Available Online: 10 March 2026 DOI: 10.54254/2977-3903/2026.32167

A review of In-Memory Computing with Non-Volatile Memory: challenges and opportunities

Youlin Wei

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China

youlin_ivy@outlook.com

Abstract. Traditional von Neumann architectures suffer from severe energy and latency overheads due to intensive data movement between memory and processing units. In-Memory Computing (IMC) integrates computation within memory arrays, greatly mitigating this bottleneck. This paper provides a comprehensive review of IMC principles, implementations, and challenges across Static Random-Access Memory (SRAM), Dynamic Random-Access Memory (DRAM), and Non-Volatile Memories (NVMs) such as Resistive Random Access Memory (RRAM), Magnetoresistive Random-Access Memory (MRAM), and Phase-Change Memory (PCM). We summarize architectural advances, device-level constraints, and system-level opportunities, with emphasis on the emerging class of resistive NVM-based IMC accelerators. Furthermore, we highlight engineering trade-offs, real-world application scenarios, and current industrial standardization efforts, offering guidance toward large-scale deployment of IMC technologies.

Keywords: In-Memory Computing, RRAM, neuromorphic computing, resistive switching, relaxation effect, AI accelerators

1. Introduction

The rapidly growing data-intensive workloads such as Artificial Intelligence (AI), high-performance computing, and 3D rendering are consuming amounts of data, and conventional computing architectures are struggling to keep up. Since its conception in 1945, the von Neumann architecture has dominated modern computer design, separating the processing unit and the memory unit, requiring the processor to repeatedly fetch data through a bus with limited bandwidth. An empirical study shows that data transfer has been reported to account for nearly 62.7% of the total energy consumption in Google workloads [1].

To overcome this difficulty, often called the "memory wall", researchers have been considering alternative computing paradigms to reduce data movement by bringing computation closer or integrating it with the memory. In-Memory Computing (IMC) is an emerging computational paradigm designed to overcome the fundamental limitations of the von Neumann architecture, where data must shuttle continuously between physically separated memory and processing units. IMC addresses this challenge by performing arithmetic or logical operations directly inside or near memory arrays, dramatically reducing data movement and enabling massive parallelism. The core idea of IMC is to exploit the intrinsic physical properties of memory devices—such as charge sharing in DRAM [2, 3], bitline current summation in SRAM, or conductance accumulation in

resistive memories—to execute operations like Multiply-Accumulate (MAC), bitwise logic, or vector-matrix multiplication where the data already resides. This allows IMC architectures to deliver significant improvements in throughput and energy efficiency while retaining compatibility with Complementary Metal-Oxide-Semiconductor (CMOS) processes and memory-centric system designs. As memory technologies such as SRAM [4], DRAM, RRAM, MRAM, and PCM continue to evolve, IMC provides a scalable pathway toward energy-efficient computing for both edge and cloud applications, making it a promising foundation for next-generation AI accelerators and data-intensive systems.

In the following sections, we first introduce IMC architectures based on mature Dynamic Random-Access Memory (DRAM) technology, and then turn our focus to Non-Volatile Memory (NVM) solutions that offer improved scalability, efficiency, and computational capability.

2. DRAM-IMC

DRAM is widely used for in-memory computing due to its high speed, low latency, and mature fabrication technology. In-memory computing with DRAM leverages the intrinsic charge storage and charge sharing mechanisms of DRAM cells to perform simple logic or arithmetic operations directly within the memory array [2, 3], reducing the need for data movement between the memory and the processor [5]. This approach helps overcome the traditional von Neumann bottleneck, where data transfer limits system performance and energy efficiency. However, DRAM's volatile nature means that data must be frequently refreshed and cannot be retained without power, limiting its suitability for long-term data storage or energy-efficient standby operation [6].

A 2017 study conducted by Seshadri et al. [2] reveals a promising architecture of IMC design based on DRAM, which is called Ambit. Ambit is an architecture that aims to exploit existing DRAM analog operation to perform computation inside memory with only minor modifications. Ambit can execute AND, OR and NOT operations in DRAM arrays, on which more complicated logic can be built. By activating three rows (denoted as A, B, C) in the same subarrays simultaneously, the operation is electrically equivalent to $bit\_out = MAJ(A, B, C)$. This mechanism is called Triple-Row Activation (TRA). The final bitline voltage is determined by the majority of charge states:

• Bitwise AND implementation: The initial values of C control the logic functions of Ambit. If all of cells in C are filled with zeros, the output is at 1 only when A and B are set to 1.

• Bitwise OR implementation: If all of cells in C are filled with ones, the output is at 1 if either A or B is set to 1.

Since three rows are utilized simultaneously each time, the address communicating and decoding place a great pressure on address bus and row decoder. The problem is resolved by reserving some designated rows. Each set of three designated rows is mapped to one single address, dramatically reducing the burden on the address bus.

