Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Memristor-based adaptive analog-to-digital conversion for efficient and accurate compute-in-memory

Zhengwu Liu
zwliu@eee.hku.hk
The University of Hong Kong https://orcid.org/0000-0001-7968-9469
Haiqiao Hong
The University of Hong Kong https://orcid.org/0009-0009-6392-1564
Zhiyuan Du
The University of Hong Kong
Mingrui Jiang
The University of Hong Kong https://orcid.org/0009-0000-2364-3904
Ruibin Mao
University of Hong Kong https://orcid.org/0000-0001-6085-0486
Yuan Ren
The University of Hong Kong https://orcid.org/0000-0002-6373-2685
Wei Mao
Xidian University
Muyuan Peng
The University of Hong Kong
Wei Zhang
Hong Kong University of Science and Technology
Can Li
The University of Hong Kong https://orcid.org/0000-0003-3795-2008
Ngai Wong
The University of Hong Kong https://orcid.org/0000-0002-3026-0108

Article

Keywords:

Posted Date: March 25th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6187057/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

Version of Record: A version of this preprint was published at Nature Communications on November 6th, 2025. See the published version at https://doi.org/10.1038/s41467-025-65233-w.

Memristor-based adaptive analog-to-digital conversion for efficient and accurate compute-in-memory

Haiqiao Hong[1], Zhiyuan Du[1], Mingrui Jiang[1], Ruibin Mao[1], Yuan Ren[1], Wei Mao[2], Muyuan Peng[1], Wei Zhang[3], Zhengwu Liu[1,*], Can Li[1,*], Ngai Wong[1,*]

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China.

[2]Hangzhou Institute of Technology, Xidian University, Hangzhou, China.

[3]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.

[*]Email: zwliu@eee.hku.hk (Z.L.); canl@hku.hk (C.L.); nwong@eee.hku.hk (N.W.).

Abstract:

Compute-in-memory (CIM) technology offers promising solutions for neural network acceleration with high energy efficiency. However, the potential of existing CIM systems is severely limited by inflexible and resource-intensive analog-to-digital converters (ADCs). Here, we present a memristor-based ADC featuring adaptive quantization for diverse CIM output distributions. Our design employs analog content-addressable memory cells with programmable overlapped boundaries to establish optimized quantization thresholds, demonstrating excellent integral and differential non-linearity of 0.319 and 0.419 least significant bit, respectively. Extensive experiments validate the robustness of our approach by achieving 89.55% accuracy on CIFAR-10 (VGG8) at 5-bit adaptive quantized precision and maintaining competitive performance on ImageNet (ResNet18) through a proposed super-resolution strategy under memristor variations observed in our experiment. Compared to state-of-the-art ADCs, our design achieves a 15.1× improvement in energy efficiency and a 17.3× reduction in area. Furthermore, the integration of our ADC into CIM systems reduces the energy and area overhead by 57.2% and 34.4% respectively for the CIFAR-10 (VGG8) task, and by 56.9% and 27.8% for the ImageNet (ResNet18) task. This work could establish a new paradigm for efficient and accurate signal quantization in practical CIM systems.

Introduction

As artificial intelligence (AI) advances rapidly, the increasing complexity of deep neural networks exposes critical limitations in traditional von Neumann architectures [1,2]. These architectures, with their separate computation and memory units[3], suffer from massive data movement overheads in modern neural network operations[4,5], resulting in high energy consumption and latency[6–8]. Compute-in-memory (CIM) with emerging memory devices, particularly memristors, offers a promising solution by integrating computation directly within memory crossbar arrays[9–19]. This integration overcomes the von Neumann bottleneck and enables efficient vector-matrix multiplications (VMMs) for deep learning operations[15,20–23]. A typical mixed-signal CIM system (**Fig. 1B**) consists of digital-to-analog converters (DACs), analog crossbar arrays, analog-to-digital converters (ADCs), and digital activation circuit implementing linear operations, quantization, and nonlinear activation functions in neural network layers, respectively.

Despite these advantages, analog-to-digital signal conversion emerges as a critical bottleneck in CIM systems[10,24–27]. Current ADCs, as fundamental components for signal conversion, consume excessive energy and circuit area, severely limiting system scalability. For example, in a state-of-the-art implementation[28], ADCs occupy up to 87.8% of total energy and 75.2% of chip area (**Fig. 1C**). Moreover, existing quantization methods struggle to handle diverse output distributions in CIM architectures[29,30]. In CIM implementations, different neural network layers and even channels are mapped to separate crossbar arrays, each carrying distinct weight patterns. This mapping, combined with varying input patterns during inference, leads to significantly different output distributions across arrays (**Fig. 1A**), demanding quantization schemes that are both adaptive and efficient. Current solutions present a clear trade-off: at low ADC precision (≤6 bits), uniform quantization offers hardware-friendly implementation but suffers from significant precision loss[2], while adaptive approaches based on cumulative distribution function[31] and Lloyd-Max algorithm[32–34] achieve higher accuracy through dynamic boundary adjustment but lack practical hardware implementations. At higher ADC precision, uniform quantization provides sufficient accuracy, making adaptive quantization less advantageous. However, the significant energy and area overhead of high-precision ADCs substantially diminishes the efficiency benefits of CIM systems, making low-bit implementations crucial for maintaining CIM's advantages in hardware efficiency.

