An 8Mb Learning-Aware RRAM Compute-in-Memory Accelerator for Embodied Self-Supervised Learning

Longhao Yan [1#], Zhe Zhan[1#], Zelun Pan[1], Bowen Wang[1], Yaoyu Tao[1,3], and Yuchao Yang[1,2,3,4*]
[1]Beijing Advanced Innovation Center for Integrated Circuits, School of Integrated Circuits, Peking University, Beijing, China.
[2]Guangdong Provincial Key Laboratory of In-Memory Computing Chips, School of Electronic and Computer Engineering, Peking University, Shenzhen, China. [3]Center for Brain Inspired Chips, Institute for Artificial Intelligence, Peking University, Beijing, China.
[4]Center for Brain Inspired Intelligence, Chinese Institute for Brain Research (CIBR), Beijing, China.
[#]Equally contributed authors. Email: yuchaoyang@pku.edu.cn,

2025 IEEE International Electron Devices Meeting (IEDM) | 979-8-3315-6785-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/IEDM50572.2025.11353856

Abstract—Embodied Self-Supervised Learning (E-SSL) enables intelligent agents to autonomously adapt to changing environments without human annotations, a key requirement for real-time, edge-based autonomy. However, conventional computing architectures suffer from energy and latency inefficiencies due to frequent data movement, impeding on-device learning. To address this, we present an 8Mb learning-aware RRAM-based Compute-in-Memory (CIM) accelerator tailored for E-SSL. Fabricated in a 40nm process, the CIM chip introduces two significant innovations: a Two-Stage Analog Weight Programming (TSAWP) unit for fast, precise, and relaxation-mitigated device conductance programming, and a Neural-Optimized Gradient-Aware Programming Scheduler (NoGAPS) for lifetime-aware, adaptive programming. The proposed chip achieves 10.2× improvements in programming speed and 11.7× system lifetime improvement compared to conventional schemes. Deployed on a quadruped robot for terrain adaptation, the system demonstrates 347× higher energy efficiency and an 8.7× reduction in latency compared to GPU baselines, showcasing robust, low-power, real-time online learning in dynamic edge scenarios.

I. Introduction

Embodied self-supervised learning empowers intelligent agents to continuously adapt to dynamic environments using self-generated feedback. By eliminating the need for human annotations, E-SSL is a crucial enabler of long-term autonomy in edge scenarios [1]. However, conventional von Neumann architectures hinder efficient hardware implementation of E-SSL due to excessive data movement, causing high energy consumption and latency unsuitable for power-constrained, real-time edge scenarios. RRAM-based CIM architectures offer a compelling alternative by performing computation directly where data is stored, dramatically reducing data movement. With intrinsic parallelism, low latency, and high storage density [2], RRAM CIM is well-aligned with the demands of E-SSL. However, several limitations have prevented its practical adoption in applications like E-SSL: (1) Slow analog conductance programming during weight updates limits continuous learning and inference capability, (2) RRAM relaxation effect leads to computational accuracy degradation, and (3) Limited device endurance restricts system lifetime under frequency weight updates [3]. These challenges pose critical barriers to real-time, on-chip learning with RRAM CIM (**Fig. 1a**).

In this work, an 8Mb learning-aware RRAM compute-in-memory accelerator is fabricated using the 40nm foundry process, featuring two key innovations: (1) Two-Stage Analog Weight Programming for rapid, precise RRAM conductance tuning and relaxation mitigation; (2) Neural-Optimized Gradient-Aware Programming Scheduler to balance weight updates, enhancing RRAM lifetime and system efficiency. These innovations enable the first demonstrated RRAM CIM accelerator tailored for E-SSL, supporting real-time model updates directly on edge hardware (**Fig. 2**). The proposed chip achieves 10.2× faster programming and over 11.7× system lifetime improvement compared to conventional schemes. Tested in a quadruped robot performing terrain adaptation, it delivers 347× higher energy efficiency and 8.7× lower latency relative to GPU baselines, demonstrating effective, robust, low-power online learning in real-world scenarios.

