Invited Paper: Towards the Efficiency, Heterogeneity, and Robustness of Edge AI

Bokyung Kim
Electrical and Computer Engineering
Duke University
Durham, USA
bokyung.kim828@duke.edu

Zhixu Du
Electrical and Computer Engineering
Duke University
Durham, USA
zhixu.du@duke.edu

Jingwei Sun
Electrical and Computer Engineering
Duke University
Durham, USA
jingwei.sun@duke.edu

Yiran Chen
Electrical and Computer Engineering
Duke University
Durham, USA
yiran.chen@duke.edu

2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) | 979-8-3503-2225-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCAD57390.2023.10323922

***Abstract*—Over the past decade, there has been a persistent trend in edge computing, driving the migration of intelligence closer to the edge. The increasing need to process data locally has fueled the deployment of highly efficient computing hardware and artificial intelligence (AI) models onto edge devices. The performance and robustness of edge computing systems are significantly influenced by the heterogeneity of computing systems and the diverse nature of data to be processed by each edge device. This paper aims to explore the principles of software/hardware co-design for edge computing systems in AI applications. We will delve into the robustness concerns faced by edge AI due to the inherent heterogeneity of systems and data. Furthermore, we will present various solutions that effectively mitigate these adverse effects and enhance the resilience of edge AI systems.**

***Index Terms*—Artificial intelligence, Edge AI, Edge computing, Efficiency, Heterogeneity, Robustness**

I. Introduction

Artificial intelligence (AI) has become a cornerstone of a myriad of applications covering image classification, speech recognition, language processing, activity detection, etc. Especially, AI-applied edge computing brings a pinnacle of sensation by realizing a dramatic advance in technologies, such as self-driving, real-time recommendation, and the construction of smart cities, by executing machine learning onto distributed edge devices. Through local computing of AI close to users and data sources, edge AI takes advantage of the obviation of the delay latency in data transfer and the promise of environmental consistency.

Among state-of-the-art technologies fortified for edge AI, two mainstreams stand out in the software and hardware fields: federated learning (FL) and processing-in-memory (PIM). FL is a popular framework for distributed machine-learning systems. The straightforward training of a distributed system is gathering local data from mobile devices and training in the server, but it raises privacy and efficiency concerns. Instead, FL orchestrates participating clients to train a global model in collaboration, preserving secure communication. Hardware has also been developed by taking a new paradigm, PIM. Breaking the rule of conventional hardware, i.e., distinct separation of memory and computing units, PIM has emerged by infusing arithmetic capability into storage units. Since the missing *memristor* [23] has been found with nano-scale thin-file structures, the realization and utilization of emerging non-volatile memories (eNVMs) notably propel fruitful outcomes in PIM research.

However, edge computing is producing a prodigious amount of data beyond expectation. For instance, a modern manufacturing plant constructs a production line with 2,000 different pieces of equipment, and each piece might gather data through hundreds of sensors, generating 2,200 TB of data per month [20]. Accordingly, edge AI in resource-constrained environments faces several challenges spanning extensive perspectives. This paper focuses on three challenges—*efficiency, heterogeneity, and robustness*, which researchers actively address through software, hardware, and co-design approaches to guarantee sustainable edge AI.

Efficiency has been a primary goal regardless of the field, but the drastic data increase on edge devices especially accents its importance because data communication and processing induce high costs. Algorithms have developed besides accuracy by introducing cutting-edge techniques like sparsification through pruning and quantization. In addition, the development has evolved to satisfy FL demands to guarantee communication and computation efficiency simultaneously. The compute-intensive machine-learning algorithms have pushed PIM accelerators to match the advances of algorithms. Further exploration regarding PIM architecture dimension and dataflow improves hardware efficiency.

However, FL has struggled with the inherent ***heterogeneity*** of distributed data. FL personalization strategies like device selection have tackled the heterogeneity. In integrating components, PIM accelerator systems also encounter heterogeneity across different hardware levels and hierarchies. Undesirable

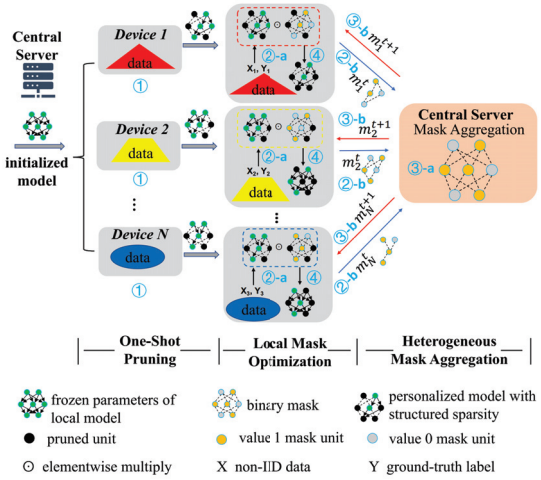


Fig. 1: Pipeline of FedMask designed for efficient FL Learning (Figure 4 in [14]).

and unexpected issues and noises occur via integration stages, e.g., device issues like nonlinearity and process/device variations, circuit issues involving IR drop, thermal/shot noises, and sneak-path current, to the system level. The failure of in-depth consideration of heterogeneity causes unreliability in systems.