Besides bitwise AND-OR, Ambit also supports NOT operation. To enable NOT operation, Ambit introduces a specific structure called Dual-Contact Cells (DCCs). A DCC is a DRAM cell that consists of Two Access Transistors and One Capacitor (2T1C). Compared to commodity 1T1C structure, the additional transistor enables DCC cell to connect to the bitline or its complementary bitline-bar. When a source row is activated, the sense amplifier detects the charge gap. Consequently, the bitline and bitline-bar are driven simultaneously with value $\pm A$ respectively. The activation of wordlines will copy the stored inverted data into a target row.

The researchers declare that only less than one percentage of DRAM chip area is required to reserve for designated rows and DCC rows, which is spatially efficient [2]. Compared with DDR3-based systems, Ambit achieves an estimate of 35 × energy reduction.

A later study by Xin et al. proposed the Reduced Operation Cycle (ROC) [7], another DRAM-based in-memory computing design. It features higher operational cycle efficiency. Unlike Ambit, which requires three activated rows to share their charge states, ROC introduces an embedded diode-connected transistor to enable AND, OR and NOT logical functions. Multiple rounds of activation and pre-charge rounds are involved in Ambit, whereas ROC requires only two commands to perform AND and OR operations. ROC also achieves and optimizes both uni-directional and bi-directional propagation along with shift, extending bitwise to word-wise operations. ROC maintains 3% area overhead, which is slightly higher than that of Ambit (less than 1%), but it greatly mitigates the inaccuracy by replacing triple-row charge sharing requirements with a proposed computation unit.

Overall, DRAM-based in-memory computing demonstrates that substantial computational capability can be unlocked within commodity memory arrays using only minimal architectural modifications. Designs such as Ambit [2] and ROC [7] highlight two complementary directions: exploiting intrinsic DRAM charge-sharing behaviors for bulk bitwise logic, and incorporating lightweight compute-aware circuitry to improve accuracy and reduce cycle overhead. These architectures preserve DRAM's core advantages—high density, low cost, and full compatibility with existing manufacturing and interface standards—while enabling significant reductions in data movement and system-level energy consumption. Despite unavoidable limitations such as refresh requirements and volatility, DRAM-IMC represents a practical and immediately deployable pathway toward near-data computation, particularly for bitwise, bulk memory operations that dominate many data-intensive workloads. As future DRAM generations evolve and system designers increasingly seek low-overhead compute integration, DRAM-IMC is poised to play an important transitional role in bridging conventional memory hierarchies with emerging compute-in-memory paradigms.

3. Non-volatile memory IMC

3.1. Promise of NVM IMC: an overview

Resistive Random-Access Memory (RRAM), a non-volatile memory technology, offers significant potential for in-memory computing due to its ability to store data through variable resistance states and retain information without power [8]. RRAM operates by changing the resistance of a dielectric material—typically a metal oxide—through the formation and rupture of conductive filaments when a voltage is applied across the device. A Low-Resistance State (LRS) represents a logic "1," while a High-Resistance State (HRS) represents a logic "0." These resistance changes are reversible and can be precisely controlled by tuning the applied voltage or current [8]. RRAM can be classified as two categories: analog RRAM and binary RRAM according to the number of functional resistance states. Analog RRAM can adjust its conductivity and produce any value between the LRS and HRS. The other type of RRAM is called binary RRAM, serving as a dichotomous device with only 1's and 0's.

A representative example of binary RRAM to accelerate Multiply-and-Accumulate (MAC) operations is the parallel XNOR-RRAM architecture proposed by Sun et al. [9]. This work investigated a special neural network, Binary Neural Network, whose weights and activations are polarized as +1 or -1. Given this feature, BNN multiplications can be expressed as XNOR operations followed by bit-counting. Specifically, a pair of complementary RRAM cells in 2T2R structure is used to represent a single synaptic weight. The upper cell is programmed to HRS to represent a weight of -1, while the reversed pattern was used for a positive weight.

Similarly, two adjacent complementary WLs are used to represent the input activations, where $(1,0)$ encodes $+1$ and $(0,1)$ encodes -1. During read-out, the bit line current is dependent on whether LRS or HRS RRAM cell is activated. Theoretically, XNOR-RRAM can even reduce data movements more significantly than XNOR-SRAM architecture due to its non-volatility characteristic and smaller cell footprints (2T2R vs 6T). The experimental results confirm the expectation by demonstrating that XNOR-RRAM improves energy and area efficiency by $10\times - 30\times$ over XNOR-SRAM architecture.

The implementation of binary-RRAM is more practical and less likely to incur bit errors without error-correcting codes, because its polarized resistance is easy to manage. And analog-RRAM is more susceptible to device variations and conductance drift. However, binary RRAM's single-bit weight precision restrains task complexity and limits its ability to support complicated networks. In contrast, analog-RRAM offers an advantage in improving weight precision, because it can represent weights with 4-8 bits.