II. Two-Stage Analog Weight Programming Unit for Fast and Precise Weight Tuning

Fig. 3a shows the integrated TSAWP unit, which combines a self-terminating module, clamping circuit, and 12-bit SAR-ADC for on-chip analog conductance tuning with over 8-bit effective precision. The programming flow in **Fig. 3b** has two stages: (1) Self-Terminating Coarse-Grained Programming (STCP): TSAWP monitors the programming current and automatically stops the pulse to adjust conductance to a coarse range quickly. (2) Low-Voltage Fine-Grained Programming (LVFP) then fine-tunes it to the target value G_{final} using low-voltage pulses. Conventional RRAM programming faces trade-offs (**Fig. 4a**). One-Shot programming offers high speed by utilizing the access transistor for compliance, but suffers from low precision due to device variation. Write-Verify (WV) enhances precision using read-verify loops at the cost of speed. To address these issues, we propose STCP, which mirrors the programming current and compares it to a reference, terminating the pulse once the target is reached (**Fig. 4**). This approach combines One-Shot speed with near-WV accuracy. As shown in **Fig. 4b**, STCP reduces conductance variation by ~75% vs. One-Shot. STCP enables fast and accurate coarse tuning and simplifies the subsequent fine-tuning. As shown in **Fig. 5a-b**, in the WV method, the total pulse duration required to program an RRAM device increases with the conductance change $\Delta G = G_{target} - G_0$. By applying STCP first, the conductance is rapidly brought close to the target, significantly reducing ΔG for LVFP (**Fig. 5b**). This results in fewer write pulses and higher efficiency of the LVFP phase. For 8-bit (256-level) programming, TSAWP achieves ~3× speedup over the conventional WV approach. The RRAM relaxation effect post-programming correlates with the conductance change during

the last pulse (ΔG_{last}), where a larger ΔG_{last} induces greater relaxation and degrades accuracy (Fig. 6b). TSAWP mitigates this effect via STCP in the first stage, constraining conductance near the target (Fig. 6a), thereby reducing ΔG_{last} before fine-tuning. The second stage, LVFP, uses low-voltage pulses to minimize the conductance change per pulse. Compared to conventional methods, TSAWP significantly lowers ΔG_{last} (**Fig. 6b**), suppressing relaxation. As depicted in **Fig. 6c**, TSAWP reduces the relative deviation (RD) at 10^3s by ~52% and delays relaxation onset by over 100×.

Fig. 7a-b shows rapid conductance convergence during STCP and the conductance fine-tuning in the LVFP stage. Each STCP range is divided into 16 fine levels, yielding 256 states. **Fig. 8a** highlights TSAWP's programming capability across RRAM cells, demonstrating accurate weight distribution and strong cell-to-cell uniformity. The mean error remains under 2% at 10^2 s post-programming (**Fig. 8b**). With 3× faster speed than conventional WV and 100× longer retention, TSAWP is well-suited for E-SSL applications.

III. Neural Optimized Gradient-Aware Programming Scheduler for RRAM-based CIM

As shown in **Fig. 9**, uneven gradient distributions in NNs create three key challenges in RRAM-based CIM: (1) Imbalanced write time: A small set of weights (~5%) dominate updates, slowing overall programming; (2) imbalanced precision: rarely updated cells accumulate relaxation errors; and (3) imbalanced endurance: overused cells wear out early, risking failure. To mitigate gradient-induced imbalance, we propose NoGAPS (Neural Optimized Gradient-Aware Programming Scheduler), comprising a Wear-Leveling Unit and an Adaptive Programming Configurator (**Fig. 10**). NoGAPS uses on-chip row/column counters to track write frequency, periodically swapping 'hot' and 'cold' rows/columns for even wear distribution. These same counters also guide the dynamic tuning of programming parameters. Infrequently written cells receive stricter settings (e.g., lower voltage, longer delays) to improve precision and suppress relaxation, while frequently updated cells use loose settings for speed. This dual strategy balances speed, precision, and endurance, boosting efficiency for RRAM-based NN inference and online learning. **Fig. 11** shows that NoGAPS reduces total write time by up to 5.1× by selectively relaxing constraints for frequently updated cells. **Fig. 12** demonstrates that NoGAPS still maintains Vector-Matrix-Multiplication (VMM) accuracy over long-term weight updates, matching TSAWP and mitigating relaxation issues seen in conventional Write-Verify. With the strategy summarized in **Fig. 13**, the combined NoGAPS+TSAWP approach achieves up to 5.1× faster programming and a 71.4% reduction in accuracy degradation compared to conventional methods. Without wear leveling, imbalanced gradient updates in neural networks cause specific RRAM cells to endure excessive write stress. As shown in **Fig. 14a**, the most frequently updated cell undergoes up to 2.6×10^4 writes—nearly 30× the average—leading to failures and degraded array functionality. NoGAPS addresses this by periodically swapping hot and cold regions to redistribute write load. This wear-leveling mechanism reduces the maximum write count by 12.4×, significantly extending device lifespan. **Fig. 14b** confirms a much more uniform write distribution, effectively mitigating endurance bottlenecks in RRAM-based inference systems.

