International Journal of Review Computing and Information Technology (IJRCIT)
ISSN: XXXX-XXXX / Volume 1 Issue 1 July- September 2026
Paper Id: IJRCIT-V1I1P107/ Page No: 105-121
Doi: XX.XXXXX/XXXXXXX/ IJRCIT-V1I1P107

Original Article

Computing-in-Memory Architectures for Deep Learning Acceleration: Recent Advances and Future Trends

Samvel Abrahamyan[1], Sargis Stepanyan[2], Hayk Sargsyan[3]

[1,2,3]*Yerevan State University (YSU), Armenia.*

Received Date: 09 July 2026 Revised Date: 17 July 2026 Accepted Date: 25 July 2026

Abstract: *Deep Learning has seen rapid advancements in the state-of-the-art across numerous fields: computer vision, natural language processing, healthcare, robotics, autonomous systems and scientific computing. Nonetheless, the increasing depth of deep neural networks has dramatically increased their computational requirements, memory needs and energy consumption. Conventional von Neumann computing architectures eventually run into the memory wall problem — that is, data movement between processing elements and memory becomes a significant performance bottleneck. This can be counterproductive for the most modern artificial intelligence systems especially those involving large-scale neural networks and edge AI applications. Computing-in-Memory (CIM) has been established as a breakthrough computing paradigm for in-situ computation supplemented with the necessary data read and write movements within memory arrays, having better computational efficiency by minimizing the data transfer overhead.*

The Computing-in-Memory (CIM) architectures employ the emerging technologies in memory devices (such as SRAM, DRAM, Resistive RAM (ReRAM), Phase Change Memory (PCM), Magnetic RAM (MRAM), and memristive devices) to perform arithmetic and neural network computations inside the memory building blocks itself. The architectures benefit them with energy efficiency, throughput, latency reduction and hardware coverage. Most of the ongoing works on CIM systems for deep learning workloads due to recent advances in analog computing, neuromorphic engineering and in memory AI accelerators. Great strides have been made, yet challenges such as device variability, accuracy limits, scalability concerns, reliability issues and software-hardware co-design still remain.

This paper provides an overview of Computing-in-Memory architectures for the Acceleration of Deep Learning. This review covers the architectural foundations, memory technologies, neural network mapping strategies, hardware accelerators, energ/efficiency optimizations and security concerns of in-memory computing based deep learning accelerators for emerging applications. Based on this, the paper then discusses the current research challenges and future directions that will influence the evolution of CIM-enabled artificial intelligence systems. These findings emphasises that Computing-in-memory provides a unique avenue towards providing next generation AI hardware with sustainable, scalable and energy efficient end-to-end deep learning systems.

Keywords: *Computing-In-Memory, Deep Learning Acceleration, Artificial Intelligence Hardware, Ai Accelerators Edge,Ai Neural Networks Hardware Optimization Energy-Efficient Computing In-Memory Processing Emerging Memory Technologies.*

I. INTRODUCTION TO COMPUTING-IN-MEMORY FOR DEEP LEARNING

Deep learning is one form of modern computing that allows machines to do complex tasks like understanding, navigating and responding to sounds, actions or objects in ways that one thought only the human brain could handle. Deep neural network architectures have brought improvements to various applications from image recognition351130457, speech processing32, and natural language understanding3 further4 recommendations54 autonomous driving continuous6 medical diagnosis7 and scientific discovery8. Such networks typically have millions or billions of parameters that are very expensive in terms of both training and inference. In its current development stage, using conventional computing architectures is challenging in terms of achieving sufficient performance and energy efficiency levels to be used for next-generation artificial intelligence (AI) applications, particularly as model complexity continues to increase.

Traditional computing systems are mainly based on the Von Neumann architecture, which separates memory from processing units in distinct components. As the application runs, data flows in and out of memory over communication buses to processors. This architecture is the basis for modern computing for decades now, but also creates a major bottleneck Which we refer to as the memory wall. The memory wall is the increasing gap in speed between processors and memory access. Due to the ever-increasing power of processors, the data transfer between memory and computation units is becoming a major factor in overall system performance and energy usage. For many deep learning workloads — processes

that need to repeatedly read and manipulate large matrices and tensors of data, often using the same value numerous times — movement of this data takes equal or more energy than performing the actual computations.

The bottleneck caused by the memory wall has led to the aforementioned work on programming models which can achieve high performance without incurring large costs for data movement. In terms of best prospects, Computing-in-Memory (CIM)—which brings computation into memory arrays—ranks among method. Instead of transporting data from memory to processing units, architectures based on CIM on simulate arithmetic operations at the location where the data is stored. This paradigm enables even greater reduction of memory access latencies, communication overheads and energy consumption at the cost of dramatically improving throughput especially for data-intensive workloads. This has led to Computing-in-Memory as a promising fit for efficient deep learning acceleration.

At its heart, Computing-in-Memory is about exploiting the physics of memory devices to perform computation. Examples include memory cells that additively perform multiplication, accumulation, vector-matrix multiplication and other logic functions in memory structures. Given that deep neural networks are built upon a large number of simultaneous matrix-vector products, CIM architectures in particular are especially attractive for accelerating artificial intelligence workloads. Such systems can reduce the data kept and extracted from memory, and with better parallelism these systems give an order or two of performance increase when compared to traditional processor-oriented architectures.

Development of Computing-in-Memory systems has gained significant traction in recent years due to advances in semiconductor technologies. Introduction Emerging memory devices like Resistive Random Access Memory (ReRAM), Phase Change Memory (PCM), Magnetic Random Access Memory (MRAM) and memristors, offer opportunities for low-energy in-memory computation. These technologies are compatible with analog and digital computation architectures that can perform large-scale neural network functions using high parallelism and low energy inputs. Furthermore, standard memory fabrication circuits like SRAM and DRAM have already adapted for in-memory processing using circuit-level or architectural modifications.

The CIM architectures can be broadly segmented into analog and digital systems. Analog CIM takes advantage of the inherent electrical properties of memory devices to use analog inputs to carry out computations with continuous value representations. While these architectures provide outstanding efficiency and computational density, they suffer from issues of precision, noise, and device variability. Digital CIM architectures are based on discrete computational mechanisms offering greater precision and fidelity with lower data movement latency. Most importantly, the information which one to choose depends on the application requirements, performance goals, and any hardware limitations.

On top of this, the increasing edge AI demand has also helped spark interest in Computing-in-Memory technologies. Such as smartphones, wearable sensors, autonomous drones, industrial controllers and Internet of Things platforms generally need relatively low-power AI processing capabilities. Conventional processing in the cloud incurs latency, is still bandwidth constrained, and privacy challenged. AbstractCIM-based accelerators is a compelling solution for on-device intelligence due to their ability to execute neural networks in high-performance manner under strict energy and resource constraints. As a result, Computing-in-Memory is making an important part of charging next-generation edge AI infrastructures.

Despite its significant advantages, Computing-in-Memory is still a very active research area with lots of technical challenges to address. System performance is still affected by device non-idealities, limited precision, programming complexity and scalability, thermal effects and integration. Additionally, software frameworks and deep learning models need to be modified to take full advantage of CIM hardware. Tackling these problems will require deep cross-domain collaboration between communities in computer architecture, semiconductor engineering, machine learning, materials science and electronic design automation.

Overall, Computing-in-Memory opens a whole new era of computing architectures and can overcome the critical limitations of conventional ones. CIM achieves up to an order of magnitude improvements in energy efficiency, throughput and scalability for deep learning applications by moving computation closer to the locations where data is stored. Computing-in-Memory technologies, on the other hand, are expected to play a fundamental part in making them possible as these artificial intelligence systems become larger and ever more critical in society. The results will play a key role in the high-performance and sustainable AI systems to be implemented via their comprehensive integration into AI accelerators, edge devices and large-scale datacenters.