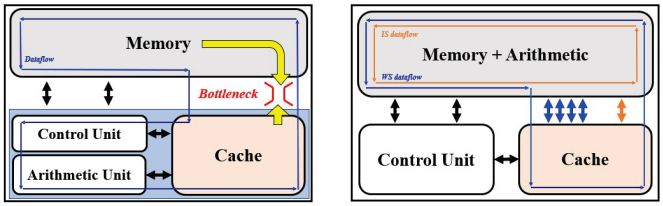
Grafting software and hardware strategies, co-design solutions are gaining ground for ***robustness***. To soothe the concern toward device disengagement, we introduce a pioneered research outcome in FL, vertical FL (VFL) for robustness. From the hardware perspective, hardware enhancement itself is crucial for robustness by implementing a separate unit to detect errors and designing accelerators without heterogeneity. At the same time, hardware-aware solutions utilize software to consider hardware issues in advance and deal with the robustness issue in PIM. We elaborate on software techniques, domain-specific accelerators, and co-design solutions for three critical challenges in the following sections.

II. Towards Efficiency

A. Efficient Federated Learning

One of the main challenges in federated learning (FL) is to achieve communication and computation efficiency concurrently. Research progress on communication efficiency has focused on two main strategies: quantization [30] and sparsification [29]. Both strategies aim to compress the parameters for transmission between devices. On the other hand, efforts to boost computation efficiency are made to reduce local training costs [13], [14] and accelerate global training convergence [12], [19]. In this context, we highlight two prominent methods, Hermes [13] and FedMask [14], which simultaneously address efficiency in both communication and computation.

1) Hermes: Hermes draws inspiration from the lottery ticket hypothesis [4]. Instead of updating the entire network on edge devices, Hermes identifies a structured-sparse subnetwork for each device via pruning. It then solely trains and communicates this subnetwork. Acknowledging that only segments



(a) Conventional Architecture (b) Processing-in-Memory

Fig. 2: Conceptual diagram of the difference between (a) conventional architecture and (b) PIM. Two dataflow types (WS and IS) are feasible in a PIM accelerator.

of the parameters overlap across devices, Hermes introduces an innovative aggregation scheme. Empirical data demonstrate that Hermes elevates inference accuracy by margins of 0.53% to 32.17% relative to conventional baselines. It also curtails communication costs by factors of 1.92× to 3.48×. Additionally, Hermes realizes a 1.83× acceleration in inference latency and achieves a notable 70% reduction in memory footprint.

2) FedMask: FedMask adeptly accomplishes both communication and computation efficiencies. At its core, it harnesses the over-parametrization inherent to deep neural networks (DNNs). Rather than training and transmitting the entirety of neural networks, FedMask centers on the training and communication of a binary mask. Subsequently, this mask is applied element-wise to a neural network with fixed parameters (Fig. 1). Contrasting with baseline methods that convey network parameters in 32-bit float32 format, FedMask astoundingly reduces overheads by a factor of 32, utilizing a concise 1-bit binary mask. Empirical results showcase that FedMask enhances inference accuracy by margins ranging from 2.43% to 28.47% when compared to conventional baselines. In tandem, it slashes communication costs by factors ranging from 32.25× to 34.48×, while also yielding computational savings from 1.37× to 2.44× during the training phase.

