ADVANCED SCIENCE Open Access
www.advancedscience.com

RESEARCH ARTICLE OPEN ACCESS

In Situ Quantization with Memory-Transistor Transfer Unit Based on Electrochemical Random-Access Memory for Edge Applications

Zhen Yang[1] | Yuxiang Yang[1] | Baiqian Wang[2] | Yaoyu Tao[3] | Zelun Pan[1] | Lei Cai[1] | Teng Zhang[1] | Longhao Yan[1] | Xianbin Li[2] | Yuchao Yang[1,3,4,5]

[1]New Cornerstone Science Laboratory, Beijing Advanced Innovation Center for Integrated Circuits, School of Integrated Circuits, Peking University, Beijing, China | [2]State Key Laboratory of Integrated Optoelectronics, College of Electronic Science and Engineering, Jilin University, Changchun, China | [3]Center For Brain Inspired Chips, Institute For Artificial Intelligence, Frontiers Science, Center For Nano-optoelectronics, Peking University, Beijing, China | [4]New Cornerstone Science Laboratory, Guangdong Provincial Key Laboratory of In-Memory Computing Chips, School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China | [5]Center For Brain Inspired Intelligence, Chinese Institute for Brain Research (CIBR), Beijing, China

Correspondence: Xianbin Li (lixianbin@jlu.edu.cn) | Yuchao Yang (yuchaoyang@pku.edu.cn)

Received: 30 October 2025 | **Revised:** 31 December 2025 | **Accepted:** 13 January 2026

Keywords: ECRAM | edge computing | in-memory computing | weight quantization

ABSTRACT

In-memory computing based on nonvolatile synaptic arrays with computing functions has significantly improved the computing energy efficiency of neural networks. However, current synaptic devices are mostly limited to accelerating matrix-vector multiplication operators, and the differentiated requirements for device characteristics in the training/inference stage have led to a sharp increase in the integration complexity of hybrid synaptic units. Hence, for low-precision quantization calculations of networks, a compact synaptic unit based on ionic nonvolatile memory-transistor coupling integration, which enables in situ approximate weight quantization without additional binary programming while maintaining parallel MVM computing capabilities, is developed. Results show that the quantization function, derived from the cell's physical electrical properties, achieves classification accuracy in binary neural networks comparable to the ideal quantization function. This approach supports low-precision continual learning, mitigates catastrophic forgetting, and enables efficient computations for binary/ternary large language models. At a 4 Mb array scale, ECRAM- and RRAM-based units achieve energy consumption advantage of 25.51× and 4.84×, respectively, over traditional digital platforms, offering a robust in situ quantization framework for low-precision edge training.

1 | Introduction

Recently, with rapid progress in deep learning (DL), a range of complicated tasks have been incredibly resolved, such as image classification [1–3], natural language processing [4–6], speech recognition [7, 8], and autonomous decision-making [9–11]. By continually increasing the depth and size of the deep neural network (DNN) models, DL will even outperform the human's level in some cognitive tasks [3, 12]. However, exponentially growing network parameters impose unsustainable hardware resource

Zhen Yang and Yuxiang Yang contributed equally to this work.

and power demands, particularly for edge-based data-intensive applications [13–15]. In conventional computing systems, frequent data transfers between memory and processing units further degrade energy efficiency, posing a critical bottleneck for scaling future models [16, 17].

To address the trade-off between massive parameters and limited computing power, brain-inspired computing-in-memory (CIM) architectures leveraging emerging nonvolatile memory (eNVM) devices have been proposed to improve energy efficiency in DL applications [18]. By integrating these eNVM devices into the crossbar arrays, matrix-vector multiplication (MVM) based on electrical principles can be physically in situ conducted without frequent data movements, which has been broadly utilized to accelerate the neural network inferences by mapping the weights into device conductance [19–21]. However, modern high-performing DNN models often rely on high-precision parameters (e.g., 32-bit floating-point), demanding complex computations and large memory capacity [22]. For most eNVM devices, such as resistive random-access memory (RRAM) [23–26], phase-change memory (PCM) [27–29], magnetoresistive random-access memory (MRAM) [30, 31], and ferroelectric tunnel junction (FTJ) [32–34], the limited number of conductance states require adjusted weight quantization. Although some researchers have proposed more advanced programming method to significantly enhance storage state numbers in single devices [21, 35], low-bit weight quantization in hardware employment is still necessary. As demonstrated in Figure 1a, for the limited computational and memory resources in edge platforms, such as mobile phones and embedded devices, neural network quantization, in which original floating-point weights are represented with highly low precision while keeping performance degradation extremely low, has been utilized to compress the model size [36]. By quantizing the neural network weights into 1-bit precision, which is also termed as binarization, a recent study replaced the heavy MVM operations well with bitwise XNOR operations and bitcount operations, and realized up to 32× memory saving and 58× speedup on CPUs [37].

As shown in Figure 1b, the integration of CIM architecture with hardware-friendly binary neural networks (BNNs) utilizes two distinct arrays to store different precision weights: real-valued analog and quantized binary weights [36]. For edge applications in adapting to complex environments, frequent neural network finetuning is necessary, which requires lots of quantization operations during online training. This process demands simultaneous storage of both weight types due to quantization's irreversible nature. However, hardware quantization still involves abundant data transfer between analog memory arrays and digital processing units, as well as between digital units and binary programming arrays, incurring additional energy and latency overheads [38]. Although in situ MVM has been realized by adopting CIM architecture, plenty of weight quantization computation and low-precision programming operations, which dominate in neural network quantization algorithms, are still performed separately due to the lack of in situ physical operators of quantization function in the CIM paradigm [39].

Many works about on-chip implementation of BNN focus on the inference stage, by mapping the trained weights into CIM arrays, but few take account of enhancing low-precision neural network training efficiency [36, 37]. For the hardware implementation of training low-precision neural networks, such as BNN, D. Shang et al. proposed a hybrid analog-digital hardware system equipped with an RRAM chip to deploy a mixed-precision continual learning (MPCL) model [40]. In their MPCL hardware employment, binary weights are represented by the normalized conductance of RRAM differential pairs while high-precision floating-point weights and related operations are carried out on the general digital processor. Through in situ MAC accelerations in the RRAM CIM arrays, about 200× energy consumption has been reduced during the inference phase compared with the conventional digital systems, but separate storage of different-precision weights still limits their online training efficiency, and the RRAM-digital mixed storing methods are broadly adopted in other works about BNN online training [36, 41]. To solve the separated weight transfer issues, Martemucci et al. proposed a hybrid FeRAM/RRAM synaptic circuit to enable on-chip inference and learning of BNN, utilizing the excellent writing characteristics of FeRAMs and the nondisruptive reading capability of RRAMs [38]. Nevertheless, extra binary programming and low-precision reading for performing ex situ inference are still needed, and the large footprint of FeRAM consumes significant chip area.

Electrochemical random-access memory (ECRAM), a three-terminal eNVM with metal-oxide-semiconductor field-effect transistor (MOSFET)-like structure, has emerged as a promising synaptic device for artificial neural network (ANN) accelerators [42–47], whose conductance can be accurately regulated through electrochemical ionic doping and dedoping processes. In contrast to conventional two-terminal memristive devices, ECRAM enables multi-state programming in a more linear and symmetrical manner, accompanied by low variability and highly deterministic tuning characteristics. Owing to its nearly ideal conductance programming attributes, ECRAM exhibits significant potential in addressing the challenges associated with on-chip training of ANNs.

In this study, we aim to address the inefficiency of separate data handling in low-precision DNN quantization within conventional CIM architectures. Based on the superior linear and stable programming characteristics of WO_x-ECRAM, we develop an analog-binary weight transfer unit called memory-transistor transfer (MTT) unit, enabling in-situ approximate quantization without extra binary programming while preserving parallel MVM capability in CIM-like arrays. The weight transfer functions are also systematically explored under the impacts of different bias conditions. Furthermore, compatible programming and inference schemes of MTT-based arrays were developed and experimentally demonstrated to verify the availability. The neural network testing results reveal that the quantization function derived from the inherent physical electrical properties of the proposed cell architecture maintains classification accuracy in binary neural networks on par with the ideal function. This approach facilitates low-precision continual learning model computations, effectively mitigating the catastrophic forgetting problem inherent in conventional neural networks and enabling efficient computations for binary/ternary large language models (LLMs). At a 4 Mb array scale, the ECRAM-based and RRAM-based quantization units exhibit significant enhancements in energy consumption, achieving improvements of 25.51× and 4.84×, respectively, over traditional digital platforms in quantization-

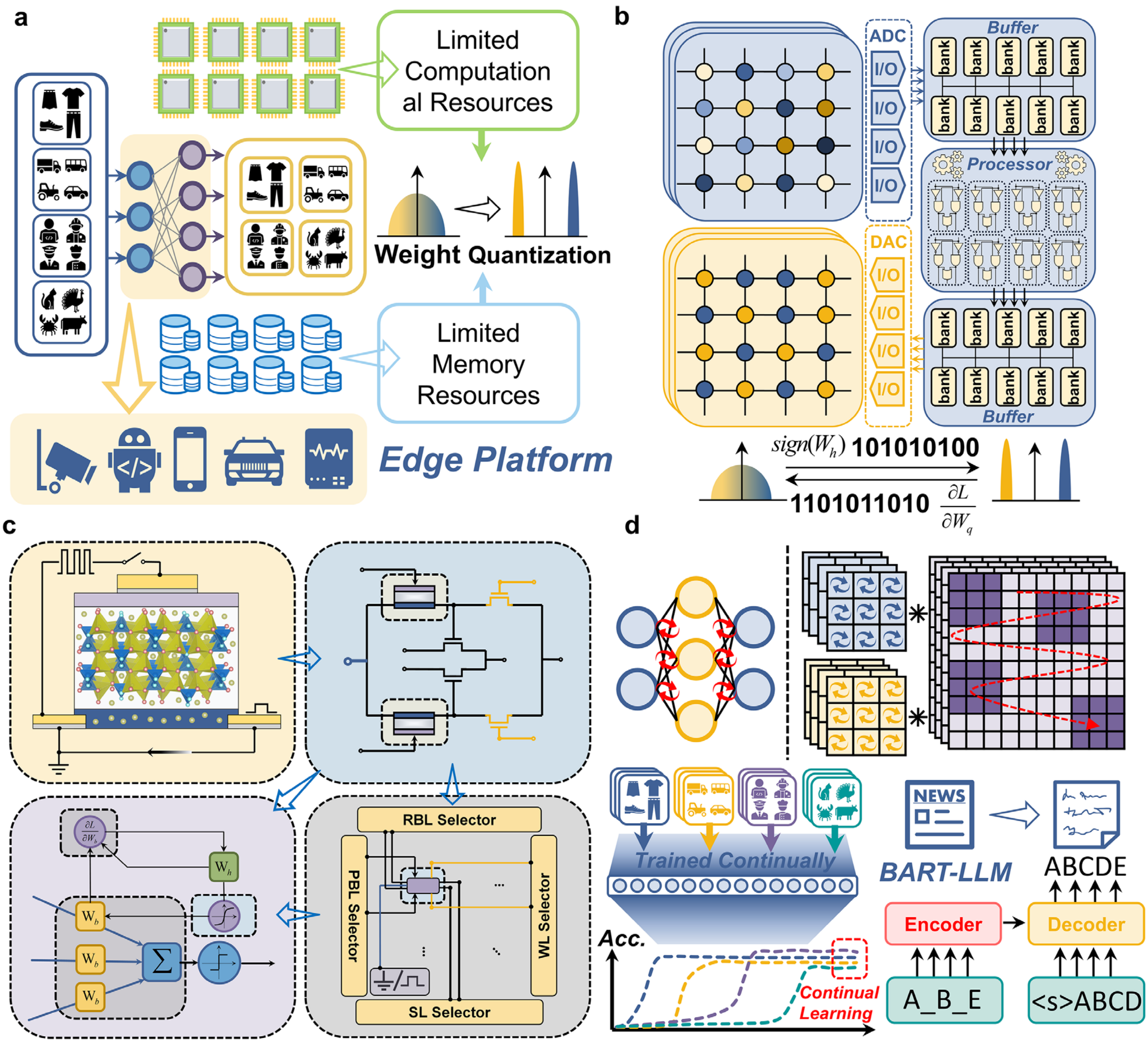

FIGURE 1 | In situ quantization for edge computing based on MTT unit. a) A schematic diagram of the weight quantization of neural networks deployed at the edge, aimed at accommodating limited computational and storage resources. b) During the network quantization-aware training process, data needs to be transferred between two weight storage arrays with different precision frequently, where the access reads, quantization computations, and low-precision programming contribute significantly to the additional latency and energy consumption. c) The context of this article is based on novel non-volatile memory devices leading to the proposed memory-transistor transfer unit, as well as the array operations and reading methods based on this unit structure, including the final application of network algorithms. d) The relevant applications addressed in this paper include lightweight network models for edge deployment, metaplasticity-inspired continual learning models, and large language models based on low-precision quantization.

related computation and storage. This advancement offers a highly energy-efficient in situ quantization computing framework for low-precision network training in edge computing environments.