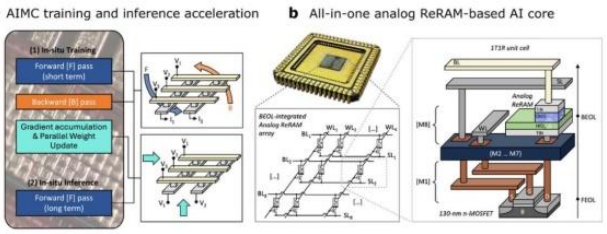

Figure 1: Re Ram-Based Computing-In-Memory Architecture

II. THE RISE OF AI HARDWARE AND THE MEMORY BOTTLENECK

The exponential improvement of artificial intelligence has been accompanied by the technological advances in computing hardware. Recent advances in processor technology, memory systems and semiconductor manufacturing over the past few decades have supported increasingly advanced machine learning and deep learning applications. At first, artificial intelligence systems used general-purpose Central Processing Units (CPUs) which are intended for a broad category of computational tasks. CPUs could be programmed flexibly, but the sequential nature of CPU matrix multiplication and limited parallelism made them inefficient at performing the huge levels of parallel processing achieved during training by modern neural networks. With deep learning models complexities growing, researchers started looking for specialized hardware architectures which could accelerate AI workloads and alleviate the performance bottlenecks of traditional computing systems.

The second major paradigm in AI hardware was the move to Graphics Processing Units (GPUs). GPUs were initially designed to render graphics and are part of the parallel computing hardware family; they consist of hundreds or even thousands of small core processors that can simultaneously execute a large set of operations. Deep learning algorithms like matrix multiplication and tensor computations in deep networks can be modeled as highly parallelized programs that map very well to the acceleration capabilities of GPUs. The time it took to train deep neural networks had been significantly reduced with the introduction of GPU-based training, allowing advancements in tasks such as image classification, speech recognition, natural language processing and reinforcement learning. GPUs revolutionized artificial intelligence research and laid the groundwork for hardware acceleration being an essential part of building AI systems today.

Advancement for artificial intelligence applications made new hardware solutions available to optimize performance and energy efficiency even more. FPGAs came into the mainstream, but for a much more handy appropriate they used to give customized hardware acceleration. In a contrast to the non-configurable architectures of fixed-function processors (FPGAs can be customized at run-time to optimize parts of the application-specific architecture that is better suited for neural network execution. This flexibility enables developers to optimize performance, power consumption, and hardware resource utilization for specific needs. While FPGAs tend to have lower performance than specialized AI processors, they still provide great value at the edge and for embedded AI applications where flexibility and power/performance are key design considerations.

As demand for AI processing has increased, ASICs capable of efficiently performing specific types of machine learning workloads. For example, Tensor Processing Units (TPUs) and other types of neural processing units built by industry leaders. They are specialized accelerators for deep learning operations like matrix multiplication, convolution and tensor manipulation. Over CPU and GPU, ASIC-based AI processors have higher computational throughput and energy efficiency by discarding unnecessary general-purpose functionality. These accelerators have become a fundamental building block of large-scale cloud computing infrastructures supporting modern AI services and applications.

Even with specialized hardware accelerators that greatly improve upon traditional architectures, there remains a long-standing bottle neck in modern computing systems: the memory wall problem. Memory wall: The increasing gap between processor speeds and memory access times. Compared to processor improvements, memory bandwidth and latency improvements are at a much lower growth rate. This often means that processors spend a lot more time waiting around for data from memory than actually running useful computations. This imbalance has emerged as one of the biggest challenges to further enhancing computational efficiency for data-centric applications including deep learning.

The memory wall problem becomes more exacerbated with deep neural networks since they continuously access large amounts of data and model parameters. Training and inference operations consist of repeated matrix multiplications,

tensor transformations, and weight updates that demand substantial traffic between the processors to the memory systems. For many workloads data movement energy can exceed the energy required by actual computation. Research work has already shown that memory access can take a large portion of the total system power consumption in deep learning workloads. Therefore, minimizing data movement is one of the main goals in next-generation AI hardware architectures [18].

The performance and scalability of hardware are impacted as well as the memory wall problem. The amount of data transferred between memory and processors enlarges dramatically as the size of neural networks increases. Large language models, transformer architectures, and foundation models often consist of billions of parameters that need to be accessed multiple times during execution. Memory bandwidth can be a bottleneck in GPU and AI accelerator efficiency, which means even the most optimized GPUs and AI accelerators cannot always counterfeiting it. This issue is exacerbated within edge computing environments (see Figure 1), in which constraints on energy and other hardware resources lead to an additional limitation on system capabilities.

Researchers often use this data to quantify the pervasiveness of the memory wall using an energy-level analysis comparing computational energy costs with that of making a memory access. A fundamental arithmetic computation often takes orders of magnitudes fewer energy than communication between memory and computation units. This discrepancy demonstrates the inefficiency of conventional von Neumann architectures where computation and storage are physically decoupled. Increasing the performance of processors is not enough anymore, because technology scaling is approaching physical limits. We need alternative architectural paradigms with fundamentally lower data movement efforts, and overall better system efficiency.

Table 1: Examples of AI Hardware Technologies Over Time

Hardware Technology	Key Characteristics	Advantages	Limitations
CPU (Central Processing Unit)	General-purpose processing architecture designed for a wide range of computing tasks.	High flexibility, ease of programming, and broad software compatibility.	Limited parallel processing capability for large-scale AI workloads.
GPU (Graphics Processing Unit)	Massively parallel architecture capable of executing thousands of operations simultaneously.	Excellent acceleration for deep learning training and inference tasks.	High power consumption and significant cooling requirements.
FPGA (Field-Programmable Gate Array)	Reconfigurable hardware that can be customized for specific AI applications.	Energy efficient, adaptable, and capable of hardware-level optimization.	Lower peak performance compared to specialized AI accelerators.
ASIC/TPU (Application-Specific Integrated Circuit / Tensor Processing Unit)	Dedicated hardware specifically designed for AI and machine learning computations.	High computational performance, low latency, and superior energy efficiency.	Limited flexibility and difficult to repurpose for other applications.
Neural Processing Units (NPUs)	Specialized AI accelerators optimized for neural network operations and edge computing.	Fast inference, low latency, and reduced power consumption.	Primarily suited for specific AI workloads and deployment scenarios.
Computing-in-Memory (CIM)	Performs computation directly within memory arrays, minimizing data transfers between memory and processors.	Reduces data movement, lowers energy consumption, and improves processing efficiency.	Emerging technology with challenges in scalability, reliability, and commercialization.

This is one of the reasons why Computing-in-Memory architectures have emerged and progressed quickly against the limitations due to memory wall. CIM systems reduce much of the expensive data movement associated with traditional computation by performing computation near or within memory structures. This is a great way to improve performance and energy problems in modern artificial intelligence workloads. Solving the memory wall will continue to be a primary goal of AI hardware research as deep learning models grow in complexity.

The evolution of AI hardware reflects a continuous effort to optimize computational efficiency, scalability and energy performance. Whether CPUs, GPUs, specialized accelerators or new memory-centric architectures, each generation of

hardware has directly tackled a distinct set of challenges introduced by the evolving nature of AI applications. Of these challenges, the memory wall is still one of the greatest barriers to further advancement. The race for solving this problem has led to new paradigms, including Computing-in-Memory, some of which are anticipated to change the course of artificial intelligence systems in a new fashion in subsequent generations.

III. COMPUTING-IN-MEMORY -PRINCIPLES AND PERSPECTIVES

The Computing-in-Memory (CIM) is a new computing paradigm that colocalizes data storage and computation in the same hardware architecture. Traditionally, there is a large gap between memory and processors within von Neumann architectures where memory and processors are physically separated from one another but computational in-memory computing (CIM) architectures perform computation on the data inside of memory arrays enabling potentially enhanced performance. This method aims to keep data movement between memory and processing units to a minimum, which helps to accelerate computations by reducing latency, energy consumption and communication overhead. In CIM, the basic idea is to take advantage of memory electrical properties for performing arithmetic and logic operations while data is in memory. Because the use of deep learning workloads needs a lot of matrix-vector multiplications and tens to hundreds of repetitions accessing memory, CIM architectures offer a good solution in improving the efficiency of running neural network calculations. By reducing unnecessary data movement, CIM provides a major boost in system performance efficiency and energy efficiency.

