(Invited Paper) Co-Designing NVM-based Systems for Machine Learning and In-memory Search Applications

Jörg Henkel*, Lokesh Siddhu*, Hassan Nassar*, Lars Bauer, Jian-Jia Chen†, Christian Hakert†, Tristan Seidl†, Kuan Hsun Chen‡, Xiaobo Sharon Hu§, Mengyuan Li§, Chia-Lin Yang¶, Ming-Liang Wei¶
*Karlsruhe Institute of Technology, Germany, †TU Dortmund, Germany, ‡University of Twente, the Netherlands, §University of Notre Dame, Indiana, USA, ¶National Taiwan University, Taiwan

***Abstract*—With the rapid development of the Internet of Things, machine learning applications on edge devices with limited resources face challenges due to large data scales and irregular memory access patterns. Non-volatile memory (NVM) technologies provide promising solutions by offering larger capacity, low leakage power, and data persistence. In this paper, we discuss the potential of NVM technology in enhancing machine learning applications by improving energy efficiency and reducing latency through in-memory computation and different NVM write modes. The insights from this analysis provide valuable guidance to device researchers and system architects working to develop high-performance systems for machine learning and accelerators in large-scale search applications using NVMs.**

I. Introduction

In the rapidly evolving field of machine learning (ML), especially on edge devices with constrained resources, the integration of Non-Volatile Memories (NVMs) such as Phase Change Memory (PCM) offers notable advantages. PCM's larger capacity, low leakage power, and data persistence make it a promising candidate for enhancing ML capabilities at the edge. However, integrating PCM into edge devices is not without challenges, particularly concerning (slow) write performance and endurance. To address these obstacles, researchers have suggested using various PCM write modes that provide a balance between write latency and data retention. For instance, fast write mode significantly reduces write latency and energy consumption, but due to low retention time, it demands data refreshing. Addressing these trade-offs is essential for optimizing the role of PCM in edge ML applications.

Another NVM technology widely used in edge devices is Ferroelectric RAM (FeRAM), which offers fast access speeds and large capacities suitable for edge contexts. The ability of FeRAM to be used as byte-addressable memory allows software to directly store important runtime information inside of the FeRAM, which can assist fast transfer between power-saving states of the system. Especially for timing predictable ML applications, minimal overhead transitions between power-saving states allow for energetic optimization while not affecting the timeliness of the system. On the downside, FeRAM is a *read-destructive* memory, which increases wear and shortens its lifespan, necessitating careful management in critical ML applications.

Beyond edge ML applications, NVMs have also demonstrated potential in accelerating large-scale search operations, a common requirement in many machine learning tasks. The vast data scale and irregular memory access patterns involved in these operations present significant challenges. NVM-based content addressable memories (CAMs) have emerged as a promising compute in memory (CIM) search solution for these applications. However, selecting the optimal NVM devices and architectures for CAM-based accelerator is a complex task. At the architecture level, factors such as the size of the basic CAM array and the partition and merge schemes used to combine array results significantly influence application-level accuracy as well as the area, latency, and energy efficiency. Furthermore, read/write costs and device variability associated with different NVMs affect overall performance and accuracy. It is crucial to examine how various NVM device and architecture choices impact the performance of in-memory search accelerators. A thorough examination/research of how various NVM devices and architectural choices affect in-memory search accelerators helps both device researchers and system architects working to develop high-performance accelerators for large-scale search applications.

Deep learning models need advanced memory technologies like NVMs for efficient weight access and movement. Extensive computations between input vectors and weight matrices worsen efficiency challenges as matrices grow. In convolutional neural networks (CNNs), deeper layers heavily depend on weight access, causing performance bottlenecks. Similarly, in multi-label classification [1], the expansion of the number of categories results in a larger set of weights, further straining system performance. Memory-augmented neural networks [2], which are designed to support lifelong learning and prevent 'catastrophic forgetting' also require frequent access to large weight data, highlighting the need to address weight movement for efficient model inference.

Traditional approaches, such as using eDRAM in systems [3], attempt to mitigate these challenges but come with drawbacks including high area costs and increased energy consumption. Emerging techniques like processing-in-non-volatile memory offer a promising alternative by potentially eliminating the overhead associated with weight movement [4]. Non-

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10.
https://doi.org/10.1145/3676536.3697114

volatile memory crossbars, for example, can be used not only for storage but also for processing tasks, enabling vector-matrix multiplication directly in the analog domain through Kirchhoff's current law. However, processing in the analog domain presents challenges such as current variation and sensing accuracy issues, which must be addressed when deploying neural models on memory crossbars. Although prior research has focused mainly on optimizing the processing of static neural models and assumes operation at room temperature, the challenges associated with dynamic neural networks [5] and the reliability issues induced by thermal variations remain unexplored. This gap necessitates a rethinking of deployment methodologies to achieve superior performance and enhanced reliability across all neural models.

The remainder of the paper is organized as follows. Section II presents a comprehensive literature survey. Section III discusses various techniques for optimizing Non-Volatile Memory (NVM) usage, including write modes, data mapping, and in-memory computation. Section IV highlights the key challenges associated with these approaches, while Section V explores open research problems in the field. Finally, Section VI concludes the paper with a summary of the findings and insights.