2 | Results and Discussion

2.1 | Characteristics of ECRAM Used in MTT Units

High-performance memory devices with minimal programming stochasticity and high endurance are essential for reliable weight conversion in MTT units. In this work, we use ECRAM as hidden weights, where conductance modulation enables indirect control of the inference transistor's linear region resistance, facilitating weight conversion and in situ approximate quantization. Figure 2a shows the fabricated ECRAM array (fabrication details in Figure S1 and the Experimental Section). An additional SiO_2 layer is employed to isolate the gate line from the drain line, preventing short-circuiting between the two. The device structure (Figure 2b) features: (1) a WO_x channel with tunable band structure via ionic doping, (2) lithium phosphorus oxynitride (LiPON) solid electrolyte, in which abundant moveable Li ions can be controlled by the gate voltage, and (3) an SiO_x passivation layer fabricated along with LiPON protecting LiPON from

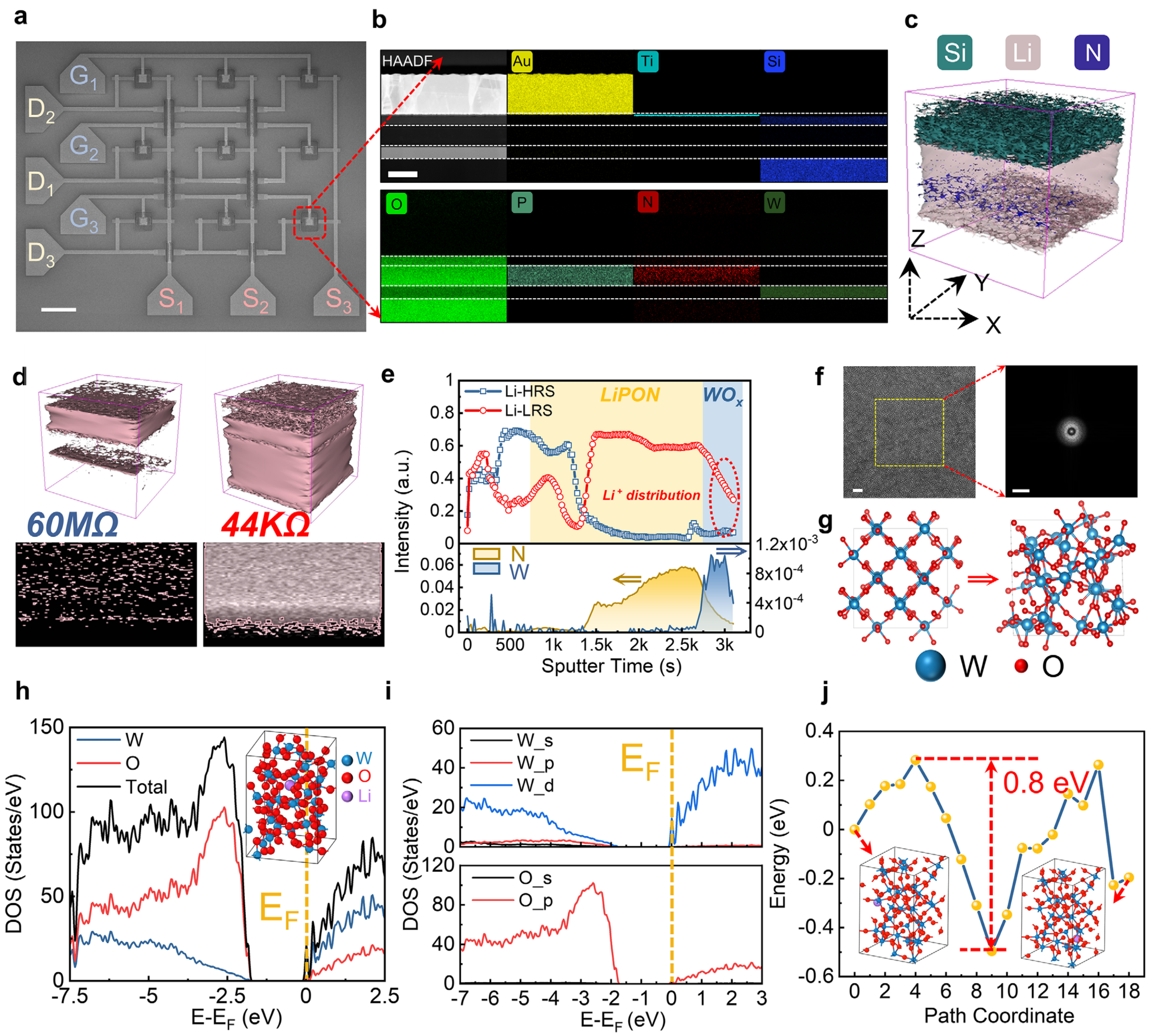

FIGURE 2 | Characterization of ionic regulation mechanisms within ECRAM. a) SEM image of the ECRAM array, which contains nine independent ECRAM devices that can be programmed and read by different gate, drain, and source lines. Scale bar, 50 μm. b) EDS distribution map of the layers in the core region of the ECRAM device. Scale bar, 100 nm. c) 3D SIMS distribution for Si, Li, and N elements when Li ions were intercalated into a channel layer. d) Comparison of Li spatial distribution for ECRAM at different states. e) SIMS depth profiles for the ECRAM in low-resistance and high-resistance states, where the yellow and blue backgrounds represent the LiPON and WO_x layers, respectively. f) HRTEM image of the WO_x layer and the corresponding diffraction pattern extracted by fast Fourier transformation for the yellow rectangular regions, the scale bar for the two images are 2 nm and 10 nm^{-1}, respectively. g) The transformation from crystalline structure to amorphous structure of WO_x by using molecular dynamics. h) The density of states (DOS) of amorphous WO_x after Li element doping. The Fermi level is set to zero. i) Projected DOS of W and O elements in panel (h). j) Energy changes of Li ions during migration.

ambient degradation (elemental analysis in Figure S2a,b). Time-of-flight secondary ion mass spectrometry (TOF-SIMS) analysis is utilized to better reveal the spatial distribution of Li ions (see the Experimental Section for experimental details). As shown in Figure 2c, the three-dimensional (3D) spatial distribution of Si, Li, and N elements can be well visualized, and a large number of mobile Li ions are the basis of ECRAM programming. A more intuitive comparison of Li element distribution within ECRAM of different conductance states can be seen in Figure 2d, showing the Li ions are more concentrated near the channel region when the ECRAM is at higher conductance states. The SIMS depth profile along the gate stack is shown in Figure 2e accordingly, where many Li ions will be driven to the WO_x-based channel and results in the depletion at the upper region of the LiPON electrolyte.

To reveal the ionic modulation mechanisms in the ECRAM programming more clearly, the first-principles calculations are performed based on the characterization data. Different from the previously used crystalline material in the ECRAM channel [44, 48], the WO_x-based channel is produced by the reactive sputtering without subsequent annealing to simplify the process flow and reduce the thermal budget, so its amorphous structure will be kept, verified by Figure 2f and Figure S2c,d. Therefore, we first generate an amorphous WO_x cell model through the rapid quenched molecular dynamics, with construction process depicted in Figure S3a. The process mainly includes four stages, that is initialization, melting-diffusion, quenching and relax, and the changes in the final microstructure are compared in Figure S3b. The WO_x structure (Figure 2g) is obtained to conduct the following electronic structure calculations, whose isotropy can help improve uniformity during ionic doping. Then, the density of states (DOS) of the Li-WO_x system is calculated, and the results are shown in Figure 2h, in which the Fermi Level is located at the conduction band. These results, compared with Figure S4a, show that Li ions lead to n-type doping in WO_x. The projected DOS in Figure 2i reveals that the main contributions to the conduction band are from W_{5d} and O_{2p} electrons. As shown in Figure S4b,c, the DOS with the Li ion at different site are both similar to the results in Figure 2h, which accords the isotropy of the amorphous WO_x. The migration energy barrier for Li ions in the WO_x matrix is shown in Figure 2j, whose maximum barrier is only 0.8 eV, which indicates the Li ions can be easily doped into the WO_x without excessively high gate voltage, which is critical for low supply voltage applications.

Under the guidance of the electrochemical ion doping mechanism, the channel conductance can be well controlled by the gate voltages, where doping and dedoping are realized by the direction of the electric field. The basic transfer characteristics of the ECRAM under 50 continuous cycles are illustrated in Figure 3a, with switching principle shown in the inset. The channel conductance increases under positive sweeping, while it decreases at negative sweeping region, and an average switching ratio with gate voltage at 0 V reaches 56. Besides WO_x, amorphous NbO_x can also be utilized [49, 50] as ECRAM channel for its outstanding stability and electronic tunability. However, for the more stable nonvolatility in serving as hidden weights in MTT units, we finally choose WO_x as the ECRAM channel material. The relationship between the formation energy of interstitial Li defects and Fermi level for NbO_x and WO_x systems are compared in Figure S5, and the lower overall formation energy in WO_x helps improve the stability of doped Li ions.

Furthermore, the size effects on the characteristics of ECRAM are explored in Figure 3b and Figure S6. The device on-state current can be enhanced by widening or shortening core areas, owing to the uniform ionic doping over the channel region, and the overall conductance G_{on} can be well expressed by:

$$G_{on} = g\frac{W}{L} \tag{1}$$

where g is proportional to the on-state conductivity under the ionic doping, which is almost the same for devices of different sizes, and W and L denote the width and length of device core region, respectively. This provides guidance for choosing proper device sizes under the maximum on-state current requirement. In addition, as shown in Figure S7, the switching window can be enhanced either by increasing the sweeping intervals or by increasing the sweeping amplitudes, indicating that the electrochemical doping is a gradual process in which ion migration is mainly driven by electric field.

Pulse-based programming, whose basic operation is diagrammatically depicted in Figure 3c, with source grounded and drain monitored by 0.1 V bias voltage, positive and negative programming pulses are implemented at the gate, which corresponds to potentiation and depression of ECRAM, respectively. The typical 180-cycle long-term potentiation and depression programming is demonstrated in Figure 3d, in which 10 V/2 ms and −7 V/2 ms pulses are used to write and erase, respectively. Different from the common programming stochasticity in two-terminal nonvolatile memory devices, especially for filament-based RRAM, 100% switching accuracy can be ensured in ECRAM, in which the conductance changes during each pulse are shown in Figure 3e. The high switching accuracy helps improve the programming efficiency, which is critical to the edge applications. In addition, by adopting the fitting model from the work of Chen et al. [51], the statistics of linearity in the conductance updating are concluded in Figure 3f, demonstrating excellent uniformity during the 180-cycle updating. More systematic results of programming sensitivity under different amplitudes and widths of the programming pulse are illustrated in Figure 3g,h, and we developed a write-verify scheme based on the increasing pulse amplitude accordingly to accelerate conductance convergence, whose flowchart is shown in Figure S8. In addition to the device programming in training, the retention of different states is important in inference, with the results shown in Figure 3i. It should be noted that the nonvolatility of the hidden weights also ensures that of the inference weights in the proposed MTT units. As shown in Figure 3j, the ECRAM reached a high switching ratio of 632 with updating process and applied pulses in Figure S8, which enables binary quantization fitting in the MTT unit. Another key characteristic of the memory device used in online training is ultrahigh endurance. Thanks to the reversible dynamic ionic doping and dedoping, the damage to the lattice structure can be negligible during repeated programming, and more than 10^8 pulse endurance is shown in Figure 3k. Detailed programming by µs-level pulse stimulation can be found in Figure S9a, and the evolution of maximum conductance changes and linearity fitting parameters during the whole programming in Figure 3k are demonstrated in Figure S9b,c, respectively. In addition, conductance updating processes in the endurance tests for the first 16×10^6 pulses and the last 16×10^6 pulses are compared in Figure S10, indicating that the ECRAM can keep excellent performances during the endurance tests. The above characteristics of the ECRAM ensure this kind of analog memory device is highly appropriate to serve as hidden weights in the following discussed MTT units.