IV. Implementation of E-SSL on Proposed Learning-Aware RRAM Compute-in-Memory Accelerator

Based on the proposed learning-aware RRAM compute-in-memory accelerator (**Fig. 15**) and test platform and application verification system (**Fig. 16**), we implemented a real-time E-SSL Traversability Estimation system (**Fig. 17**). The neural network model deployed on the RRAM CIM accelerator processes the visual input from the robot dog's onboard camera. During operation, self-supervision signals are generated to drive the learning loop, enabling the model to adapt online to complex and unseen terrains, facilitating real-time, robust navigation.

Fig. 18a validates the system's learning under dynamic situations: during indoor–outdoor transitions, the system undergoes a short adaptation phase and learns new environmental features. **Fig. 18b** shows the online adjustability of RRAM conductance during learning. **Fig. 19** analyzes the accuracy under abrupt environmental changes. Results show our RRAM-based E-SSL system can quickly recover and maintain high accuracy after environmental transitions, outperforming non-adaptive baselines. NoGAPS ensures system endurance. As shown in **Fig. 20**, Linear extrapolation indicates a >10× improvement in lifetime, alleviating lifetime concerns under intensive online learning workloads. TSAWP and NoGAPS also reduce the average programming time per learning cycle to 1/10 of that of conventional methods (**Fig. 21**), minimizing interruption during inference and enhancing system availability. Our chip can achieve a 347× improvement in energy efficiency and an 8.7× reduction in latency compared to GPU platforms (**Fig. 22**).

V. Conclusion

This work presents the first 8Mb RRAM CIM accelerator designed for efficient on-device E-SSL. To address the inefficiencies of conventional computing, the 40nm chip features two innovations: The TSAWP unit for fast and precise updates, and the NoGAPS for extending hardware lifetime, adaptive programming. The design achieves over 10× improvement in both programming speed and system lifetime compared to conventional methods. Deployed on a robot, it demonstrates 347× higher energy efficiency and an 8.7× latency reduction versus GPU baselines, enabling robust, real-time online learning.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (92164302, 8206100486 and 62404004), Guangdong Provincial Key Laboratory of In-Memory Computing Chips (2024B1212020002), Shenzhen Science and Technology Program (JCYJ20241202125907011), Beijing Natural Science Foundation (L234026, 25D40029, 25FY3314) and the 111 Project (B18001).

References

[1] G. Kahn, et al. IEEE Robot. Autom. Lett, 2021. [2] P. Yao, et al., Nature, 2020. [3] S. Yan, et al., TED, 2023. [4] J. Yang, et al., IEDM, 2024. [5] L. Yan, et al., IEDM, 2024. [6] Y. Chen, et al., IEDM, 2024. [7] W. Zhang, et al., Science, 2023.