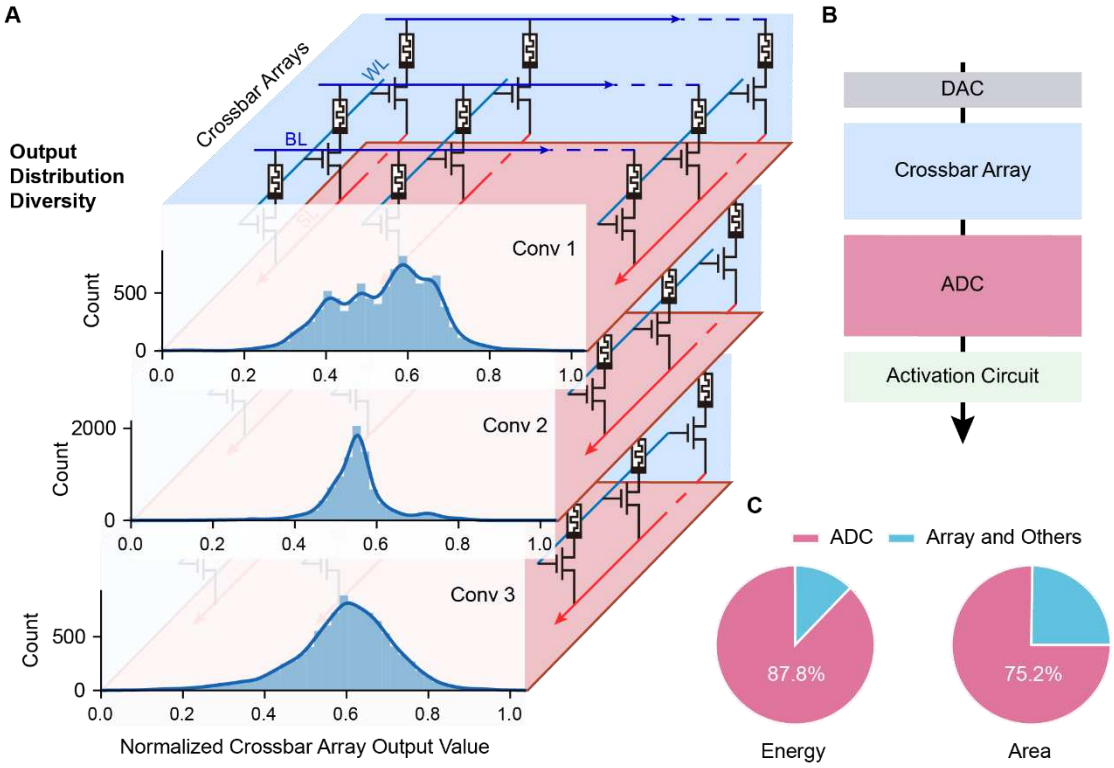


Fig. 1. Challenges in CIM systems for neural network computation. (A) Output distributions from crossbar arrays for three convolutional layers in VGG8 show significant diversity across layers. This diversity presents challenges for precision-constrained uniform quantization. **(B)** Schematic of a typical mixed-signal CIM system, showing the key components involved in neural network computation. The crossbar array executes vector-matrix multiplication, followed by ADC converting analog outputs into digital signals for subsequent digital nonlinear activation functions. **(C)** Resource breakdown in a state-of-the-art CIM implementation illustrates the substantial portion of energy and area consumed by ADCs.

In this work, we introduce a memristor-based ADC using analog content-addressable memory (CAM) cells for adaptive quantization in CIM systems, significantly reducing quantization errors compared to uniform approaches. By establishing quantization thresholds with overlapped CAM boundaries, our ADC demonstrates excellent quantization performance with integral non-linearity (INL) and differential non-linearity (DNL) of 0.319 and 0.419 least significant bits (LSB) for 5-bit configuration. In neural network applications, our ADC exhibits robust performance across different architectures under practical memristor variations. The 5-bit ADC achieves 89.55% accuracy on CIFAR-10 using VGG8. For more complex networks like ResNet18 on

ImageNet, our super-resolution strategy effectively mitigates hardware variations, achieving near-ideal accuracies of 34.76% (4-bit), 61.81% (5-bit), and 65.50% (6-bit), respectively. Moreover, the ADC exhibits superior hardware efficiency, delivering a 15.1× lower energy consumption while reducing the chip area by 17.3× compared to state-of-the-art solutions. Furthermore, compared to memristor ramp-based ADC implementation[35], our one-step conversion ADC achieves 48.5× lower latency and 26.6× lower energy consumption while maintaining comparable area efficiency. When integrated into CIM systems and compared to conventional successive approximation register (SAR) ADC, our design significantly reduces ADC overhead, decreasing energy and area contribution by 57.2% and 34.4% respectively for VGG8, and by 56.9% and 27.8% for ResNet18.

Results

Design of the memristor-based ADC

Our ADC design leverages analog CAM[36–38] as the core to create a programmable voltage quantizer, termed a quantization cell (Q-cell) (**Fig. 2A**). The Q-cell consists of two memristors (M_1, M_2), five transistors (T_1 to T_5), and two inverters (INV_1, INV_2). V_H and V_L are applied across voltage dividers formed by M_1-T_1 and M_2-T_3 pairs, while restricted to avoid mis-programming the device conductance. By tuning the memristor conductance, quantization boundaries can be defined through these dividers. Specifically, the M_1-T_1 divider controls the lower boundary via discharge transistor T_2, while the M_2-T_3 divider, coupled with INV_1, determines the upper boundary through discharge transistor T_4. This dual-boundary configuration enables voltage quantization based on programmed thresholds, as verified through circuit simulations using the UMC 180 nm technology library. The voltage responses at nodes A and B were obtained with a sampling time of 2 μs, with V_H and V_L set to 0.8 V and 0.3 V, respectively, across the M_1-T_1 and M_2-T_3 pairs. The input voltage was swept from 0.75 V to 1.30 V to characterize the Q-cell's response at nodes A and B (**Fig. 2B**). A PMOS transistor (T_5) initializes the circuit by charging the match line. The match line voltage, after passing through INV_2, generates a binary output: '0' when the input voltage falls within the programmed boundaries, and '1' otherwise.

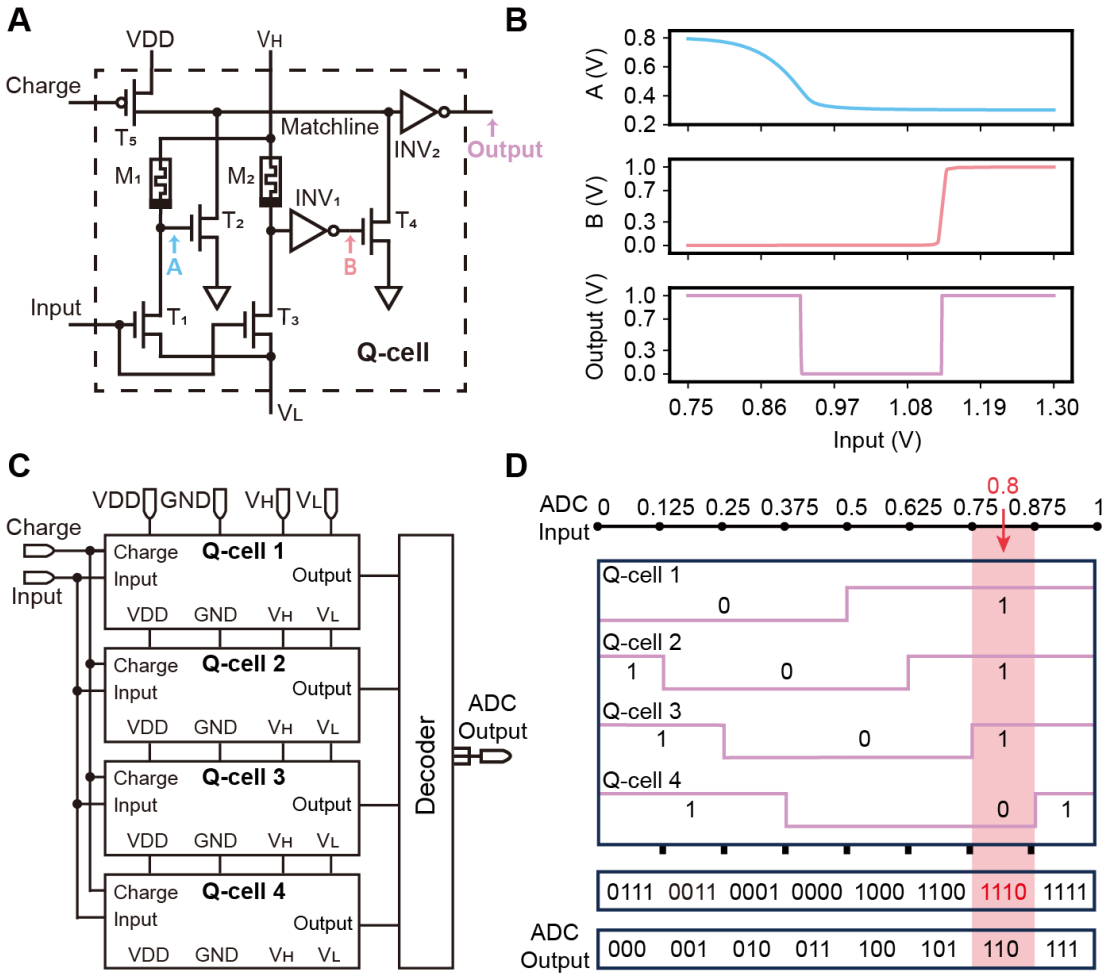


Fig. 2. Memristor-based ADC design. (A) Schematic of a memristor-based Q-cell. **(B)** Responses to the input voltage, including voltages at nodes A and B, and the output of the Q-cell. **(C)** Schematic of the proposed ADC at 3-bit precision, showing the arrangement of four Q-cells and decoder to achieve 3-bit precision quantization. **(D)** Illustration of the proposed ADC at 3-bit precision. When input 0.8V is applied, the cell output is 1110, and the ADC output is decoded to 110.