2.2 | Weight Transfer Functions in MTT Units

The excellent analog programming characteristics discussed above ensure highly efficient weight mapping in the MTT units. As the theoretically derived model in Figure S11a, the symmetrical structure is adopted to form a differential weight unit. It should be noted that reference resistors denoted by yellow rectangles with conductance G_{Ref} is used to simplify model derivation, though they are implemented by transistors with fixed gate voltage in actual circuits. We consider the resistance value R_{Ref} of the reference resistors to be 1. The purple variable resistor represents the analog memory device, and the transistors whose gates are controlled by the voltage division between the variable resistors

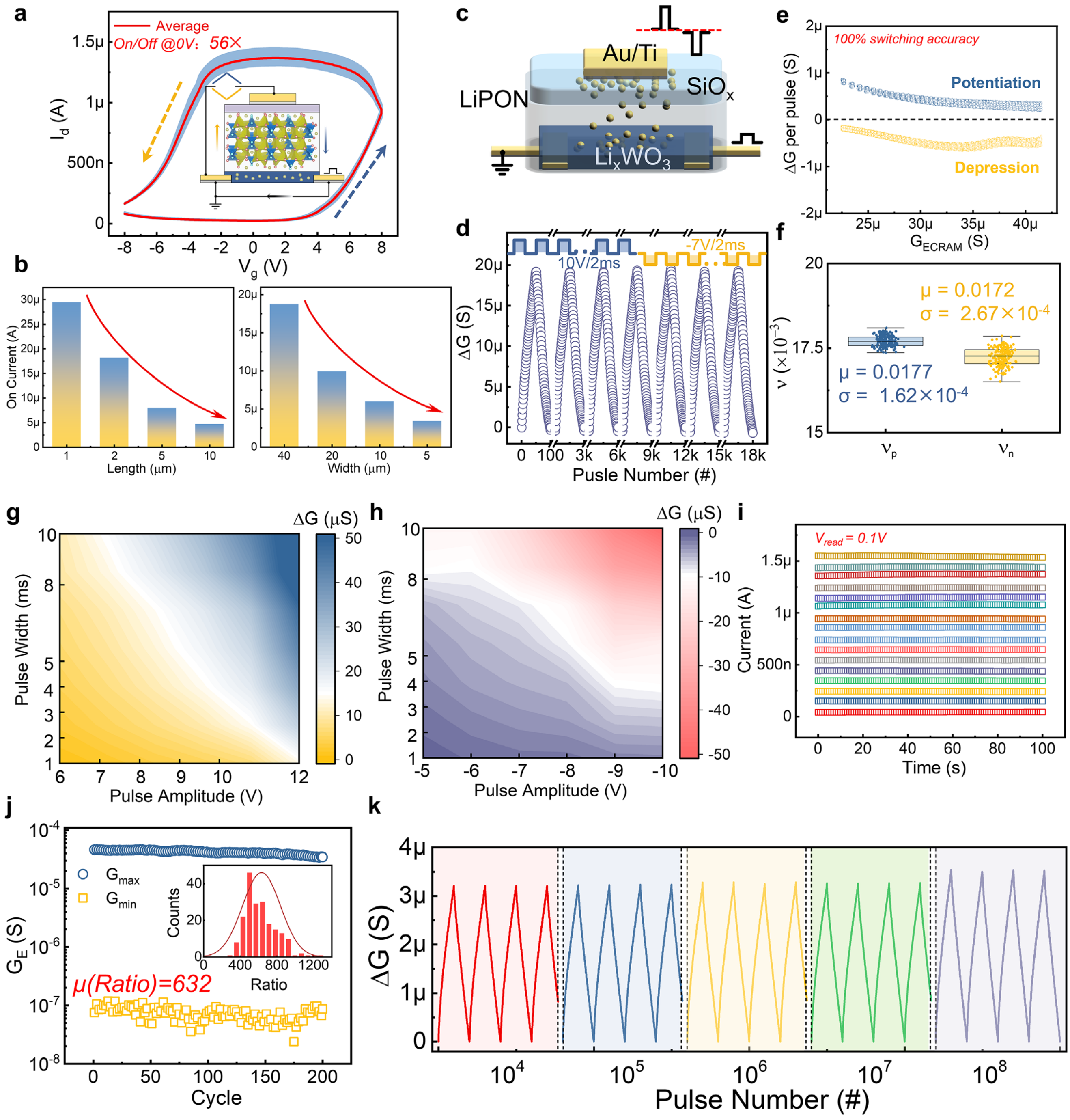

FIGURE 3 | Characterization of the basic properties of ECRAM. a) Transfer characteristics of the ECRAM under 50 continuous cycles, with a maximum scanning voltage amplitude of 8 V for both positive and negative regions, yielding an average switching ratio of 56 at a gate voltage of 0 V. b) The on-state current of ECRAM in different sizes, including different lengths and widths. c) Schematic diagram of the basic operations of the ECRAM device, illustrating the doping and de-doping of the channel by mobile ions in the electrolyte, controlled by the electric field applied between the gate and the channel. d) Variation of long-term potentiation and depression during 180 continuous cycles, where the potentiation and depression pulse amplitudes are 10 and −7 V, respectively, each with a width of 2 ms. e) Statistics of conductance variation for each programming pulse during potentiation and depression. f) The distribution of fitting parameters of programming linearity for both potentiation and depression. g-h) Statistical data on the potentiation and depression of ECRAM conductance under different programming pulse amplitudes and widths. i) Retention characteristics of 16 different states of the ECRAM, with a channel read voltage of 0.1 V. j) Statistical data on the high switching window of the device during continuous conductance modulation, achieving an average ratio of 632. k) Endurance test for over 10^8 programming pulses, showing no device degradation during these pulse stimuli, with specific details demonstrating the conductance modulation characteristics of the device after 10^4, 10^5, 10^6, 10^7, and 10^8 pulses.

and fixed resistors provide the binary inference weights, realizing the purpose of in situ weight transfer.

Considering V_T to be the threshold voltage of the transistors used in MTT units and their transfer characteristic described in the Experimental Section, the hidden weight reading bias voltage V_{read_H} is set as 2 × V_T. The reading bias voltages applied at the drains of the inference transistors, whose conductances are denoted by G_{B_+} and G_{B_-}, are 0.1 and −0.1 V, respectively. Thus, the final binary inference weight G_B is the difference between the two, that is:

$$G_B = G_{B_+} - G_{B_-} \tag{2}$$

For the mapping and programming of the analog hidden weight W_h, when $W_h \geq 0$, negative analog memory device G_{A_-} will be immediately programmed to R_{Ref}, then G_{A_+} will be programmed to the corresponding values according to a linear mapping relationship. When $W_h < 0$, positive analog memory device G_{A_+} will be fixed at value 1, then we will program the negative analog memory device. The final analog conductance G_A can be expressed as max{G_{A_+}, G_{A_-}}, and the quantized hidden conductance G_H can be written as:

$$G_H = \begin{cases} \dfrac{G_A - G_{Ref}}{K \times G_{Ref}}, & W_h \geq 0 \\ \dfrac{-G_A + G_{Ref}}{K \times G_{Ref}}, & W_h < 0 \end{cases} \tag{3}$$

Here, K is a scaling factor constraining G_H to [−1, 1], effectively representing the dynamic range (on/off ratio) of the analog memory device. Due to the MTT unit's symmetrical design, Figure S11b shows how the binary inference weight G_B relates to the quantized hidden conductance G_H for varying K. Larger K values make this relationship approximate an ideal binary quantization function. The electrical behavior of the devices translates this to a functional mapping between binary inference weight W_b and analog hidden weight W_h as follows:

$$W_b = \begin{cases} 2 \times \left(\dfrac{K \times W_h + 1}{K \times W_h + 2} - 0.5 \right), & W_h \geq 0 \\ -2 \times \left(\dfrac{-K \times W_h + 1}{-K \times W_h + 2} - 0.5 \right), & W_h < 0 \end{cases} \tag{4}$$

By varying K, the weight transfer functions are shown in Figure S11c. As before, larger K values make the function approach an ideal Sign function for binarization. However, practical device programming faces trade-offs: excessive dynamic range degrades device performance, necessitating a balanced operating range that preserves quantization accuracy. To evaluate this, we test different K values in a continual learning neural network model [52] inspired by synaptic metaplasticity (simulation details are described in the Experimental Section). Results (Figure S11d–f) show that continual learning performance saturates at $K \geq 20$, suggesting this as the optimal lower bound for subsequent weight transfer studies.

When the hidden weight reading bias voltage V_{read_H} is below 2 × V_T, the inference transistor operates in the subthreshold region at low analog memory conductance. As simulated in Figure S12a, this creates a zero-gradient interval (with half its length defined as the radius). Figure S12b shows the abstracted weight transfer function under different radii, while Figure S12c–e presents the corresponding network performance. Under lower V_{read_H}, larger radii exacerbate deviations from the ideal Sign function, degrading performance, which likely results from constrained hidden weight updates in the subthreshold regime. Thus, V_{read_H} should exceed twice V_T to ensure robust binary quantization.

The MTT unit can alternatively employ an asymmetric differential structure (Figure S13a). A potential advantage of this configuration is the consistency of the two analog hidden weights during weight mapping. The quantized hidden conductance G_H is then determined by the analog conductance G_A through the following relationship:

$$G_H = \frac{G_A - G_{Ref}}{K \times G_{Ref}} \tag{5}$$

where all the parameters are the same as before. With the hidden weight reading bias voltage set to 2 × V_T, electrical characterization (Figure S13b) reveals the asymmetric weight transfer function between binary inference weight G_B and quantized hidden conductance G_H across different on/off ratios. The derived mathematical relationship is:

$$W_b = \frac{K \times W_h}{K \times W_h + 2 \times G_{ref}} \tag{6}$$

where we set G_{ref} at 50 for convenience, and the specific function graph is shown in Figure S13c. The asymmetry between positive and negative weight transfer functions intensifies with larger operating ranges, even saturating even when W_h gets close to 1, which results in significant deviation from the ideal Sign function. Network-level performance tests (Figure S13d–f) confirm this limitation. While the asymmetric MTT unit eliminates hidden weight programming identification, its poor binary quantization fidelity renders it unsuitable for in-situ weight transfer applications.

In experimental implementation, we replace the reference element with a transistor biased at constant gate voltage (V_{ref_G} = 0.6 V) for CMOS compatibility and operational flexibility, while also improving the weight transfer function's dynamic range. Figure 4a,b shows the measured relationships between ECRAM conductance G_e and inference transistor current I_{ts} at V_{read_H} = 1.4 V, with experimental setups illustrated in insets. Figure 4c presents the multi-cycle binary weight W_b versus quantized hidden weight W_h relationship, with quantization details in the Experimental Section and parameter distributions shown in the inset. This leads to the generalized model:

$$W_b = \begin{cases} 2 \times \left(\dfrac{A_p \times W_h + 1}{B_p \times W_h + 2} - 0.5 \right), & W_h \geq 0 \\ -2 \times \left(\dfrac{A_n \times W_h + 1}{B_n \times W_h + 2} - 0.5 \right), & W_h < 0 \end{cases} \tag{7}$$

where A_p, B_p, A_n, and B_n parameters take the similar role of K, indicating the effective operating dynamic range for the

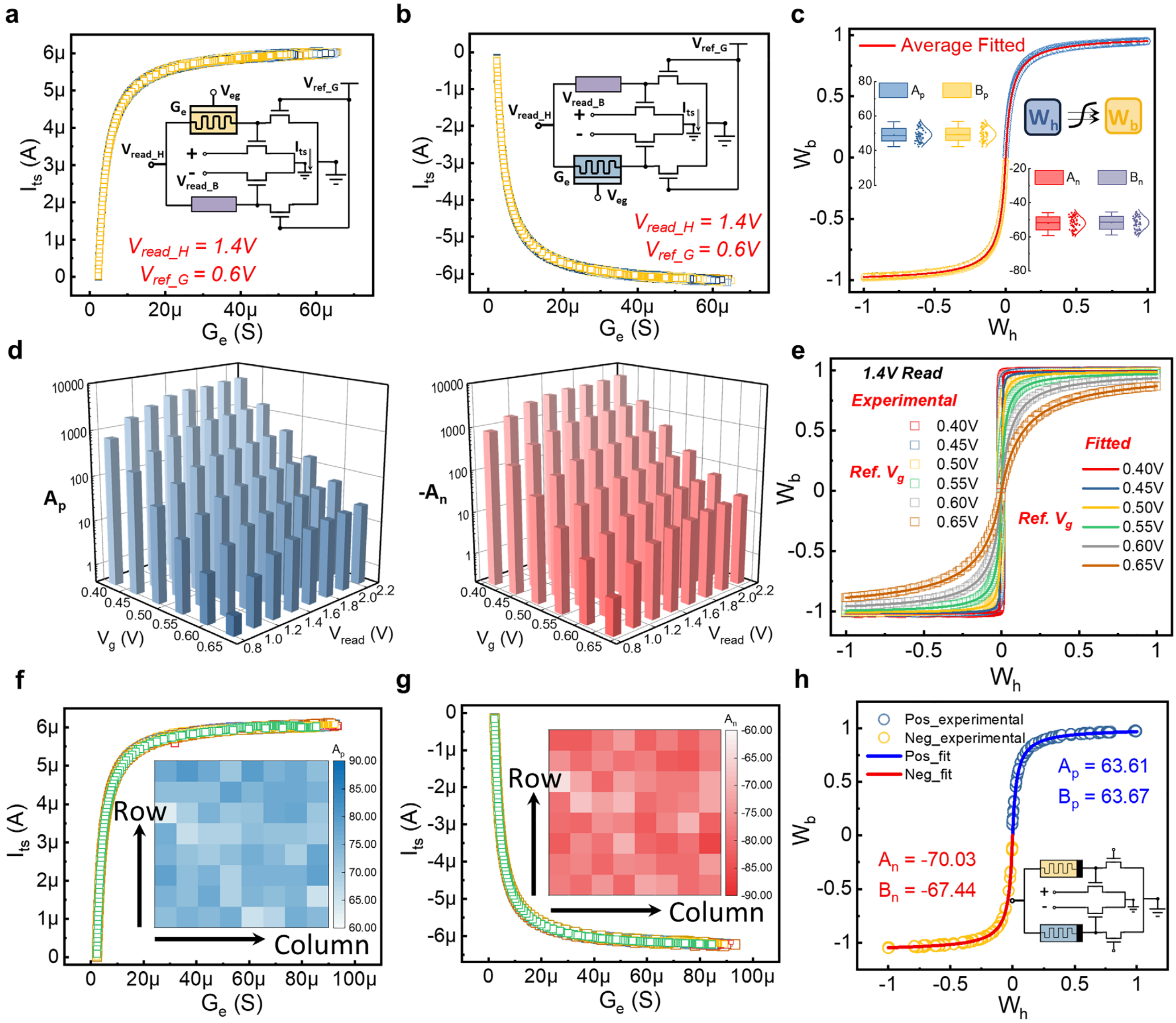

FIGURE 4 | The performances under different biases for MTT units. a,b) The relationship between ECRAM conductance and the channel current of the inference transistor under positive and negative bias conditions, respectively. The hidden weight bias voltage is set at 1.4 V, with the gate voltage of the reference transistor set at 0.6 V. The positive and negative read voltages at the drain of the inferred transistor are set at 0.01 and −0.01 V, respectively. c) The relationship between the quantized hidden weights and the binary weights corresponding to (a) and (b), with the inset showing the distribution of the fitted parameters under different cycle stimuli. d) Parameter fitting results for the weight transfer function of the MTT unit at different hidden weight bias voltages and gate voltages of the reference transistor. e) Relationships between the quantized hidden weights and the binary weights in the MTT unit, with the hidden weight bias voltage fixed at 1.4 V and varying gate bias voltages of the reference transistor. f,g) Statistical data on the positive and negative relationships between the conductance of ECRAM and the channel current of the inference transistor among the 64 different MTT units, with the inset displaying the distribution of the corresponding fitted parameters. h) The relationship between the quantized hidden weights and binary weights in the RRAM-based MTT unit, with the fitted curve indicated by the solid line in the figure.

