NV-Memory Centric Liquid Neural Network Accelerator for Reflexive Robotic Control

***Abstract*— This paper proposes a non-volatile memory (NV-memory) centric Liquid Neural Network (LNN) accelerator for reflexive robotic control. LNNs are well suited for low-level robotic control under strict edge constraints by virtue of its adaptive neuron dynamics. However, LNN inference requires a huge number of neuron state updates via iterative operations, consuming substantial energy. The proposed accelerator reduces iterative data movement and maximizes hardware density. In the proposed accelerator implementations, three approaches are optimized for their respective functions involved in the iterative operations of Neural Circuit Policies (NCPs), a class of LNN: (I) 6-bit digital SRAM CAM and 6-bit digital ReRAM LUT for sigmoid activation, (II) 6-bit analog multi-level cell (MLC) ReRAM Computation-in-Memory (CiM) for matrix-vector multiplication (MVM), and (III) a hybrid near-memory digital processor with 8-bit ReRAM and 8-bit SRAM buffer for arithmetic operations. As a result, the proposed accelerator reduces energy consumption to 1/12.4 and memory area by 88.1% compared to the SRAM-only configuration on a representative robotic manipulation task, while maintaining a high success rate. Furthermore, the energy consumption is reduced to 1/688 of that of the Jetson Orin Nano 8 GB.**

Keywords—Physical AI, Robotic control, Liquid Neural Network (LNN), Computation-in-Memory (CiM), ReRAM

I. Introduction

Physical AI requires low-level policies that provide reflexive adaptability, i.e., fast, reactive responses, under edge constraints. However, existing methods exhibit a fundamental trade-off: behavior cloning is efficient but static, while adaptive approaches incur significant latency (Fig. 1(a)). Liquid Neural Networks (LNNs) [1] offer a promising solution as compact and adaptive low-level policies within a hierarchical control architecture, based on biologically inspired neurons (Fig. 1(b)). These neurons can dynamically vary their internal states in response to sensory inputs. This intrinsic adaptability enables history-dependent responses, increasing temporal expressivity and allowing the system to flexibly react to environmental changes. While some models [2, 3] aim to reduce the computational cost of LNNs, they often sacrifice biological interpretability and reflexive dynamics. Neural Circuit Policies (NCPs) are a representative application of LNNs, which are organized into a biologically derived four-layer functional hierarchy (sensory, inter, command, and motor neurons), enforcing high sparsity, resulting in significantly fewer parameters [4].

However, the underlying neuron dynamics of NCPs, governed by Liquid Time-constant Networks (LTCs) as formulated in Eq. (1) pose significant challenges (Fig. 1(c)). To update neuron states while ensuring stability for stiff dynamics, NCPs solves LTCs using the semi-implicit Euler method with a fixed timestep Δ (Eq. (2)) [5]. Because the neuron state updates require multiple iterative evaluations per timestep, neuron states must be repeatedly read from and written back to memory during inference. To mitigate this data movement bottleneck, recent studies [6, 7] have explored solving ODEs directly with ReRAM-based memory-centric architectures. However, these solvers do not fully address the diverse computational requirements of NCPs like nonlinear activations $\sigma_i(x_j(t))$ (Eq. (3)). These complex operations are costly in terms of latency and energy.

To address this, a non-volatile memory (NV-memory) centric LNN accelerator is proposed for reflexive robotic control (Fig. 1(d)). The proposed accelerator implements three distinct approaches optimized for primary functional blocks of LNN, which requires diverse computational characteristics. Proposal I is a digital SRAM CAM-ReRAM LUT block that accelerates nonlinear operations via a lookup table for sigmoid activation. Proposal II is an analog multi-level cell (MLC) ReRAM Computation-in-Memory (CiM) that performs intensive matrix-vector multiplication (MVM). Proposal III is a hybrid SRAM-ReRAM digital near-memory processor that executes arithmetic operations efficiently while reducing data movement using an SRAM buffer and ReRAM. Furthermore, the proposed accelerator optimizes the bit precision of the components in each functional block to reduce memory footprint and energy consumption.