To illustrate the fundamental operation of our ADC architecture, we first present its implementation with uniform quantization, which allows us to validate the basic functionality and reliability of our design through well-established ADC metrics. This uniform configuration serves as a foundation for understanding the device's capabilities before exploring its adaptive quantization features.

The ADC precision determines the required number of Q-cells, with an n-bit ADC requiring 2^{n-1} Q-cells to cover the input range (see **Supplementary Note S1** for details). **Figs. 2C-D** shows a 3-bit ADC implementation using four Q-cells, where seven quantization boundaries divide the normalized input range into eight uniformly

distributed segments. Each Q-cell employs a progressive boundary distribution pattern based on LSB, with unique lower and upper boundary pairs. To maximize conductance range utilization, we implement a dual-bias strategy: the left input (T_1) receives a bias voltage matching the minimum input range of the ADC, while the right input receives a smaller bias (half of the input range). This approach ensures balanced conductance utilization across both sides, enabling the same conductance range to define both lower and upper quantization boundaries. In our 3-bit implementation, the Q-cells are programmed with voltage ranges of (0,0.5), (0.125,0.625), (0.25,0.75), and (0.375,0.875). This overlapping configuration generates unique digital codes through distinct Q-cell activation patterns. For example, an input of 0.8V activates only the last Q-cell (falling within 0.75-0.875), producing an output code of '1110', which is then decoded to the final binary output '110' (see **Supplementary Note S2** for details).

Although the illustrative example shows uniform quantization, the quantization boundary in our design is reconfigurable to achieve adaptive quantization. The key advantage of our design lies in its hardware efficiency and adaptability, where memristor programmability enables dynamic boundary reconfiguration to accommodate varying system requirements[39].

Experimental benchmarking of uniform quantization ADC

The experimental characterization of our memristor-based ADC architecture necessitates a rigorous evaluation framework that encompasses both conventional ADC performance metrics and adaptive quantization capabilities. Given that the fundamental operation of our design relies on precise analog-to-digital conversion, we first metrics, particularly integral non-linearity established its baseline performance through industry-standard static characterization (INL) and differential non-linearity (DNL) under uniform quantization conditions. These metrics provide quantitative validation of the device's intrinsic conversion accuracy and linearity, which underpin both conventional and adaptive quantization functionalities.

To experimentally demonstrate our ADC design, we first fabricated memristors with the structure shown in the microscopy image (**Fig. 3A**). The fabricated devices exhibit stable conductance states that can be effectively controlled through programming compliance current (**Fig. 3B**). This multi-level conductance programmability serves dual purposes: it enables the implementation of uniform quantization boundaries for

standardized ADC characterization while establishing the foundation for adaptive boundary modulation in neural network applications.

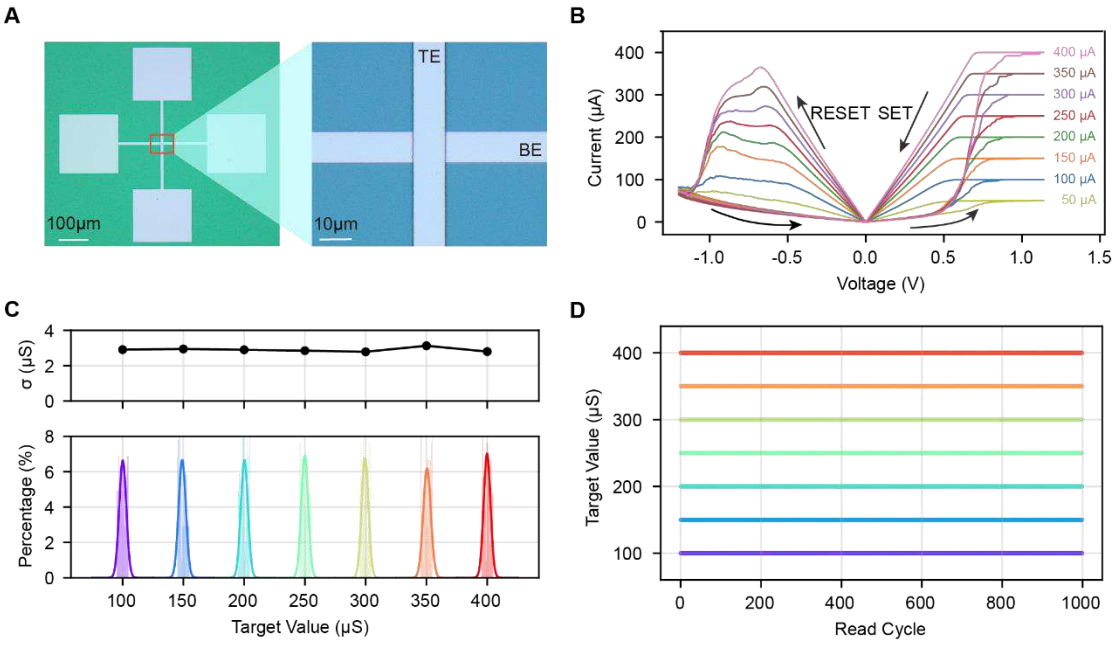


Fig. 3. Device characterization. (A) The microscopy images show the fabricated memristor device. **(B)** SET and RESET curves with multiple memristor conductance levels realized by controlling the compliance current. **(C)** Program cycle-to-cycle variation measurement showing conductance variation (SD: 2.90 μS, <1%) within 100-400 μS range. **(D)** Read cycle-to-cycle variation characterization showing stable conductance (SD: 0.14 μS, 0.047%) over 1000 cycles.

Our memristor devices demonstrated stable performance characteristics: program variation measurements showed 2.90 μS conductance variation (< 1%) within 100-400 μS range (**Fig. 3C**), while read stability tests revealed 0.14 μS (0.047%) variation over 1000 consecutive cycles (**Fig. 3D**) (see **Methods** for details).

Further analysis of device variability under programming conditions[40] extended our static characterization to encompass both INL and DNL[41] across the full operating range (see **Methods** for details). INL quantifies deviation from ideal linear response, while DNL measures uniformity between successive codes. In CIM systems, INL affects feature map computations, where high values lead to reduced inference accuracy, while DNL determines precision in distinguishing activation levels. The impact of these variations is amplified in CIM architectures, as ADC quantization across array lines can

cascade through network layers, causing significant accuracy loss in deep learning tasks.