Motivation and System Design of Learning-Aware RRAM Compute-In-Memory Accelerator

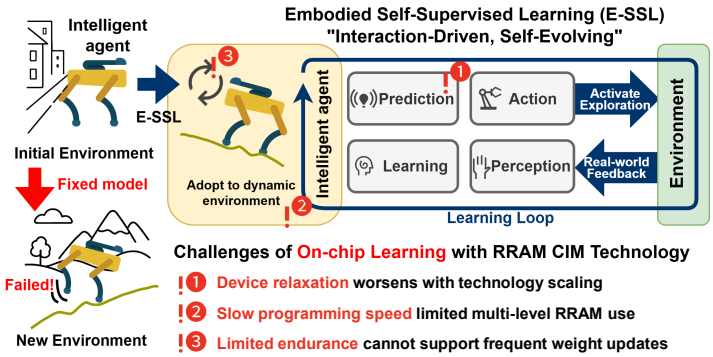

Fig. 1 RRAM-based CIM accelerators offer low power and latency for edge intelligence (e.g., robotics), but fixed-weight models lack adaptability. E-SSL enables perception-driven adaptation, yet RRAM limitations hinder its use in such applications.

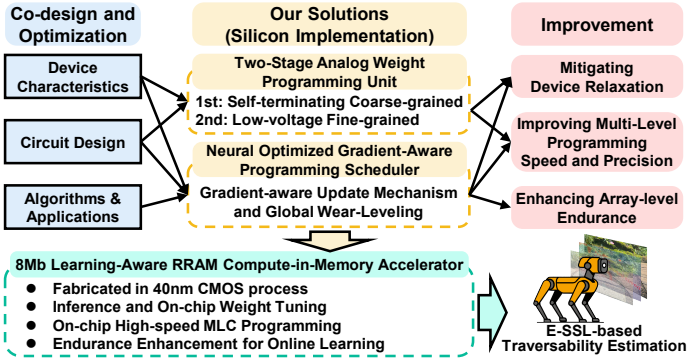

Fig. 2 Proposed learning-aware RRAM CIM accelerator for E-SSL, featuring Two-Stage Analog Weight Programming (TSAWP) and Neural-Optimized Gradient-Aware Programming Scheduler (NoGAPS), which together address key challenges in RRAM-based learning.

Two-Stage Analog Weight Programming Unit for Fast and Precise RRAM Conductance Tuning

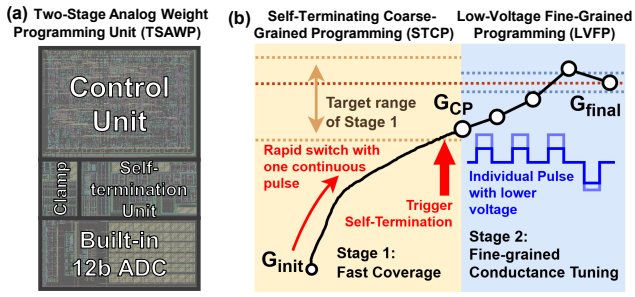

Fig. 3 (a) Layout of the proposed TSAWP Unit. (b) Operating procedure of TSAWP Unit: STCP is first used to perform a fast, coarse programming step, followed by LVFP, which carries out fine-grained, precise programming.

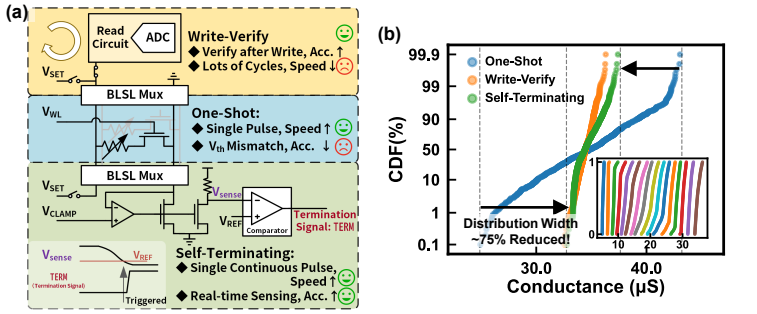

Fig. 4 (a) Comparison of conventional one-shot programming, write-verify and the proposed self-terminating mechanism. (b) Conductance distributions after programming, while self-terminating programming shows tighter distribution.