B. Processing-in-Memory Accelerators

As shown in Fig. 2(a), conventional architecture distinguishes memory and computing units into individual components. The data transfer across the separation incurs a bottleneck phenomenon because of the mismatch with computing speed, drawing a line in efficiency. Breaking through the limitation, hardware designers have focused on domain-specific accelerators on edge devices. Domain-specific accelerators can leverage existing hardware like graphics processing units (GPUs) or novel paradigms. Processing-in-memory (PIM) infusing the computing capability to memories (Fig. 2(b)) is an emerging but promising paradigm, as PIM alleviates resource waste by data movement. Resistive random-access memory (RRAM) is a representative eVNM, providing device merits like low power and small area consumption. More importantly, the integrability of eNVM with the array configuration boosts the efficiency of the matrix operations, as illustrated in Fig. 3(a). The memories remember weights as their conductance and adjust the stored conductance according to supplied voltage. According to Ohm's law and Kirchhoff's

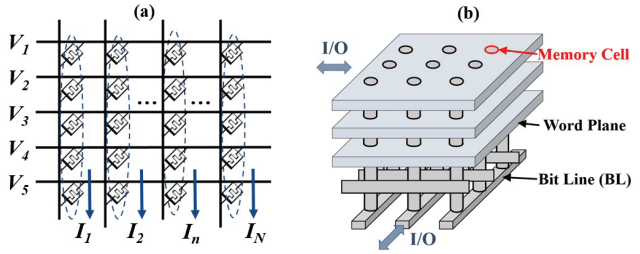


Fig. 3: Conceptual diagram of the difference between Von Neumann architecture and PIM.

law, RRAM produces currents with the supplied voltage, and the accumulated current of an array signifies the multiplication and sum of matrix operations. RRAM-applied PIM has been explored from extensive perspectives.

1) Supporting Advanced Learning Models: In early exploration, Hu et al. demonstrated the analogy of RRAM to the synaptic function [6] in the implementation of the brain-state-in-a-box (BSB) model. With the prosperity of deep neural networks (DNN), many RRAM-based PIM accelerators aim for diverse DNN models. For convolutional neural networks (CNNs), PipeLayer [21] proposes an accelerator for training through a pipeline technique. However, the inefficiency of training in inference, or vice versa, was captured in AtomLayer because of large on-chip buffers and bubbles in pipelining [18]. AtomLayer proposes a universal accelerator with atomic layer computation, that is, computing only one network layer each time. Memory-intensive CNNs bring about the appearance of software techniques like compression, binarization, and lightweight convolutions. Shortly after neural networks involve numerous but mostly unnecessary zeros (sparsity), pruning/compression techniques have been researched and supported in hardware. According to [5], sparse-skipping mechanisms are conducive to saving sparsity-relevant resources but introduce irregular data access patterns which demand large input buffers. The proposed cascading structured pruning technique induces predictable sparsity patterns so that the buffer size can be reduced only for unique activation data access. On the other hand, Kim et al. [9] design an accelerator optimized for binarization in networks. In [9], a simplified computing process is developed with presumption, and binarized weight convolution is accelerated through 3D architecture. As convolution variances in compact networks cause inefficiency in array utilization, Mobilattice [38] addresses the under-utilization issue of depth-wise convolution by hybrid digital/analog mode. Besides CNNs, other machine-learning techniques are also realized through RRAM-based PIM accelerators. GraphR [22] is designed with memory and graph engine (GE), and GE conducts graph computations in sparse matrix format. ReTransformer [35] accelerates the scaled dot-production attention for transformer models by removing data dependency through computation decomposition. The personalized recommendation algorithm is accelerated in [28] by optimization at the architecture level, inner-product engines, and at the algorithm level, an access-aware mapping algorithm.

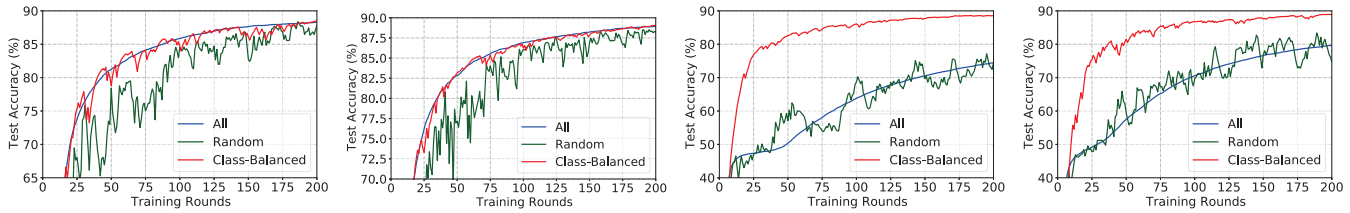
2) 3D Architecture: While PIM accelerators have widely adopted the 2D array configuration for efficiency in matrix multiplication, the potential of 3D architecture is witnessed because of its high bit-cost scalability and diverse designs for efficiency. Fig. 3(b) displays a vertically-stacked architecture, one of the representative and fundamental 3D designs. Different 3D RRAM structures have been proposed and demonstrated for pattern recognition in [10], [25], [26] by utilizing a reinforcement learning algorithm. The studies adopted the double-cell structure, where two devices are dedicated to a single value to express positive and negative values. While the double-cell design typically improves the signal margin but sacrifice area in 2D, the architectural advantage of 3D, i.e., area efficiency, compensated for the sacrifice, even achieving better efficiency than the 2D array design. Beyond simply reducing the occupied area, 3D can be exploited for efficiency with various design strategies. A 3D RRAM array design accelerates the binarized network by concurrently propagating computations with different inputs into corresponding layers in [9]. INCA [11], a state-of-the-art study, proposes a novel 3D RRAM design with a two-transistor-one-RRAM cell structure for input-stationary (IS) dataflow, as discussed in the subsequent explanation.