In practice, analog-RRAM computing requires Digital-to-Analog (DAC) conversion to encode input vectors, as well as Analog-to-Digital (ADC) conversion for digitizing output. Analog-RRAM can support a wide range of advanced neural networks. Hu et al. [10] validated this capability by developing a fully integrated neuro-inspired HfO_x memristor chip based on that facilitates in-chip learning.

RRAM-based in-memory computing exploits the resistive switching properties of memory cells to perform parallel analog or digital computations, such as vector-matrix multiplications, which are essential for artificial intelligence and machine learning applications. Its non-volatility enables instant-on computing and improved energy efficiency, as data persistence eliminates the need for data reload after power loss. Furthermore, the cell area of RRAM is significantly smaller than that of SRAM ($> 100F^2$) and DRAM ($6F^2$), which is less than $4F^2$. This spatial efficiency enables it to be fabricated into a three-dimensional stacked structure [11].

RRAM presents a promising path toward highly efficient, scalable, and low-power computing architectures beyond traditional CMOS limits. In a recent demonstration by Wan et al. [5], ~3 million analogue-programmable RRAM cells (1T1R configuration) are integrated monolithically with CMOS and supports diverse AI models including CNNs, LSTMs and RBMs, thereby achieving versatility in data-flow mapping across multiple cores. The authors have been able to achieve an Energy-Delay Product (EDP) approximately 1.6–2.3$\times$ better and computational density 7–13 $\times$ higher than previous RRAM-CIM chips. High inference accuracy can be achieved by quantizing the model to 4-bit weights, achieving 99.0% on MNIST, 85.7% on CIFAR-10 and 84.7% on Google Speech Commands.

Phase-Change Memory (PCM) is another promising non-volatile memory technology for IMC, leveraging the reversible phase transition between the amorphous and crystalline states of chalcogenide materials [12]. The large and controllable resistance contrast between these two phases enables analog or multi-level storage, making PCM particularly attractive for implementing high-precision vector–matrix multiplications. Unlike binary RRAM, PCM naturally supports incremental programming, allowing fine-tuned conductance states that correspond to multi-bit weights. This capability has been widely explored for neuromorphic and in-memory acceleration of neural networks, where multi-level analog precision can significantly reduce quantization loss and improve model accuracy.

PCM also exhibits strong temperature retention and excellent cycling endurance compared to filamentary RRAM devices. However, its principal challenges stem from high programming energy—which results from the need to heat the phase-change material—and temporal conductance drift, especially in the amorphous state. This drift follows a power-law decay, gradually shifting programmed weights and degrading accuracy over time. Despite these limitations, recent PCM-based IMC prototypes have demonstrated competitive performance for inference workloads, with high computational parallelism, robust analog weight storage, and energy benefits when amortized across large batch computations. As programming schemes and drift

compensation algorithms continue to improve, PCM remains a compelling candidate for high-precision and large-scale in-memory computing.

Spin-Transfer Torque MRAM (STT-MRAM) is another emerging NVM technology with potential IMC applications, particularly in digital and reliability-critical workloads. MRAM stores information using the magnetic orientation of a free layer relative to a fixed layer in a Magnetic Tunnel Junction (MTJ), resulting in resistance states defined by tunneling magnetoresistance. Because this switching relies on magnetic rather than ionic movement, MRAM offers excellent endurance, fast read access, and strong CMOS manufacturing compatibility. These characteristics make MRAM suitable for IMC architectures that require frequent updates or robust binary operations.

Although MRAM is less naturally suited for analog weight storage compared to PCM or RRAM, recent studies have shown that MTJ arrays can implement bitwise logic operations—such as NAND, NOR, or majority functions—directly through current-mode sensing. These digital IMC primitives can accelerate workloads dominated by binary operations, such as graph processing, search, encryption, or binary neural networks. The main challenges of MRAM-based IMC include relatively large MTJ cell size (limiting density compared to RRAM) and the difficulty of achieving stable analog-level programming due to stochastic switching behaviors. Still, MRAM's high endurance and read stability make it an appealing option for IMC accelerators that prioritize reliability, non-volatility, and low standby power.

In conclusion, each memory technology exhibits distinct trade-offs in energy efficiency, speed, density, and computational accuracy, driven by its underlying physical operating principles. Table 1 provides a comparative summary of these characteristics across the major IMC memory technologies.

Table 1. Comparison of major memory technologies for In-Memory Computing

Technology	SRAM	DRAM	RRAM	PCM	STT-MRAM
Volatility	Volatile	Volatile	Non-Vol.	Non-Vol.	Non-Vol.
Density	Low	Medium	High	Med-High	Medium
Speed	Very Fast	Fast	Medium	Medium	Fast
Analog Cap.	Strongly Analog-Friendly	Limited Analog Ops	Excellent	Strong MLC	Binary
Advantages	High accuracy; mature CMOS	High density; minimal process change	3D-stackable; analog MAC	Temperature stability; commercial maturity	Endurance; CMOS compatibility
Limitations	Large area (6T); poor scalability	Refresh overhead; charge-based drift	Variability; endurance; relaxation	High programming power	Limited analog operation