II. Literature Survey

This section provides a comprehensive overview of recent advancements in NVM technologies and their applications in machine learning and search operations. To enable edge ML, we first explore strategies for optimizing NVM write modes and addressing associated challenges, including techniques for balancing write performance and data integrity. Then, we address the read-destructive issues of FeRAM and present solutions at both the technology and software levels. Next, we examine the use of CAMs in large-scale search tasks, highlighting CAM architecture and scalability innovations. Finally, we discuss the role of CIM technology in enhancing deep learning models, focusing on memory precision and performance improvements.

To accelerate execution when NVM is used as main memory, prior works focused on NVM write modes, Qiu et al. [6] introduce a loop-tiling method aimed at reducing retention time and utilizing the fast write mode extensively. Pan et al. [7] focus on addressing slow writes in embedded systems with NVMs by using scratchpad RAM to minimize latency. Chen et al. [8] investigate data encoding techniques to effectively manage retention time violations. Li et al. [9] propose the use of fast write instructions for single-core processors without caches, which are determined during compilation. Siddhu et al. [10] employ retention time profiling to add fast write instructions for single-core architectures. However, this approach lacks data refresh, posing a risk to data integrity during runtime variations. Finally, for multicore systems, Zhang et al. [11] propose QuicknDirty (QnD), a technique that refreshes fast written data during memory idle periods to ensure data integrity.

The read-destructive issue of FeRAM is, despite widely reported, tackled as a speicifc problem by Kato et al. on the technology level [12], where a nondestructive readout operation is proposed for FeRAM. Hakert et al. approach the issue from a software level, where they design explicit wear-leveling mechnaisms for destructive read and write systems [13]. Exploiting the fast transition between power states in a timing critical system with byte adressable FeRAM is proposed by Günzel et al. [14], where a timing model of task execution on different memory types is used to maximize hibernation period safely.

Another acceleration possibility with NVM is the usage of CAM for CIM. It is particularly suited for applications that require rapid and energy-efficient search operations [15]. CAMs excel in performing search operations by simultaneously comparing an input query against all stored data entries, enabling the identification of the matching memory entry in constant time. CAM can be built with different device technologies, including both conventional CMOS and emerging NVM devices. Ni et al. [16] summarize the TCAM designs built with various technologies. The conventional 16T CMOS-based CAM, which is volatile and occupies a large area, incurs footprint and leakage penalties. Emerging NVM technologies such as ReRAM, FeFET, and STTRAM have led to several TCAM designs that offer low-power, high-speed, and high-density benefits.

To overcome scalability challenges in CAM design, particularly for large-scale datasets, researchers have developed hierarchical CAM architectures [17], [18]. These architectures decompose the CAM-based accelerator into multiple layers, typically organized into a hierarchy of banks, mats, arrays, and subarrays. This hierarchical design allows CAMs to handle larger datasets by distributing data across multiple subarrays, which can operate in parallel.

In addition to the general acceleration possibilities from NVM, application-specific acceleration of Machine Learning via NVM also exists. With advancements in CIM technology, MAC operations can now be executed directly within memory crossbars or blocks, enabling storage devices to function as computational units. The deployment of CIM systems depends on the specific operations that they are designed to target.

When CIM targets an encoder, such as convolutional layers, the output precision becomes critical. Challenges such as cell current noise, OFF-state current, and sensing current limitations can impact the quality of the output [19], [20]. To prevent noise accumulation and manage the diverse current ranges required for sensing, only a limited number of inputs are processed at a time [4], [21]. The subset of the weight matrix corresponding to these inputs, which can be sensed by the analog to digital converter (ADC), is defined as an operational unit (OU). Consequently, the entire weight matrix is divided into sub-arrays, distributed across multiple OUs.

To enhance reliability without reducing OU size, one approach is to reorder the weight matrix to minimize accumulated current within the OU [22], thereby reducing cell noise. Another strategy involves adjusting weight values to decrease the number of cells in the ON state or employing bit-flipping techniques to increase sparsity [23]. For performance improvements, the weight layout can be reorganized to ensure full utilization of each OU [24], [25], thus reducing the total

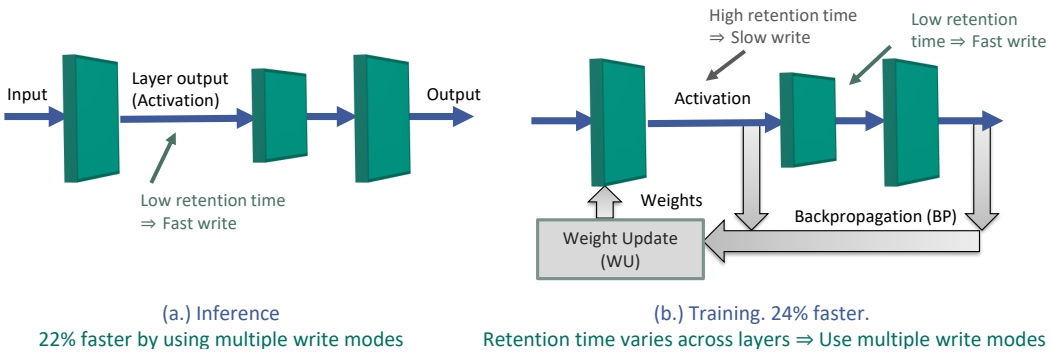

Fig. 1. CNN inference and training. Layer-wise write modes are suggested based on RTR.

number of OUs required. Additionally, remapping weights to achieve uniform sparsity across OUs helps maintain a consistent current distribution, preventing scenarios where near-zero currents dominate sensing time and prolong ADC sensing durations [26].