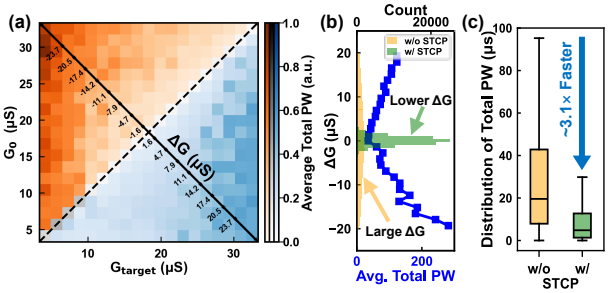

Fig. 5 (a) Total pulse width, PW vs. initial (G_0) and target conductance (G_{target}); $\Delta G = G_{target} - G_0$ shown diagonally. (b) ΔG positively correlates with PW (For STCP, G_0 is the conductance after STCP). STCP significantly reduces ΔG and thus (c) shortening programming time.

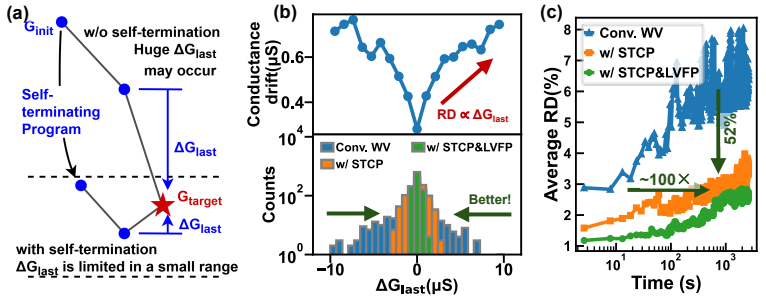

Fig. 6 (a) Self-termination programming constrains the final conductance jump (ΔG_{last}). (b) Relaxation versus ΔG_{last} and its distributions across methods. (c) Evolution of relative error (RD) for three methods; STCP + LVFP notably suppresses conductance drift.

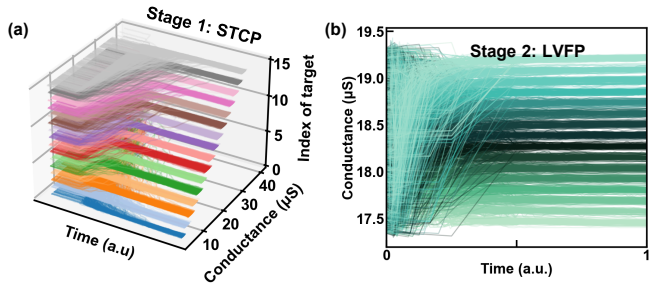

Fig. 7 (a) Conductance evolution during STCP with time normalized to highlight the transition. (b) Conductance tuning via LVFP after STCP. Each of the 16 STCP targets is further divided into configurable LVFP targets—16 in this case—yielding 256 total levels.

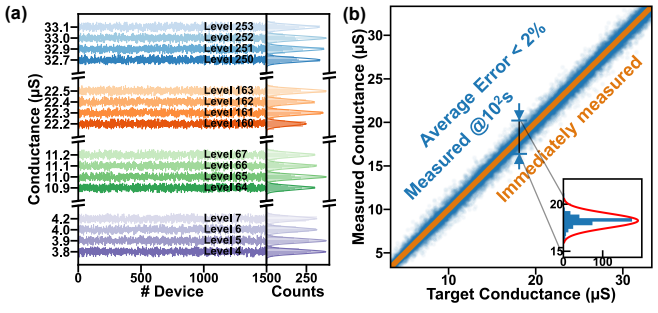

Fig. 8 (a) The proposed TSAWP unit achieves excellent MLC capability (>8 bits)—with high cell-to-cell uniformity. (b) Target vs. measured conductance @0s and @10^2 s. The average write error @10^2 s is <2%.

Neural Optimized Gradient-Aware Programming Scheduler for RRAM-Based CIM

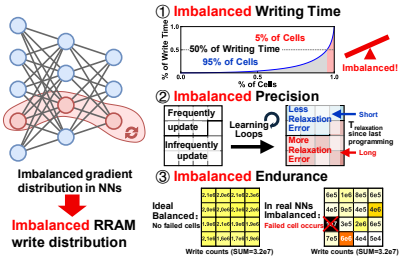

Fig. 9 The inherent gradient imbalance in neural networks leads to uneven writing time, precision, and endurance in RRAM-based CIM systems.