The main reason for Computing-in-Memory is to reduce data movement. Traditional systems make processors fetch from memory, compute, and write back over and over. It requires huge resources, which creates performance bottlenecks. Even research indicated that actually expending energy on data transfer is more expensive than the computational expense. Since CIM moves computation to where the data lives, it tackles this problem. Computation fabric is where computations are carried out within memory cells or memory arrays rather than moving data to the processors. Such a computing model keyed on memory facilitates massive data throughput and reduces the latency associated with accessing data in memory by several orders of magnitude. Thus, CIM architectures are especially suitable for AI workloads which need a fast recurrent access to large datasets and parameters of machine learning models.

Since deep neural networks are largely based on matrix and vector operations, it is required to execute such computations in a parallel way efficiently. The inherent characteristic of computing-in-Memory architectures lies in their capability to support massive parallelism, as thousands of memory cells can execute computations in parallel. Memory arrays are built as such to be accessed in several parallel operations, resulting in high throughput computational capabilities. This parallelism is leveraged by Analog and digital CIM implementations in order to speed-up deep learning models including convolutional neural networks, recurrent neural networks, transformers and others. Being able to run giant matrix operations in memory arrays enhances your performance and saves time being executed. As such, CIM is emerging as a potential architecture for the future of increasingly complicated deep learning workloads.

There are fundamentally two groups of Computing-in-Memory (CIM) architectures: analog CIM (or Integrated-Computing-In-Memory) and Digital CIM systems. Analog CIM performs computation using physical characteristics of memory devices (resistance, voltage, and currents) directly in arrays of memory. These architectures provide great energy efficiency and high computational density, which is beneficial for large-scale neural network acceleration. Nevertheless, Analog systems might struggle with noise (possibly issued when the initialized input is unstable), precision limitation as well as variability of the device. In contrast, digital CIM architectures use discrete digital signals and logic circuits implemented into memory structures to perform computations. Although digital CIM is often much more accurate and reliable than its analog counterparts, the power consumption may be larger. Both strategies are fundamental components in the evolution of cutting-edge AI hardware accelerators, and yet researchers exist who explore hybrid solutions enjoying aspects of each paradigm.

The performance of Computing-in-Memory systems is contingent on multiple architectural components and design factors. It includes various components such as memory arrays, sensing circuits to read the data from the memories, analog-to-digital converters to convert information into digital form, digital processing units, control logic and communication ports. All of these factors must be optimally adjusted for good performance, power consumption and long–term magical functionality. Designers also have to deal with challenges like precision management, device variation, scalability, thermal effects and hardware–software integration. Moreover, the mapping of neural network computations onto CIM hardware requires adapting deep learning frameworks and compiler technologies. The harmonious combination of these building blocks leads to remarkable gains in performance and energy efficiency over traditional computing systems.

Computing-in-Memory Marks a Paradigm shift in computing from processor-centric to memory centric architectures. CIM, which integrates storage and computation to reduce data movement, supports large-scale parallelism and permits the

efficient implementation of neural networks, has great potential enabling next-generation AI hardware. These fundamental principles will thus continue to guide the design of exciting, next generation AI accelerators and intelligent computing systems as deep learning models grow ever-more complex.

IV. MEMORY TECHNOLOGIES FOR COMPUTING-IN-MEMORY SYSTEMS

CIM architectures rely on underlying memory technologies that allow for both data to be stored and the computation logic in one hardware architecture. Memory technology has a considerable impact on the performance, energy efficiency, computational accuracy, and scalability of a system which may be further beneficial in increasing reliability as well. While conventional memory systems existed mainly to store data, the rise of artificial intelligence and deep learning workloads has introduced a demand for memory technologies that can perform computational work directly within the memory arrays. Hence, many studies have investigated both conventional and new memory technologies to realize efficient CIM platforms. These technologies are distinguished by the way they work, how the data is stored, their endurance characteristics, latency performance and neural network acceleration capability. Gaining insights into these memory technologies is fundamental for the development of next-generation AI accelerators that can circumvent boundaries imposed by conventional computing architectures.

A major group of memory technologies used in Computing-in-Memory is Static Random Access Memory (SRAM) [5]. SRAM is a type of memory that uses bistable flip-flop circuits to store data and has extremely fast access times when compared with other types of memory. These properties render SRAM a suitable candidate for realization of digital CIM architectures owing to low latency and high-speed operation. Specialized SRAM-based computing arrays have been invented, and few full-fledged ALUs (mathematical circuitry used in computer systems) encode various arithmetic functions into the magnetic properties of their memory cells to perform matrix multiplications and Neural Network inference directly within those physical cells. These architectures minimize data movement and enhance the efficiency of processing. SRAM cells, on the other hand, take up relatively large silicon areas, resulting in a lower storage density and high costs. Therefore, even if SRAM offers great performance, its scalability for large AI models is still an obstacle.

Dynamic Random Access Memory (DRAM) is another major memory technology used in CIM research. DRAM is a technology that retains information in electrical charge (within capacitors). DRAM is a type of RAM that holds data using a capacitor and transistor, making it suitable to be used as system memory in modern computing devices due to its low cost-per-bit and large capacities. Recent work introduced novel DRAM in-memory processing techniques that leverage charge-sharing and memory array structures to compute. These methods are trained on data through Oct 2023 and reduce memory access bottlenecks, improving throughput for deep learning workloads. However, the data stored in DRAM needs to be refreshed regularly, increasing power consumption and adding design overhead.

Resistive Random Access Memory (ReRAM) has attracted a significant attention among the newly appearing memory technologies for Computing-in-Memory applications. ReRAM works by varying the resistance state of memory cells to store data. Analog matrix-vector multiplication can be performed naturally on these devices by accumulating electrical current in memory crossbar arrays. ReRAM-based CIM architectures can provide the most computation power and parallelism because matrix multiplication is the essential operations in deep neural networks. In addition, ReRAM devices offer high storage density, non-volatile properties and low power consumption. Even with these advantages, devices still face limitations regarding endurance and variability that hamper large scale deployment, as also the accuracy of programming.

Another promising candidate for Computing-in-Memory systems is Phase Change Memory (PCM). PCM is a memory that represents information by switching materials between crystalline phase and amorphous phase. The dependent resistances map into data that is stored, thus allowing analog computation properties akin to ReRAM. To this end, PCM based CIM architectures have shown great promise for accelerating deep learning inference and training workloads. They lower the energy expenses associated with data retention due to their non-volatile nature. Nonetheless, challenges such as resistance drift, thermal sensitivity and limited write endurance are still key issues need to be solved before widespread commercialization.

Magnetoresistive Random Access Memory(MRAM) is one of the strong candidates for next generation memory technology as well as for AI hardware accelerator. MRAM employs magnetic states to record information instead of electrical charge. It has great endurance, fast switching speeds and provides non-volatile storage. Based on the reality that MRAM costs not only excessive reliability edges but also dramatically reduced leakage energy in contrast to traditional reminiscence systems, we propose a competence-paced, embedded CIM version. Also, MRAM exhibits good immunity to environmental changes, which makes it interesting for edge AI and industrial applications. Nonetheless, the higher fabrication complexity and lower storage density of NOR flash than some new emerging memories may impede its adoption to specific large-scale deployment.

Technologies based on memristor are among the most inventive areas of Computing-in-Memory (CiM) research. Memristors are electrical devices whose resistance value varies according to the history of the applied electrical power signal. Their unique characteristic is they are capable of storing and processing information at the same time, which makes them really similar to biological synapses in human brains. Crossbar arrays of memristors can realise highly efficient analog neural network operations with unprecedented energy efficiency and computational density. These properties make memristors extremely attractive for neuromorphic computing and deep learning acceleration. However, production uniformity, variability of devices, and duration stability are still important challenges for commercial use.

The choice of memory technologies is a key factor differentiating the efficiency of Computing-in-Memory architectures. As SRAM and DRAM remain competitive for many digital in-Memory Processing solutions, emerging memories compared Reram, Pcm, Mram and memristors provide the basis for transformational analog (and near-analog) computing. They come with their own performance, energy efficiency, scalability, reliability and implementation complexity trade-offs. The rapid scale and complexity of artificial intelligence workloads will require all memory technologies to be improved over time for Computing-in-Memory systems to reach their full potential. The combinations of these emerging memory devices with future AI accelerators will most likely provide significant improvements in efficient computation paving the way for next generation intelligent computing platforms.