3) Dataflow: Other hardware types of accelerators (e.g., systolic arrays, GPUs) have been studied and verified the dataflow importance with different performances according to dataflow. However, PIM-based accelerators have shown little attention to dataflow by keeping weights in PIM units and fetching/storing inputs (activations) to separate memory units, which only induces a single dataflow, called weight-stationary (WS). INCA [11] catches the importance of dataflow in PIM accelerators and proposes an input-stationary (IS) dataflow for the first time (Fig. 2(b)). In INCA, inputs (activations) are placed RRAM PIM units to retain generated outputs in PIM, which will be consumed by the subsequent layer in a short time. By eliminating redundant accesses for activations and only demanding to fetch weights from buffers, INCA could save the number of buffer accesses and improve hardware efficiency. INCA further highlights the dataflow importance, providing insights that dataflow impacts the number of necessary RRAM cells, array utilization, and accuracy. According to INCA, RRAM cells can hardly be recycled in WS because weights are necessary until the end of the computations. WS also drops the array utilization in compact CNNs and degrades the accuracy due to the sensitivity of accuracy to weight parameters, which are affected by device noises in WS. INCA proves the RRAM-saving, utilization-, and accuracy-keeping effects in IS, by enabling to recycle RRAMs during the computations, fully utilizing RRAMs regardless of light weights, and less affecting accuracy which is immune to input (activation) variations by device noises.

III. Challenges on Heterogeneity

A. Heterogeneous data

Data heterogeneity has persistently posed challenges in FL, prompting a myriad of research endeavors. Historically,



(a) Global Balanced One-class (b) Global Balanced Two-class (c) Global Imbalanced One-class (d) Global Imbalanced Two-class

Fig. 4: Comparison of three client selection strategies: 'All', 'Random', and 'Class-Balanced', evaluated under four distinct scenarios (Figure 1 in [37]). Random selection harms the training performance as the selected clients are not guaranteed to provide class-balanced data in (a) and (b). Class-balanced selection can achieve better results under global imbalance scenarios.

the bulk of these efforts can be grouped into three main categories: modification of local training loss, refinement of global aggregation algorithms, and strategic device selection. The issue with data heterogeneity in FL arises as models tend to update in considerably varied directions. Consequently, some researchers [15] incorporate regularization terms into the local training loss to ensure training stability. Others [7] fine-tune the aggregation algorithms to curtail variance, while a distinct group [1], [2], [17] focuses on selective device iteration. However, when system heterogeneity appears, especially in such a computational resource variance, device selection emerges as a potent solution that addresses both forms of heterogeneity. This section will spotlight two contemporary exemplars, FedCor [27] and FedCBS [37], of device selection within FL.

1) Fed-CBS: Fed-CBS addresses the performance degradation resulting from global class imbalance across disparate devices. The authors introduce several foundational concepts: the "Local dataset", denoting individual device datasets; the "Global dataset", representing the amalgamation of all devices' datasets; and the "Group dataset", referring to the combined datasets of select devices. Initial experiments revealed a noteworthy impact of class imbalance on MNIST. Random device selection was found to adversely affect training performance, while a purposeful selection of a group dataset boasting balanced classes exhibited superior results, as depicted in Fig 4. Building on these observations, Fed-CBS advocates for client selection based on the *Quadratic Class-Imbalance Degree* — a novel metric presented in the paper that elucidates how a client's local label distribution influences global class imbalance in pairwise terms. Empirical evidence shows that Fed-CBS can enhance accuracy on CIFAR-10 [46] by margins of 2% to 7% and expedite the convergence rate by factors ranging from 1.3× to 2.8× when juxtaposed against the leading contemporary method [33].