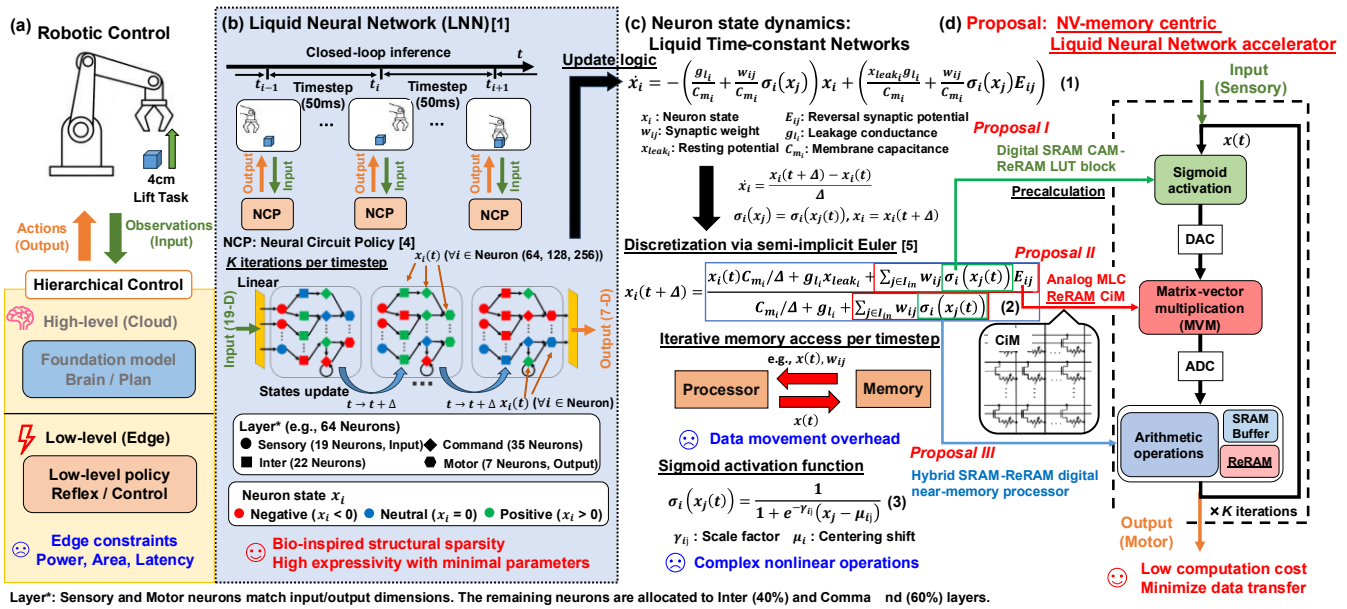


Fig. 1 Overview of the proposed NV-memory centric Liquid Neural Network (LNN) [1] accelerator in (a) reflex robotic control. (b) Neural Circuit Policy (NCP) [4], a class of LNN, is adopted in hierarchical control as low-level policy. (c) Iterative neuron states update by Liquid Time-constant Networks (LTCs) via semi-implicit Euler [5], leading to computational bottlenecks. (d) Proposal: NV-memory centric accelerator for NCPs. Prop. I: Digital SRAM CAM-ReRAM LUT block for nonlinear functions, Prop. II: Analog MLC ReRAM CiM for matrix-vector multiplication (MVM), and Prop. III: Hybrid SRAM-ReRAM digital near-memory processor for arithmetic operations.

II. Proposed NV-Memory Centric Liquid Neural Network (LNN) Accelerator

The proposed NV-memory centric LNN accelerator processes NCPs, a class of LNNs, in a compact array area with low energy consumption. Fig. 2 shows the performance evaluation of NCPs with varying neuron count (N = 64, 128, 256) alongside conventional MLP and RNN baselines [8] using the Robosuite framework [9]. These models were configured to achieve comparable task performance on the reflexive Lift Task. The performance is quantified by the average success rate across 10 inference trials. The results demonstrate that NCPs achieve high average success rate while reducing the number of parameters by 65.3× compared to RNN, with only a 1.5% degradation in average success rate.

In the proposed accelerator, the NCP update equation (Eq. (1)) is decomposed into three dominant computational primitives, each mapped onto dedicated hardware functional blocks. Table I summarizes the computational specifications and design requirements for each functional block (N = 256). Sigmoid activation and MVM blocks dominate computation and array configuration, requiring high-density and energy-efficient memory technologies. To address these, Table II compares SRAM and analog/digital ReRAM. ReRAM exhibits a substantial advantage in cell size, achieving 4-10F^2 for memory cells and 10-50F^2 for CiM [10, 11], compared to 120-200F^2 for standard 6T SRAM cells. For proposal I, 10T SRAM CAM cell size is estimated at 200-330F^2, scaled from 6T-SRAM cell proportionally. ReRAM significantly reduces leakage power and achieves superior energy efficiency (< 0.1pJ/bit) relative to SRAM (> 1pJ/bit) [12]. By optimizing the design requirements and memory characteristics, three hardware functional blocks are optimized as follows.