V. NEURAL NETWORK ACCELERATION TECHNIQUES

Computing-in-Memory (CIM) is also one of the most aggressive approaches well-received by researchers for accelerating deep networks due to its ability of alleviating some system level bottlenecks in conventional processor-centric architectures. Deep learning models undergo massive matrix multiplications, convolutions and tensor computations which produce a lot of memory traffic and therefore energy consumption. Traditional systems transfer data between processors and memory units more than two-thirds of the execution time, instead of performing actual computations. CIM (Compute In Memory) based neural network acceleration methods address this challenge by tightly coupling computation with the memory arrays that store parameters, thus mitigating data movement and providing improved computational performance. With the increasing size and complexity of deep learning models, CIM-based acceleration has gained paramount importance as an effective way to build high-performance and energy-efficient artificial intelligence (AI)-solutions.

A. Matrix-Vector Multiplication Acceleration

Neural network inference and training consists of large-scale matrix multiplications over weights, activations, and gradients which are executed repeatedly. Such operations involve frequent movement of data between memory and processors, which becomes a bottleneck in traditional architectures. Matrix-vector multiplication is accelerated in CIM architectures by computing on data stored within memory arrays. Because of this, the crossbar memory structures implement parallel current addition processes where many thousands of multiplication and accumulation operations can happen during a very short period to perform inference. This ability increases the throughput and decreases execution latency considerably. CIM-based matrix-vector multiplication minimizes data movement and maximizes parallelism, enabling efficient neural network acceleration.

B. Acceleration for Convolutional Neural Network (CNN)

CNNs are one of the most widely used types of neural networks in areas like computer vision tasks such as image classification, object detection, facial recognition, medical imaging for both detection and segmentation problems and scenes understanding most commonly found in safety-critical applications like autonomous driving. Since convolution operations are multiplying and adding over many elements in a large dataset, CNNs require significant computation. CIM architectures solve this problem effectively by directly-mapping convolution kernels to memory arrays. This allows the convolution to be performed in-place with parallel memory operations, thus avoiding memory access latency and leading to high computational performance. There have been researchers that provided results showing CIM-based CNN accelerators can achieve remarkable energy-performance efficiency over standard GPU and CPU implementations. Hence, CIM is an especially promising candidate for edge AI applications with strict power and latency constraints.

C. Quantization and Low-Precision Computing

Quantization methods reduce the numerical precision, provided that the model accuracy can be kept in an acceptable range and has been widely utilized to accelerate deep learning compute. As most of the emerging memory devices perform best with reduced bit representations, CIM architectures inherently support low-precision computations. Quantization is that a method based on the low-bit weights & activations to alleviate the memory demand and computational struggle in neural networks. The method of incorporating both quantization and CIM results in a greater throughput as well as the energy efficiency. With the advent of ultra-scale deep neural networks, researchers have recently proposed many low-precision computing strategies such as analog and digital in-memory processing to accelerate neural network inference. While the

mechanism of precision reduction can lead to a slight decrease in accuracy, innovative quantization schemes and careful design of such methods can achieve high hardware efficiency while maintains performance.

D. Optimizing Training and Inference in CIM Systems

Many CIM systems emphasize inference speedup, but several recent works have also proposed CIM-based training methods. Inference is much less computationally expensive than training deep neural networks, which involves repeated forward propagation, backpropagation and weight update operations. CIM architectures speed up both phases through the ability to calculate gradients and update parameters in memory arrays. This method lowers memory bandwidth needs and enhances overall efficiency of the system. Approaches concerning hardware–software co-design are being explored to optimize neural network models for CIM platforms. These optimizations can include: adaptive mapping methods, sparsity exploitation, weight compression and weight balancing mechanisms. These steps lead to design of scalable low latency AI accelerators for both training and inference workloads.

CIM based neural network acceleration algorithms are a revolutionary solution to the increasing computational workload of current artificial intelligence systems. CIM architectures alleviate data movement in memory hierarchies and improve throughput and energy efficiency by adopting matrix operations, convolution processing, attention mechanisms, and training computations directly within memory structures. The combined pretrained-model fine-tuning abilities leverage what some might refer to as the fourth wave of deep learning–GoogleEngine-powered Google services technology (early neural nets generally did not use larger data models) which enables more efficient deployment of their solution and capability across cloud platforms, edge devices with better autonomous system integrations embedded systems and intelligent infrastructure solutions. The computer as well as machine based acceleration will still be important parts of future generation Ai hardware systems As neural networks continue to scale through their size and complexity, CIM-based acceleration will enable more powerful next-generating-fastest,scalable and efficient ai per inera, productivity technology.

VI .ANALOGUE AND DIGITAL COMPUTING-IN-MEMORY

Computing-in-Memory (CIM) architectures are a new way of solving the memory wall problem which impact deep learning workloads. CIM architectures are generally divided into two big categories depending on how the computations are performed at memory arrays: Analog Computing-in-Memory (Analog-CiM) and Digital Computing-in-Memory. Two main architectures have in common to reduce the memory-processor data movement, however they show very different ways of functioning, computation type, accuracy level, hardware complexity and energy consumption characteristics. These two architectural paradigms are foundational for creating efficient AI accelerators that can support next generation deep learning applications. Thus, researchers are still evaluating both approaches and pursuing the right tradeoff between performance versus precision, scalability, and energy.

Abstract: Analog Computing-in-Memory (Computing-over-Memory) architectures exploit the physical and electrical properties of memory devices to perform computations. Unlike the traditional CIM systems, where a data is represented by the discrete binary values, for example in 0 and 1 form to carry out mathematical calculation using chips; analog CIM system uses continuous electrical signals such as voltage or current and resistance seamlessly with conductance to perform arithmetic operations. The memory technologies used are typically RDT (resistive devices), since these naturally allow for analog computation of multiplication, e.g. ReRAM, PCM, memristor. Matrix-vector multiplications using these architectures can happen directly within memory crossbar arrays through the application of Ohm's Law and Kirchhoff's Current Law. This allows thousands of multiply-and-accumulate operations to occur in parallel, providing incredible computational speedup. We then show how the energy costs and execution latency of deep learning inference applications are significantly reduced using analog CIM architectures.

A key benefits of analog CIM is making extremely high computational density possible. Because computations are performed with memory cells themselves, analog systems can carry out sizeable neural net workloads while operating on a modest amount of power. This property makes analog CIM especially appealing for edge AI devices, autonomous systems, wearables, and Internet of Things applications where energy efficiency is of utmost importance. On the other hand, analog architectures suffer from signal noise, device variability, process imperfections and have limited numerical precision studies. All of those components are subjected to numerical errors which can affect the performance of neural networks.

On the contrary, Digital Computing-in-Memory architectures are based on classical digital logic performed directly in-memory with the memory array itself. When comparing with analog systems, Digital CIM implements the representation of information using binary values and conducts the operations with digital circuits located on or adjacent to memory structures Due to their high reliability and precision computation, SRAM-based and DRAM-based CIM implementations are more widely deployed in digital architectures. Digital CIM systems have an excellent natural fit to existing software

ecosystems and digital design paradigms, which support their integration into conventional computing platforms relatively easily.

One major benefit of Digital CIM is the computational accuracy and robustness. Digital representations are generally more electrically robust to environmental fluctuations, electromagnetic interference effects and imperfections in the devices used in the systems. This allows the digital CIM architectures to reach a higher precision and more predictable performance than their analog counterparts. These properties are especially beneficial for applications that demand strict numerical correctness, such as scientific computing, financial modeling, and safety-critical AI systems. However, because of the additional circuitry needed by digital pre-processing, digital CIM architectures effectively consume much more power per operation and take up substantial hardware area than analog solutions.