Note: Cap. = Capability; Non-Vol. = Non-Volatile

3.2. Challenges: a case study with RRAM

3.2.1. Large energy and area overhead of ADC and DAC circuits

The major challenge of RRAM-based IMC lies in the large energy and area overhead associated with the peripheral digital-to-analog converters and analog-to-digital converters. Although RRAM crossbars provide massive parallelism for Multiply-Accumulate (MAC) operations, the interface circuitry needed to communicate between the analog memory core and digital processing units often dominates total system cost. In particular, multi-bit DACs are required to apply precise input voltages, while high-resolution ADCs are needed to sense and digitize output currents from each array column. These converters consume a significant portion of the total energy and chip area—often exceeding that of the RRAM array itself—thus limiting the achievable energy efficiency and scalability of IMC systems [13]. For example, in the NeuRRAM architecture,

although substantial co-optimization reduced peripheral overhead, converter energy and area remain critical bottlenecks to further scaling and integration density [5].

3.2.2. Conductance relaxation effect

One of the key reliability challenges in RRAM devices for IMC is resistance relaxation, also known as temporal instability [8]. After programming an RRAM cell to a target resistance state, its conductance gradually changes over time due to thermally activated diffusion and spontaneous rearrangement of oxygen vacancies and metal ions [11, 13] within the dielectric layer. This relaxation can occur over time scales ranging from milliseconds to hours, even under no electrical bias, leading to deviations from the originally intended weight values in neuromorphic and IMC applications. As a result, the effective synaptic weights stored in RRAM crossbars may lose precision or even change polarity, directly degrading inference accuracy [5, 10] and long-term stability of neural network models. Because IMC systems rely on analog conductance states to represent numerical weights, such relaxation undermines both computational accuracy and device reliability, especially for large-scale arrays with millions of cells.

The seriousness of resistance relaxation stems from its cumulative and stochastic nature, which makes it difficult to compensate purely through calibration or redundancy. In inference workloads requiring static weight storage, slow drift can progressively distort vector-matrix multiplication outputs; in training or on-chip learning, where weight updates are frequent, relaxation further introduces asymmetric programming errors that slow convergence. Moreover, relaxation effects often exhibit device-to-device and cycle-to-cycle variability, complicating circuit-level compensation schemes. As RRAM IMC architectures continue to scale, the relative impact of such variability grows due to the analog accumulation of small conductance errors across thousands of parallel devices, making resistance relaxation one of the most serious barriers to achieving stable, high-precision analog computation.

Although the negative effect caused by conductance relaxation effect is prominent and has been recognized, and several physical models have been provided to measure the severity of oxygen vacancy drift and filament morphology, most of the existing physical models are too complicated to implement or reproduce in large-scale neural networks. Therefore, instead of evaluating on a complex analytical model, Zhai et al. [14] established a more feasible method utilizing trained neural networks for simulating conductivity drift to quantitatively evaluate how severe this problem is.

To shed more light onto the impact of conductance relaxation on IMC performance, we emulate conductance drift by introducing stochastic perturbations to the trained neural network in two scenarios:

• weights alone are perturbed, and

• weights along with biases are perturbed.

A fully connected neural network has been trained on the MNIST dataset and used as the benchmark model. The input layer of this architecture includes 784 neurons, followed by a hidden layer consisting of 128 ReLU-activated neurons, and then an output layer out of 10 neurons corresponding to 10 digit classes.

For each experiment, a non-perturbed baseline model is trained and converges. Then the trained weights are perturbed following the aforementioned noise model, and the perturbed model is straightly evaluated without retraining. The accuracy gap between the non-perturbed and perturbed model is measured.

In order to ensure the computational efficiency of this simulation, we adopt a simplified version of the conductance relaxation model based on stochastic perturbations. Instead of explicitly modeling oxygen-vacancy drift or filament morphology, weight w in the network is perturbed by zero-mean Gaussian noise:

$$w' = w + \Delta, \Delta \sim N(0, \sigma^2), \tag{1}$$

where σ stands for levels of conductance degradation severity.

(a) Weight Perturbation

Python code:

```
with torch.no_grad():
    for name, param in model.named_parameters():
        if 'weight' in name:
            param.add_(sigma * torch.randn_like(param))
```

Figure 1. Impact of weight perturbation on model ac curacy

(b) Weight + Bias Perturbation

Python code:

```
with torch.no_grad():
    for name, param in model.named_parameters():
        param.add_(sigma * torch.randn_like(param))
```