memory device. Figure S14a–c shows the transistor characteristics, where linear region operation is crucial for analog processing since inference transistor conductance is not purely binary. To systematically study the weight transfer function, we varied V_{ref_G} and V_{ref_G} while fixing maximum device conductance at 40 µS. The resulting fitting parameters (Figure 4d, Figure S15) reveal consistent trends: lower V_{ref_G} and higher V_{read_H} produce better approximation of ideal binary quantization. Notably, replacing the reference transistor with a fixed resistor (Figure S16) yields transfer functions insensitive to bias voltages, with nearly identical fitting parameters across all tested dynamic ranges (Figure S16f). Figure S14d,e shows enhanced scaling factors in the reference-transistor-based MTT unit, particularly at high V_{read_H} The nonlinear channel resistance increase with drain-source bias (Figure S14c) accelerates node voltage rise, causing faster transfer function saturation as $|W_h|$ approaches 1. This nonlinearity effectively expands the dynamic range, reflected in the fitting parameters. Parameter pair consistency (A_p, B_p) and (A_n, B_n) in Figure S14f ensures transfer symmetry. Reducing V_{ref_G} lowers overall channel conductance, including the minimum memory device conductance (equal to reference transistor conductance at $\frac{V_{read\_H}}{2}$), thereby enhancing dynamic range under fixed upper limits. Figure S17 shows the ECRAM conductance-current relationship across V_{ref_G} values, with corresponding weight transfer functions in Figure 4e showing excellent model agreement. While lower V_{ref_G} improves Sign function approximation (crucial for BNNs), it demands higher switching ratios, requiring trade-offs between fitting accuracy and programming efficiency. Uniformity tests across MTT units (Figure 4f,g, Figure S18a,b) demonstrate consistent weight transfer functions, enabling low-variation array implementation. Notably, Figure S18c,d reveals a near-linear relationship between transfer parameters and maximum operational range.

While ECRAM serves as an excellent analog hidden weight in the MTT unit due to its superior programming characteristics, the concept can be extended to other nonvolatile memory devices. RRAM, a two-terminal memory technology, has gained prominence in AI inference chips for its CMOS compatibility, high density, and compute-in-memory potential [53, 54]. However, its stochastic and nonlinear programming behavior has limited its adoption in edge training, despite device-level improvements [55]. Here, we explore RRAM-based MTT units to broaden hidden memory device options. Figure S19a shows DC switching characteristics of TaO_x-based RRAM devices, exhibiting filamentary switching [56]. The RRAM-MTT unit (Figure 4h inset) uses the 1T1R intermediate node voltage to control the inference transistor, while the reference transistor modulates RRAM analog switching. Multi-state programmability (16 states, ~75 on/off ratio) is demonstrated in Figure S19b–e. Despite excellent analog switching, RRAM suffers from nonlinear read behavior (Figure S19f), particularly at high resistance states under large read voltages, which can disrupt stored states. Unlike ECRAM's linear resistance (Figure S20), RRAM requires careful selection of V_{read_H}. For V_{read_H} = 0.8 V and V_{ref_G} = 0.7 V, the RRAM–inference transistor current relationship yields a weight transfer function (Figure 4) with fitting parameters >50, meeting transfer requirements. In advanced CMOS nodes with lower threshold voltages, RRAM performs well even with V_{read_H} at 0.2 V and V_{ref_G} at 1.2 V, as validated in Figure S21c–e.

2.3 | Basic Operations for MTT-Unit-Based Arrays

In addition to the static weight transfer function of different hidden memory device states, dynamic programming and reading operations are also important in MTT units. To reduce hardware overhead, the reference transistor will be utilized as the selector in array-level programming. However, for conventional ECRAM-selector-based arrays, the selector transistor is mostly connected to the gate terminal of ECRAM [50], which is termed as vertical 1E1T unit (Figure S22a). In this unit, by controlling the switching of the selector transistor, the programming signal can be applied to the ECRAM or not. The selector transistor can also be connected to drain/source terminal of the ECRAM (Figure S22b), and we call this horizontal 1E1T unit (H-1E1T). A selective programming scheme is proposed based on the H-1E1T, and the applied signals in each terminal during different programming processes are summarized in Table S2, including potentiation, remaining and depression stages. During all the stages, the V_{ED} terminal and V_{ES} terminal will be kept unchanged, biased at depression voltage V_D and grounded, respectively. The electrochemical doping or dedoping within ECRAM is determined by the electric field between gate terminal and drain/source terminal. For potential programming, V_{EG} terminal is applied by the potential voltage V_P while V_{TG} terminal is at ground state, and the selector transistor is at Off state; thus, the bias voltage of the ECRAM drain terminal is approximately 0 V, the voltage difference between gate and drain is almost V_P, which enables conductance to increase. If the selector transistor is at On state, V_{TG} terminal is applied by source supply voltage V_{SS}, thus the bias voltage of the ECRAM drain terminal is approximately V_D, and the effective programming voltage of ECRAM is $V_P - V_D$, based on previously half-selected ECRAM programming method [57], the device state can hardly be changed. On the contrary, during the depression stage, the V_{EG} terminal is grounded state and the V_{TG} terminal is applied by source supply voltage V_{SS}, then the effective programming voltage of ECRAM is -V_D, and the backward gate-channel electric field will help drive ions back into electrolyte and enable depressed programming. However, when V_{TG} is also grounded, there will be negligible voltage differences within ECRAM. Compared with the selective programming in vertical 1E1T unit, there is no use of any negative bias signals, reducing the complexity of peripheral CMOS circuits.

The proposed two-terminal selective programming method enables efficient array-level parallel operations in our CIM architecture, supporting both row-wise and column-wise programming with separate potentiation and depression phases (Figure 5a). Experimental results demonstrate precise conductance control under varying programming voltages (V_P and V_D), showing an exponential relationship between conductance change per pulse and voltage amplitude (Figure S22c,d). Remarkably, this two-terminal approach maintains excellent programming linearity (Figure S22e), as evidenced by the 100% accuracy achieved in long-term potentiation/depression cycles (Figure 5b) and its superior linear characteristics (Figure S22f). The system's robustness is further confirmed through continuous selective programming tests across all operational stages, where device states remain stable during retention periods (Figure S23). More complex programming scenarios (Figure 5c,d) successfully validate the architecture's capability for reliable online training applications.

During inference operations, the selector transistors transition into reference transistors with uniformly biased gate terminals, while all ECRAM gates are grounded or left floating to prevent state degradation. Input data is applied to the drain terminals of inference transistors, enabling in situ weight transfer through voltage division between ECRAM devices and reference transistors—a process governed by the previously discussed ECRAM conductance relationship (Figure 5e). This architecture allows dynamic regulation of binary inference weights through modulation of hidden analog memory states (Figure 5f), achieving idealized linear multiplication between input voltages and hidden weights via the inference transistors operating in their linear region. The system's analog computation capability is demonstrated in Figure 5g, showing precise linear transformation of randomly generated input waveforms.

To demonstrate the application of MTT units in mapping neural networks, firstly, a small two-layer fully connected BNN (2×3×1) is utilized to classify a self-made XOR dataset (Figure 5h). The

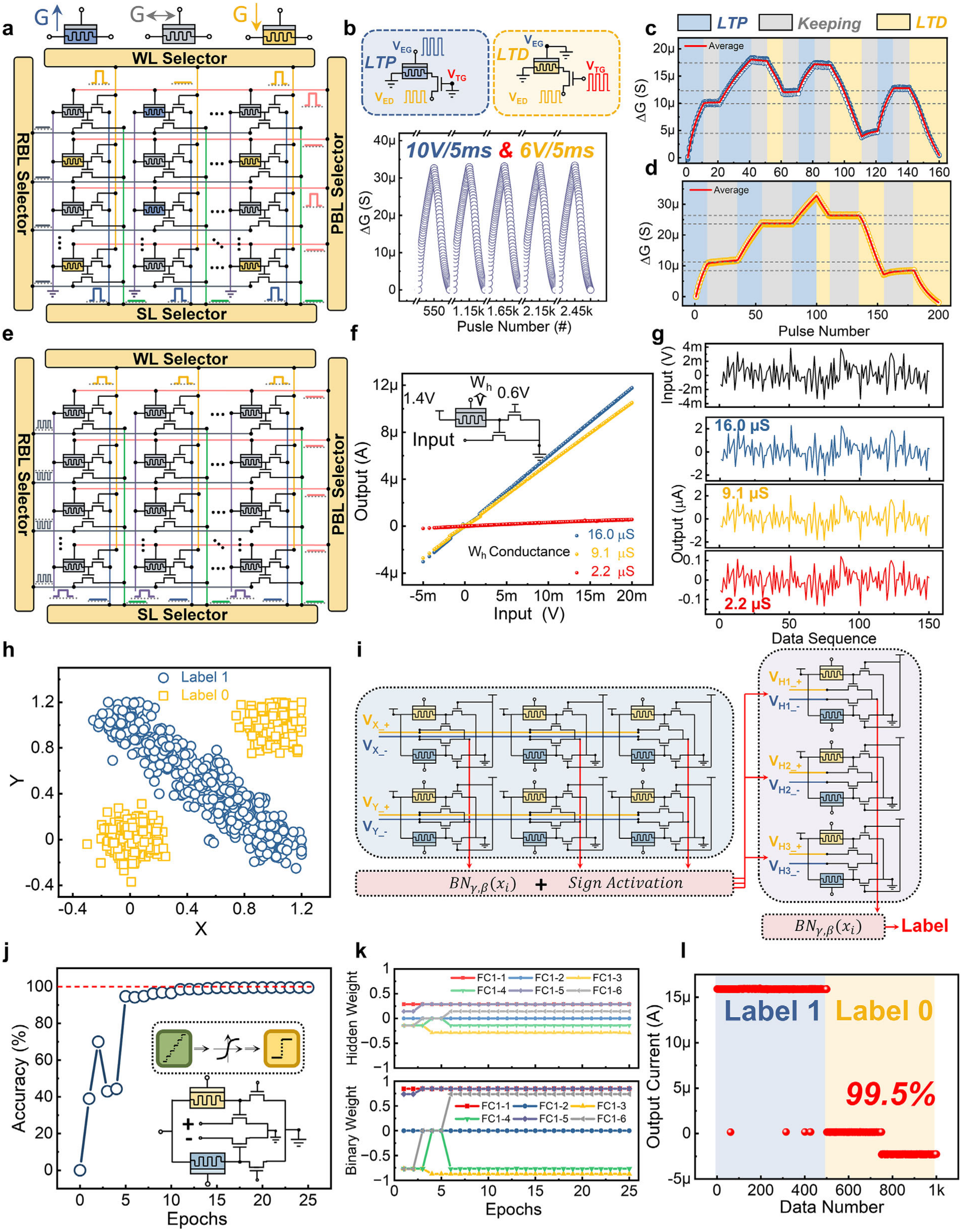

FIGURE 5 | Programming and inference of MTT-based arrays. a) Schematic of array-level programming, where programming pulse signals at different ports are labeled alongside the respective signal lines. The blue, gray, and yellow blocks represent increases, maintenance, and decreases in the conductance of the ECRAM after stimulus, respectively. b) Variation of long-term conductance potentiation and depression in the H-1E1T unit. The upper panel illustrates the application of pulse signals for conductance potentiation and depression in the H-1E1T unit, with VEG and VED pulse amplitudes set at 10 and 6 V, respectively, and a pulse width of 5 ms for both. c,d) Selective programming results of the H-1E1T unit based on the pulse programming scheme shown in panel (a), where the red line indicates the average conductance change during programming. e) Schematic diagram of inference based on the MTT-unit-based array, where the inference input waveform is applied at the drain terminal of the inference transistor controlled by the voltage division in the MTT unit. f) Relationship between the simulated input voltage and output current under different hidden weight conductance. g) Specific simulated input voltage waveforms and output current waveforms from panel (f), demonstrate a strong linear relationship that enables analog

hardware architecture based on MTT unit is shown in Figure 5i, which also include batch normalization layers and Sign activation function units, with the data flow indicated by red arrows. Based on the weight transfer function in the MTT unit, the neural network training converges to nearly 100% accuracy within 25 epochs (Figure 5j), and more specific details of the used weight transfer function can be found in the Experimental Section. The evolution of hidden weights and binary inference weights of the first layer in training is compared in Figure 5k, and approximate discrete binarization of analog weights is successfully realized. By mapping the trained hidden weights into ECRAM conductance, the final measured classification accuracy can reach 99.5%, with another similar example (Figure S24) achieving accuracy of 98.5%.