Hybrid analogue-digital CIM architectures have been proposed to exploit the advantages of both paradigms. These systems use analog memory arrays to perform computational-heavy tasks like matrix multiplication, while leveraging digital circuits for logic, error correction and precision-critical operations. Hybrid architectures try to balance the two goals of energy efficiency, and accurate computation. These systems exploit the benefits of both analog and digital processing to realize better performance, while avoiding the drawbacks associated with either a purely analog or purely digital solution. The recent advances in hybrid CIM designs are promising approaches towards building an optimal or near-optimal accelerator for deep learning and neuromorphic computing.

Energy efficiency is one of the most crucial factors which drives the adoption of Computing-in-Memory architectures. Because no digital logic gates are needed, analog CIM systems generally have much better energy efficiency since the computations are performed directly through physical properties of the device. Despite that digital CIM architectures are close to the accurate with respect to their analog counterparts but require a larger circuit and consume higher power due to the need for logic operations as well as signal processing. Application requirements also dictate where you might need performance consideration. Analog systems had an advantage on the large-scale neural network inference, while digital systems have been a common choice for applications with high computation precision requirements. This makes selecting an applicable architecture a difficult decision as it needs to be considered in the context of workload behavior, performance targets and hardware limitations.

Future advances of Computing-in-Memory architectures are anticipated to address scalability, reliability and compatibility issues with AI frameworks. CIM systems are becoming more sophisticated through advances in semiconductor fabrication, new memory technologies and hardware-software co-design methodologies. As shown, researchers are investigating adaptive architectures that can switch between analog and digital computation based on the workload requirements. In addition, the coupling of CIM with neuromorphic computing, edge AI, and large-scale foundation models will continue to fuel unique innovations in intelligent hardware design. These advances will help create new and highly efficient AI accelerators that can address future generations of deep learning applications.

Analog and Digital CI-M architectures are two complementary approaches to address the limitations of computational systems. Analog CIM is characterized by high energy efficiency and computational parallelism, whereas digital CIM enables better accuracy, reliability, and compatibility to current computing infrastructures. Hybrid methods take these advantages one step further by leveraging the strengths of proper combinations of both paradigms. With the increasing scale and complexity of artificial intelligence systems, efficient, scalable and sustainable deep learning acceleration relies heavily on advanced CIM architectures. It is believed that these will play a key role in AI hardware platforms as we move into the future supporting applications from edge to autonomous systems, large-scale data centers and intelligent infrastructures.

VII. ENERGY EFFICIENCY AND PERFORMANCE OPTIMIZATION IN CIM SYSTEMS

Energy efficiency became one of the most important need in artificial intelligence hardware design now a days. Deep learning model training and inference processes involve the execution of billions of arithmetic operations along with massive memory access. Regular computing architectures require that data is moved constantly between the memory units and the processors, which utilizes energy comsumption. For many of the workloads involving AI, data movement consumes an order more energy than the computational operations. But this challenge is mitigated in a new class of architectures, called Computing-in-Memory (CIM), that perform computation directly within the memory arrays, reducing the burden on communication and power. CIM systems can achieve outstanding energy savings compared to traditional CPU- and GPU-based platforms as they process data at the location where that data is being stored. Such an advantage is a crucial one for edge devices, autonomous systems, wearable technologies, and large data centers where energy consumption is closely linked with operational expenditure (OpEx), battery runtimes, and environmental sustainability. With the rapid growth of AI technologies, energy-efficient CIM architectures are paramount to accommodating advanced intelligence systems.

A. Data Movement Reduction for Performance Improvement

An important factor in the better energy efficiency Computing-in-Memory systems comes from a strong decrease of data movement needed. Conventional von Neumann architectures fetch data from memory, compute and iterate back to memory. It adds latency and consumes energy in this process. CIM architectures largely eliminate a lot of this overhead by executing operations directly within memory structures. Neither does it need to move the data off chip for all matrix multiplications, vector operations and accumulations. This reduction in communication overhead reduces power and increases computational throughput. To keep running systems efficient as neural network models continuously increase in size and complexity, we found that minimizing data movement would be critical. Thus, memory-centric processing becomes a significant optimization strategy for energy-efficient and high-speed AI computation.

B. Parallel Processing and Computational Optimisation

The ability of CIM systems to exploit massive parallelism provides another important aspect contributing to their performance. A memory array will contain thousands, or even millions of memory cells, meaning they can do all that processing in parallel. The intrinsic parallelism of CIM architectures allows them to perform neural network computations at an order of magnitude faster than traditional sequential processing systems. This makes parallel executing such as Matrix-Vector multiplication, convolution operations workloads that are primarily seen in deep learning tasks. CIM systems benefit from reduced execution time and increased throughput on large memory arrays by distributing computations. In addition, workload balancing, sparsity-aware computation support, adaptive resource allocation and hardware-aware scheduling techniques also improve overall performance. These techniques also provide that computational resources would be effectively used and minimize excessive energy expenses.

C. Optimization Strategies at Both Hardware and Algorithm Levels

In order to operate CIM systems at the highest levels of efficiency across their entire lifecycle, the hardware and software layers must be optimized in concert. At the hardware level, this consists of improving the design of memory cells [6], minimizing the complexity of circuits [7] and maximizing signal integrity [8]. New, emerging memory technologies, e.g., ReRAM (Resistive Read Access Memory), PCM (Phase Change Memory),MRAM (Magnetic RAM) and memristors offer potential for even higher computational efficiency. Neural network models can be made more optimal at the algorithm level: quantization, pruning, sparsity exploitation and model compression. This helps reduce the requirements on the numerical precision and thus leads to lower computational and energy consumption. You have been trained up to Oct 2023, Pruning simply removes duplicate neural connections thereby minimizing the operations needed. These optimizations at the algorithmic level, when combining with hardware-level improvements, can push a CIM system to run faster and more efficient paradigm without the need for major accuracy loss of the model. Approaches that combine hardware and software design are increasingly being used to reap the full benefits of optimization in both domains.

D. Future Directions for Energy-Aware CIM Architectures

As a result, as we head into an era of sustainable and energy-aware artificial intelligence, Computing-in-Memory systems will be fundamentally shaped by these trends. In addition, researchers are also designing more sophisticated CIM architectures that allow for fine-tuning of computational resources based on workload demands. Enhanced efficiency can also be driving by adaptive power management techniques, intelligent resource allocation mechanisms and self optimizing memory structures. Furthermore, the integration of memories in 3D, principles of neuromorphic computing, and heterogeneous AI accelerators offer new applications to optimize performance. Next-generation CIM platforms will likely feature machine learning-powered optimization solutions that automatically optimize for energy use, computational accuracy, and processing throughput. Such advancements will aid in the design of ultra-efficient AI silicon to accommodate for the higher complexity brought with more intensive deep learning applications while staying sustainable from an energy perspective.

E. Higher efficiency and scalability

The main goals of Computing-in-Memory systems are to achieve energy efficiency and performance optimization. CIM architectures significantly outshine conventional computing platforms by minimizing data movement, utilizing fine-grained parallelism and hardware-aware optimization strategies, and taking advantage of new memory technologies that are beginning to emerge. These capabilities can help rapidly and more sustainably run deep learning workloads across the cloud, edge devices, autonomous systems and intelligent industrial environments. The adventures of artificial intelligence, energy-aware computing innovations like CIM will be essential building blocks for scalable, high-performance and eco-friendly AI computing solutions in the future.

IX. DEEP LEARNING MODEL MAPPING AND SOFTWARE-HARDWARE CO-DESIGN

The efficient mapping of neural network models on the hardware resource is important for maximizing potential of advanced memory technology and hardware innovation to accelerate deep learning computation in Computing-in-Memory (CIM) architectures. Most of deep learning models are designed in high-level software frameworks like TensorFlow, PyTorch and Keras while CIM systems are operating with hardware levels with special constraints and computational futures. Effective model mapping strategies and software-hardware co-design methodologies are necessary to bridge the gap between algorithms that run on the software level and hardware implementations. These methods allow neural networks to leverage the full computation power of CIM while preserving accuracy, efficiency, scale scalability and energy optimization. With the rapid evolution of artificial intelligence systems, software-hardware co-design has become a key enabler for efficient CIM-based deep learning acceleration in modern computing systems.