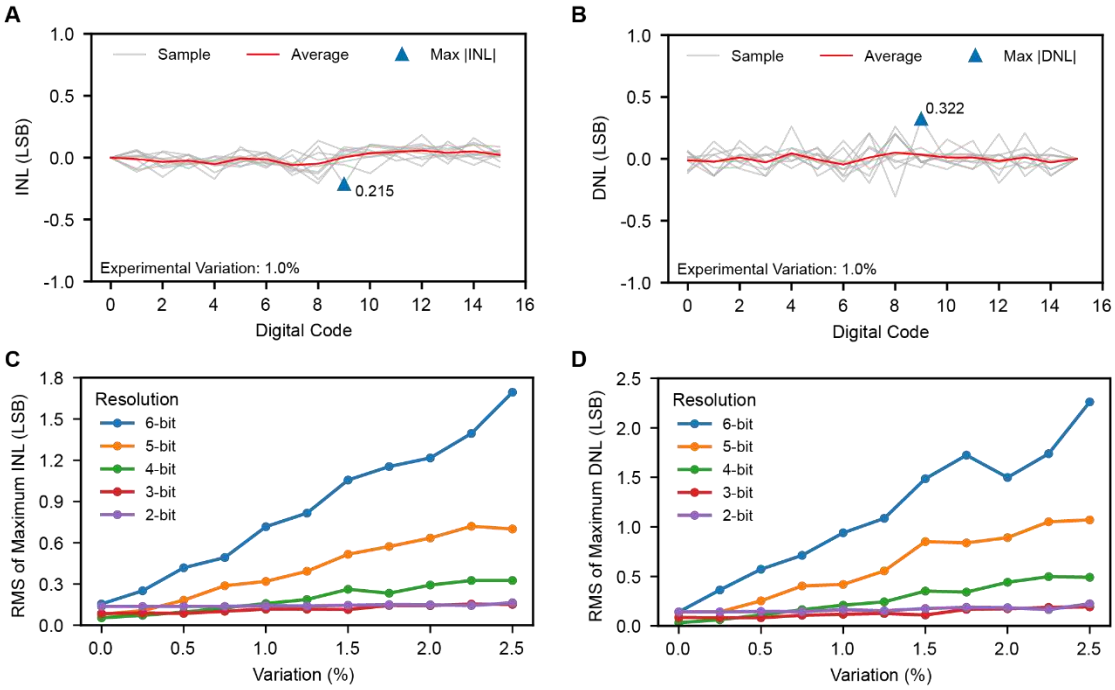


Fig. 4. Performance metrics of the memristor-based ADC. **(A)** INL of 4-bit ADC under 1% variation across 16 digital codes, maximum 0.215 LSB. **(B)** DNL under the same conditions, maximum 0.322 LSB. **(C)** RMS of maximum INL vs. device variation for 2-6 bit precisions. **(D)** RMS of maximum DNL vs. device variation for 2-6 bit precisions.

For our 4-bit ADC implementation, maximum INL and DNL were limited to 0.215 LSB and 0.322 LSB, respectively (**Figs. 4A-B**), validating design robustness under experimental variation. Scalability analysis across 2-6 bit resolutions (**Figs. 4C-D**) using root mean square (RMS) analysis of maximum INL and DNL revealed consistent performance (see **Methods** for details). As detailed in **Table S2**, all bit precisions maintained sub-1-LSB linearity metrics under experimental variation. Specifically, at 5-bit precision, our architecture achieved RMS values of 0.319 LSB and 0.429 LSB for maximum INL and DNL, respectively. These measurements, conducted under uniform quantization, establish the fundamental precision and stability of our ADC architecture, providing critical attributes that enable subsequent implementation of adaptive quantization schemes across diverse CIM applications.

Adaptive quantization ADC for efficient neural networks

Although our ADC design has demonstrated superior benchmark performance, its full potential has yet to be exploited. What makes it special is that the quantization boundaries can be programmed arbitrarily using the memristors in the Q-cell, opening up the possibility for adaptive quantization in neural networks and achieving non-degraded performance even with low-bit quantization. Our adaptive quantization implementation integrates hardware-based boundary adjustment with software optimization. This process encompasses both offline analysis and dynamic operation during inference. Initially, we collect output statistics from each convolution layer by running inference on a representative training dataset, capturing the distinct characteristics of various network layers. Utilizing these statistics, we apply a modified Lloyd-Max algorithm[32] to determine optimal quantization boundaries (**Fig. S1**), iteratively adjusting them to minimize mean squared error (MSE) while adhering to hardware constraints. Subsequently, we program these optimized boundaries into the memristor devices through precise conductance modulation, employing either compliance current control or programming pulses with program-and-verify calibration. This approach accounts for the non-linear relationship between conductance and voltage thresholds in our Q-cell design. During inference, each Q-cell operates with these optimized boundaries, enabling adaptive quantization that closely aligns with the actual distribution of CIM array outputs.

This software-hardware co-optimization methodology ensures the effective realization of adaptive quantization's theoretical benefits in physical implementation. The hardware component exploits memristors' programmable conductance states for dynamic quantization boundary setting, while the software optimization process fine-tunes these boundaries based on actual data distributions and hardware constraints, resulting in a highly efficient and adaptable CIM system.

To validate our approach, we evaluated our strategy using the VGG8 network[42] on the CIFAR-10 dataset[43]. The experimental results demonstrated the effectiveness of our adaptive quantization method. Specifically, the reconfigurable ADC achieved superior boundary alignment with data distribution compared to uniform quantization (**Fig. 5A**), reducing MSE from 14.99 to 3.10 (**Fig. 5B**).