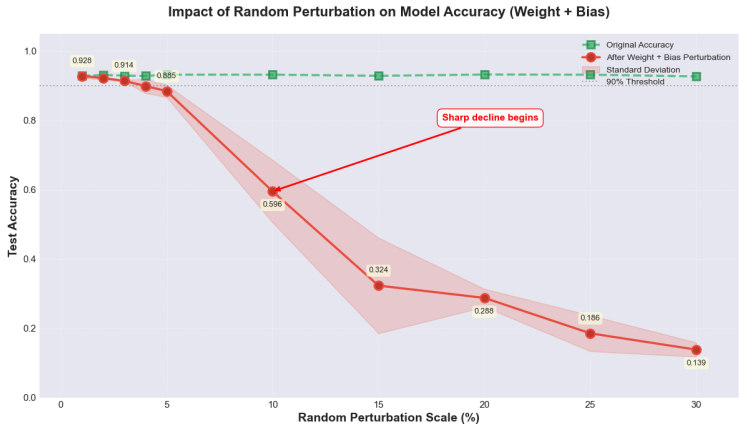

Figure 2. Impact of weight + bias perturbation on model accuracy

Figure 1 shows the impact of perturbations on model accuracy when the biases remain unchanged. As the σ increases from 0.01 to 0.30, when $\sigma < 0.05$, the accuracy drops less than 2%, which is tolerable. A break

of the stability occurs when σ reaches 0.10, where accuracy plummets to 54.1%. At the highest perturbation level ($\sigma = 0.30$), accuracy drops to 15.6%, barely higher than the probability of random guessing.

Figure 2 illustrates the results when both weights and biases are perturbed at the same time. The degradation resembles that of weight-only perturbation. When σ increases to 10%, accuracy drops to 59.6%. At $\sigma = 0.30$, it falls to 13.9%. The similarity suggests that weight perturbations contribute to the majority of the accuracy degradation, while the impact of perturbative noise on bias is minor.

In summary, the relationship between perturbation strength and accuracy degradation does not conform to linearity. When σ is not larger than 0.05, the accuracy maintains over 90%, but performance stability collapses when σ exceeds 0.10. Consequently, to maintain a desirable accuracy, we should constrain the conductance to $\sigma \lesssim 0.05$.

3.3. Mitigation of challenges: new opportunities

3.3.1. DAC and ADC circuits challenges

To mitigate the large energy and area overhead of peripheral DAC/ADC circuits, recent research has explored several engineering strategies spanning device materials, circuit architectures, and algorithm–hardware co-design. At the circuit level, one practical direction is to reduce the precision requirement of converters through quantization-aware training or mixed-precision techniques, enabling the use of low-bitwidth DACs and ADCs (e.g., 3–6 bits) [15]. Such converters are significantly smaller and less power-hungry, and their impact on inference accuracy is minimal when training is adapted to the reduced precision. For example, NeuRRAM demonstrates that model retraining under device-constrained quantization can reduce ADC energy by more than 50 percent, while maintaining state-of-the-art accuracy [16]. Another approach is the development of charge-domain [5] or time-domain [17] CIM architectures, which replace analog current integration with switched-capacitor or pulse-width–modulated encoding. These schemes shift accumulation into the digital domain, eliminating the need for high-resolution ADCs at every column and drastically lowering total converter power.

From a circuit-design standpoint, Successive Approximation Register (SAR) ADCs with shared comparator arrays, in-memory digital accumulation buffers, and column-parallel yet time-multiplexed converter banks have been proposed to amortize ADC cost across many rows. For instance, using a shared 6-bit SAR ADC [18] per subarray instead of per column reduces area overhead by nearly an order of magnitude, though designers must carefully manage timing and routing congestion. Similarly, hierarchical conversion, where coarse analog results are digitized locally and fine-grained corrections are computed digitally, can reduce both conversion time and energy. At the device-material level, improved linearity and conductance ranges in RRAM cells enable simpler low-resolution ADCs, since the dynamic range of sensed currents becomes narrower and easier to quantize accurately. However, such material optimization introduces trade-offs with endurance and retention, requiring careful balancing of forming voltage, filament stability, and read-disturb robustness.

Finally, algorithm–hardware co-design plays a critical role in converter mitigation. Techniques such as analog-aware training, weight clipping, input normalization, and error-tolerant mapping allow neural networks to operate reliably under noisy or low-precision conversion. Some architectures adopt hybrid analog–digital computing, performing MAC operations in analog while shifting nonlinear activations, normalization layers, or error accumulation into the digital domain. This reduces the number of ADC operations per inference cycle and improves robustness against analog noise. Together, these material-, circuit-, and algorithm-level strategies substantially alleviate DAC/ADC overhead and point toward practical, scalable RRAM-based IMC architectures that can be deployed across both edge devices and data-centric computing platforms.

3.3.2. Resistance relaxation challenges

To mitigate resistance relaxation, researchers have explored a combination of materials engineering, circuit compensation, and algorithm–hardware co-design approaches. On the materials side, introducing more stable switching oxides—such as $HfAlO_x$ or TaO_x —helps reduce oxygen vacancy mobility, thereby suppressing filament reshaping after programming. These materials also enable narrower conductance distributions, which reduces the burden on peripheral sensing circuits. However, incorporating such materials often requires additional ALD precursors or modified annealing processes, increasing fabrication complexity and potentially impacting BEOL thermal budgets. Optimized electrode interfaces (e.g., TiN/TaO_x stacks) can further stabilize the filament by creating stronger vacancy reservoirs, but this also introduces challenges in achieving uniform device-to-device performance due to variations in interfacial stoichiometry.