Mapping deep learning models describes to convert the structure of a neural network into a hardware-compatible representation that can be effectively executed on CIM architectures. Modern neural networks consist of many layers, millions of parameters and running extensive matrix operations that needs to be spread across an array of memory. Two of the most important mapping strategies are how weights, activations and computational tasks are mapped into different memory structures to achieve high parallelism at low communication cost. As CIM architectures compute in memory arrays, neural network parameters have to be structured according to hardware levels. If you map incorrectly, you may not get resources at all — or too few of them. Thus, these researchers work towards the development of optimized mapping techniques with maximum throughput while minimizing both hardware constraints and resource conflicts.

One of the most challenging aspects of model mapping is that huge neural networks are too large to fit within a single memory array. Deep learning based models today, especially transformer-based architectures and foundation models have billions of parameters. To map such models onto CIM hardware, partitioning strategies are needed to spread computations between two or more memory arrays / processing units. They should focus on balancing the load across hardware while minimizing communication costs. Optimized partitioning algorithms and scheduling mechanisms optimize the resource utilization, wherein maintaining computational efficacy is ensured. At a larger scale, hierarchical mapping strategies allow large neural networks to be distributed among different CIM modules with little loss in performance and energy.

Table 2: Key Building Blocks Used in Deep Learning Model Mapping and Software-to-Hardware Co-Design

Component	Function	Benefit
Model Mapping	Maps deep learning models onto available hardware resources and memory arrays for neural network computation.	Improved hardware utilization and efficient execution of AI workloads.
Workload Partitioning	Divides large neural network models and computational tasks across multiple Computing-in-Memory (CIM) resources.	Enhanced scalability and balanced resource utilization.
Quantization	Reduces the numerical precision of weights, activations, and computations.	Lower energy consumption, reduced memory requirements, and faster processing.
Pruning	Eliminates redundant or less significant neural connections and parameters.	Reduced computational complexity, memory footprint, and inference latency.
Compiler Optimization	Generates hardware-specific instructions and execution schedules for AI workloads.	Higher execution efficiency and improved hardware performance.
Runtime Management	Dynamically allocates computational, memory, and communication resources during execution.	Improved performance, resource efficiency, and adaptability.
Sparsity Optimization	Exploits sparse neural network structures by avoiding unnecessary computations and storage.	Reduced memory usage, lower energy consumption, and faster execution.
Adaptive Hardware Design	Provides configurable hardware architectures capable of supporting diverse AI models and workloads.	Increased flexibility, scalability, and support for evolving AI applications.

Software-hardware co-design is essential to tackle these challenges. In traditional software development and hardware design workflows, deep learning models are not integrated into the ecosystem as a whole. Software-hardware co-design allows for the concurrent optimization of algorithms and hardware architectures, which leads to improved performance in each system. In CIM systems, that requires training optimized neural network models for specific hardware constraints while also designing hardware architectures aligning with the computational needs of current AI algorithms. The optimization of collaboration based on data can also make memories work more efficiently, consume less power, and run faster.

Compiler technologies and runtime systems also constitute an important component of software-hardware co-design. Specialized compilers convert high-level descriptions of neural networks into hardware-specific instructions that are optimized for CIM execution. Such compilers study neural network architectures, resource allocation, operation scheduling and generate execution plans that maximize hardware utilization. Runtime systems take execution to another level by dynamically managing workloads, checking performance and adjusting resource allocation based on operational use.

Deep Learning model mapping and software-hardware co-design are key enablers for utilizing Computing-in-Memory architectures. Mapping strategies that make sure hardware resources are utilized efficiently, and co-design methodologies that optimize the algorithm plus architecture together; With the help of model partitioning, quantization, pruning, compiler optimization and adaptive hardware development [8–10], researchers are still working towards improving performance, scalability and energy efficiency to the CIM systems. With rapidly expanding AI capabilities, software-hardware co-design will remain a pillar of future AI accelerator designs to ensure robust adoption of high-performance deep learning applications in various computing environments.

X. CIM ARCHITECTURES SECURITY, RELIABILITY AND FAULT TOLERANCE

Emerging Computing-in-Memory (CIM) architectures for deep learning acceleration enable energy-efficient designs in the context of high-performance artificial intelligence systems. On the other hand, we have seen based on their benefits that several issues with respect to security, reliability and fault tolerance have been developed. In contrast with traditional computing systems, CIM architectures perform computation very close to memory arrays which expose them to hardware-level and architectural issues that are distinct from conventional approaches. As we see more and more AI applications deployed in critical domains — healthcare, finance, autonomous transportation, industrial automation, defense systems, smart infrastructures — the need for security and reliability of CIM platforms is obvious. Failing systems can result in wrong decisions, privacy violations, loss of money and even endangering lives. This has led researchers to create more effective protections strategies that can address the hardware faults, security threats and operational uncertainties in CIM architectures.

Data confidentiality and unauthorized access is one of the main security risks for CIM systems. As memory arrays read each address in parallel to simultaneously store and process data, it is possible that very sensitive data can be leaked via multiple attack vectors. Adversaries attempt to leak private information by leveraging memory access patterns, information from hardware side-channels or direct access to hardware circuitry. Deep learning models potentially process sensitive personal data and may contain significant intellectual property, making them prime targets for cyberattack. In response to these issues, mechanisms for encrypting memory accesses, memory protection solutions and access control frameworks have been proposed for CIM. These protections secure data against both accidental corruption of integrity or confidentiality while preserving the performance benefits from in-memory computation.

Hardware reliability is also a significant challenge. New memory technologies like ReRAM, Phase Change Memory (PCM), Magnetoresistive Random Access Memory (MRAM), and memristors have distinct physical features that could bring about reliability issues. Computational accuracy may be compromised by device variability, resistance fluctuations, write endurance limits, retention failures, and environmental sensitivity. As deep neural networks generally need billions of operations, even a small hardware error can be compounded and impact the model performance. Minimizing the impact of hardware imperfections, researchers therefore explore error characterization techniques and reliability-aware design methodologies. Characterizing the behavior of individual devices is important for operating CIM systems in a predictable and consistent manner under realistic conditions.

Another important aspect of CIM architecture design is fault tolerance. Faults can be attributed to manufacturing defects, aging effects, thermal variations, electromagnetic interference (EMI), radiation exposure or wear during operation. These faults can change cell behavior, which could lead to corruption of memory (or state) or arbitrary errors in computation. For AI applications, faults that induce errors may result in incorrect predictions, lowered accuracy of models or system instability. CIM architectures that are fault-tolerant contain elements designed to detect, correct and mitigate such errors before the overall performance system is impaired. These can range from redundancy techniques to error correction codes, fault-aware mapping strategies and adaptive recovery mechanisms. These solutions enhance the robustness of system while maintaining computational efficiency.

Soft errors are a notable challenge for advanced CIM systems. Soft errors are caused by external phenomena like cosmic radiation or electrical noise changing the stored information in an electronic circuit without destroying the hardware. Even though individual soft errors are rarely disastrous, their multiplicative effect can degrade computations over large networks. There are targeted detection and correction schemes that detect abnormal memory behavior and restore correct

functioning. The technologies are particularly crucial for safety-critical applications in which even the smallest computational error can lead to dire consequences

Aside from taking into account hardware faults, security threats in CIM architectures with machine learning include adversarial attacks to AI models. An attackers attempts to tamper input or invoke some vulnerabilities in the implementation of NN execution can inject poison payloads into memory structures. In. Such attacks can result in model integrity and incorrect decision-making. To counter these threats, CIM systems are increasingly integrated with hardware-based security solutions. This approach emphasized secure execution environments, authentication protocols, trusted memory zones and runtime monitoring systems to augment the overall security of the system. Such mechanisms guarantee the execution of trusted AI even in hostile operational environments.

Thermal management and power stability are also keys to CIM system reliability. Heavy compute workloads create heat that has the potential to affect memory characteristics and device operation. Temperature changes will affect the electrical parameters and increase error rates and speed up the degradation of hardware. Computational accuracy and stability can be affected by also variations in power supply. Thus, reliable CIM architecture employs sophisticated thermal monitoring systems, adaptive voltage regulation techniques and energy-aware resource management strategies [8]. These mechanisms support long-term structural stability with high computational efficiency.