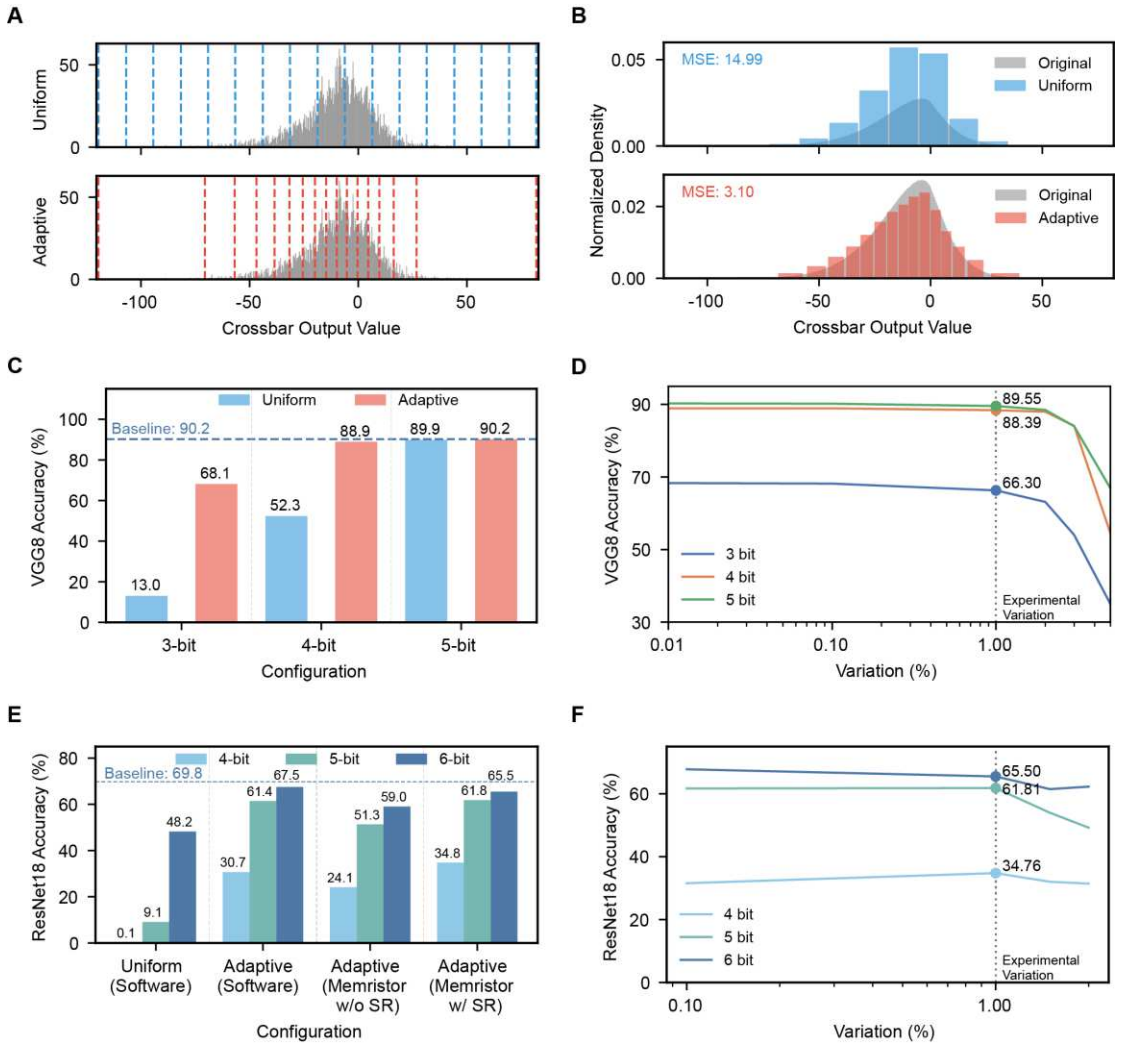


Fig. 5. Analysis of adaptive quantization and network performance with our ADC. (A) Distribution of crossbar array output values showing adaptive *vs.* uniform quantization boundaries. **(B)** MSE comparison between adaptive (3.10) and uniform (14.99) quantization on the full dataset. **(C)** VGG8 performance comparison: uniform *vs.* adaptive quantization at 3-5 bit precisions. **(D)** VGG8 accuracy across ADC precisions. **(E)** ResNet18 accuracy comparison: uniform quantization (software), adaptive quantization (software), our ADC without and with super-resolution (SR) strategy (under 1% experimental memristor variation) at 4-6 bit precisions. **(F)** ResNet18 accuracy with super-resolution strategy at 4-6 bit precisions.

Across different ADC precisions (**Fig. 5C**), our adaptive approach achieved 68.1%, 88.9%, and 90.2% accuracy at 3-bit, 4-bit, and 5-bit, respectively, significantly outperforming uniform quantization (13.0%, 52.3%, 89.9%). Under experimental variations (**Fig. 5D**), the network maintained robust performance with 66.30%, 88.39%, and 89.55% accuracy at respective bit levels.

For the more complex ResNet18 network[44] on ImageNet[45], we conducted a comprehensive evaluation across different quantization scenarios. First, we established baseline performance using ideal adaptive quantization, achieving 30.7%, 61.4%, and 67.5% accuracy at 4-bit, 5-bit, and 6-bit precisions, respectively. These results exhibit the theoretical benefits of adaptive quantization at different bit precisions (**Fig. 5E**). However, when implementing the ADC with real memristor devices under experimental variations (1% conductance variation), the accuracies dropped to 24.1%, 51.3%, and 59.0%. This accuracy degradation (average 6.6% drop) reveals the impact of device variations on quantization precision.

To address this challenge, we developed a super-resolution strategy that leverages the inherent variation between paired ADCs. When two ADCs with nominally identical boundaries produce different outputs due to device variations, it suggests that the input lies within the boundary region affected by these variations. We identify such boundary-proximate states, assign appropriate boundary values, and thereby utilize device variations for super-resolution (see **Methods** for implementation details). This approach effectively improved accuracies to 34.8%, 61.8%, and 65.5% for 4-bit, 5-bit, and 6-bit configurations respectively (**Fig. 5F**). Notably, these results not only recover the accuracy loss from device variations but, in some cases (e.g., 4-bit: +4.06% improvement), surpass the theoretical adaptive quantization baseline. This improvement shows how our super-resolution strategy transforms device variations from a limitation into an advantage, thereby enhancing quantization precision.

Hardware efficiency evaluation of memristor-based ADC

To evaluate the hardware benefits of our approach, we conducted a comprehensive analysis comparing our ADC design with state-of-the-art SAR, Flash, and other ADCs[46]. Given the emerging nature of low-bit ADCs at advanced technology nodes, direct comparisons at 16 nm node with 5-6 bit precision are challenging due to the limited number of published works in this specific domain. To provide a comprehensive evaluation of our design's capabilities, we extended our comparison across different technology nodes and precision levels, enabling a broader assessment of the advantages offered by our memristor-based approach.

As illustrated in **Fig. 6A**, our memristor-based ADC demonstrates remarkable

improvements in both energy efficiency and area utilization at 5-bit ADC precision across this comprehensive comparison space. A detailed breakdown of both energy consumption and area utilization (**Fig. 6B**) reveals the relative contributions of the ADC core components and the decoder, demonstrating the efficient resource allocation in our design. The design achieves an energy efficiency of 12.58 fJ per conversion (see **Methods** for detail), representing a 15.1× improvement over conventional designs[47]. In terms of area, our ADC occupies only 18.07 μm², which is 17.3× smaller than the most compact conventional ADCs[48].

