laser light spot; it was equipped with a camera to capture environmental images, steering motor for direction control, and driving motor for throttle control. The memristor chip received the input features of the environmental images from the PC and provided the output control signals for the steering angles and driving throttles [see materials and methods section 12 (Motion control task of learning new samples) for details].

As illustrated, a convolutional neural network (CNN) that included six convolution layers and two fully connected (FC) layers with the dimensions of 512×100×10 was first trained off-chip with old scene data (i.e., dark scene data), and then the weights of these two FC layers were transferred to the two corresponding arrays of the memristor chip. Next, improvement learning of the new scene (i.e., a bright scene) was performed on the chip by tuning the weights of the last FC layer. The details can be found in materials and methods section 16 (The hybrid system for running the motion control task).

Before improvement learning, the car could have lost track of the target (i.e., light spot) in the bright scene: It deviated from the target or moved forward even if no target was present. After improvement learning, the car adapted well to the bright scene and still performed well in the dark scene. The evolution of the scores during improvement learning is presented, where a score of 1.0 denotes the best performance [see materials and methods section 12 (Motion control task of learning new samples)]. The scores became stable after improvement learning with 500 training samples from the new scene. After improvement learning with all the training samples, the average score in the new scene significantly increased from 0.605 to 0.912, and that of the old scene increased from 0.951 to 0.963, showing that no degradation occurred.

B. Image Classification Task of Learning a New Class

Next, we demonstrated the learning of a new class in an image classification task involving the MNIST dataset. The base model was trained to recognize images of the digits 0 and 2-9 (old classes) and then transferred to the memristor chip. Next, the improvement learning of the new class (i.e., the digit 1) was performed on the chip. The accuracy achieved for the new class increased substantially during the improvement learning with only a few training samples, and the accuracy for the remaining nine old classes was not significantly reduced. The accuracies yielded for the new class and old classes stabilized after 100 training samples. After improvement learning with 100 training samples, the average accuracy for the new class increased from 7.02 to 93.0%, and that achieved for the old classes slightly decreased from 95.3 to 93.2%.

C. Audio Recognition Task of Learning New Samples

In addition, the memristor chip was also used to implement an audio recognition task of learning new samples. In the audio recognition task, the on-chip improvement learning improved the recognition accuracy for female audio samples based on the weights pretrained with male audio samples (fig. S8A).

D. Motion Control Task of Learning a New Class

In the motion control task of learning a new class, the on-chip improvement learning of the "moving backward" action helped the car to accomplish the tough task based on the model only knowing when to move forward or stop. After improvement learning, the accuracy achieved for the new class (i.e., moving backward) increased from 0 to 95.2%, and that for the old classes (i.e., moving forward or stopping) decreased from 89.5 to 89.4%.

Moreover, we performed another simulation with a memristor ResNet20 network for CIFAR-100 image recognition, which required 20 unseen classes to be learned after training on the remaining 80 categories. The accuracies on old, new, and whole datasets for this task were similar to software results with the floating-point precision [see materials and methods section 17 (The scalability of the method to larger neural networks) and fig. S10]. These results illustrated that the proposed STELLAR architecture could be scalable to larger neural networks and could realize efficient improvement learning with high-precision software accuracies.

V. CONCLUSIONS

We developed a fully integrated memristor chip with the improvement learning ability and low energy cost. The schemes in the STELLAR architecture, including its learning algorithm, hardware realization, and parallel conductance tuning scheme, are general approaches that facilitate on-chip learning by using a memristor crossbar array, regardless of the type of memristor device. We demonstrated the improvement learning of both new samples and a new class across various tasks, including motion control, image classification, and speech recognition, which showed that the STELLAR architecture accommodated the device nonidealities and equipped the memristor chip with improvement learning ability to adapt to new scenarios. With further circuitry engineering [41] based on advanced fabrication technology, the STELLAR architecture could enable on-chip learning memristor chip with an energy efficiency about 75 times higher than that of the digital

accelerator [36]. More details can be seen in materials and methods section 18 (The energy efficiency estimation of the memristor-based learning chip). This study is an important step toward future chips with high energy efficiency and extensive learning capabilities.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637-646, Oct. 2016.

[2] D. E. Rumelhart, G. Hinton, R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533-536, Oct. 1986.

[3] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, May 2015.

[4] A. Coates et al., "Deep learning with COTS HPC systems," in Proc. 30th Int. Conf. Mach. Learn., 2013, pp. 1337-1345.

[5] M. Horowitz, "Computing's energy problem (and what we can do about it)," in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, Feb. 2014, pp. 10-14.

[6] H.-S. P. Wong, S. Salahuddin, "Memory leads the way to better computing," Nat. Nanotechnol., vol. 10, no. 3, pp. 191-194, Mar. 2015.

[7] E. Strubell, A. Ganesh, A. McCallum, "Energy and policy considerations for deep learning in NLP," in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 09, 2020, pp. 13693-13696.

[8] D. Ielmini, H.-S. P. Wong, "In-memory computing with resistive switching devices," Nat. Electron., vol. 1, no. 6, pp. 333-343, Jun. 2018.

[9] M. A. Zidan, J. P. Strachan, W. D. Lu, "The future of electronics based on memristive systems," Nat. Electron., vol. 1, no. 1, pp. 22-29, Jan. 2018.