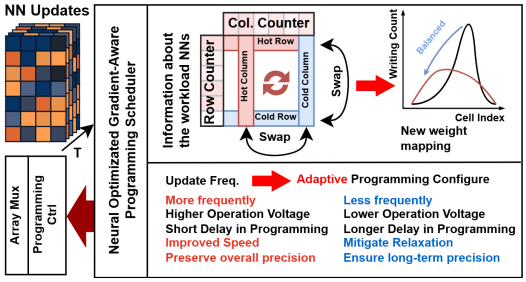

Fig. 10 Illustration of the Neural Optimized Gradient-Aware Programming Scheduler (NoGAPS).

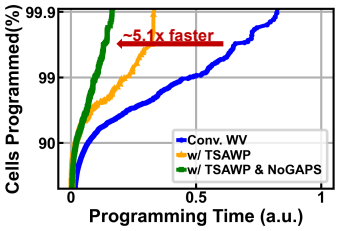

Fig. 11 Total pulse width comparison across programming schemes. NoGAPS improves write speed by 5.1× compared to conventional methods.

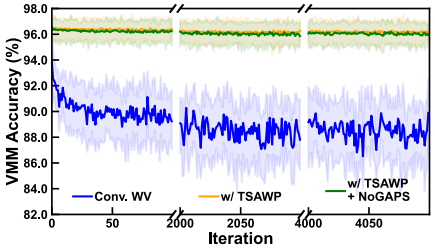

Fig. 12 VMM accuracy over time. NoGAPS achieves significantly faster programming without sacrificing accuracy, matching TSAWP's accuracy.

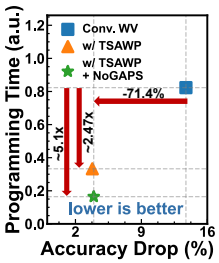

Fig. 13 Comparison of different programming strategies.

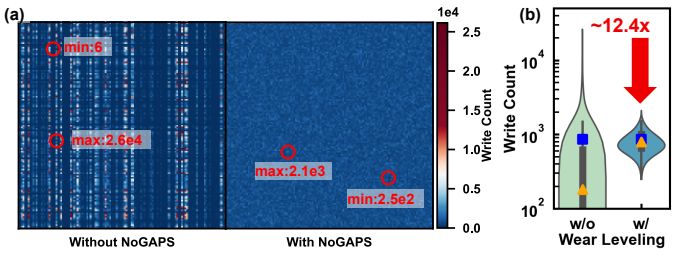

Fig. 14 (a) Write count distribution across the RRAM array with and without NoGAPS; extrema highlighted. (b) NoGAPS reduces write counts by 12.4×, significantly enhancing endurance.

Implementation of E-SSL on Learning-Aware RRAM Compute-in-Memory Accelerator

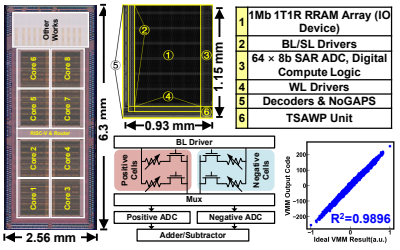

Fig. 15 Chip micrograph and system architecture of the proposed learning-aware compute-in-memory accelerator.

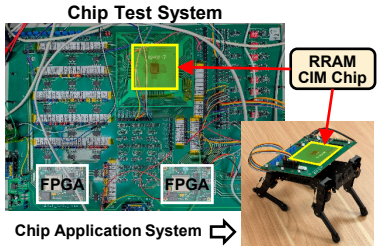

Fig. 16 Experimental test setup used to evaluate the proposed learning-aware compute-in-memory accelerator

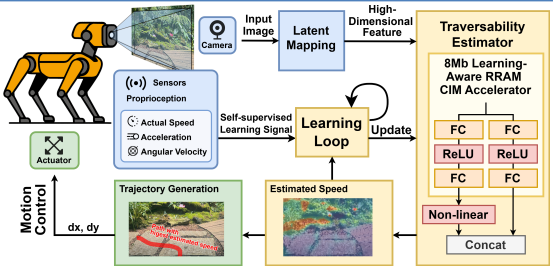

Fig. 17 Demonstrated E-SSL-based Traversability estimation framework using an 8Mb RRAM CIM accelerator. The robot learns from camera and sensor data to adapt motion control in real time.