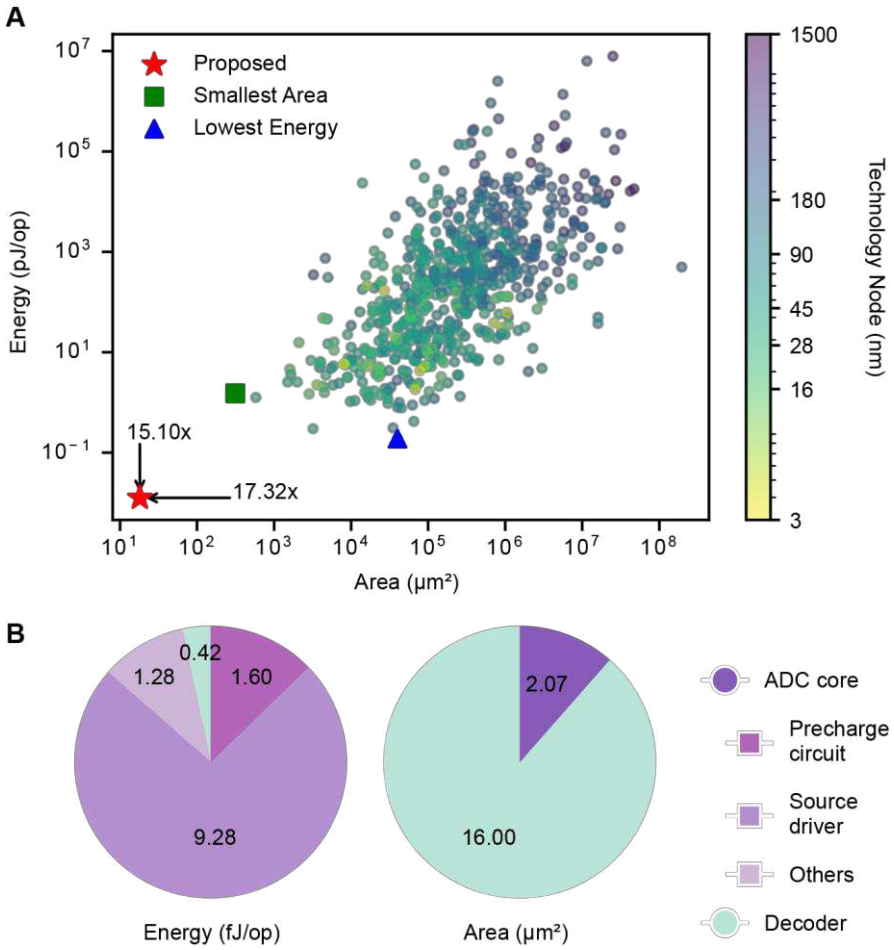


Fig. 6. Energy and area comparison of memristor-based ADC. (A) Scatter plot comparing energy efficiency versus area of the state-of-the-art ADCs[46], with technology nodes indicated by the color bar. Our memristor-based ADC (red star) demonstrates superior performance in both metrics, compared to conventional designs, including lowest-energy design[47] (blue triangle) and smallest-area design[48] (green square). **(B)** Breakdown of energy consumption and area utilization of our ADC implemented at 5-bit precision, showing the proportion between ADC core (precharge circuit, source driver, and other circuits) and decoder.

Compared to existing memristor ramp-based ADC implementations[35], our one-step conversion mechanism demonstrates advantages in both speed and energy efficiency (**Table. S3**). Ramp-based ADCs achieve complete parallelism by allocating a dedicated comparator to each column, enabling simultaneous quantization of all column outputs. This architecture is exceptionally well-suited for CIM systems that require high readout bandwidth. Consequently, these systems become optimal for data-intensive and high-throughput applications where the simultaneous processing of large information volumes is critical. In scenarios requiring lower latency, the architecture may be less optimal due to its operational principle, which requires sequential comparisons as the ramp signal progresses through each quantization. The prolonged comparator activation during this comparison period leads to elevated energy dissipation. For example, this principle results in a reported latency of 32 ns and energy consumption of 42.82 pJ at 5-bit precision. In contrast, our design achieves quantization in a single step, with all thresholds evaluated simultaneously. Our implementation demonstrates a latency of 0.66 ns and energy consumption of 1.61 pJ per VMM computation in a configuration of 32 ADCs per macro, while maintaining area efficiency (578.24 μm^2). These improvements represent a 48.5× reduction in conversion latency and a 26.6× reduction in energy consumption.

The benefits of our design extend beyond component-level comparisons to system-level improvements. Integration into CIM systems exhibits a significant reduction in analog-to-digital conversion overhead. Compared to energy- and area-efficient SAR ADC[49] in CIM systems, our design achieves substantial improvements: for VGG8 implementations, the energy contribution of ADC components decreases from 79.8% to 22.5% of total system energy (a 57.2% reduction), while the area overhead drops from 47.6% to 13.2% of total chip area (a 34.4% reduction). Similar improvements are observed in ResNet18 implementations, where ADC energy overhead is reduced from 71.5% to 14.6% (a 56.9% reduction), and area contribution decreases from 36.6% to 8.8% (a 27.8% reduction), as illustrated in **Figs. S2-S3 and Tables S4-S5**.

These comprehensive improvements across different metrics and comparison scenarios show the superiority of our architecture, enabling substantially higher system operating frequencies and better energy efficiency for CIM applications. The dramatic reductions in both energy and area overhead effectively address the conversion bottleneck in CIM systems, highlighting the transformative potential of our memristor-

based ADC for efficient and accurate neural network inference acceleration.

Discussion

This work presents a memristor-based ADC architecture for neural network acceleration. Our ADC employs analog content-addressable memory cells with overlapped boundaries to establish optimized quantization thresholds, implements a super-resolution strategy to maintain competitive accuracy under practical memristor variations, and achieves a 15.1× improvement in energy efficiency and a 17.3× reduction in area compared to state-of-the-art designs. When integrated into CIM systems, it dramatically reduces ADC overhead, with energy and area reductions of 57.2% and 34.4% for VGG8, and 56.9% and 27.8% for ResNet18, respectively. The ADC supports adaptive quantization while maintaining robust performance, making it particularly suitable for edge computing applications. Beyond immediate applications in inference acceleration, this design opens new possibilities for adaptive hardware-algorithm co-design in next-generation AI systems, bridging the gap between algorithmic requirements and hardware constraints.

Methods

Device characterization

The electrical characterization of the memristor device is conducted using a semiconductor parameter analyzer (Keysight B1500A) with the chip mounted on a probe station at room temperature. Conductance programming is performed through quasi-static DC sweeps, employing a customized program-and-verify scheme.

For program cycle-to-cycle variation measurement (**Fig. 3C**), we employed a program-and-verify scheme. During each program-and-verify cycle, the compliance current is incrementally increased based on the difference between the measured conductance and the target value. This adjustment continues as long as the measured conductance remains below the target value. The step size for the compliance current is determined by the difference between the measured conductance and the target value (in μS), multiplied by an empirical scaling factor. A scaling factor of 0.5 is employed

for the initial 10 iterations, after which it is reduced to 0.25 to enhance success rates. The process terminates after a maximum of 20 iterations, regardless of whether the target conductance has been achieved. If the measured conductance exceeds the target value beyond the tolerance range (±5 μS), a reset sweep (starting from -1.2V) is applied, and the programming process is restarted using the newly calculated compliance current value.

Target conductance values are selected from 100 μS to 400 μS in 50 μS steps, with each target programmed 100 times. After each attempt, the device is reset before the next program-and-verify cycle. Experimental results show that 91.6% of attempts converge to the target within 10 iterations, with an average of 5.57 iterations required.

For read cycle-to-cycle variation characterization (**Fig. 3D**), the device conductance is first programmed to the target values using the same program-and-verify method. Once achieved, the conductance is read 1000 times for each target value. All measurements are performed on the same device to ensure consistency.

Adaptive ADC design

The reconfigurable ADC design was implemented using Spice and Cadence Virtuoso tools. The quantization cell with memristor devices was modeled and simulated. The performance metrics of the memristor-based ADC were evaluated based on the simulation results, including INL, DNL, RMS of maximum INL, and RMS of maximum DNL.

The INL, which represents the maximum deviation of the actual conversion characteristic from the ideal straight line, was calculated using:

$$INL_k = \frac{V(k) - V_{best-fit}(k)}{LSB} \quad (1)$$

where $V(k)$ is the actual voltage corresponding to output code k, $V_{best-fit}(k)$ is the voltage on the best-fit line. The best-fit line is determined through linear regression:

$$V_{best-fit}(k) = ak + b \quad (2)$$

where a and b are coefficients calculated using the least squares method.