When CIM is targeted for search purposes, memory functions as a coarse filter with greater noise tolerance compared to the encoding phase [20]. However, multiple-bit sensing remains an issue to be addressed. One approach to addressing this issue is to train both input and weight data into binary forms [20], eliminating the need for multiple-bit sensing to digitize the partial sums of each input and weight bit. Since the output current represents the overall current, multiple-bit sensing can be replaced with a single-bit comparator.

Another approach with a similar concept involves projecting real number vectors or strings into a hyperspace [2], [27] or converting them into vector space by Locality Sensitive Hashing (LSH) [28] where each element is binary. Previous work utilized memory crossbars in memory-augmented neural networks, first projecting vectors into binary space and then conducting searches within this binary space. This method alters the sensing behavior by focusing on finding the maximal value instead of binarization, which helps to bypass the challenges of multiple-bit sensing.

III. Enhancing Computation and System Performance with Non-Volatile Memory

Integrating NVM enhances computation and performance in edge machine learning, search acceleration, and memory-centric deployments. Leveraging its persistent storage and energy efficiency, NVM addresses key architectural constraints. We focus on optimizing NVM with specialized write modes for edge applications, partitioning in search accelerators, and CIM for neural network processing.

A. Advancing Edge Machine Learning with Non-Volatile Memory Technologies

To advance edge machine learning, we explore first using PCM write modes to boost performance. PCM offers different write modes, balancing between write speed and retention time. Fast write modes, while quick, have a shorter retention [11]. We suggest utilizing fast writes for intermediate variables in machine learning programs, which allows us to reduce the overall execution time [10]. This is particularly vital for data-heavy tasks like CNNs when deployed at the edge e.g., federated learning applications [29], [30]). CNNs have data-independent access patterns that help us create reliable memory profiles and provide compile-time hints on the appropriate PCM write modes. The runtime system then uses these hints to assign write modes, improving performance.

As shown in Figure 1, for inference tasks, subsequent layers use data from previous layers, reducing the need for long retention times and allowing more use of fast writes. During training, layer outputs are used in backpropagation to calculate weights. The weight update happens last for layers near the input, so these layers have a higher retention time requirement and might need slow writing, while layers near the output can use fast writes. This flexible approach, with multiple write modes, helps meet varying retention time requirements and improves performance by more than 20% for both inference and training tasks [10]. By optimizing memory accesses in this way, we significantly enhance machine learning efficiency at the edge, addressing the growing demands of data-intensive applications.

A different angle of performance optimization is to adopt properties of the underlying memory in the executed software. Especially for NVM technologies, where latency and energy consumption depend on the memory access behavior of the ML application, software optimization can make a huge impact. ROLLED [31], considers domain-wall racetrack memories (DW-RTM) as scratchpad memory, where shift operations are required to align the memory with the access port. Depending on the distance of two subsequent memory accesses, the latency and energy overhead for the shift operation increases. As a machine learning model with a highly customizable memory access latency, ROLLED explores decision trees and random forests, as a normal inference of the model only requires a single path of a tree to be accessed. Proper memory layouting minimizes the expected amount of shift operations during execution of the trees.

B. Accelerating large-scale search problems using NVM-based in-memory computing

Designing CAM-based search accelerators for large-scale search tasks presents significant challenges, particularly due to the inherent limitations of NVM array sizes. These arrays are

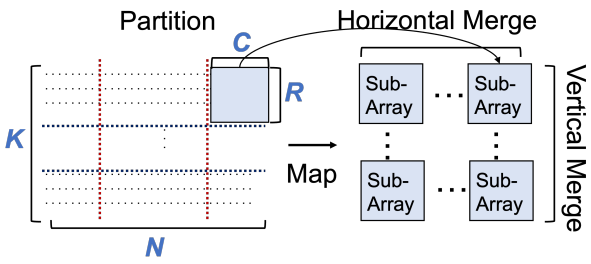

Fig. 2. Partition and merge for CAM-based in-memory accelerators.

typically too small to hold an entire dataset, making it crucial to partition the data across multiple arrays and efficiently merge the search results. The way this partitioning and merging is handled directly impacts the accuracy, speed, and overall efficiency of the search accelerator.

1) Partitioning and Merging: Given the limited size of NVM arrays, data must be divided across multiple subarrays within the CAM architecture. The partitioning process depends on the dimensions of the data and the capacity of the subarrays. Once the data is partitioned, the next step is to merge the search results from these subarrays to produce a coherent application-level result.

Horizontal Partition and Merge: When the data dimensions N exceeds the column C of a subarray as shown in Figure 2, the data must be horizontally split across several subarrays. For exact matches, merging the results with a simple AND operation across all subarrays can maintain high accuracy. However, for best match operations, where each subarray identifies a match independently, merging the results becomes more complex. A voting scheme can be used to approximate the global best match, but this introduces additional hardware complexity, increasing area and energy consumption. This trade-off is especially pronounced in threshold matching scenarios, where current horizontal merge strategies are inefficient, forcing designers to choose between accuracy and hardware simplicity.

Vertical Partition and Merge: When the number of entries K exceeds the row count R of a subarray Figure 2, vertical partitioning is required. For exact and threshold matches, simply aggregating results from all subarrays is often sufficient, though it may not be the most hardware-efficient approach. However, in best match scenarios, a comparator-based merge is necessary to determine the most accurate result across all entries. Although this method improves accuracy, it requires more complex circuitry, leading to increased latency and power consumption. The trade-off here lies in balancing the need for precise search results against the desire to minimize hardware costs.