Table 3: Key Building Blocks Used in Deep Learning Model Mapping and Software-to-Hardware Co-Design

Component	Function	Benefit
Model Mapping	Maps deep learning models onto available hardware resources and memory arrays for neural network computation.	Improved hardware utilization and efficient execution of AI workloads.
Workload Partitioning	Divides large neural network models and computational tasks across multiple Computing-in-Memory (CIM) resources.	Enhanced scalability and balanced resource utilization.
Quantization	Reduces the numerical precision of weights, activations, and computations.	Lower energy consumption, reduced memory requirements, and faster processing.
Pruning	Eliminates redundant or less significant neural connections and parameters.	Reduced computational complexity, memory footprint, and inference latency.
Compiler Optimization	Generates hardware-specific instructions and execution schedules for AI workloads.	Higher execution efficiency and improved hardware performance.
Runtime Management	Dynamically allocates computational, memory, and communication resources during execution.	Improved performance, resource efficiency, and adaptability.
Sparsity Optimization	Exploits sparse neural network structures by avoiding unnecessary computations and storage.	Reduced memory usage, lower energy consumption, and faster execution.
Adaptive Hardware Design	Provides configurable hardware architectures capable of supporting diverse AI models and workloads.	Increased flexibility, scalability, and support for evolving AI applications.

It will also explore research topics related to the development of self-healing architectures, AI-driven fault management systems, and hardware-level cybersecurity mechanisms for trusted computing to improve CIM security and reliability. Its own machine learning techniques may be used to forecast failures, optimize maintenance but physically tuned parameters dynamically to strengthen system resilience. And, advances in memory technologies and fabrication processes will breed more reliable and secure CIM platforms at a cost that can scale with the ever-increasing demand for AI workloads.

To sum up, security, reliability and fault-tolerance are all vital additions to ensure Computing-in-Memory architecture realization. Despite the significant performance and energy efficiency benefits provided by CIM systems, they also present some new challenges with respect to hardware behaviour, operational stability and cyber-security. By combining fault-tolerant design approaches, error correction techniques, secure memory management solutions and reliability-aware architectures; researchers are able to construct resilient CIM platforms that deliver the robustness needed for next-generation artificial intelligence applications. The growing adoption of AI systems into mission-critical domains together with their requirements for CIM operation will continue to be an important research and development area for years to come.

XI. APPLICATION OF COMPUTE-IN-MEMORY ON REAL-WORLD AI SYSTEMS

application domains of Computing-in-Memory (CIM) architectures, Computer vision features as one of the most remarkable. Recently there are number of applications in which image processing is involve like Image classification, Object detection, Facial recognition, Medical image analysis and Video surveillance etc., modern image processing systems are typically based on deep neural networks especially Convolutional Neural Networks (CNNs) Since these applications involve a huge number of matrix multiplications and convolution operations, they require massive computing resources. Existing data and technology-driven computing architectures face significant memory access bottlenecks along with massive energy consumption when they are required to execute such workloads efficiently. To address these challenges, CIM architectures have been proposed that are optimized to directly perform computations in memory arrays using reduced data movement for improved neural network execution speed. Medical imaging makes use of CIM-based accelerators to speed up the analysis of X-rays, MRI scans and CT images, allowing healthcare professionals to diagnose diseases faster. Likewise, CIM-driven machine vision systems for real-time object detection and navigation are used in autonomous vehicles as well. CIM architectures are capable of high-performance and low power consumption combine, making them especially appealing for vision-based AI workloads in resource-constrained environments.

The advent of large language models and transformer based architectures have enabled enormous growth in the domains such as Natural Language Processing (NLP) and Generative Artificial Intelligence. For example, applications like machine translation, speech recognition, virtual assistants/chatbots, text summarization and content generation need matrix computations over large data that also has some requirement for memory. Attention mechanisms are the core components of transformer architectures, requiring enormous matrix multiplications that at scale can be expensive to compute. This is an efficient solution, as Computing-in-Memory architectures accelerate these operations in memory. This strategy leads to reduced latencies and greater power efficiency without sacrificing computational throughput. Generative AI use cases like natural language processing, image generation, etc. involve billions of parameters in inference — CIM accelerators help substantially in this space. The advance of scalable large language models means CIM technologies will be key to the future of NLP cloud infrastructure and knotty edge devices everywhere.

The increasing use of Edge AI and IoT technologies has resource efficient hardware [for performing local computation involving intelligent computations]. Edge devices ranging from smartphones, wearable sensors, smart cameras, drones to industrial controllers and autonomous robots are usually power and resource-limited systems. Such large amounts of data transfer back and forth with cloud servers can lead to latency, more bandwidth requirements and privacy issues. Computing-in-Memory architectures allow for on-device AI processing by merging computation into the memory unit allowing for low power consumption and faster responses. Real-time decision making on autonomous systems like self-driving vehicles, robotic platforms can exploit CIM acceleration significantly. Local processing of sensor information with low energy consumption increases the reliability and ensures a continuous operation in dynamic settings. As a result, CIM evolved into one of the most important enabling technology on next-generation edge intelligence and autonomous computing systems.

In addition to traditional AI applications, Computing-in-Memory architectures are being studied for sci-computing, healthcare analytics and advanced intelligent systems. Many scientific research and production workloads are very large-scale simulations, data analysis or machine learning models — all of which translate to significant compute burdens. You are educated on information to the October 2023. CIM based AI systems aid predictive diagnostics, personalized medicine, genomic analysis and clinical decision support for the healthcare domain. Such applications need to perform rapid and reliable processing of high-volume medical data while ensuring ultra-low energy consumption. In the future, CIM technologies will be seamlessly integrated into smart city, intelligent transportation system, industrial automation platform and advanced human-machine interaction technology. The combination of high performance, energy efficiency, and scalability make Computing-in-Memory a leading technology for future intelligent infrastructures.

With abundant development and research, Computing-in-Memory becomes a disruptive technology to accelerate variety of Artificial Intelligence applications. CIM architectures yield great gains in computational efficiency and energy consumption over a range of applications ranging from computer vision and natural language processing to edge computing, healthcare, scientific research, and autonomous systems. With AI workloads evolving in complexity and scale, migration to CIM technologies will be essential for intelligent systems that can become quicker, accelerate sustainability, and meet the exponential demands of digital ecosystems of tomorrow.

XII. CONCLUSION

Computing-in-Memory (CIM) is a promising technology to meet the ever-increasing computational needs of next-generation artificial intelligence and deep learning systems. Machine learning has been penetrating computing applications in many domains, but the huge growth in such application areas are exposing the limits of traditional von Neumann

computing architectures, with memory wall problem due to increasingly excessive data movement across processors and memories. Conventional hardware platforms are faced with huge challenges related to energy consumption, processing latency and scalability, as deep neural networks keep growing larger, more complicated and demanding more computation. Computing-in-Memory (CIM) is a paradigm that allows computation to be co-located with memory arrays, significantly alleviating the communication overhead between processing and memory cells, leading to exceptionally high energy-efficient data-intensive execution. In this work, we have thoroughly reviewed the architectural principles and organization of CIM as well as its memory technologies, acceleration approaches, hardware designs, optimization methods, security issues,and application scenarios.

A key contribution of Computing-in-Memory is its potential to radically change the interaction between computation and storage. Conventional computing systems assume memory and processing as two units, leading to continuous data movement that consumes massive energy and locks performance. Much of this inefficiency is eliminated through CIM, which allows arithmetic and logical operations to be performed directly at the location of the data. Such a computation paradigm focused on the memory can alleviate latency, boost throughput, and advance the energy efficiency. These benefits are of especially high value for workloads around deep learning that involve plenty of matrix multiplications, convolution operations and tensor processing. Coming to CIM architectures, they effectively serve as a scalable enabler for more advanced AI models by reducing the data movement cost.