At the device-operation level, techniques such as self-compliance programming, Incremental Step Pulse Programming (ISPP) [19], and post-forming annealing [20, 21] help suppress relaxation by minimizing over-programming and residual thermal stress in the filament. For example, ISPP schemes apply gradually increasing pulses and verify conductance after each step, significantly reducing initial drift but at the cost of longer programming latency and increased peripheral control overhead. Thermal stabilization through mild annealing can also reduce relaxation amplitude by encouraging more stable filament geometry, but this must be carefully controlled to avoid altering programmed conductance states.

At the circuit and algorithm levels, practical compensation techniques such as closed-loop write–verify, on-chip refresh, and periodic recalibration are used to restore drifting weights. A representative example is the NeuRRAM architecture, where weight drift is mitigated by occasional low-energy reprogramming pulses triggered based on column-level sensing statistics. While effective, this introduces nontrivial scheduling challenges, as refresh operations must be interleaved with active inference without exceeding energy or latency budgets. Algorithmically, drift-aware training, bounded weight clipping, and robust quantization improve resilience by shaping the neural network such that small conductance variations produce minimal accuracy degradation. Some recent works even exploit the logarithmic time-dependence of drift to implement predictive compensation, where weights are pre-distorted during programming so that they decay toward target values over time.

Together, these material-, device-, circuit-, and algorithm-level strategies represent a multi-layer mitigation framework that significantly improves long-term stability and accuracy retention in RRAM-based IMC. Although each solution introduces trade-offs—such as increased fabrication complexity, added circuit control overhead, or additional training cost—these approaches collectively enhance the viability of RRAM IMC for practical AI inference and edge computing systems.

4. Future outlook

In practice, the selection of an IMC architecture depends heavily on the characteristics and constraints of the target application. For edge AI accelerators—including smart sensors, drones, wearables, and mobile devices—energy efficiency and compact footprint are the dominant considerations. RRAM- and PCM-based analog IMC architectures excel in these scenarios because their massively parallel vector–matrix multiplications reduce data movement and enable sub-milliwatt inference operation. These architectures are especially beneficial for always-on workloads such as keyword spotting, gesture recognition, and event-driven anomaly detection, where low-leakage non-volatility reduces standby power.

In contrast, data center and cloud environments prioritize throughput, reliability, and compatibility with existing high-bandwidth memory systems. DRAM-based IMC solutions such as Ambit and ROC are

particularly effective for accelerating bitwise operations that dominate large-scale analytics, database scanning, search, and memory-intensive AI workloads. Their ability to operate directly within commodity DRAM arrays allows data centers to gain significant bandwidth and energy advantages without requiring disruptive changes to system architecture. Meanwhile, MRAM-based digital IMC has gained interest for reliability-critical workloads such as secure search, encryption, and memory-bound sparse computations due to its endurance and deterministic switching behavior.

Together, these application-driven examples illustrate that the practical deployment of IMC depends on workload characteristics: analog NVM IMC is favored for resource-constrained edge inference, whereas DRAM-based IMC suits large-scale, throughput-intensive cloud operations. Hybrid IMC architectures that combine NVM analog MACs with DRAM-based digital bitwise acceleration are likely to play an increasingly important role in heterogeneous computing systems.

IMC is also beginning to gain traction in industry, supported by early standardization efforts. The IEEE P3103 initiative is currently working toward defining a unified IMC taxonomy, architectural abstraction, and test methodology, providing a foundation for broader interoperability. At the memory-system level, JEDEC working groups have begun exploring IMC extensions within future LPDDR, DDR, and HBM standards, recognizing the increasing need for near-memory compute in AI and data-centric workloads. These standardization efforts are essential for ensuring that IMC subsystems can be integrated seamlessly into existing SoC and accelerator platforms.

Several semiconductor companies have demonstrated early IMC prototypes, underscoring growing industrial interest. For example, IBM has released PCM-based analog compute chips demonstrating high-precision VMM operations [22, 23]. Samsung has also contributed to PCM device engineering [24]. Meanwhile TSMC and Intel provide embedded RRAM/MRAM platforms that support IMC research and low-power AI inference. DRAM manufacturers have also explored integrating Ambit-like mechanisms into commodity DRAM arrays with minimal area overhead, indicating a realistic pathway toward large-scale adoption in data centers. Despite this progress, challenges remain in software–hardware co-design, including compiler support, device-aware quantization frameworks, model mapping tools, and integration with mainstream machine learning libraries.

Cost and compatibility also play critical roles in IMC deployment. For NVM-based IMC, the introduction of additional fabrication steps—such as oxide doping, interface engineering, or 3D stacking—can raise manufacturing cost and impact yield. DRAM IMC offers the lowest integration barrier but provides more limited compute precision. Furthermore, IMC architectures must maintain compatibility with conventional memory interfaces, thermal constraints, and system-level power budgets. Addressing these issues through standardization, streamlined toolchains, and process co-optimization will be critical for enabling widespread industrial promotion and long-term sustainability of IMC technology.