Moreover, for RRAM-based MTT units, the programming scheme resembles conventional 1T1R. Array-level programming (Figure S25a) leverages bipolar characteristics: set/reset operations are performed via the PBL and SL selectors, respectively. Programming proceeds row-by-row or column-by-column, with separate potentiation and depression phases. Inference in RRAM-MTT arrays mirrors ECRAM-MTT arrays—an analog input waveform is applied at the drain terminal of the inference transistor, regulated by voltage division in the upper MTT unit (Figure S25b). The two-terminal compact structure of RRAM-MTT enables significant area reduction.

With the weight transfer function enabled in both ECRAM- and RRAM-based MTT units, we extend their application to practical binary neural networks (BNNs), which are ideal for edge platforms due to their lightweight nature. Simulations on fully connected (FCNN) and convolutional neural networks (CNN) (see the Experimental Section) show comparable performance between ECRAM- and RRAM-based in-memory-binarization (IMB) functions when using idealized Sign binarization (Figure S26). Final testing accuracies across multiple trials are summarized in Table S3. Notably, binary CNN performance improves when employing the physical weight transfer function in MTT units, suggesting that the fuzzy binary quantization can somehow keep more precision at the original hidden weights in CNN and that the physical weight transfer in MTT unit is not just a simple approximation of Sign function.

Prior energy-efficient computing units often trade functionality for hardware complexity. In contrast, our MTT unit's programming scheme maintains quantization capability while minimizing components. For fair comparison, we evaluate within an in-memory computing architecture using crossbar-like arrays. As shown in Figure S27a,c, conventional differential structures require 8 components (4T4E or 4T4R) to store both analog hidden weights and binary weights. However, our MTT unit, optimized for edge quantization networks, achieves this with only 4T2E (Figure S27b) or 4T2R (Figure S27d), enabling simultaneous analog and binary weight storage. Thus, MTT units not only enable in situ weight transfer but also reduce hardware complexity, improving both energy efficiency and footprint.

2.4 | In Situ Weight Quantization in MTT Units for Continual Learning and LLM

To verify and expand the versatility of the MTT unit in other kinds of important edge applications, two other important applications, continual learning and LLM, are employed to do the examination.[58]

Notably, A. Laborieux's metaplasticity-inspired BNN [52, 59] demonstrates particular promise for edge implementation, combining synaptic-centric training with hardware-friendly characteristics suitable for emerging learning approaches.

Edge applications should adapt to unpredictable real-world variations, requiring neural networks to continually learn new scenarios without degrading performance on prior tasks, which is a challenge known as catastrophic forgetting. While standard deep learning methods fail to learn incrementally without access to old data, simply retraining on all data (both old and new) is computationally prohibitive for resource-constrained edge devices. Recent research has developed continual learning algorithms to address this, including regularization, experience replay, dynamic architectures, and knowledge distillation [58]. Among these methods, a metaplasticity-inspired BNN proposed by A. Laborieux [52, 59], can be generally used in a range of emerging learning approaches. The synaptic-centric training with hardware-friendly features make it more suitable for implementation at edge devices.

Metaplasticity, inspired by neural mechanisms enabling task retention (Figure 6a), governs synaptic strength changes through hidden states, effectively representing the plasticity of synaptic plasticity. This synaptic consolidation mechanism is crucial for memory storage, where a synapse's metaplastic state quantifies its task importance, preventing catastrophic forgetting. Laborieux et al. [52] incorporated this concept into BNNs by linking real-valued hidden weights to synaptic importance, enabling continual multi-task learning. As shown in Figure 6b, their modified BNN training alters gradient computation; that is, when binary inference and hidden weight signs align, the gradient is scaled by a nonlinear meta function f_{meta}:

$$f_{meta}\ (m, W_h) = 1 - tanh^2 (m \times W_h) \tag{8}$$

multiplication computations controlled by hidden weights. h) Linearly nonseparable datasets are used for testing the array composed of MTT units, where shapes and colors represent different classes. i) Schematic representation of a 2×3×1 array composed of MTT units, which also includes batch normalization layers and activation function units based on the Sign function. j) Training performance visualization of the dataset in panel (h) within a BNN based on a 2×3×1 architecture. k) Changes in hidden weights and binary inference weights of the first layer neural network throughout the training process, where the hidden weights are quantized to 3 bits during training. l) Inference results mapped to hardware from the final weights in panel (k), achieving a practical inference accuracy of 99.5%.

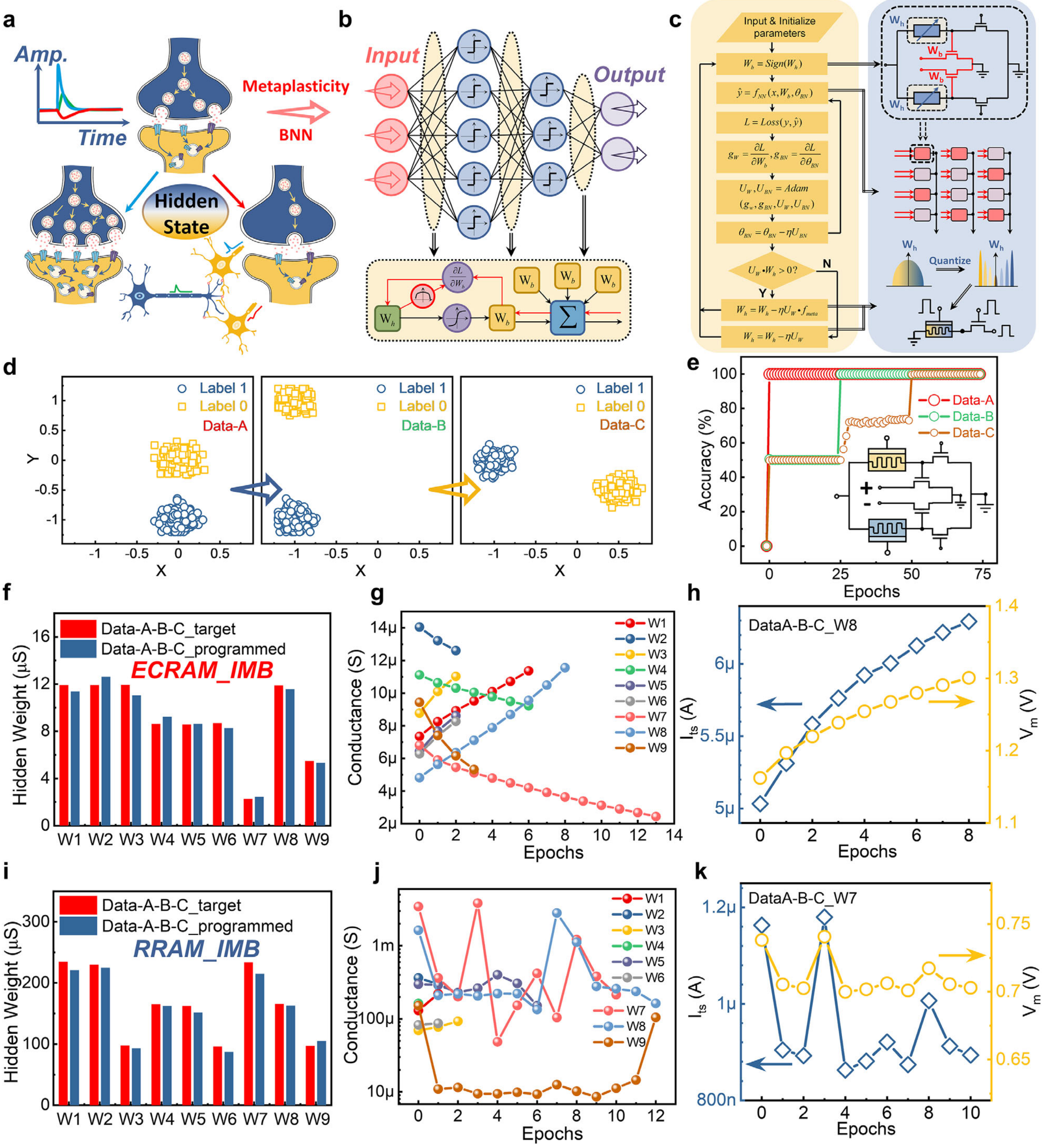

FIGURE 6 | Applications of weight transfer relationships in MTT units for continual learning. a) Schematic diagram of the relationship between synaptic metaplasticity and changes in synaptic weights in biological neural networks. b) Schematic diagram of a continual learning model that combines metaplasticity with binary neural networks, where each weight in the model consists of an analog hidden weight for training and a binary-value inference weight for inference. During network training, the gradient computation is modulated by the value of hidden weights. c) Algorithm flowchart for the binary neural network based on metaplasticity and its corresponding hardware acceleration, which primarily includes weight binary quantization and in-memory computation inference components. d) Datasets are used to test the continual learning performance of small networks, comprising three distinct datasets: Data-A, Data-B, and Data-C. e) Training performance of the BNN based on metaplasticity on the different datasets in panel (d), where hidden weights are quantized to 3 bits during training, and the binary quantization curve utilizes weight transfer relationships from ECRAM-based MTT units. f–h) Statistical distributions of target conductance versus actual conductance for hidden weights in the first layer of the network; the variation of hidden weight conductance during the write-verify programming process; and the read current of an inference transistor for a typical weight, along with the variation of the intermediate node voltage during the programming process, where weights are implemented using ECRAM-based MTT units. i–k) Statistical distributions of target conductance versus actual conductance for hidden weights in the first layer of the network; the variation of hidden weight conductance during the programming process; and the read current of an inference transistor for a typical weight, along with the variation of the intermediate node voltage during the programming process, where weights are implemented using RRAM-based MTT units.

Here, W_h represents the real-valued hidden weight and m controls the decay rate in non-zero regions. The gradient computation for binary inference weights W_b incorporates W_h to minimize sign changes, particularly for larger $|W_h|$ values, thereby preserving performance on learned tasks. As shown in left of Figure 6c, conventional CIM architectures face significant overhead from frequent weight binarization between W_h and W_b, as gradient computation and weight quantization occur in separate arrays. This leads to both increased latency/energy from data movement and reduced efficiency from post-quantization programming. Our solution (Figure 6c, right) integrates MTT units with CIM architecture to eliminate this quantization bottleneck, enabling direct mapping of quantized hidden weights to analog memory devices.

To evaluate continual learning performance, we created three custom datasets (Data-A/B/C, Figure 6d) and trained a 2×3×1 binary FCNN using the metaplasticity algorithm (see the Experimental Section). ECRAM-MTT units enabled low-precision weight quantization. As shown in Figure 6e, the model successfully learned new tasks without catastrophic forgetting. The larger model, which consists of a large number of weight parameters, will be more stable in continual learning for the excess weights to learn new things and keep memory for old things, and the continual learning capacity of larger-level neural network will be also explored later.

Weight mapping only requires programming analog hidden weights into MTT units, eliminating additional binarization steps. To address device non-idealities, we employ a write-verify scheme (Figure S28) with adaptive programming intensity for reliable Set/Reset operations. This universal approach applies to various emerging memories (RRAM, PCM, FeFET, ECRAM, etc.) [18, 19], differing only in specific Set/Reset implementations. For ECRAM-MTT units, Figure 6f demonstrates precise conductance programming with minimal error, while Figure 6g shows the deterministic switching from random initial states. Figure 6h illustrates the coordinated binary/hidden weight updates in MTT operation. Additional validation (Figure S29a–f) confirms software-equivalent inference accuracy. RRAM-MTT implementations (Figure 6i–k, Figure S30) achieve similar accuracy but suffer from nonlinear conductance changes, increasing latency and energy consumption, which highlights the need for RRAM with improved linearity for efficient online training.