The DNL, which measures the difference between the actual and ideal code step

widths, was determined using:

$$DNL_k = \frac{V(k+1) - V(k)}{LSB} - 1 \tag{3}$$

To assess the overall nonlinearity errors, we calculated the RMS INL and RMS DNL:

$$INL_{RMS} = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} INL_k^2} \tag{4}$$

$$DNL_{RMS} = \sqrt{\frac{1}{N-1}\sum_{k=0}^{N-2} DNL_k^2} \tag{5}$$

where N is the total number of ADC codes.

The design was optimized by precisely controlling the programming compliance current or pulse to adjust memristor conductance. To measure INL and DNL, a range of input voltages was applied, and the ADC's output was compared to the ideal linear response. The impact of device variation was analyzed by introducing controlled conductance variations in the memristors and observing the resulting changes in INL and DNL.

Neural network evaluation

We evaluated our ADC strategy using two neural network architectures: VGG8 for CIFAR-10 classification and ResNet18 for ImageNet classification. The networks were implemented using a modified NeuroSim framework[49], which we enhanced with detailed ADC behavioral models to accurately capture the impact of quantization and device variations. Our simulation framework incorporates precise modeling of ADC characteristics, providing realistic performance estimates for practical CIM implementations. The networks were implemented with different ADC configurations and quantization strategies.

Super resolution strategy

To address device variations in practical implementations, we developed the super-resolution strategy, which utilizes two ADCs with nominally identical quantization boundaries. Under memristor variations, these boundaries exhibit slight misalignments. When an input voltage falls near a quantization boundary, the two ADCs may produce different quantization results. We leverage this disagreement to identify and precisely quantize these boundary-proximate values. Specifically, when the two ADCs produce different outputs for the same input, we assign the input to the corresponding boundary value, effectively achieving super-resolution at quantization boundaries. This approach not only improves quantization accuracy but also takes advantage of inherent device variations to enhance precision.

Implementation details for 5-bit ADC design

Our design consists of the ADC core (precharge circuit, source driver, and other circuits) and a decoder circuit, with their respective contributions to energy and area shown in **Fig. 6B**. For the ADC core, which comprises 16 Q-cells for a 5-bit ADC, is based on a previously demonstrated analog CAM in TSMC 16 nm technology[36]. To estimate the baseline energy consumption, we started with this analog CAM design, excluding the DAC portion, which yielded 0.47 fJ per operation with a latency of 45 ps. To enhance the linearity of our ADC, we expanded the memristor conductance range from the original 200 μS to a wider range of 100-400 μS, primarily impacting the energy consumption of the source driver within each Q-cell. The original source driver consumed 0.29 fJ per operation, but with the expanded conductance range, this value doubled to 0.58 fJ. Consequently, after this adjustment, the total energy consumption for the ADC core increased to 12.58 fJ. The physical footprint was derived from a 2×2 CAM array occupying 0.72 μm × 0.72 μm in 16 nm technology. The decoder circuit was synthesized using Synopsys Design Compiler with TSMC 28 nm technology library, achieving a latency of 0.12 ns. The total conversion latency of 0.165 ns per ADC serves as the basis for our system-level timing analysis in CIM implementations. While this mixed-technology implementation represents a practical engineering compromise, the performance impact of using 28 nm for the decoder is conservative - an eventual full 16 nm implementation would likely achieve even better energy efficiency and a smaller area. By extrapolating these values based on the number of

quantization cells and operational frequency, we calculated the total energy and area metrics for various ADC configurations.

CIM system evaluation

To comprehensively evaluate the system-level benefits of our proposed memristor-based ADC design, we conducted a detailed performance analysis using a modified NeuroSim framework[49]. The CIM architecture was implemented with memristor crossbar arrays, each configured as a 128×128 matrix. To optimize the balance between performance and hardware overhead, we adopted a resource-sharing strategy where every four columns share one ADC, resulting in 32 ADCs per array. All ADCs were configured with 5-bit precision (32 quantization levels) to maintain consistent comparison across different network architectures.

The complete system architecture incorporates several key components, including the core crossbar array, ADC, adder, and other peripheral circuits. The adder units include adder trees and shift register pairs for accumulating partial products. Other circuits comprise essential peripheral components such as word line decoders and their corresponding drivers, multiplexer circuits with associated decoders, and word line switch matrices. The entire system, including the conventional SAR ADC model, was evaluated under a 14 nm technology node to ensure fair comparison with state-of-the-art implementations. The detailed energy consumption and area utilization of each component were analyzed for both VGG8 and ResNet18 implementations, as presented in **Figs. S2-S3**, respectively. The comprehensive breakdown of system-level metrics is provided in **Tables S2-S3**, demonstrating the significant reduction in ADC overhead achieved by our design across different network architectures.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Code availability

The codes that support the findings of this study are available from the corresponding

author upon reasonable request.

Reference

1. Li, C. *et al.* Analogue signal and image processing with large memristor crossbars. *Nat Electron* **1**, 52–59 (2018).
2. Aguirre, F. *et al.* Hardware implementation of memristor-based artificial neural networks. *Nat Commun* **15**, 1974 (2024).
3. Von Neumann, J. First draft of a report on the EDVAC. *IEEE Annals Hist. Comput.* **15**, 27–75 (1993).
4. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
5. Bassey, J., Qian, L. & Li, X. A Survey of Complex-Valued Neural Networks. Preprint at http://arxiv.org/abs/2101.12249 (2021).
6. Chen, Y.-H., Emer, J. & Sze, V. Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks. in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* 367–379 (2016). doi:10.1109/ISCA.2016.40.
7. Chen, Y.-H., Yang, T.-J., Emer, J. & Sze, V. Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **9**, 292–308 (2019).
8. Ganguly, A., Muralidhar, R. & Singh, V. Towards Energy Efficient non-von Neumann Architectures for Deep Learning. in *20th International Symposium on Quality Electronic Design (ISQED)* 335–342 (2019). doi:10.1109/ISQED.2019.8697354.
9. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
10. Shafiee, A. *et al.* ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* 14–26 (IEEE, Seoul, South Korea, 2016). doi:10.1109/ISCA.2016.12.
11. Ielmini, D. & Wong, H.-S. P. In-memory computing with resistive switching devices. *Nat Electron* **1**, 333–343 (2018).
12. Ni, K. *et al.* In-Memory Computing Primitive for Sensor Data Fusion in 28 nm HKMG FeFET Technology. in *2018 IEEE International Electron Devices Meeting (IEDM)* 16.1.1-16.1.4 (IEEE, San Francisco, CA, 2018). doi:10.1109/IEDM.2018.8614527.