5. Conclusion

In-Memory Computing (IMC) has become a promising solution to the von Neumann bottleneck by enabling computation directly within memory arrays. This review examined IMC across both mature and emerging memory technologies. DRAM-based architectures such as Ambit and ROC demonstrate that substantial computational capability can be unlocked with minimal modifications to commodity memory, providing an immediate and practical path for accelerating large-scale bitwise and memory-centric operations in data centers. Meanwhile, emerging non-volatile memories—RRAM, PCM, and MRAM—offer higher density, non-volatility, and support for analog or multi-level computation, enabling more efficient vector–matrix

multiplication for edge AI, scientific computing, and low-power inference. Each memory type exhibits distinct strengths and limitations shaped by its switching physics, precision, scalability, and peripheral circuit requirements.

Realizing robust IMC systems also requires addressing device- and circuit-level challenges. Techniques such as materials engineering, filament stabilization, controlled programming, and drift compensation improve device reliability, while circuit strategies—including reduced-precision conversion, charge-domain accumulation, hierarchical ADC sharing, and hybrid analog–digital designs—help mitigate DAC/ADC overhead. Algorithm–hardware co-design further enhances system robustness through quantization-aware and drift-aware training. Practical deployment scenarios show that analog NVM IMC is well suited for energy-constrained edge workloads, whereas DRAM IMC naturally fits bandwidth-bound data center tasks.

Standardization and industrial efforts are accelerating the transition of IMC from research to commercial adoption. Early IEEE and JEDEC activities, along with prototypes from major semiconductor companies, indicate strong momentum toward integrating IMC into future computing systems. As memory technologies mature and cross-layer optimization continues, IMC is poised to play an increasingly central role in next-generation AI accelerators and data-centric architectures.

References

[1] Mutlu, O., Ghose, S., Gómez-Luna, J., & Ausavarungnirun, R. (2019). *Enabling practical processing in and near memory for data-intensive computing*. arXiv preprint. https: //arxiv.org/abs/1905.04376

[2] Seshadri, V., Lee, D., Mullins, T., Hassan, H., Boroumand, A., Kim, J., Kozuch, M. A., Mutlu, O., Gibbons, P. B., & Mowry, T. C. (2017). Ambit: In-memory accelerator for bulk bitwise operations. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-50)* (pp. 273–287). ACM. https: //doi.org/10.1145/3123939.3124544

[3] Seshadri, V., Kim, Y., Fallin, C., Lee, D., Ausavarungnirun, R., Pekhimenko, G., Luo, Y., Mutlu, O., Gibbons, P. B., Kozuch, M. A., & Mowry, T. C. (2013). RowClone: Fast and energy-efficient in-DRAM bulk data copy and initialization. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-46)* (pp. 185–197). ACM. https: //doi.org/10.1145/2540708.2540725

[4] Verma, N., Jia, H., Valavi, H., Tang, Y., Ozatay, M., Chen, L.-Y., Zhang, B., & Deaville, P. (2019). In-memory computing: Advances and prospects. *IEEE Solid-State Circuits Magazine, 11*(3), 43–55. https: //doi.org/10.1109/MSSC.2019.2922889

[5] Wan, W., Kubendran, R., Schaefer, C., Eryilmaz, S. B., Zhang, W., Wu, D., Deiss, S., Raina, P., Qian, H., Gao, B., Joshi, S., Wong, H.-S. P., & Cauwenberghs, G. (2022). A compute-in-memory chip based on resistive random-access memory. *Nature, 608*, 504–512. https: //doi.org/10.1038/s41586-022-04992-8

[6] Mutlu, O. (2023). *Retrospective: RAIDR: Retention-aware intelligent DRAM refresh*. arXiv preprint. https: //arxiv.org/abs/2306.16024

[7] Xin, X., Zhang, Y., & Yang, J. (2019). ROC: DRAM-based processing with reduced operation cycles. In *Proceedings of the 56th ACM/IEEE Design Automation Conference (DAC)* (pp. 1–6). IEEE. https: //doi.org/10.1145/3316781.3317766

[8] Yan, B., Li, B., Qiao, X., Xue, C.-X., Chang, M.-F., Chen, Y., & Li, H. (2022). Resistive memory-based in-memory computing: From device and large-scale integration system perspectives. *IEEE Transactions on Circuits and Systems I: Regular Papers, 69*(12), 5021–5036. https: //doi.org/10.1109/TCSI.2022.3201382

[9] Sun, X., Liu, R., Peng, X., & Yu, S. (2018). Computing-in-memory with SRAM and RRAM for binary neural networks. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–5). IEEE. https: //doi.org/10.1109/ISCAS.2018.8351806