2) FedCor: FedCor addresses the deceleration in convergence induced by class imbalance. Empirical findings underscore that an astute client selection strategy can markedly enhance the convergence rate of the Federated Learning (FL) procedure. The authors discerned that prior research [2] tends to undervalue the correlation of loss across clients, thereby only attaining incremental improvements over uniform selection. In response, the authors introduce FedCor—a novel FL framework grounded in a correlation-centric client selection approach, designed to amplify the FL convergence rate. At its core, FedCor is developed on the empirical observation that fluctuations in client losses across communication rounds can be aptly represented by a Gaussian Process (GP), depicted in Fig. 5. Intriguingly, correlations amongst clients are unveiled through the GP's covariance. The devised algorithm judiciously opts for clients that augment global training (characterized by high correlation coefficients relative to other clients) while eschewing repetitive or redundant selections (signified by low correlation coefficients with already chosen clients). Empirical evaluations elucidate that, when benchmarked against prevailing methods, FedCor bolsters convergence rates by 34%∼99% and 26%∼51% for FMNIST and CIFAR-10 datasets, respectively.

B. Heterogeneity Causing Noises and Costs in Hardware

As Fig. 6 displays, the integration of hardware components into a system should face and overcome heterogeneity across different levels; otherwise, it could fail to design a reliable and robust system. Other than the strategies introduced here, co-design methodologies to combat the combination of the issues are discussed in Section IV.

1) Device Level: Memory devices have diverse non-ideal properties, such as nonlinearity and process/device variation. Nonlinear devices generate output currents that are not linearly proportional to the applied voltage and could distort the multiplication results, causing inaccuracy. Process and device variations occur owing to the random diffusion and drift of ions and vacancies in devices. Process variation refers to varying data whenever accessing devices for reading and writing. The device-to-device variation is also unignorable because different cells of an array engender a variation range around

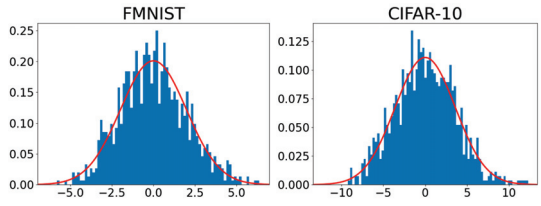


Fig. 5: Through the depicted histograms, the first principal component approximates a Gaussian distribution in FL under heterogeneous data conditions (Figure 2 in [27]).

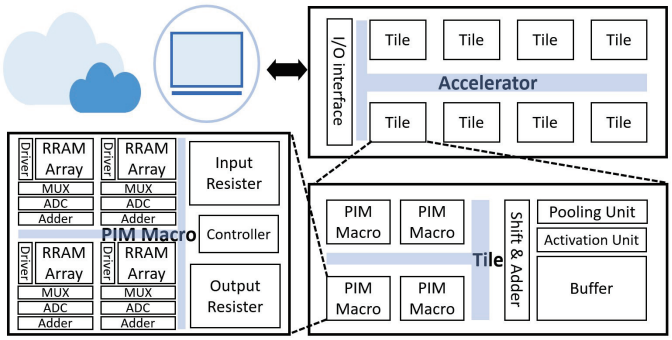


Fig. 6: Heterogeneity in the hardware integration into a system across hierarchy levels and components. The figure shows a typical RRAM-based PIM accelerator system hierarchy.

a targeted value, instead of a uniform value. Furthermore, it is well-known that eNVMs have low endurance, the number of program/erase operations to guarantee reliable data. The advance in device materials mitigates the issues by providing higher endurance [8].

2) Circuit Level: Circuit-level issues, i.e., IR drop, thermal/shot noises, and sneak-path current, are imperative when integrating memory devices into an array. The IR drop becomes severe as the array size increases, insufficiently delivering the applied voltage to the target device. Also, the electron's random movement by heat in conductance materials is unpredictable and uncontrollable, which causes thermal noise (a.k.a, Johnson–Nyquist noise). The shot noise occurs because of the discrete nature of electrons. Both noises stand out in the high operating frequency system, which is preferred for computing efficiency. The array structure with one memory device introduces the sneak-path current. The absence of "*switch*" induces the participation of unselected cells because the untargeted current is created independently and flows along with a targeted current. Nonlinearity can function as a switch in the 1R-based array structure, and accordingly, it is intentionally utilized to overcome the sneak path issue [10].