A larger-level neural network simulation is necessary to demonstrate the practicality of MTT units. Four non-overlap datasets are shown in Figure 7a, which are Kuzushiji MNIST (KMNIST) [60], Fashion MNIST (FMNIST), MNIST and Kannada MNIST (KaMNIST) [61]. All the image sizes in the datasets are 28×28×1, thus the neural network can keep the same structure in training sequentially. In this metaplasticity-inspired algorithm, the analog hidden weights serve as important factors behind the binary weights, and two sets of mixed-precision weights are essential in continual learning, which is verified in Figure S31a where the catastrophic forgetting happens without the binarization. This demonstrates that low-precision quantization is crucial not only for resource efficiency but also for maintaining fundamental network performance in continual learning scenarios.

To optimize memory efficiency, we quantize hidden weights into discrete states with a boundary of ±1. As pretested (Figure S32), 3-bit precision maintains continual learning capability, and is therefore adopted for both ECRAM-MTT and RRAM-MTT units. Figure 7b and Figure S31b show the training processes, while Figure 7c compares final accuracies across binarization methods, confirming the IMB function's effectiveness. Programming error tolerance is critical for weight mapping. Figure 7d demonstrates that ECRAM-MTT units tolerate up to 20% programming noise (with weight distributions in Figure 7e). However, larger errors accumulate during training, causing significant performance degradation with increasing epochs (Figure S33). Table S4 quantifies this trade-off between training duration and accuracy under varying programming errors, highlighting the need for careful balance.

Apart from continual learning of different tasks, another variant is stream learning, which is utilized to tackle the situation where only part of the training data are available at a given time. As shown in Figure S34 and Figure 7f, both FCNN and CNN architectures (see the Experimental Section for details) demonstrate successful stream learning using the weight transfer function, confirming its utility in both ECRAM-MTT unit and RRAM-MTT unit.

For low-precision quantization, ternary quantization possesses better expressive abilities than binary precision counterparts, and its basic function is:

$$W_T = \begin{cases} +1, & W_h > \Delta \\ 0, & |W_h| \le \Delta \\ -1, & W_h < -\Delta \end{cases} \tag{9}$$

where Δ is the positive threshold parameter, and W_T and W_h are ternary inference weights and real-valued hidden weights, respectively. The zero-value region is an important feature in ternary quantization, whereas it does not exist in binarization weight transfer in MTT units. However, by biasing the MTT unit at sub-threshold regions, as previously predicted in Figure S12 where W_b will always be about 0 at a small value of W_h, it is possible to serve as a physical ternary weight transfer function. By controlling the hidden weight reading bias voltages, which usually are smaller than V_T of the inference transistor, ternary weight quantization function can be well fitted, as experimental tested in Figure 7g and Figure S35a, where the relation between transferred ECRAM conductance G_e' and original ECRAM conductance G_e is:

$$G_e' = \begin{cases} G_e, & I_{ts} \ge 0 \\ -G_e, & I_{ts} < 0 \end{cases} \tag{10}$$

Here, I_{ts} is the channel current of the inference transistor measured at the source terminal, and the fitting model is based on:

$$W_T = \begin{cases} 2 \times \left(\dfrac{A_p \times (W_h - r) + 1}{B_p \times (W_h - r) + 2} - 0.5 \right), & W_h > r \\ 0, \quad |W_h| \le r \\ -2 \times \left(\dfrac{A_n \times (W_h + r) + 1}{B_n \times (W_h + r) + 2} - 0.5 \right), & W_h < -r \end{cases} \tag{11}$$

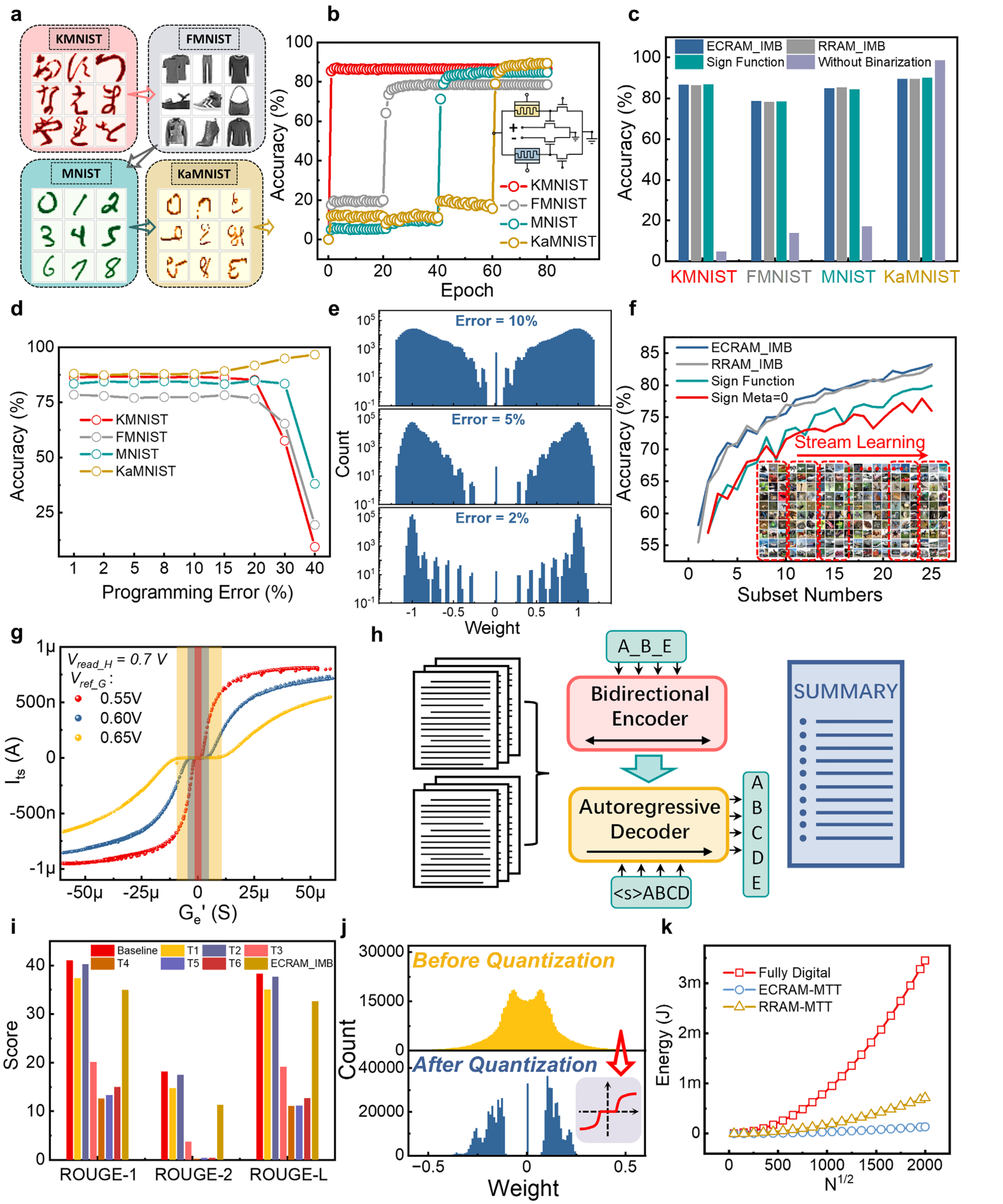

FIGURE 7 | Large network applications and validation of in situ weight quantization relationships based on MTT units. a) Continual learning datasets for simulation validation, with each image in the different datasets sized at 28×28×1, and the datasets being independent of one another. b) Training results of continual learning for the datasets in panel (a), where the learning sequence is KMNIST-FMNIST-MNIST-KaMNIST, employing a weight binary quantization function based on ECRAM MTT units. c) Statistical results of the final training accuracy across different datasets using various binary weight conversion functions. d) Network training performance is based on different relative programming error levels, where the network performance is almost unaffected when the error level is below 20%. e) The final distribution of hidden weights at the first layer of the network after training under different programming errors. f) Stream learning performances based on different binary weight conversion functions, with the dataset using Cifar-10 and the network model being VGG-7. g) Weight ternary quantization relationships based on MTT units, where the hidden weight bias voltage is set at 0.7 V, and the gate bias voltages of the reference transistor are set at 0.55, 0.60, and 0.65 V, respectively. h) Schematic diagram of the

where r is the threshold parameter or the radius of zero regions, and as for other parameters, the previous model can be referred to. Fitting parameters of the ternary weight transfer function in the different bias conditions with corresponding notes are listed in Table S5. For the network-level verification of the ternary weight transfer function in the sub-threshold ECRAM-MTT unit, BART, a kind of LLM featuring a bidirectional encoder similar to BERT and a left-to-right decoder akin to GPT, whose low-bit version are used in resolving text summarization tasks [62], is employed to test (Figure 7h). The statistics-based quantization algorithm in Liu's work [62] is adopted, including ternary and binary version of the BART model. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is employed to evaluate automatic text summarization, which is based on comparing the overlap between generated text (e.g., abstracts) and reference text (e.g., human-written abstracts). The variants include ROUGE-1, ROUGE-2 and ROUGE-L, and the higher the scores are, the better the model performance is. When the model input and activation are under the relevant quantization, the LLM performances of the different weight transfer functions in MTT units are summarized in Figure 7i, including both ternary and binary quantization. T2-bias based ternary weight transfer function corresponds to the most outstanding performance among all the tested conditions, where hidden weight bias voltage is 0.7 V and reference gate bias voltage is 0.6 V. Compared with baseline in software, final results only lag behind by 0.80, 0.75, and 0.67 at ROUGE-1, ROUGE-2, and ROUGE-L, respectively. The weights before and after T2 ternary quantization are partly demonstrated in Figure 7j, in which the zero-region feature is reflected in weight transfer. In addition, when the model input and activation functions are both in 8-bit float format, as tested in Figure S35b, ECRAM-MTT-based binarization is the best choice for quantization. Thus, the physical quantization scheme should be adjusted according to the specific neural network.

In the energy estimation during the weight quantization and saving process, for the digital hardware platform, A100 GPU is adopted as the processor with Samsung 980 PRO NVMe solid-state drive as the nonvolatile memory for storage of analog weights and binary weights. The evaluation of the MTT-unit-based array is presented in Note S1, which is mainly based upon the experimentally measured device data, including the average number of Write-Verify times and the average power consumption of programming and reading once. Figure 7k shows the energy comparison in the quantization computation and storage process based on different hardware architectures. As the array scale (N) increases, the MTT unit saves more energy in the weight quantization process. When the array scale is 4Mb, the ECRAM-MTT and RRAM-MTT platforms save 25.51× and 4.84× of energy consumption, respectively. To evaluate the performance of the MTT unit in the synaptic unit function, it is compared with some related works based on hybrid synaptic units, mainly from the aspects of unit device type, structural complexity, programming characteristics and network computing characteristics. As shown in Table 1, this work only requires 6 components with significant reduction in unit complexity [25, 38, 40, 63, 64]. It supports in situ weight conversion without additional secondary programming, supports mixed precision calculations, possesses scalable unit structure, and is able to be applied to RRAM memory types with a wider range of application scenarios, including BNN-based continual learning and low-precision LLM.

3 | Conclusion

In this work, for in situ accelerating low-precision quantization computations of edge networks, we have developed an in situ quantization hardware unit based on ionic non-volatile memory-transistor coupling integration. By in-situ quantization of analog weights, the traditional in-memory computing paradigm has been expanded. Our research focuses on revealing the physical mechanism of linear conductance regulation of ECRAM devices, promoting the precise mapping of network analog weights in online training, and clarifying its stable conductance regulation mechanism through a combination of experimental characterization and theoretical modeling. By establishing a theoretical model for device parameter optimization, we systematically study the impact of bias conditions on the in situ quantization function of weights. Moreover, this research successfully extends the technical solution to the RRAM system, and based on its comprehensive performance optimization of switching ratio tolerance, it can also replace the ideal quantization function in software. Experiments show that the classification accuracy of the quantization function based on the electrical characteristics of the unit structure in the binary neural network is consistent with the ideal function, which can support low-precision continual learning to overcome the catastrophic forgetting problem of traditional neural networks and binary/ternary large language model calculations as well. At the 4 Mb array scale, during the computation and saving of the quantization process, the quantization units based on ECRAM and RRAM achieve 25.51 and 4.84 times energy consumption reductions compared to traditional digital platforms, respectively, providing a more energy-efficient in situ quantization calculation solution for low-precision network training at the edge.

4 | Experimental Section

4.1 | ECRAM Array Fabrication

The fabrication of the ECRAM array utilized silicon substrates featuring 300 nm of thermally grown silicon dioxide. The process began with electron-beam lithography to create patterns for the source-drain contacts, after which 5 nm of Ti and 25 nm of Au were deposited using electron-beam evaporation. The contact electrodes, essential for monitoring changes in channel conductance, were finalized through a lift-off procedure. Following this, a second round of electron-beam lithography was

BART text summarization model application, which primarily consists of a bidirectional encoder and an autoregressive decoder. i) LLM learning results of the weight conversion relationships of MTT units under different bias conditions, where both the model input and activation functions undergo relevant quantization. j) Distribution diagrams before and after quantization of some weights based on the T-2 weight conversion relationship. k) Energy consumption comparison between in-place quantization based on MTT units and weight conversion on traditional CMOS digital platforms.

employed to define the channel area, where 60 nm of tungsten trioxide (WO_x) was deposited by radio frequency magnetron sputtering, acting as the ECRAM channel. After the lift-off, the electrolyte region was patterned using a third round of electron-beam lithography, during which 120 nm of LiPON was deposited from a Li_3PO_4 target using radio frequency magnetron sputtering under a nitrogen flow of 10 sccm and at a power setting of 150 W. Subsequently, 40 nm of SiO_x was deposited to shield the electrolyte from environmental moisture. To isolate the different wires, 40 nm of SiO_x was used after the fourth round of electron-beam lithography. Finally, a fifth round of electron-beam lithography was utilized to pattern test pads, which were then deposited with 10 nm of Ti and 220 nm of Au via electron-beam evaporation.