2) Case Study: In this case study, we explore the performance of a CAM-based search accelerator designed for a classification task that involves high-dimensional embeddings. Given the high-dimensional nature of the data, where the embedding length N is significantly larger than the subarray column counts C, a horizontal partitioning and merging approach is necessary. Specifically, we evaluate the effectiveness of a voting scheme used to merge results across arrays.

We assess the accuracy and Energy-Delay Product (EDP)

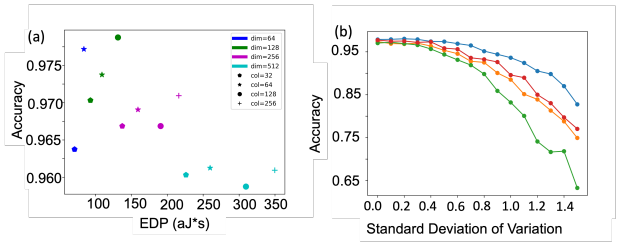

Fig. 3. (a) The impact of array column count on EDP and accuracy for different embedding length. (b)

across different embedding dimensions and subarray sizes (see Figure 3(a)). It can be seen that smaller embedding dimensions tend to achieve higher accuracy when using the same subarray column size. This is because fewer dimensions reduce the complexity and potential errors introduced by the voting scheme during merging. In addition, larger subarrays generally yield better accuracy for the same number of dimensions, as they allow more data to be compared within each subarray, minimizing partitioning errors. Additionally, by incorporating partition and merge schemes, we examine how device variation impacts overall accuracy Figure 3(b). For a fixed number of dimensions, smaller subarrays, despite offering better EFP, tend to be less resilient to these non-idealities. Conversely, for a fixed subarray column size, a smaller dimension increases vulnerability to these non-ideal effects.

C. Memory-Centric Deployment of Machine Learning Models

As neural networks become larger, the data movement between memory and processing units has emerged as a significant bottleneck for both performance and energy consumption. By eliminating memory access during inference, Computing-in-Memory (CIM) offers a substantial speedup and enhanced energy efficiency compared to traditional NPU designs. Figure 4 illustrates the performance comparison of the CIM-based accelerator, PRIME [32], and prior digital accelerators. The compared architectures include a CPU integrated with a co-processor NPU (denoted as pNPU-co) and an NPU stacked with DRAM using 3D stacking (denoted as pNPU-pim). To get an idea of the benefit we compare the latency and energy as follows.

1) Latency Comparison: Memory access time in both convolutional layers (CNN) and multi-layer perceptrons (MLP) dominates the processing time in conventional von Neumann architecture (pNPU-co), as shown in Figure 4(c). With the implementation of CIM, processing time can be significantly reduced by eliminating memory access. The evaluation of PRIME indicates that memory access time accounts for approximately 85% of the total time in CNN layers and up to 95% in MLP layers due to lower weight reuse. It also shows that with the CIM technique, memory access time becomes negligible. Additionally, CIM offers competitive parallelism compared to 64 NPU cores (pNPU-pim-x64). The latency comparison in Figure 4(a) demonstrates that PRIME achieves a 1.8x to 2.7x speedup for CNN layers and a 5x to 7.8x speedup for MLP layers compared to pNPU-pim-x64. Finally, in the VGG-D application, CIM significantly reduces memory access time by

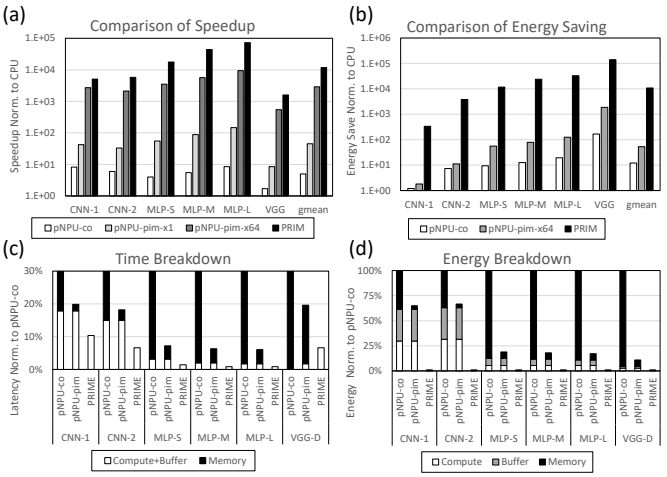

Fig. 4. Energy and Latency Evaluation from PRIME [32]

over 95%. Despite a 10% increase in computation time, PRIME still achieves around 3x speedup for VGG-D.

2) Energy Comparison: PRIME also demonstrates superior energy efficiency by reducing memory access and improving computing efficiency, as shown in Figure 4(b). The evaluation reveals that the CIM technique provides, on average, 206x greater energy efficiency than the conventional von Neumann architecture (pNPU-pim). The breakdown in Figure 4(d) shows that stacking the NPU with DRAM reduces processing energy by approximately 35% for CNNs and around 82% for MLPs due to near memory process. With the CIM technique, overall processing energy is reduced to less than 1.5%. Finally, by reducing the energy consumption of weight access and computation, PRIME achieves a remarkable 75x improvement in energy efficiency for VGG-D compared to pNPU-pim.

IV. Challenges for Deployment of NVM

Integrating NVM into large-scale computing systems offers significant opportunities and challenges. NVM technologies like PCM, ReRAM, and racetrack memories provide advantages such as persistent storage and potential energy efficiencies. However, these benefits are offset by challenges including limited write endurance, high energy consumption during writes, scaling issues in CAM-based search accelerators, and thermal variability affecting access patterns. Addressing these obstacles is crucial for advancing NVM technologies and integrating them into future computing systems.