3) System Level: Peripheral circuits are necessary for smooth communication in a system, e.g., row selection circuits like a decoder and switch matrix for arrays and amplifiers not to lose signals during the propagation. In particular, co-existing heterogeneous analog and digital approaches in a system necessitate converters like analog-to-digital/digital-to-analog (ADC/DAC) for communication with other components. However, ADC/DAC are power-hungry components and demand huge areas, presenting a bottleneck in hardware. Although the advantage of analog computing in RRAM devices lies in multi-bit precision computing of high performance, the interpretation for multi-bit precision could cost even more than the computing expense. The circuit complexity of converters is also one of the design issues. To address the ADC/DAC overhead, only a single domain is adopted between analog and digital. Digital approaches with RRAM are presented by [9], [35]. Spiking-based circuits are another solution in the analog domain to eliminate the need for converters [16].

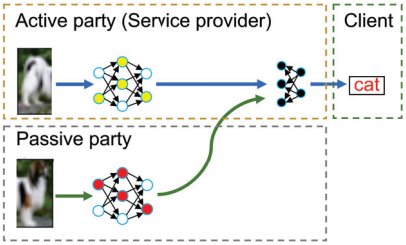


Fig. 7: Vertical Federated Learning (Figure 1 of [24]).

IV. Enhancement of Robustness

A. Improve Device Quitting Robustness

Robustness has been a prominent research focus within FL. Nonetheless, scant attention has been dedicated to robustness concerning unexpected device disengagement. The authors of [24] pioneer this avenue, concentrating on a unique FL context termed Vertical Federated Learning (VFL) [24]. Contrary to standard FL, which distributes data samples across clients, clients possess varying feature sets of overlapping individuals in VFL. Typically, one party, which has the labels of the overlapping samples and a subset of features, is designated as the active party. Meanwhile, other parties of only feature subsets are termed passive parties. While passive parties primarily hone feature extractors, training-wise active parties refine both feature extractors and classifiers which incorporate features from passive entities, as shown in Fig. 7.

At the inference juncture in VFL, it is imperative for clients to disclose their held features. This mandates that the aggregate performance be particularly vulnerable to device disconnections. As illustrated in Fig. 8, a conspicuous performance dip from 63% to 53% is observed when a party withdraws from collaboration in a standard setup. Addressing this, the authors advocate a 'party-specific dropout' strategy. The active party's training loss undergoes a transformation, culminating in multi-objective training, detailed as:

$$\mathbb{E}_{p^2,\ldots,p^K}\left[\mathcal{L}(\Theta; D)\right] = \sum_{\boldsymbol{z}=(z^2,\ldots,z^K)\in\{0,1\}^{K-1}} \prod_{k=2}^{K} (p^k)^{z^k}(1-p^k)^{(1-z^k)}\ell^{\boldsymbol{z}}, \quad (1)$$

where

$$\ell^{\boldsymbol{z}} = \sum_{i=1}^{N} \mathcal{L}(\mathcal{S}_{\theta_S}(H_i^1, \mathbb{1}_{\{z^2=0\}}H_i^2, \ldots, \mathbb{1}_{\{z^K=0\}}H_i^K), y_i),$$

and p^k is the probability to drop out party k, $\mathcal{S}_{\theta_S}$ is the predictor on active party, and H^k is the feature extracted by the feature extractor of k-th party. Empirical assessments in a two-party CIFAR10 context reveal that despite a modest performance decrement (from 75.1% to 73.7%), robustness, in the face of party 2's dropout, is considerably bolstered (from 53% to 62.7%).

B. Approaches in Hardware

Robustness can intensify by a separate physical unit (hardware-enhanced) and/or the exploit of software (hardware-

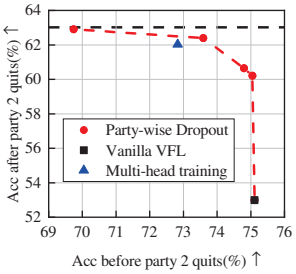


Fig. 8: Party-wise Dropout Impact on CIFAR10 (as referenced in Figure 3 of [24]). The black dashed line and 'Vanilla VFL' signify the accuracy achieved without party 2 and from training without resorting to party-specific dropout, respectively.

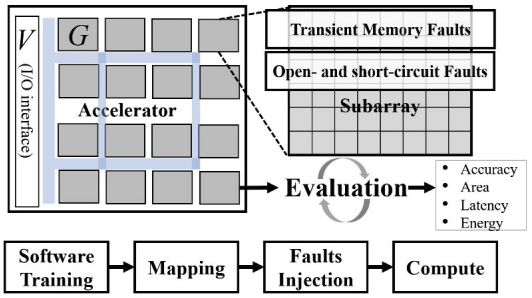


Fig. 9: Software/hardware co-design method for robustness.

aware) by reflecting the impacts of hardware issues in advance, as Fig. 9 exhibits the typical process of the co-design method.