4.2 | Electrical Measurements

All the electrical measurements were performed using an Agilent B1500A semiconductor analyzer and the electrode pad lead-out was realized by the TS3000 Probe System. Except for device size exploration, the devices size used in this paper are all kept at 10 µm × 10 µm.

4.3 | Material Characterization

To analyze the material components of fabricated devices, the TEM samples were prepared first by the focus ion beam technique (Helios G5 UX). After that, TEM testing as well as EDS imaging were performed on Talos F200X G2 systems. TOF-SIMS tests of WO_x-based ECRAM were conducted with a PHI nano TOF II (ULVAC-PHI Inc., Japan). The sputter etching was performed using an Ar^+ beam (3 kV 100 nA) to obtain the depth profile of tested samples for different resistance states.

4.4 | Transistor Characteristics Used in Model Prediction

For a nMOSFET, while $V_{gs} \geq V_T$, if $V_{ds} \ll V_{gs} - V_{th}$, the drain current I_{ds} can be written as:

$$I_{ds} = \mu_{eff} \times C_{ox} \times \frac{W}{L} \times \left(V_{gs} - V_T\right) \times V_{ds} + bias \qquad (12)$$

where V_T is the threshold voltage of the transistor, V_{gs} and V_{ds} are the gate-source voltage drop and drain-source voltage drop, respectively, μ_{eff} is the effective carrier mobility, C_{ox} is the oxide capacitance per unit area, W and L are the width and length of the transistor, respectively, and the *bias* is the compensation. When $V_{gs} < V_T$, the transistor will be in the subthreshold region, the channel current I_{ds} can be written as:

$$I_{ds} = \mu_{eff} \times C_{ox} \times \frac{W}{L} \times (m-1) \times T'^{2}$$
$$\times exp\left(\frac{V_{gs} - V_T}{m \times T'}\right) \times \left(1 - \exp\left(-\frac{V_{ds}}{T'}\right)\right)$$

Here, $T' = \frac{k \times T}{q}$, where k is the Boltzmann constant, T is the absolute temperature, q is the charge of electrons, and m is

TABLE 1 | Evaluation of hybrid synaptic unit for separate training and inference.

Training device (TD)	Inference device (ID)	Unit complexity	Bit precision of TD	Nonvolatility of TD	Programming-free ID	In situ weight transfer	In situ inference	Mixed-precision unit	Application	Refs.
Capacitor	PCM	2×3T1C+2T2P	>8	No	No	Yes	Yes	No	Transfer Learning	[63]
RRAM	Transistor	2×2T1R	1	Yes	Yes	Yes	Yes	No	Transpose, Margin Enhancement	[25]
FeFET	FeFET	2×2T1D-FeFET	8	Yes	No	Yes	Yes	No	Transfer Learning	[64]
Digital	RRAM	FPGA+2×1T1R	Floating-point	No	No	No	Yes	Yes	BNN	[40]
FeRAM	RRAM	8T8C+1T1R	3	Yes	No	Yes	No	Yes	BNN	[38]
ECRAM	**Transistor**	**2×2T1E**	**5**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**BNN, TNN-LLM**	**This work**

Note: For a more fair comparison, unit complexity is based on the differential structure in CIM.

the subthreshold slope factor. The above parameter values are concluded in Table S1.

4.5 | Neural Network Simulations in MTT Model Derivation

The metaplasticity-inspired BNN is adopted to test the weight transfer functions in the MTT model, where the main body of this algorithm is the BNN, whose weight values and neuron activations are limited to be 1 or –1. The distinctions between this approach and a traditional binary neural network are in the training process, where the updated gradients must be multiplied by a nonlinear function, which is associated with the values of the hidden weights and is influenced by a hyperparameter m, which will be fixed at 12. The tested image sizes are all set as 28×28; the network structure is kept at 784×1000×500×10; the learning rate is set at 0.005; the hidden weights are uniformly quantized into 3 bits with quantization boundary of ±1.

4.6 | Quantized Hidden Weights and Binary Weights from Physical Electrical Signals

By taking the weight transfer function into neural network simulation, the relationship between quantized hidden weights W_h and ECRAM conductance G_e is:

$$W_h = \begin{cases} \dfrac{G_e - G_{e,min}}{G_{e,max} - G_{e,min}}, \; I_{ts} \geq 0 \\ \dfrac{-G_e + G_{e,min}}{G_{e,max} - G_{e,min}}, \; I_{ts} < 0 \end{cases} \tag{14}$$

where I_{ts} is the channel current of the inference transistor, $G_{e,max}$ and $G_{e,min}$ are maximum and minimum ECRAM conductance during programming, respectively. The relationship between quantized binary weights W_b and channel current of the inference transistor I_{ts} is:

$$W_b = \frac{I_{ts}}{I_{ts,max}} \tag{15}$$

where $I_{ts,max}$ is the channel reading current when the gate is biased at V_{read_H}, which is also the maximum channel current at infinite ECRAM conductance.

4.7 | Weight Transfer Function Used in Neural Network Training

During all the network simulations unless otherwise specified, the MTT transfer function in ECRAM-MTT units is selected when the maximum operating conductance is 50 μS, and the specific function is:

$$W_b = \begin{cases} 2 \times \left(\dfrac{41.36 \times W_h + 1}{41.62 \times W_h + 2} - 0.5 \right), \; W_h \geq 0 \\ -2 \times \left(\dfrac{-44.67 \times W_h + 1}{-44.45 \times W_h + 2} - 0.5 \right), \; W_h < 0 \end{cases} \tag{16}$$

While for RRAM-based MTT units, the weight transfer function is:

$$W_b = \begin{cases} 2 \times \left(\dfrac{63.61 \times W_h + 1}{63.67 \times W_h + 2} - 0.5 \right), \; W_h \geq 0 \\ -2 \times \left(\dfrac{-70.03 \times W_h + 1}{-67.44 \times W_h + 2} - 0.5 \right), \; W_h < 0 \end{cases} \tag{17}$$

4.8 | ECRAM- or RRAM-Based IMB for Edge Neural Network Simulation

For testing IMB in MTT units in edge light-weight neural networks, two kinds of common models, FCNN and CNN, are employed. As for FCNN, Fashion-MNIST is chosen for classification, with neural network size of 784×1000×500×10 and learning rate of 0.005. For CNN simulations, the binary VGG-7 with six convolution layers of 128-128-256-256-512-512 filters and kernel sizes of 3 is employed, and the learning rate is 0.0005. Both neural network simulations of FCNN and CNN are run for 10 rounds, the average final accuracies and variations are shown in Table S3.

4.9 | Neural Network Simulation for Continually Learning Small Tasks

For neural network simulation on the continual learning of different small tasks, neural network structure is set as 2×3×1; learning rate is 0.005; meta-parameter m is 10; the quantized precision and boundary of hidden weights are 3 bits and ±1, respectively.

4.10 | Stream Learning

Basic neural network structures are the same as edge neural network simulations. In the FCNN stream learning, the Fashion-MNIST dataset is split into 40 subsets, and each subset is learned for 10 epochs, with meta parameter m set as 5. In the CNN stream learning, the Cifar-10 dataset is split into 25 subsets, and each subset is learned for 20 epochs, with meta parameter m set as 10.

4.11 | DFT Calculation

The structural properties of the material before and after Li atom doping were investigated through density functional theory (DFT) calculations using the Vienna Ab initio Simulation Package (VASP) [65, 66]. The projector augmented wave (PAW) plane-wave basis set with the Perdew–Burke–Ernzerhof (PBE) [67] exchange-correlation functional was employed in the computational framework. To account for the strong correlation effects of d electrons in transition metal elements, DFT+U [68] correction was implemented with Hubbard U values of 4.0 eV for Nb and 6.2 eV for W, respectively. Prior to electronic property calculations, structural relaxation was performed until the atomic forces converged below 0.02 eV/Å. In addition, the migration energy barriers of Li ions within different materials were determined using the climbing image nudged elastic band (CI-NEB) method [69]. The amorphous structures were generated via melt-quench molecular dynamics simulations starting from their crystalline phases. The amorphous WO_x structure was generated using a 128-atom WO_3 supercell, while the amorphous NbO_x structure was generated using a 168-atom Nb_2O_5 supercell.

Acknowledgements

This work was supported by the National Key R&D Program of China (2023YFB4502200, 2023YFA1407200), the Guangdong Provincial Key Laboratory of In-Memory Computing Chips (2024B1212020002), the Shenzhen Science and Technology Program (JCYJ20241202125907011), the National Natural Science Foundation of China (8206100486, 92164302), the Beijing Natural Science Foundation (L234026, L257010), Financial Support for Outstanding scientific and technological innovation Talents Training Fund in Shenzhen, and the New Cornerstone Science Foundation.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

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

References

1. M. Kraeter, S. Abuhattum, D. Soteriou, et al., "AIDeveloper: Deep Learning Image Classification in Life Science and Beyond," *Advanced Science* 8 (2021): 2003743, https://doi.org/10.1002/advs.202003743.

2. S. Shahali, M. Murshed, L. Spencer, et al., "Morphology Classification of Live Unstained Human Sperm Using Ensemble Deep Learning," *Advanced Intelligent Systems* 6 (2024): 2400141, https://doi.org/10.1002/aisy.202400141.

3. K. He, X. Zhang, S. Ren, and J. Sun, 2016 IEEE Conference on Computer Vision and Pattern Recognition CVPR, (IEEE, 2016), 770–778, https://doi.org/10.1109/CVPR.2016.90.

4. D. Yigci, M. Eryilmaz, A. K. Yetisen, S. Tasoglu, and A. Ozcan, "Large Language Model-Based Chatbots in Higher Education," *Advanced Intelligent Systems* 7 (2025): 2400429, https://doi.org/10.1002/aisy.202400429.

5. Z. Wei, Y. Chen, Q. Zhao, et al., "Implicit Perception of Differences Between NLP-Produced and Human-Produced Language in the Mentalizing Network," *Advanced Science* 10 (2023): 2203990, https://doi.org/10.1002/advs.202203990.

6. H. Li, "Deep Learning for Natural Language Processing: Advantages and Challenges," *National Science Review* 5 (2018): 24–26, https://doi.org/10.1093/nsr/nwx110.

7. H. Ding, Z. Zeng, Z. Wang, et al., "Deep Learning-Enabled MXene/PEDOT:PSS Acoustic Sensor for Speech Recognition and Skin-Vibration Detection," *Advanced Intelligent Systems* 4 (2022): 2200140, https://doi.org/10.1002/aisy.202200140.

8. Y. Liu, H. Li, X. Liang, et al., "Speech Recognition Using Intelligent Piezoresistive Sensor Based on Polystyrene Sphere Microstructures," *Advanced Intelligent Systems* 5 (2023): 2200427, https://doi.org/10.1002/aisy.202200427.

9. Y. Yang, M. A. Bevan, and B. Li, "Hierarchical Planning With Deep Reinforcement Learning for 3D Navigation of Microrobots in Blood Vessels," *Advanced Intelligent Systems* 4 (2022): 2200168, https://doi.org/10.1002/aisy.202200168.

10. J. Yao, Q. Cao, Y. Ju, et al., "Adaptive Actuation of Magnetic Soft Robots Using Deep Reinforcement Learning," *Advanced Intelligent Systems* 5 (2023): 2200339, https://doi.org/10.1002/aisy.202200339.

11. M. Zhao, G. Wang, Q. Fu, X. Guo, and T. Li, "Deep Reinforcement Learning-Based Air Defense Decision-Making Using Potential Games," *Advanced Intelligent Systems* 5 (2023): 2300151, https://doi.org/10.1002/aisy.202300151.

12. K. He, X. Zhang, S. Ren, and J. Sun, 2015 IEEE International Conference on Computer Vision ICCV, (IEEE, 2015), 1026–1034, https://doi.org/10.1109/ICCV.2015.123.

13. H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things With Edge Computing," *IEEE Network* 32 (2018): 96–101, https://doi.org/10.1109/MNET.2018.1700202.

14. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal* 3 (2016): 637–646, https://doi.org/10.1109/JIOT.2016.2579198.

15. F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey," *IEEE Access* 8 (2020): 58322–58336, https://doi.org/10.1109/ACCESS.2020.2982411.

16. A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer, "AI and Memory Wall," *IEEE Micro* 44 (2024): 33–39.

17. K. Sun, J. Chen, and X. Yan, "The Future of Memristors: Materials Engineering and Neural Networks," *Advanced Functional Materials* 31 (2021): 2006773, https://doi.org/10.1002/adfm.202006773.