13. Xue, C.-X. *et al.* 24.1 A 1Mb Multibit ReRAM Computing-In-Memory Macro with 14.6ns Parallel MAC Computing Time for CNN Based AI Edge Processors. in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)* 388–390 (IEEE, San Francisco, CA, USA, 2019). doi:10.1109/ISSCC.2019.8662395.
14. Deng, B. L., Li, G., Han, S., Shi, L. & Xie, Y. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proc. IEEE* **108**, 485–532 (2020).
15. Yu, S., Jiang, H., Huang, S., Peng, X. & Lu, A. Compute-in-Memory Chips for Deep Learning: Recent Trends and Prospects. *IEEE Circuits Syst. Mag.* **21**, 31–56 (2021).
16. Mao, R. *et al.* Experimentally validated memristive memory augmented neural network with efficient hashing and similarity search. *Nat Commun* **13**, 6284 (2022).
17. Le Gallo, M. *et al.* A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference. *Nat Electron* 1–14 (2023) doi:10.1038/s41928-023-01010-1.
18. Jiang, M., Shan, K., He, C. & Li, C. Efficient combinatorial optimization by quantum-inspired parallel annealing in analogue memristor crossbar. *Nat Commun* **14**, 5927 (2023).
19. Wang, Z. *et al.* A dual-domain compute-in-memory system for general neural network inference. *Nat Electron* 1–12 (2025) doi:10.1038/s41928-024-01315-9.
20. Yu, S. Neuro-Inspired Computing With Emerging Nonvolatile Memory. *Proc. IEEE* **106**, 260–285 (2018).
21. Hu, M. *et al.* Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. in *Proceedings of the 53rd Annual Design Automation Conference* 1–6 (ACM, Austin Texas, 2016). doi:10.1145/2897937.2898010.
22. Gallo, M. L. *et al.* Mixed-Precision In-Memory Computing. *Nat Electron* **1**, 246–253 (2018).
23. Wang, Z. *et al.* In situ training of feed-forward and recurrent convolutional memristor networks. *Nat Mach Intell* **1**, 434–442 (2019).
24. Chi, P. *et al.* Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory. *ACM SIGARCH Computer Architecture News* **44**, 27–39 (2016).
25. Zhu, Z. *et al.* A Configurable Multi-Precision CNN Computing Framework Based on Single Bit RRAM. in *Proceedings of the 56th Annual Design Automation Conference 2019* 1–6 (ACM, Las Vegas NV USA, 2019). doi:10.1145/3316781.3317739.
26. Jiang, H. *et al.* Analog-to-Digital Converter Design Exploration for Compute-in-Memory

538 Accelerators. *IEEE Des. Test* **39**, 48–55 (2022).
539 27. Wan, W. *et al.* A compute-in-memory chip based on resistive random-access memory. *Nature* **608**,
540 504–512 (2022).
541 28. Yao, P. *et al.* Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–
542 646 (2020).
543 29. He, Y. *et al.* InfoX: an energy-efficient ReRAM accelerator design with information-lossless low-bit
544 ADCs. in *Proceedings of the 59th ACM/IEEE Design Automation Conference* 97–102 (ACM, San
545 Francisco California, 2022). doi:10.1145/3489517.3530396.
546 30. Yue, J. *et al.* 15.2 A 2.75-to-75.9TOPS/W Computing-in-Memory NN Processor Supporting Set-
547 Associate Block-Wise Zero Skipping and Ping-Pong CIM with Simultaneous Computation and
548 Weight Updating. in *2021 IEEE International Solid- State Circuits Conference (ISSCC)* 238–240
549 (IEEE, San Francisco, CA, USA, 2021). doi:10.1109/ISSCC42613.2021.9365958.
550 31. Sun, H. *et al.* An Energy-Efficient Quantized and Regularized Training Framework For Processing-
551 In-Memory Accelerators. in *2020 25th Asia and South Pacific Design Automation Conference (ASP-*
552 *DAC)* 325–330 (IEEE, Beijing, China, 2020). doi:10.1109/ASP-DAC47756.2020.9045192.
553 32. Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 129–137
554 (1982).
555 33. Sun, X. *et al.* XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural
556 networks. in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)* 1423–1428
557 (IEEE, Dresden, Germany, 2018). doi:10.23919/DATE.2018.8342235.
558 34. Cai, Y. *et al.* Low Bit-Width Convolutional Neural Network on RRAM. *IEEE Trans. Comput.-Aided*
559 *Des. Integr. Circuits Syst.* **39**, 1414–1427 (2020).
560 35. Yang, J. *et al.* Efficient nonlinear function approximation in analog resistive crossbars for recurrent
561 neural networks. *Nat Commun* **16**, 1136 (2025).
562 36. Li, C. *et al.* Analog content-addressable memories with memristors. *Nat Commun* **11**, 1638 (2020).
563 37. Pedretti, G. *et al.* Tree-based machine learning performed in-memory with memristive analog CAM.
564 *Nat Commun* **12**, 5806 (2021).
565 38. Pedretti, G. *et al.* Differentiable Content Addressable Memory with Memristors. *Adv Elect Materials*
566 **8**, 2101198 (2022).
567 39. Huang, S., Jiang, H. & Yu, S. Hardware-aware Quantization/Mapping Strategies for Compute-in-

Memory Accelerators. *ACM Trans. Des. Autom. Electron. Syst.* **28**, 1–23 (2023).

40. Feinberg, B., Wang, S. & Ipek, E. Making Memristive Neural Network Accelerators Reliable. in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)* 52–65 (IEEE, Vienna, 2018). doi:10.1109/HPCA.2018.00015.
41. Jespers, P. G. *Integrated Converters: D to A and A to D Architectures, Analysis and Simulation*. (Oxford University Press, Oxford New York, 2001).
42. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Preprint at http://arxiv.org/abs/1409.1556 (2015).
43. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. *Technical report*.
44. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. Preprint at http://arxiv.org/abs/1512.03385 (2015).
45. Deng, J. *et al.* ImageNet: A Large-Scale Hierarchical Image Database. in *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (2009). doi:10.1109/CVPR.2009.5206848.
46. Boris, M. ADC Performance Survey 1997-2024. *GitHub* [Online]. Available: https://github.com/bmurmann/ADC-survey.
47. Bindra, H. S., Annema, A.-J., Louwsma, S. M. & Nauta, B. A 0.2 - 8 MS/s 10b flexible SAR ADC achieving 0.35 - 2.5 fJ/conv-step and using self-quenched dynamic bias comparator. in *2019 Symposium on VLSI Circuits* C74–C75 (2019). doi:10.23919/VLSIC.2019.8778093.
48. Yonar, A. S. *et al.* An 8b 1.0-to-1.25GS/s 0.7-to-0.8V Single-Stage Time-Based Gated-Ring-Oscillator ADC with 2\times Interpolating Sense-Amplifier-Latches. in *2023 IEEE International Solid-State Circuits Conference (ISSCC)* 1–3 (2023). doi:10.1109/ISSCC42615.2023.10067745.
49. Peng, X., Huang, S., Luo, Y., Sun, X. & Yu, S. DNN+NeuroSim: An End-to-End Benchmarking Framework for Compute-in-Memory Accelerators with Versatile Device Technologies. in *2019 IEEE International Electron Devices Meeting (IEDM)* 32.5.1-32.5.4 (IEEE, San Francisco, CA, USA, 2019). doi:10.1109/IEDM19573.2019.8993491.

Acknowledgments

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R, the National Natural Science Foundation of China (62404187), and the General Research Fund (GRF) Project 17203224 of the Research Grants Council (RGC), Hong Kong SAR.

Author contributions

Z.L., C.L., and N.W. supervised the project. H.H., R.M., C.L., and N.W. contributed to the conception of the idea. H.H. performed the experiments and analyzed data under the supervision of Z.L., C.L., and N.W. Z.D. fabricated the memristors. Z.D., M.J., and R.M. provided guidance on experimental procedures. Y.R., W.M., M.P., and W.Z. contributed valuable insights through technical discussions. H.H., Z.L., C.L., and N.W. wrote the manuscript with input from all authors.

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformation.docx