[10] Zhang, W., Yao, P., Gao, B., Liu, Q., Wu, D., Zhang, Q., Li, Y., Qin, Q., Li, J., Zhu, Z., Cai, Y., Wu, D., Tang, J., Qian, H., Wang, Y., & Wu, H. (2023). Edge learning using a fully integrated neuro-inspired memristor chip. *Science, 381*(6663), 1205–1211. https: //doi.org/10.1126/science.ade3483

[11] Zahoor, F., Zulkifli, T. Z. A., & Khanday, F. A. (2020). Resistive random access memory (RRAM): An overview of materials, switching mechanism, performance, multilevel cell (MLC) storage, modeling, and applications. *Nanoscale Research Letters, 15*(1), Article 90. https: //doi.org/10.1186/s11671-020-03299-9

[12] Burr, G. W., Shelby, R. M., Sebastian, A., Kim, S., Kim, S., Sidler, S., Virwani, K., Ishii, M., Narayanan, P., Fumarola, A., Sanches, L. L., Boybat, I., Gallo, M. L., Moon, K., Woo, J., Hwang, H., & Leblebici, Y. (2017). Neuromorphic computing using non-volatile memory. *Advances in Physics: X, 2*(1), 89–124. https: //doi.org/10.1080/23746149.2016.1259585

[13] Yu, S. (2018). Neuro-inspired computing with emerging nonvolatile memory. *Proceedings of the IEEE, 106*(2), 260–285. https: //doi.org/10.1109/JPROC.2018.2790840

[14] Zhai, X., Kang, Y., & Tian, L. (2025). An analytical model of RRAM relaxation effect and its application for neural network weight refresh strategy in large-scale RRAM array. *IEEE Transactions on Electron Devices*. Advance online publication. https: //doi.org/10.1109/TED.2025.3526789

[15] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. In *International Conference on Learning Representations (ICLR)*. https: //openreview.net/forum?id=HkZtjibFl

[16] Krestinskaya, O., Zhang, L., & Salama, K. N. (2023). Towards efficient in-memory computing hardware for quantized neural networks: State-of-the-art, open challenges and perspectives. *IEEE Transactions on Nanotechnology, 22*, 377–386. https: //doi.org/10.1109/TNANO.2023.3291584

[17] Mayahinia, M., Singh, A., Bengel, C., Wiefels, S., Lebdeh, M. A., Menzel, S., Wouters, D. J., Gebregiorgis, A., Bishnoi, R., Joshi, R. V., & Hamdioui, S. (2022). A voltage-controlled, oscillation-based ADC design for computation-in-memory architectures using emerging ReRAMs. *ACM Journal on Emerging Technologies in Computing Systems, 18*(2), Article 32, 1–25. https: //doi.org/10.1145/3501773

[18] Spear, M., Kim, J. E., Bennett, C. H., Agarwal, S., Marinella, M. J., & Xiao, T. P. (2023). The impact of analog-to-digital converter architecture and variability on analog neural network accuracy. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 9*(2), 133–141. https: //doi.org/10.1109/JXCDC.2023.3314898

[19] Liu, J.-C., Wu, T.-Y., & Hou, T.-H. (2018). Optimizing incremental step pulse programming for RRAM through device–circuit co-design. *IEEE Transactions on Circuits and Systems II: Express Briefs, 65*(5), 617–621. https: //doi.org/10.1109/TCSII.2018.2828225

[20] He, H., Tan, Y., Lee, C., & Zhao, Y. (2023). Ti/HfO_2-based RRAM with superior thermal stability based on self-limited TiO_x. *Electronics, 12*(11), Article 2426. https: //doi.org/10.3390/electronics12112426

[21] Zhao, J., Li, Y., Li, J., & Zhou, L. (2021). Role and optimization of thermal rapid annealing in Ta/TaO_x/Ru based resistive switching memory. *Journal of Alloys and Compounds, 875*, Article 160905. https: //doi.org/10.1016/j.jallcom.2021.160905

[22] Nandakumar, S. R., Boybat, I., Yi, X., Piveteau, C., Gallo, M. L., Rajendran, B., Sebastian, A., & Eleftheriou, E. (2020). Accurate deep neural network inference using computational phase-change memory. *Nature Communications, 11*(1), Article 2473. https: //doi.org/10.1038/s41467-020-16164-9

[23] Sebastian, A., Gallo, M. L., Khaddam-Aljameh, R., & Eleftheriou, E. (2020). Memory devices and applications for in-memory computing. *Nature Nanotechnology, 15*(7), 529–544. https: //doi.org/10.1038/s41565-020-0655-z

[24] Park, S.-O., Jeong, Y., Lee, S., Kim, T., Cho, S., Kim, H., Lee, J., Kim, J., Park, J., Kim, H., Lee, J., Kim, J., Park, J., Kim, H., Lee, J., Kim, J., Park, J., Kim, H., Lee, J., … Hwang, H. (2024). Phase-change memory via a phase-changeable self-confined nano-filament. *Nature, 628*, 293–298. https: //doi.org/10.1038/s41586-024-07230-5