A. Limited write endurance

Many NVM technologies, including PCM, ReRAM, and 3D XPoint, have lower write endurance than traditional DRAM. This means that they can only withstand a limited number of write operations before degrading, typically in the range of 10^5 to 10^6 cycles [33]. Improving the write endurance of NVM is crucial for their widespread adoption and longevity in computer systems.

B. High Energy Consumption & Latency for Write Operations

A significant challenge for Non-Volatile Memory (NVM) technologies is the high energy consumption and latency associated with write operations. Compared to traditional DRAM, NVM write operations require substantially more energy, often consuming 15-20 times more power. For example, flipping a single bit in PCM can consume around 50 pJ/b, while writing a DRAM page only needs about 1 pJ/b [33]. Additionally, write latencies in NVMs are typically higher than read latencies and can be orders of magnitude slower than DRAM writes. This asymmetry between read and write performance, coupled with high energy demands, poses significant challenges to system designers. It affects overall system performance, power efficiency, and thermal management, potentially limiting NVM adoption in power-constrained environments and performance-critical applications.

C. Beyond RAM

Certain NVM technologies, namely racetrack memories [34], not only deliver a novel cell technology, that can be assembled to a conventional memory array, but in contrast come with a novel semantic of accessing memory contents. Memory domains are placed in nanotracks, which are aligned with access heads. In order to access certain memory domains, the nanotrack has to be shifted to the correct position prior to an access. This, essentially, goes beyond the semantic of random access memory and imposes a more complex overhead model to the system.

Besides the performance and the energy overheads, the reliability issue on such racetrack memories goes also beyond the prior knowledge acquired for random access memory. Due to the behavior of shifts and alignments, a misalignment fault may occur, when the targeted data is not aligned after the shift operation, to an access port and incorrect data is read from the cell [35]. The NetDrift framework [34], shows that there is a potential to balance the trade-offs between acceptable drops in accuracy and enhancements in performance in binarized neural networks (BNNs) under such faults.

D. Scaling Issues in NVM-Based CAM Search Operations

With the growing importance of retrieval-augmented generation, search operations have become increasingly vital in applications involving embedding retrieval and database searches. However, as data scales up, significant challenges arise. CAM-based search accelerators, once promising for their speed and energy efficiency, are now struggling due to architectural limitations and the constraints of current NVM-based memory.

Although CAMs excel at fast brute-force searches on relatively small datasets, their effectiveness diminishes as the data scales to gigabyte levels. The question arises whether this search algorithm remains optimal for such large datasets. Additionally, when data sizes exceed the capacity of NVM-based memory, efficiently managing and processing these large-scale datasets within CAM structures becomes a significant hurdle. These challenges highlight the need for advances in both CAM architecture and NVM technology to sustain their relevance in large-scale applications.

E. Thermal Variability and Data-Dependent Access

Despite the benefits computing in non-volatile memory can provide, e.g., energy efficiency and shorter processing time

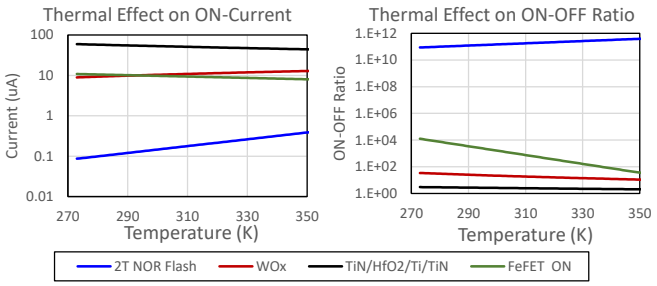

Fig. 5. Thermal Effect on ON-state current and ON-OFF Ratio

it faces several challenges that are not well explored. These challenges include the thermal issue [36] and loading balance for the trend of data-dependent access flow [1], [5], [37], [38].

1) Thermal Issue: In prior work, cell current has typically been assumed to be measured at room temperature. However, cell current can vary with environmental temperature [36]. As an example Figure 5 shows the changes in ON-state current and ON-OFF ratio for different types of emerging devices. Such variations significantly decrease the reliability of the computing system. This issue becomes even more critical with the use of 3D stacking techniques [39], [40], where heat generated by nearby logic circuits can exacerbate the problem. Thermal-aware MAC methodologies and hardware designs for MAC operations have not yet been thoroughly explored.

2) Data Dependent Flow: The neural model to be accessed depends on the output of the previous layer, which can reduce operations [5]. However, when deploying the model to a memory crossbar, the hardware may not fully benefit from this reduction. This is because the memory crossbar processes tasks where the corresponding weights are pre-programmed. When multiple tasks share hardware resources, such as buses or caches, contention can occur, leading to performance drops. Since the requirements of subsequent tasks are unknown until the previous layer's output is available, pre-programming weights to prevent resource contention remains a challenge.

V. Open Research Problems

NVM technologies offer transformative opportunities in edge computing, persistent memory, and in-memory computing. Optimizing NVM demands innovative solutions for durability, energy efficiency, and scalable algorithms in resource-limited environments. Integrating NVM with existing technologies necessitates thermal-aware model retraining and dynamic workload prediction for reliability and performance. Addressing these challenges is essential to harness NVM's potential for next-generation computing.