1) Hardware-enhanced solutions: Robustness is a particularly critical issue in some applications, e.g., robotics and automation in mechanical, because missing errors could result in terrible outcomes. The errors can be detected and corrected by separate codes and hardware modules. In [3], an RRAM-based array is utilized as an error detection module that detects potential failures of other components. Furthermore, the study also tackles the nonideal scenarios by a precision crossbar array to ensure quantization accuracy which is synchronized with weight update for the precision match. The work further enhances hardware design for the programming variation with serial-connected RRAMs instead of single cells.

Since emerging devices have a critical weakness in endurance because of device immaturity, conventional memory-based PIM designs are also researched in hardware for the immediate use of PIM with high reliability. Unlike the multi-bit storage capability of emerging devices, conventional memories typically have one-bit storage per cell. To extricate from the communication costs, a state-of-the-art work [31] proposes an ADC-less SRAM CIM macro with the digital approach. The proposed SRAM-PIM design rarely modifies the cell structure and can take advantage of the commercial fabrication process with low costs. The design shows high throughput as 1.041 Mb/mm^2 and 27.38 TOPS/W.

2) Hardware-aware design solution: While expecting that the non-ideal properties improve with advanced materials and technologies, software is leveraged for hardware robustness. For noise tolerating in PIM, Yang et al. proposed to design an RRAM-based stochastic-noise-aware (ReSNA) training method [34]. Considering thermal and shot noises, ReSNA obtains the noise distribution at specific operating frequencies and temperatures. The amplitude of the distribution is analyzed to be interpreted as impact levels on the weight parameters. Then the proposed training process reflects the interpreted impact levels and PIM hardware configurations to combat the noise repercussion. The hardware-aware method in ReSNA can also deal with noises in inference by further considering random telegraph noise and programming noise

However, the only consideration of circuit-level issues is insufficient to ensure hardware for robust intelligence. In the case of training, frequent RRAM reprogramming is necessary for weight update, which makes the variation issues at the device level severe. The variations could lead to a failure of successful training and low accuracy, Driven by the issue, ESSENCE [36] proposes an endurance-considered training with a structured sparse gradient matrix in weight update. The number of programming operations is reduced by 10×. It is noteworthy that ESSENCE even provides higher stability in training results with the presence of variations. According to ESSENCE, accuracy improves 59% than the training method without under 2.0% the Gaussian distributed random noise, which is for modeling variations.

In a recent study, HERO [32], the generalization gap between training and inference is bounded by a l_2 weight perturbation. Gradient equations with the Hessian value are conducive to the minimization of the generalization gap and quantization loss simultaneously. Hence, HERO proposes to unify and optimize the performance and robustness in quantization with the Hessian-enhance regularization optimization method. The acquired more flattened loss surface around the weight convergence area proves the effectiveness of HERO. The proposed method enables PIM designs to have various precision bits in weight parameters with high quantization robustness.

V. Conclusion

Edge AI—*fusing edge computing and AI technologies*—offers tremendous opportunities in diversified applications. This paper provided insights on three crucial points, efficiency, heterogeneity, and robustness, which edge AI research faces and deals with. Specifically, we focused on two technologies, FL and PIM. FL is a promising solution for heterogeneity in a distributed system of edge AI. On the other hand, PIM has demonstrated its effectiveness for hardware in supporting advanced machine-learning algorithms. However, the inherent heterogeneity could cause the degradation of efficiency and robustness in both technologies. Various solutions for FL and PIM were summarized, including co-design approaches beyond separate efforts made in hardware and software.

Acknowledgment

This work is supported by NSF CNS-2112562, IIS-2140247, EPMD-1955246, CNS-1822085, and ARO W911NF-19-2-0107.

REFERENCES

[1] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning: Submodularity and convergence analysis," in *ICML 2021 International Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2021.

[2] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[3] G. Feng, B. Kim, and H. H. Li, "Bionic robust memristor-based artificial nociception system for robotics," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 3552–3556.

[4] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rJl-b3RcF7

[5] E. Hanson, S. Li, H. Li, and Y. Chen, "Cascading structured pruning: enabling high data reuse for sparse dnn accelerators," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 522–535.

[6] M. Hu, H. Li, Q. Wu, G. S. Rose, and Y. Chen, "Memristor crossbar based hardware realization of bsb recall function," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1–7.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[8] T. Kempen, R. Waser, and V. Rana, "50x endurance improvement in taox rram by extrinsic doping," in *2021 IEEE International Memory Workshop (IMW)*. IEEE, 2021, pp. 1–4.