[10] M. Lanza et al., "Memristive technologies for data storage, computation, encryption, and radio-frequency communication," Science, vol. 376, no. 6597, p. eabj9979, Jun. 2022.

[11] Q. Liu et al., "33.2 A fully integrated analog ReRAM based 78.4TOPS/W compute-in-memory chip with fully parallel MAC computing," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2020, pp. 500-502.

[12] W. Wan et al., "A 74 TOPS/W compute-in-memory chip based on resistive random-access memory for AI edge applications," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2020, pp. 498-500.

[13] C.-X. Xue et al., "A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge processors," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2021, pp. 245-247.

[14] W.-H. Chen et al., "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors," Nat. Electron., vol. 2, no. 9, pp. 420-428, Sep. 2019.

[15] S. Ambrogio et al., "Equivalent-accuracy accelerated neural-network training using analogue memory," Nature, vol. 558, pp. 60-67, Jun. 2018.

[16] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, E. Eleftheriou, "Memory devices and applications for in-memory computing," Nat. Nanotechnol., vol. 15, pp. 529-544, 2020.

[17] W. Zhang et al., "Design guidelines of RRAM based neural-network accelerator for analog in-memory computing," Nat. Electron., vol. 3, no. 6, pp. 371-382, Jun. 2020.

[18] M. Prezioso et al., "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," Nature, vol. 521, pp. 61-64, May 2015.

[19] P. Yao et al., "Face classification using electronic synapses," Nat. Commun., vol. 8, p. 15199, May 2017.

[20] C. Li et al., "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," Nat. Commun., vol. 9, p. 2385, Jun. 2018.

[21] Z. Wang et al., "Resistive switching materials for information processing," Nat. Rev. Mater., vol. 5, pp. 173-195, 2020.

[22] C. Li et al., "Long short-term memory networks in memristor crossbar arrays," Nat. Mach. Intell., vol. 1, pp. 49-57, Jan. 2019.

[23] P. Yao et al., "Fully hardware-implemented memristor convolutional neural network," Nature, vol. 577, pp. 641-646, Jan. 2020.

[24] F. Cai et al., "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," Nat. Electron., vol. 2, no. 7, pp. 290-299, Jul. 2019.

[25] E. J. Fuller et al., "Parallel programming of an ionic floating-gate memory array for scalable neuromorphic computing," Science, vol. 364, pp. 570-574, May 2019.

[26] 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.

[27] H. Wu et al., "Resistive RAM for acceleration of deep neural networks," in IEEE Int. Electron Devices Meeting (IEDM), Dec. 2017, pp. 11.5.1-11.5.4.

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

[29] Y. Xie et al., "Adaptive and efficient processing for domain-specific workloads," Proc. IEEE, vol. 109, no. 1, pp. 14-42, Jan. 2021.

[30] Q. Zhang et al., "Sign-back: A cost-efficient and accurate training framework for memristor-based computation-in-memory systems," Neural Netw., vol. 108, pp. 217-223, Dec. 2018.

[31] A. Krizhevsky, G. Hinton, "Learning multiple layers of features from tiny images," Univ. of Toronto, Tech. Rep., 2009.

[32] W. Wu et al., "15.2 A 100ns ferroelectric based non-volatile compute-in-memory macro with 16 ReLu-less algorithm for AI edge devices," in IEEE Symp. VLSI Technol., Jun. 2018, pp. 103-104.

[33] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[34] P. Chi et al., "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," in Proc. 43rd Annu. Int. Symp. Comput. Archit., Jun. 2016, pp. 27-39.

[35] L. Song, X. Qian, H. Li, Y. Chen, "PipeLayer: A pipelined ReRAM-based accelerator for deep learning," in IEEE Int. Symp. High Perform. Comput. Archit. (HPCA), Feb. 2017, pp. 541-552.

[36] D. Han et al., "55.1 A 50TOPS/W 8b heterogeneous RRAM in-memory computing using 6T2R cell and accuracy-stabilizing calibration," IEEE J. Solid-State Circuits, vol. 56, no. 9, pp. 2858-2869, Sep. 2021.

[37] R. Khaddam-Aljameh et al., "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," in Proc. VLSI Circuits, Jun. 2021, pp. 1-2.

[38] W. Wan et al., "A compute-in-memory chip based on resistive random-access memory," in Proc. VLSI Technol., Jun. 2020, pp. 1-2.

[39] J.-M. Hung et al., "A four-megabit compute-in-memory macro with eight-bit precision," Nat. Electron., vol. 4, pp. 921-930, 2021.

[40] K. Weiss, T. M. Khoshgoftaar, D. Wang, "A survey of transfer learning," J. Big Data, vol. 3, p. 9, 2016.

[41] W.-H. Huang et al., "A 29TOPS/W 71b/mm² heterogeneous RRAM in-memory computing using 40nm architecture with stability checking for accurate image generation," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2023, pp. 15-17.

[42] W. Zhang, P. Yao, B. Gao, H. Wu, "Data for edge learning using a fully integrated neuro-inspired memristor chip," Zenodo, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8145521

[43] W. Zhang, P. Yao, B. Gao, H. Wu, "Edge learning using a fully integrated neuro-inspired memristor chip," Zenodo, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8151757