A. Usage in edge and IoT devices

NVMs offer significant advantages for edge and IoT devices due to their low power consumption, non-volatility, and high density. These features are well suited for environments with limited resources where energy efficiency is crucial [41]. However, there are still significant research challenges to address, such as improving write endurance for long-term reliability, developing efficient wear-leveling techniques for constrained devices, and enhancing security and privacy measures to protect sensitive data [42]. Resolving these issues is vital to maximize the effectiveness of NVMs in IoT applications and ensure their durability and security in edge computing environments. For example, to make racetrack memories more applicable, suitable light-weight countermeasures could be explored to exploit the trade-off between acceptable drops in accuracy and enhancements in performance, against the misalignment faults.

B. Persistent memory systems

NVMs enable the creation of persistent memory systems that bridge storage and memory, thereby improving performance and simplifying system designs. With direct access to persistent data structures, NVMs reduce the need for complex data movement between volatile and non-volatile memory. However, there are several research challenges that need to be addressed, including designing efficient crash consistency mechanisms, optimizing data placement and movement across memory tiers, and developing new programming models and abstractions. These advances are crucial to fully exploiting persistent memory capabilities, ensuring ease of use for developers, and maintaining system reliability and efficiency. Resolving these issues will be essential to advance persistent memory technology.

C. Specific Integration and Utilization

NVM comes in various forms, requiring specialized optimization methodologies, especially for ML applications. Deciding on memory type and configuration, and mapping it to software remains an open research question. One approach is hybrid memory systems, where multiple memory types are available to the software, necessitating proper management and holistic optimization. Another is abstracting memory properties from the software, handling optimization and management transparently in hardware or system software. A major challenge is reacting to different application behaviors.

D. Scalable In-Memory Computing for Large-Scale Search Problems: Algorithm and Hardware Co-Design

While CAMs excel at parallel searches for smaller datasets, their performance drops with gigabyte-level data due to the inefficiency of brute-force approaches. To address this, there is a growing need for algorithm-hardware co-design, where search algorithms are tailored to the hardware's capabilities and constraints. By developing hybrid search strategies and optimizing algorithms specifically for CAM architectures, it is possible to enhance scalability and maintain efficiency in large-scale applications.

E. Thermal-Aware Re-training of NVM-CIM

Deploying CIM in real-world applications presents a significant challenge due to the thermal effects on NVM cell current, which can fluctuate with temperature and undermine system reliability. A straightforward approach to mitigate this issue involves using reference memory cells as thermal sensors and designing thermal-aware sensing circuits or calibration schemes. However, this approach introduces additional circuit

overhead, increasing energy consumption, and area requirements, potentially negating the benefits of CIM. Therefore, a critical challenge lies in addressing thermal issues without incurring significant circuit design overhead. One promising solution is to train models to be resilient to temperature variations. By retraining models to tolerate thermal effects, it is possible to achieve reliable classification results with minimal changes to circuit design.

F. Dynamic Neural Model Deployment: Workload prediction

To develop a reliable and dynamic NVM-CIM based accelerator, workload prediction is a key research area. In scenarios where the execution of certain model components depends on the outputs of previous layers, accurately predicting which portions of the weight matrix will be accessed simultaneously is challenging. This uncertainty complicates the deployment of weights in a way that minimizes contention for shared resources. For instance, in systems where CIM devices are organized into multiple channels, with devices within each channel sharing a bus, assigning co-accessed weights to a single channel can lead to bus contention, leaving other channels underutilized. However, if co-accessed weights can be accurately predicted, they can be strategically distributed across different hardware resources, such as separate channels, to prevent contention. Therefore, accurately predicting co-accessed weights is crucial for optimizing deployment and ensuring efficient system operation.

VI. Summary

In edge machine learning applications with limited resources, non-volatile memories (NVMs) like Phase Change Memory (PCM) offer significant advantages, including large capacity, low leakage power, and data persistence. However, these benefits come with challenges such as slow write performance and endurance issues. To address these, researchers have explored PCM write modes that balance write latency and retention time. This paper suggests leveraging these modes to optimize performance, particularly in CNNs used in edge and federated learning. By utilizing fast write modes for intermediate variables, we achieve over 20% improvement in execution time and performance for both inference and training tasks. Although fast writes are efficient, they have shorter retention times, so we optimize their use based on data patterns and memory profiles to enhance overall system efficiency. Despite these advancements, NVMs still face challenges compared to DRAM, requiring 15-20 times more power for writes and exhibiting higher write latencies, which impact system performance and power efficiency. Continued research is crucial for adopting NVMs in power-constrained, performance-critical environments.

Extending the focus on PCM, large-scale search tasks in machine learning also encounter challenges due to vast data scales and irregular memory access patterns. NVM-based Content Addressable Memories (CAMs) offer a promising solution for fast, energy-efficient searches by eliminating data access bottlenecks. However, selecting the optimal NVM devices and CAM architectures involves balancing trade-offs among accuracy, area, latency, and energy efficiency. Hierarchical CAM architectures can manage larger datasets, but as data scales continue to grow, traditional brute-force methods and existing NVM capacities struggle to keep pace. Future research must concentrate on enhancing algorithm-hardware co-design and advancing NVM technologies to sustain the effectiveness of CAM-based accelerators in large-scale applications.

Transitioning from search tasks to broader deep learning applications, Computing-in-Memory (CIM) technology emerges as a promising solution for improving energy efficiency and reducing latency by enabling in-memory computation, thus eliminating the need for weight movement. However, several challenges remain, including the impact of thermal effects on cell current, which can compromise system reliability, and memory access contention issues in dynamic neural networks.