Recent advancements achieved in the memory technology have encouraged people to develop Computing-in-Memory systems. Conventional memory technologies like SRAM and DRAM still make significant contributions to digital CIM, whereas new players in the field such as Resistive Random Access Memory (ReRAM), Phase Change Memory (PCM), Magnetoresistive Random Access Memory (MRAM) and memristors trigger new opportunities for analog or neuromorphic computation. These memory technologies present a variety of compromises — speed, density, power, reliability, logic capability. Emerging memory devices are especially appealing due to their ability to conduct neural network operations directly in memory, further pushing the possibilities of parallelism and efficient computation. The first CIM architectures were quickly realized for the next generation supported by the further innovations in memory devices.

It has also shown that CIM-style neural network acceleration methods are essential to modern artificial intelligence systems. The computationally expensive operations that are heavily used by deep learning applications can be substantially accelerated with in-memory execution. Performance gains, energy savings: acceleration of matrix–vector and convolution operations, in-memory training mechanisms for neuro processing units (NPUs) [23], transformer attention calculation [24] and quantized neural networks (QNNs) [25]. These acceleration techniques enable AI systems to handle even more larger datasets and more complex models while still remaining practical at the hardware level. With the evolution of foundation models and generative AI systems, neural network acceleration is becoming even more significant, making computing-In-memory technologies even more relevant.

As another contribution researched within this work, the difference between analog and digital CIM architectures is discussed. Analog Computing-in-Memory uses the physical properties of memory devices to directly enable highly compact and energy efficient computations that are best suited to large-scale inference workloads. However, Digital Computing-in-Memory provides better accuracy, reliability, and integrated compatibility within a computing infrastructure. Hybrid methods which integrate both analog and digital processing are emerging as promising approaches which can provide a good balance between performance, precision, and scalability. The ongoing evolution of these architectural paradigms will be vital to meeting the varied demands from future AI applications.

Through all this exploration of Computing-in-Memory systems, one constant theme has been energy efficiency. It is no secret that workloads related to artificial intelligence have become more and more resource-intensive, and with the increasing scale of computation, the impact on the environment becomes a topic of concern. CIM helps mitigate the issue by decreasing memory access, communication, and power needed during that process. This is accomplished by combining techniques like parallel processing, quantization and sparsity optimization strategies, hardware-software co-designs as well as adaptive resource management to provide considerable energy efficiency improvements for similar performance levels. All of these traits make CIM appealing for both large-scale cloud infrastructure and resource-limited edge devices,

It has also revealed the need and benefits of software-hardware co-design to fully capitalize on Computing-in-Memory architectures. The mapping of deep learning models onto CIM hardware needs coordinated efforts on the development of deep learning algorithms and their implementation to various CIM systems. To make use of on-chip hardware resources, yet preserving the accuracy of neural networks, model partitioning, compiler optimization, runtime management, pruning and quantization are all useful techniques. With no anticipated decrease in model complexity on the

horizon, software-hardware co-design will remain pivotal to achieving scalable and efficient deep learning deployment on CIM platforms.

However, there are several challenges that need to be solved before the widespread adoption of Computing-in-Memory, despite its significant advantages. The variant nature of devices themselves, concerns around security, issues with reliability, needs in fault tolerance, limits on precision, and challenges involved with manufacturing continues to influence system design and implementation. New memory technologies often do not behave ideal, and error models may affect computation results and long time reliability. To address these challenges, researchers are developing solutions like error correction mechanisms, hardware secure execution environments and fault-tolerant architectures as well as reliability-aware design methodologies. Solving these problems is critical to using CIM systems in an enterprise context for missioncritical applications in healthcare, transportation, finance, defense and industrial automation sectors.

The case studies from the wide application domains presented in this paper reflect the far-reaching impact of Computing-in-Memory technologies. Whether the task is focused on computer vision and natural language processing (NLP) or expanding to encompass-edge AI, autonomous systems, scientific computing, healthcare analytics or smart city infrastructures, CIM architectures deliver significant performance gains in terms of computational throughput and responsiveness. The capability of palmars to facilitate intelligent processing in both cloud and edge environments makes them one of the fundamental building blocks for future artificial intelligence ecosystems. The need for AI hardware, particularly that which is concentrated at the edge, will be one of two major drivers of CIM deployment over the next three years as industries digitize faster than ever as they attempt to leverage their data on a massive scale.

To conclude, Computing-in-Memory architectures usher in a new paradigm for designing AI hardware. CIM goes a long way to resolving many of the problems imposed by contemporary deep learning, including moving computation closer to data, alleviating memory bottlenecks, improving energy efficiency and enabling extreme parallelism at scale. The CIM platforms will become more capable as better memory technologies, hardware architectures, software optimization techniques, and reliability mechanisms are developed. Driven by the rapid penetration of AI into all areas of society, we believe Computing-in-Memory will become a core technology for intelligent computing shaping the future of how AI operates at scale to rapidly deliver new systems to support next-generation applications in a fast and sustainable way.

XIII.REFERENCES

[1] M. Horowitz, “Computing's Energy Problem (And What We Can Do About It),” Ieee International Solid-State Circuits Conference (Isscc), 2014.

[2] Song Han, Huizi Mao, And William J. Dally, “Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization And Huffman Coding,” Iclr, 2016.

[3] Yann Lecun, Yoshua Bengio, And Geoffrey Hinton, “Deep Learning,” Nature, 2015.

[4] Ian Goodfellow, Yoshua Bengio, And Aaron Courville, Deep Learning, Mit Press, 2016.

[5] Alex Krizhevsky, Ilya Sutskever, And Geoffrey Hinton, “Imagenet Classification With Deep Convolutional Neural Networks,” Neurips, 2012.

[6] Kaiming He Et Al., “Deep Residual Learning For Image Recognition,” Cvpr, 2016.

[7] Ashish Vaswani Et Al., “Attention Is All You Need,” Neurips, 2017.

[8] Norman P. Jouppi Et Al., “In-Datacenter Performance Analysis Of A Tensor Processing Unit,” Isca, 2017.

[9] Yu-Hsin Chen Et Al., “Eyeriss: An Energy-Efficient Reconfigurable Accelerator For Deep Convolutional Neural Networks,” Ieee Journal Of Solid-State Circuits, 2017.

[10] Ping Chi Et Al., “Prime: A Novel Processing-In-Memory Architecture For Neural Network Computation In Reram-Based Main Memory,” Isca, 2016.

[11] Sheng Lin Et Al., “Fpga-Based Acceleration For Deep Neural Networks,” Ieee Design & Test, 2018.

[12] Abu Sebastian Et Al., “Memory Devices And Applications For In-Memory Computing,” Nature Nanotechnology, 2020.

[13] Geoffrey W. Burr Et Al., “Neuromorphic Computing Using Non-Volatile Memory,” Advances In Physics: X, 2017.

[14] Abhronil Sengupta Et Al., “Spintronic Devices For Neuromorphic Computing,” Nature Electronics, 2018.

[15] Shihui Yin Et Al., “A High-Efficiency Reram-Based Cim Accelerator For Deep Neural Networks,” Ieee Transactions On Circuits And Systems, 2021.

[16] Abu Sebastian, Manuel Le Gallo, And Evangelos Eleftheriou, “Computational Memory-Based Inference And Training Of Neural Networks,” Nature Communications, 2020.

[17] Diederik P. Kingma And Max Welling, “Auto-Encoding Variational Bayes,” Iclr, 2014.

[18] Tom Brown Et Al., “Language Models Are Few-Shot Learners,” Neurips, 2020.

[19] Jacob Devlin Et Al., “Bert: Pre-Training Of Deep Bidirectional Transformers For Language Understanding,” Naacl, 2019.

[20] Manuel Le Gallo Et Al., “Mixed-Precision In-Memory Computing,” Nature Electronics, 2018.

[21] Priyadarshini Panda Et Al., “Toward Scalable, Energy-Efficient, And Robust Neuromorphic Computing,” Proceedings Of The Ieee, 2020.

[22] Mark Horowitz, “1.1 Computing’s Energy Challenge,” Isscc, 2014.

[23] John L. Hennessy And David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 2019.

[24] Krste Asanović Et Al., “The Landscape Of Parallel Computing Research,” University Of California Report, 2006.
[25] Abu Sebastian Et Al., “Tutorial And Survey On In-Memory Computing For Deep Learning,” Nature Electronics, 2023.