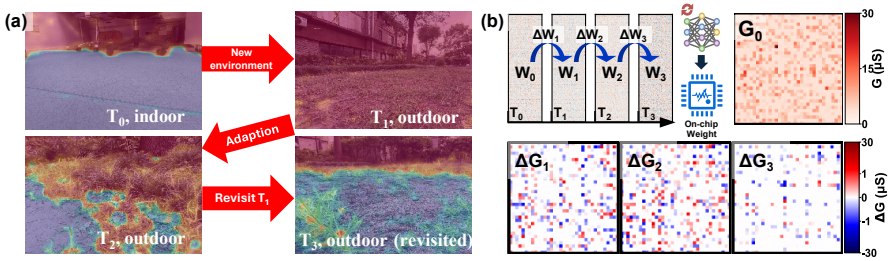

Fig. 18 (a) On-chip learning during transition from indoor (T_0) to unseen outdoor environments (T_1–T_3). Accuracy briefly drops at T_1, recovers after adaptation at T_2, and improves notably when revisiting T_1 (T_3). (b) Real-time updates in weight matrix (**W**) and RRAM conductance (**G**) reflect ongoing adaptation.

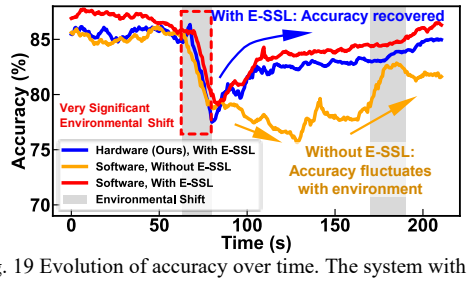

Fig. 19 Evolution of accuracy over time. The system with E-SSL retains accuracy after environmental shifts, while the system without tuning exhibits noticeable degradation.

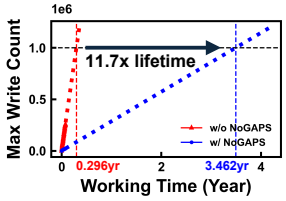

Fig. 20 Simulated maximum write count over array. Linear regression (dotted) shows extended lifetime enabled by wear leveling.

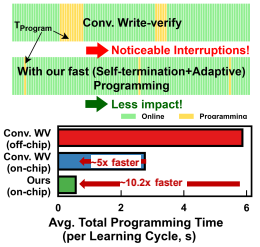

Fig. 21 Our chip significantly accelerates device programming, enhancing overall system availability and uptime.

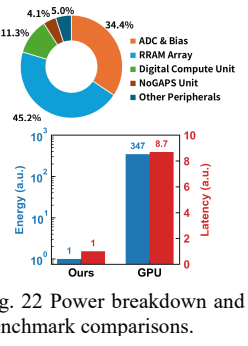

Fig. 22 Power breakdown and benchmark comparisons.

	Ours	IEDM 2024 [4]	IEDM 2024 [5]	IEDM 2024 [6]	Science 2023 [7]
Technology	40nm	180nm	40nm	28nm	130nm
Memory type	RRAM	RRAM	PCM	RRAM	RRAM
Implementation Level	Processor	Macro	Macro	Macro	Macro
Array/Core #	8	1	2	1	1
Capacity	8Mb	1Kb	288Kb	576Kb	158.8Kb
Analog Conductance Programming-Accelerated	Yes (on-chip)	No	No	No	No
Relaxation-Mitigated	Yes (on-chip)	No	No	No	No
Endurance-Enhanced	Yes (on-chip)	No	No	No	No
Energy Efficiency (TOPS/W)	36.2 (8b-IN,Analog-W)	\	26.8 (8b-IN,8b-W)	12.25 (\)	18.5 (4b-IN,Analog-W)
Peak MVM Throughput (TOPS)	17.5 (8b-IN,Analog-W)	\	\	\	\
Application	Embodied Self-supervised Learning	Differential Equation Solver	BCI (Neural Manifold)	Double Encryption	Image Classification

Table. 1 Comparison table.