This paper identifies key open research problems, emphasizing the need for thermal-aware model retraining to ensure models can tolerate temperature variations without requiring significant circuit design changes. It also underscores the importance of developing dynamic neural model deployment strategies to optimize resource utilization and prevent operational conflicts in CIM-based systems. Addressing these challenges is essential for the successful deployment of reliable and adaptive CIM-based accelerators in real-world applications.

VII. Acknowledgments

This work is supported in part by the German Research Foundation (DFG) as part of the priority program "SPP 2377: Disruptive Memory Technologies" under projects: *Reconfigurable Architectures and Real-Time Systems Co-Design for Non-Volatile Memory (ARTS-NVM)*. We thank DFG (Project Number: 405422836, NVM-OMA). We would like to acknowledge the National Science and Technology Council of Taiwan (Grant NSTC 111-2923-E-002-014-MY3 and Grant NSTC 112-2221-E-002-116-MY3). This work was supported in part by the Logic and Memory Devices program of SemiconductorResearch Corporation (SRC).

References

[1] S. Li, F. Tu, L. Liu, J. Lin, Z. Wang, Y. Kang, Y. Ding, and Y. Xie, "Ecssd: Hardware/data layout co-designed in-storage-computing architecture for extreme classification," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14.

[2] G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Robust high-dimensional memory-augmented neural networks," *Nature communications*, vol. 12, no. 1, p. 2468, 2021.

[3] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun *et al.*, "Dadiannao: A machine-learning supercomputer," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2014, pp. 609–622.

[4] C.-X. Xue, W.-H. Chen, J.-S. Liu, J.-F. Li, W.-Y. Lin, W.-E. Lin, J.-H. Wang, W.-C. Wei, T.-W. Chang, T.-C. Chang *et al.*, "24.1 a 1mb multibit reram computing-in-memory macro with 14.6 ns parallel mac computing time for cnn based ai edge processors," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 388–390.

[5] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, 2021.

[6] K. Qiu, Q. Li, and C. J. Xue, "Write mode aware loop tiling for high performance low power volatile PCM," in *DAC*, 2014.

[7] C. Pan, M. Xie, C. Yang, Y. Chen, and J. Hu, "Exploiting multiple write modes of nonvolatile main memory in embedded systems," *TECS*, 2017.

[8] Y.-S. Chen, Y.-H. Chang, and T.-W. Kuo, "DTC: A drift-tolerant coding to improve the performance and energy efficiency of multi-level-cell phase-change memory," *TCAD*, 2023.

[9] Q. Li, L. Jiang, Y. Zhang, Y. He, and C. J. Xue, "Compiler directed write-mode selection for high performance low power volatile PCM," in *LCTES*, 2013.

[10] L. Siddhu, H. Nassar, L. Bauer, C. Hakert, N. Hölscher, J.-J. Chen, and J. Henkel, "Swift-CNN: Leveraging PCM memory's fast write mode to accelerate CNNs," *ESL*, 2023.

[11] M. Zhang, L. Zhang, L. Jiang, F. T. Chong, and Z. Liu, "Quick-and-Dirty: An architecture for high-performance temporary short writes in MLC PCM," *TC*, 2019.

[12] Y. Kato, Y. Kaneko, H. Tanaka, K. Kaibara, S. Koyama, K. Isogai, T. Yamada, and Y. Shimada, "Overview and future challenge of ferroelectric random access memory technologies," *Japanese Journal of Applied Physics*, vol. 46, no. 4S, p. 2157, 2007.

[13] C. Hakert, K.-H. Chen, H. Schirmeier, L. Bauer, P. R. Genssler, G. von der Brüggen, H. Amrouch, J. Henkel, and J.-J. Chen, "Software-managed read and write wear-leveling for non-volatile main memory," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 1, pp. 1–24, 2022.

[14] M. Günzel, C. Hakert, K.-H. Chen, and J.-J. Chen, "Heart: H ybrid memory and e nergy-a ware r eal-t ime scheduling for multi-processor systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5s, pp. 1–23, 2021.

[15] X. S. Hu, M. Niemier, A. Kazemi, A. F. Laguna, K. Ni, R. Rajaei, M. M. Sharifi, and X. Yin, "In-memory computing with associative memories: A cross-layer perspective," in *2021 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2021, pp. 25–2.

[16] K. Ni, X. Yin, A. F. Laguna, S. Joshi, S. Dünkel, M. Trentzsch, J. Müller, S. Beyer, M. Niemier, X. S. Hu, and S. Datta, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.

[17] A. Kazemi, F. Müller, M. M. Sharifi, H. Errahmouni, G. Gerlach, T. Kämpfe, M. Imani, X. S. Hu, and M. Niemier, "Achieving software-equivalent accuracy for hyperdimensional computing with ferroelectric-based in-memory computing," *Scientific reports*, vol. 12, no. 1, p. 19201, 2022.

[18] H. Farzaneh, J. P. C. De Lima, M. Li, A. A. Khan, X. S. Hu, and J. Castrillon, "C4cam: A compiler for cam-based in-memory accelerators," in *International Conference on Architectural Support for Programming Languages and Operating Systems*. New York, NY, USA: Association for Computing Machinery, 2024, p. 164–177. [Online]. Available: https://doi.org/10.1145/3620666.3651386

[19] M.-L. Wei, H. Amrouch, C.-L. Sung, H.-T. Lue, C.-L. Yang, K.-C. Wang, and C.-Y. Lu, "Robust brain-inspired computing: On the reliability of spiking neural network using emerging non-volatile synapses," in *2021 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2021, pp. 1–8.