[9] B. Kim, E. Hanson, and H. Li, "An efficient 3d reram convolution processor design for binarized weight networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 5, pp. 1600–1604, 2021.

[10] B. Kim and H. Li, "Leveraging 3d vertical rram to developing neuromorphic architecture for pattern classification," in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2020, pp. 258–263.

[11] B. Kim, S. Li, and H. Li, "Inca: Input-stationary dataflow at outside-the-box thinking about deep learning accelerators," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2023, pp. 29–41.

[12] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, 2021, pp. 19–35.

[13] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 420–437.

[14] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 42–55.

[15] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[16] Z. Li, Q. Zheng, Y. Chen, and H. Li, "Spikesen: Low-latency in-sensor-intelligence design with neuromorphic spiking neurons," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[17] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[18] X. Qiao, X. Cao, H. Yang, L. Song, and H. Li, "Atomlayer: A universal reram-based cnn accelerator with atomic layer computation," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.

[19] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[20] Samsung, "3 must-haves for intelligent manufacturing," https://www.forbes.com/sites/samsungsds/2020/01/06/3-must-haves-for-intelligent-manufacturing/?sh=8b6a359670ea, 2020, accessed: 2023-08.

[21] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE international symposium on high performance computer architecture (HPCA)*. IEEE, 2017, pp. 541–552.

[22] L. Song, Y. Zhuo, X. Qian, H. Li, and Y. Chen, "Graphr: Accelerating graph processing using reram," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2018, pp. 531–543.

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

[24] J. Sun, Z. Du, A. Dai, S. Baghersalimi, A. Amirshahi, D. Atienza, and Y. Chen, "Robust and ip-protecting vertical federated learning against unexpected quitting of parties," *arXiv preprint arXiv:2303.18178*, 2023.

[25] W. Sun, S. Choi, B. Kim, and J. Park, "Three-dimensional (3d) vertical resistive random-access memory (vrram) synapses for neural network systems," *Materials*, vol. 12, no. 20, p. 3451, 2019.

[26] W. Sun, S. Choi, B. Kim, and H. Shin, "Effect of initial synaptic state on pattern classification accuracy of 3d vertical resistive random access memory (vrram) synapses," *Journal of Nanoscience and Nanotechnology*, vol. 20, no. 8, pp. 4730–4734, 2020.

[27] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang, H. Li, and Y. Chen, "Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 102–10 111.

[28] Y. Wang, Z. Zhu, F. Chen, M. Ma, G. Dai, Y. Wang, H. Li, and Y. Chen, "Rerec: In-reram acceleration with access-aware mapping for personalized recommendation," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2021, pp. 1–9.

[29] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[30] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[31] B. Yan, J.-L. Hsu, P.-C. Yu, C.-C. Lee, Y. Zhang, W. Yue, G. Mei, Y. Yang, Y. Yang, H. Li *et al.*, "A 1.041-mb/mm 2 27.38-tops/w signed-int8 dynamic-logic-based adc-less sram compute-in-memory macro in 28nm with reconfigurable bitwise operation for ai and embedded applications," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 188–190.

[32] H. Yang, X. Yang, N. Z. Gong, and Y. Chen, "Hero: Hessian-enhanced robust optimization for unifying and improving generalization and quantization performance," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 25–30.

[33] M. Yang, X. Wang, H. Zhu, H. Wang, and H. Qian, "Federated learning with class imbalance reduction," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 2174–2178.

[34] X. Yang, S. Belakaria, B. K. Joardar, H. Yang, J. R. Doppa, P. P. Pande, K. Chakrabarty, and H. H. Li, "Multi-objective optimization of reram crossbars for robust dnn inferencing under stochastic noise," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2021, pp. 1–9.

[35] X. Yang, B. Yan, H. Li, and Y. Chen, "Retransformer: Reram-based processing-in-memory architecture for transformer acceleration," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[36] X. Yang, H. Yang, J. R. Doppa, P. P. Pande, K. Chakrabarty, and H. Li, "Essence: Exploiting structured stochastic gradient pruning for endurance-aware reram-based in-memory training systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[37] J. Zhang, A. Li, M. Tang, J. Sun, X. Chen, F. Zhang, C. Chen, Y. Chen, and H. Li, "Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction," in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 354–41 381.

[38] Q. Zheng, X. Li, Z. Wang, G. Sun, Y. Cai, R. Huang, Y. Chen, and H. Li, "Mobilatice: a depth-wise dcnn accelerator with hybrid digital/analog nonvolatile processing-in-memory block," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.