18. A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory Devices and Applications for in-Memory Computing," *Nature Nanotechnology* 15 (2020): 529–544, https://doi.org/10.1038/s41565-020-0655-z.

19. Z. Wang, H. Wu, G. W. Burr, et al., "Resistive Switching Materials for Information Processing," *Nature Reviews Materials* 5 (2020): 173–195, https://doi.org/10.1038/s41578-019-0159-3.

20. W. Haensch, A. Raghunathan, K. Roy, et al., "Compute in-Memory With Non-Volatile Elements for Neural Networks: A Review From a Co-Design Perspective," *Advanced Materials* 35 (2023): 2204944, https://doi.org/10.1002/adma.202204944.

21. W. Song, M. Rao, Y. Li, et al., "Programming Memristor Arrays With Arbitrarily High Precision for Analog Computing," *Science* 383 (2024): 903–910, https://doi.org/10.1126/science.adi9405.

22. J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proceedings of the IEEE* 107 (2019): 1655–1674, https://doi.org/10.1109/JPROC.2019.2921977.

23. J. Kim, S. Lee, S. Kim, et al., "Synaptic Characteristics and Vector-Matrix Multiplication Operation in Highly Uniform and Cost-Effective Four-Layer Vertical RRAM Array," *Advanced Functional Materials* 34 (2024): 2310193, https://doi.org/10.1002/adfm.202310193.

24. B. Gao, H. W. Zhang, and S. Yu, 2009 Symposium on VLSI Technology (IEEE, 2009), 30–31.

25. L. Wang, W. Ye, and J. Lai, 2021 Symposium on VLSI Technology, (IEEE, 2021), 1–2.

26. J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, 2021 IEEE Custom Integrated Circuits Conference CICC, (2021), 1–2.

27. S.-O. Park, S. Hong, S.-J. Sung, et al., "Phase-Change Memory via a Phase-Changeable Self-Confined Nano-Filament," *Nature* 628 (2024): 293–298, https://doi.org/10.1038/s41586-024-07230-5.

28. M. Le Gallo, R. Khaddam-Aljameh, M. Stanisavljevic, et al., "A 64-Core Mixed-Signal in-Memory Compute Chip Based on Phase-Change Memory for Deep Neural Network Inference," *Nature Electronics* 6 (2023): 680–693, https://doi.org/10.1038/s41928-023-01010-1.

29. Y. Cheng, Q. Yang, J. Wang, et al., "Highly Tunable β-Relaxation Enables the Tailoring of Crystallization in Phase-Change Materials," *Nature Communications* 13 (2022): 7352, https://doi.org/10.1038/s41467-022-35005-x.

30. D. Zhang, Y. Hou, L. Zeng, and W. Zhao, "Hardware Acceleration Implementation of Sparse Coding Algorithm With Spintronic Devices,"

IEEE Transactions on Nanotechnology 18 (2019): 518–531, https://doi.org/10.1109/TNANO.2019.2916149.

31. H. Lv, J. Fidalgo, A. V. Silva, et al., “Multi-Level Switching and Reversible Current Driven Domain-Wall Motion in Single CoFeB/MgO/CoFeB-Based Perpendicular Magnetic Tunnel Junctions,” *Advanced Electronic Materials* 7 (2021): 2000976, https://doi.org/10.1002/aelm.202000976.

32. C. Ma, L. Tao, Z. Luo, et al., “Efficient Parallel Multi-Bit Logic-in-Memory Based on a Ultrafast Ferroelectric Tunnel Junction Memristor,” *Adv Electron Mater* 7 (2021): 2000988.

33. W. Kho, H. Hwang, T. Noh, H. Kim, J. M. Lee, and S.-E. Ahn, “Maximizing the Synaptic Efficiency of Ferroelectric Tunnel Junction Devices Using a Switching Mechanism Hidden in an Identical Pulse Programming Learning Scheme,” *Advanced Intelligent Systems* 6 (2024): 2400211, https://doi.org/10.1002/aisy.202400211.

34. W. Mu, C. Ke, C. Huangfu, et al., “Room-Temperature Out-of-Plane Ferroelectricity in 1T′/1H MoS_2 Heterophase Bilayer,” *Advanced Materials* 37 (2025): 2504941, https://doi.org/10.1002/adma.202504941.

35. D. Sharma, S. P. Rath, B. Kundu, et al., “Linear Symmetric Self-selecting 14-Bit Kinetic Molecular Memristors,” *Nature* 633 (2024): 560–566, https://doi.org/10.1038/s41586-024-07902-2.

36. S. Yu, Z. Li, and P.-Y. Chen, 2016 IEEE International Electron Devices Meeting IEDM, (IEEE, 2016), 16–24.

37. M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, Computer Vision—ECCV 2016, ed. B. Leibe, J. Matas, N. Sebe, and M. Welling, (Springer International Publishing, 2016), 525–542.

38. M. Martemucci, F. Rummens, T. Hirtzlin, et al., “Hybrid FeRAM/RRAM Synaptic Circuit Enabling On-Chip Inference and Learning at the Edge,” in *2023 Int. Electron Devices Meet* (IEDM, 2023), 1–4.

39. Z. Sun, S. Kvatinsky, X. Si, A. Mehonic, Y. Cai, and R. Huang, “A Full Spectrum of Computing-in-Memory Technologies,” *Nature Electronics* 6 (2023): 823–835, https://doi.org/10.1038/s41928-023-01053-4.

40. Y. Li, W. Zhang, X. Xu, et al., “Mixed-Precision Continual Learning Based on Computational Resistance Random Access Memory,” *Advanced Intelligent Systems* 4 (2022): 2200026, https://doi.org/10.1002/aisy.202200026.

41. W. Zhang, B. Gao, J. Tang, et al., “Neuro-Inspired Computing Chips,” *Nature Electronics* 3 (2020): 371–382, https://doi.org/10.1038/s41928-020-0435-7.

42. M. Huang, M. Schwacke, M. Onen, J. del Alamo, J. Li, and B. Yildiz, “Electrochemical Ionic Synapses: Progress and Perspectives,” *Advanced Materials* 35 (2023): 2205169, https://doi.org/10.1002/adma.202205169.

43. A. Sood, A. D. Poletayev, D. A. Cogswell, et al., “Electrochemical Ion Insertion From the Atomic to the Device Scale,” *Nature Reviews Materials* 6 (2021): 847–867, https://doi.org/10.1038/s41578-021-00314-y.

44. Y. Li, E. J. Fuller, J. D. Sugar, et al., “Filament-Free Bulk Resistive Memory Enables Deterministic Analogue Switching,” *Advanced Materials* 32 (2020): 2003984, https://doi.org/10.1002/adma.202003984.

45. J. Cui, F. An, J. Qian, et al., “CMOS-Compatible Electrochemical Synaptic Transistor Arrays for Deep Learning Accelerators,” *Nature Electronics* 6 (2023): 292–300, https://doi.org/10.1038/s41928-023-00939-7.

46. H. Kwak, N. Kim, S. Jeon, S. Kim, and J. Woo, “Electrochemical Random-Access Memory: Recent Advances in Materials, Devices, and Systems Towards Neuromorphic Computing,” *Nano Convergence* 11 (2024): 9, https://doi.org/10.1186/s40580-024-00415-8.

47. H. Lee, D. G. Ryu, G. Lee, et al., “Vertical Metal-Oxide Electrochemical Memory for High-Density Synaptic Array Based High-Performance Neuromorphic Computing,” *Advanced Electronic Materials* 8 (2022): 2200378, https://doi.org/10.1002/aelm.202200378.

48. M. Onen, N. Emond, B. Wang, et al., “Nanosecond Protonic Programmable Resistors for Analog Deep Learning,” *Science* 377 (2022): 539–543, https://doi.org/10.1126/science.abp8064.

49. Y. Li, J. Lu, D. Shang, et al., “Oxide-Based Electrolyte-Gated Transistors for Spatiotemporal Information Processing,” *Advanced Materials* 32 (2020): 2003018, https://doi.org/10.1002/adma.202003018.

50. Y. Li, Z. Xuan, J. Lu, et al., “One Transistor One Electrolyte-Gated Transistor Based Spiking Neural Network for Power-Efficient Neuromorphic Computing System,” *Advanced Functional Materials* 31 (2021): 2100042, https://doi.org/10.1002/adfm.202100042.

51. P.-Y. Chen, X. Peng, and S. Yu, “NeuroSim: A Circuit-Level Macro Model for Benchmarking Neuro-Inspired Architectures in Online Learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37 (2018): 3067–3080, https://doi.org/10.1109/TCAD.2018.2789723.

52. A. Laborieux, M. Ernoult, T. Hirtzlin, and D. Querlioz, “Synaptic Metaplasticity in Binarized Neural Networks,” *Nature Communications* 12 (2021): 2549, https://doi.org/10.1038/s41467-021-22768-y.

53. K. Zhu, S. Pazos, F. Aguirre, et al., “Hybrid 2D–CMOS Microchips for Memristive Applications,” *Nature* 618 (2023): 57–62, https://doi.org/10.1038/s41586-023-05973-1.

54. P. Yao, H. Wu, B. Gao, et al., “Fully Hardware-Implemented Memristor Convolutional Neural Network,” *Nature* 577 (2020): 641–646, https://doi.org/10.1038/s41586-020-1942-4.

55. W. Zhang, P. Yao, B. Gao, et al., “Edge Learning Using a Fully Integrated Neuro-Inspired Memristor Chip,” *Science* 381 (2023): 1205–1211, https://doi.org/10.1126/science.ade3483.

56. S. Chen, Z. Yang, H. Hartmann, A. Besmehn, Y. Yang, and I. Valov, “Electrochemical Ohmic Memristors for Continual Learning,” *Nature Communications* 16 (2025): 2348, https://doi.org/10.1038/s41467-025-57543-w.

57. Y. Lu, Z. Yang, Y. Tao, et al., “Energy-Efficient Online Training With In Situ Parallel Computing on Electrochemical Memory Arrays,” *Adv Intell Syst* 7 (2025): 2401068, https://doi.org/10.1002/aisy.202401068.

58. A. Soltoggio, E. Ben-Iwhiwhu, V. Braverman, et al., “A Collective AI via Lifelong Learning and Sharing at the Edge,” *Nature Machine Intelligence* 6 (2024): 251–264, https://doi.org/10.1038/s42256-024-00800-2.

59. S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, “Loss of Plasticity in Deep Continual Learning,” *Nature* 632 (2024): 768–774, https://doi.org/10.1038/s41586-024-07711-7.

60. T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, “Deep Learning for Classical Japanese Literature,” *Computer Vision and Pattern Recognition* (2018), https://doi.org/10.48550/arXiv.1812.01718.

61. V. U. Prabhu, “Kannada-MNIST: A New Handwritten Digits Dataset for the Kannada Language,” *Computer Vision and Pattern Recognition* (2019), https://doi.org/10.48550/arXiv.1908.01242.

62. Z. Liu, B. Oguz, A. Pappu, Y. Shi, and R. Krishnamoorthi, “Binary and Ternary Natural Language Generation,” *Computation and Language* (2023), https://doi.org/10.48550/arXiv.2306.01841.

63. S. Ambrogio, P. Narayanan, H. Tsai, et al., “Equivalent-Accuracy Accelerated Neural-Network Training Using Analogue Memory,” *Nature* 558 (2018): 60–67, https://doi.org/10.1038/s41586-018-0180-5.

64. H. Ning, Z. Yu, Q. Zhang, et al., “An in-Memory Computing Architecture Based on a Duplex Two-Dimensional Material Structure for In Situ Machine Learning,” *Nature Nanotechnology* 18 (2023): 493–500, https://doi.org/10.1038/s41565-023-01343-0.

65. G. Kresse and J. Furthmuller, “Efficiency of Ab-Initio Total Energy Calculations for Metals and Semiconductors Using a Plane-Wave Basis Set,” *Computational Materials Science* 6 (1996): 15–50, https://doi.org/10.1016/0927-0256(96)00008-0.

66. G. Kresse and J. Furthmüller, “Efficient Iterative Schemes for Ab Initio Total-Energy Calculations Using a Plane-Wave Basis Set,” *Physical Review B* 54 (1996): 11169, https://doi.org/10.1103/PhysRevB.54.11169.

67. J. P. Perdew, K. Burke, and M. Ernzerhof, “Generalized Gradient Approximation Made Simple,” *Physical Review Letters* 77 (1996): 3865, https://doi.org/10.1103/PhysRevLett.77.3865.

68. S. L. Dudarev, G. A. Botton, S. Y. Savrasov, C. J. Humphreys, and A. P. Sutton, “Electron-Energy-Loss Spectra and the Structural Stability of Nickel Oxide: An LSDA+U Study,” *Physical Review B* 57 (1998): 1505, https://doi.org/10.1103/PhysRevB.57.1505.

69. G. Henkelman, B. P. Uberuaga, and H. Jónsson, “A Climbing Image Nudged Elastic Band Method for Finding Saddle Points and Minimum Energy Paths,” *The Journal of Chemical Physics* 113 (2000): 9901–9904, https://doi.org/10.1063/1.1329672.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.

Supporting file: advs73957-sup-0001-SuppMat.docx.