[20] M.-L. Wei, H.-T. Lue, S.-Y. Ho, Y.-P. Lin, T.-H. Hsu, C.-C. Hsieh, Y.-C. Li, T.-H. Yeh, S.-H. Chen, Y.-H. Jhu *et al.*, "Analog computing in memory (cim) technique for general matrix multiplication (gemm) to support deep neural network (dnn) and cosine similarity search computing using 3d and-type nor flash devices," in *2022 International Electron Devices Meeting (IEDM)*. IEEE, 2022, pp. 33–3.

[21] T.-H. Yang, H.-Y. Cheng, C.-L. Yang, I.-C. Tseng, H.-W. Hu, H.-S. Chang, and H.-P. Li, "Sparse reram engine: Joint exploration of activation and weight sparsity in compressed neural networks," in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 236–249.

[22] Y. Xu, S. Jin, Y. Wang, and Y. Qi, "Aggressive fault tolerance for memristor crossbar-based neural network accelerators by operational unit level weight mapping," *IEEE Access*, vol. 9, pp. 102 828–102 834, 2021.

[23] F. Liu, W. Zhao, Z. Wang, Y. Chen, X. Liang, and L. Jiang, "Era-bs: Boosting the efficiency of reram-based pim accelerator with fine-grained bit-level sparsity," *IEEE Transactions on Computers*, 2023.

[24] C.-Y. Tsai, C.-F. Nien, T.-C. Yu, H.-Y. Yeh, and H.-Y. Cheng, "Repim: Joint exploitation of activation and weight repetitions for in-reram dnn acceleration," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 589–594.

[25] Y. Wang, F. Tu, L. Liu, S. Wei, Y. Xie, and S. Yin, "Spcim: Sparsity-balanced practical cim accelerator with optimized spatial-temporal multi-macro utilization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 1, pp. 214–227, 2022.

[26] D. E. Kim, A. Ankit, C. Wang, and K. Roy, "Samba: sparsity aware in-memory computing based machine learning accelerator," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2615–2627, 2023.

[27] Z. Zou, H. Alimohamadi, F. Imani, Y. Kim, and M. Imani, "Spiking hyperdimensional network: Neuromorphic models integrated with memory-inspired framework," *arXiv preprint arXiv:2110.00214*, 2021.

[28] R. Mao, B. Wen, A. Kazemi, Y. Zhao, A. F. Laguna, R. Lin, N. Wong, M. Niemier, X. S. Hu, X. Sheng *et al.*, "Experimentally validated memristive memory augmented neural network with efficient hashing and similarity search," *Nature communications*, vol. 13, no. 1, p. 6284, 2022.

[29] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," *NeurIPS*, 2020.

[30] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, 2021.

[31] C. Hakert, A. A. Khan, K.-H. Chen, F. Hameed, J. Castrillon, and J.-J. Chen, "Rolled: Racetrack memory optimized linear layout and efficient decomposition of decision trees," *IEEE Transactions on Computers*, vol. 72, no. 5, pp. 1488–1502, 2022.

[32] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 27–39, 2016.

[33] S. Kargar and F. Nawab, "Challenges and future directions for energy, latency, and lifetime improvements in nvms," *Distributed and Parallel Databases*, vol. 41, no. 3, pp. 163–189, 2023.

[34] B. L. David, M. Yayla, J.-J. Chen, K.-H. Chen, and A. A. Khan, "Evaluating the impact of racetrack memory misalignment faults on bnns performance," in *Embedded Computer Systems: Architectures, Modeling, and Simulation - 24th International Conference, SAMOS*, 2024, in press.

[35] C. Zhang, G. Sun, X. Zhang, W. Zhang, W. Zhao, T. Wang, Y. Liang, Y. Liu, Y. Wang, and J. Shu, "Hi-fi playback: tolerating position errors in shift operations of racetrack memory," in *Proceedings of the 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015, p. 694–706.

[36] M.-L. Wei, M. Yayla, S.-Y. Ho, J.-J. Chen, H. Amrouch, and C.-L. Yang, "Impact of non-volatile memory cells on spiking neural network annealing machine with in-situ synapse processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[37] G. Huang, "Dynamic neural networks: advantages and challenges," *National Science Review*, p. nwae088, 2024.

[38] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, pp. 275–293, 2014.

[39] K. Puttaswamy and G. H. Loh, "Thermal analysis of a 3d die-stacked high-performance microprocessor," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*, 2006, pp. 19–24.

[40] L. Siddhu, R. Kedia, S. Pandey, M. Rapp, A. Pathania, J. Henkel, and P. R. Panda, "Comet: An integrated interval thermal simulation toolchain for 2d, 2.5 d, and 3d processor-memory systems," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 19, no. 3, pp. 1–25, 2022.

[41] Y. Liu, S. Zhao, W. Chen, X. Ge, F. Liu, S. Li, and N. Xiao, "Nvm storage in iot devices: Opportunities and challenges." *Computer Systems Science & Engineering*, vol. 38, no. 3, 2021.

[42] H. Nassar, L. Bauer, and J. Henkel, "Anv-puf: Machine-learning-resilient nvm-based arbiter puf," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 5s, pp. 1–23, 2023.