A. Proposal I: Digital SRAM CAM-ReRAM LUT Block for Sigmoid Activation

The block implements sigmoid activation with deterministic precision, avoiding the inherent instability in analog implementations (Fig. 3(a)). Building upon the framework of prior associative computing architectures [13], a hybrid memory strategy is adopted to balance reliability and density. Complex arithmetic is replaced by LUT-based lookup to reduce online computation overhead [14]. SRAM is employed for CAM array to provide robust and reliable

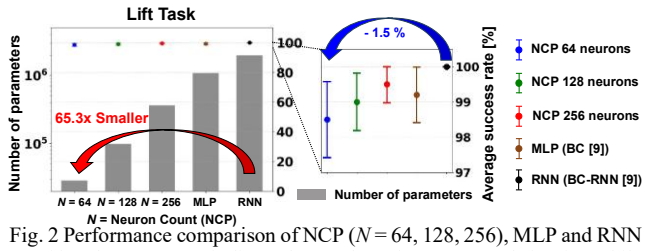


Fig. 2 Performance comparison of NCP (N = 64, 128, 256), MLP and RNN on Lift task [8, 9]. Bars indicate the number of parameters (left axis), while markers show average success rate (right axis).

Table I Computational specifications and hardware design requirements for NCP (N = 256) in Lift task.

N = 256 , Input dim = 19, CAM entries (E) = 256 (8-bit), Iteration steps per timestep (K)= 3

Functional block	Implementation	Operations (per timestep)	Complexity	Array configuration	Design requirements
I. Sigmoid activation $\sigma_i(x_j(t))$	Digital SRAM CAM	211K (275×256×3)	$O(N \cdot E)$ Computation dominant	256×1	- Low-latency search - High reliability
	Digital ReRAM LUT	211K (275×256×3)	$O(N^2)$	256×256	- High read cycles - High capacity - Long data-retention
II. MVM	Analog ReRAM CiM	422K (275×256×3×2)	$O(N^2)$	275× 256	
III. Arithmetic operations	Digital Processor	4.60K (256×6×3)	$O(N)$	Area dominant -	- Flexible calculation

Table II Comparison of SRAM and analog/digital ReRAM device characteristics [10 - 12]

	Usage	Cell size (F^2)	Set / Reset endurance	Read Latency	Leakage Power	Energy per bit access [12]
ReRAM	Memory / LUT [10]	4 – 10 (7)	10^8 – 10^{11}	~ 10 ns	Low	< 0.1 pJ (0.1)*
	CiM [11]	10 – 50 (30)				
SRAM	Buffer [10]	120 – 200 (160)	Free	~ 0.2 – 2 ns	High	> 1 pJ (1.0)
	CAM (10T)	200 – 330 (265)				

Energy per bit access* (ReRAM) = Vread (0.2V) * Iread (< 50uA) * Read latency (< 10ns) [12] = 0.1 pJ

matching, while single-level cell (SLC) ReRAM is utilized for LUT storage to satisfy high-capacity requirements. The SRAM CAM matches the quantized inputs $x_j(t)$ and activates the corresponding match-line (ML) [15], which then selects the corresponding entry in the ReRAM LUT. Activations $\sigma_i(x_j(t))$ (Eq. (3)) are precalculated and stored in ReRAM and the selected ML routes the output to the corresponding bit-line (BL). To minimize the memory footprint, the nonlinear parameters of sigmoid activation (γ_{ij}, μ_{ij}), originally defined per synapse, are constrained to be neuron-specific (γ_i, μ_i). This reduction in parameter degrees of freedom decreases the required LUT capacity from $O(N^3)$ to $O(N^2)$, enabling high-density integration while maintaining sufficient performance.

B. Proposal II: Analog MLC ReRAM CiM for MVM

Analog MLC ReRAM CiM performs multiply-accumulate (MAC) operations (Fig. 3(b)). Because MVM dominates the NCP workload, analog MLC ReRAM CiM is adopted for parallel computing, minimizing cell area and energy consumption. Leveraging the proven effectiveness of memory-centric architectures [16], the proposed CiM utilizes the high-density and parallel processing capabilities of ReRAM to overcome the data movement bottleneck. These gains are further enhanced by using MLC ReRAM instead of SLC. In MLC ReRAM, synaptic weights (w_{ij}, $w_{ij}E_{ij}$) are stored as cell conductance. w_{ij} represents the magnitude of synaptic weights, while E_{ij} is reversal synaptic potential that defines the polarity of the synapse. Activated states are converted into analog word-line (WL) voltages which drive the ReRAM cells. The computation result is obtained as the accumulated current on BL.

C. Proposal III: Hybrid SRAM-ReRAM Digital Near-memory Processor for Arithmetic Operations

The digital near-memory processor handles the remaining arithmetic operations using neuron-specific parameters (Fig. 3(c)). ReRAM stores fixed parameters (C_{mi}/Δ, g_{li}, $g_{li}x_{leaki}$) to minimize cell size, while SRAM buffer stores neuron state variables $x_i(t)$ to take advantage of high endurance. The digital near-memory processor ensures numerical stability during the update of neuron states $x_i(t+\Delta)$.

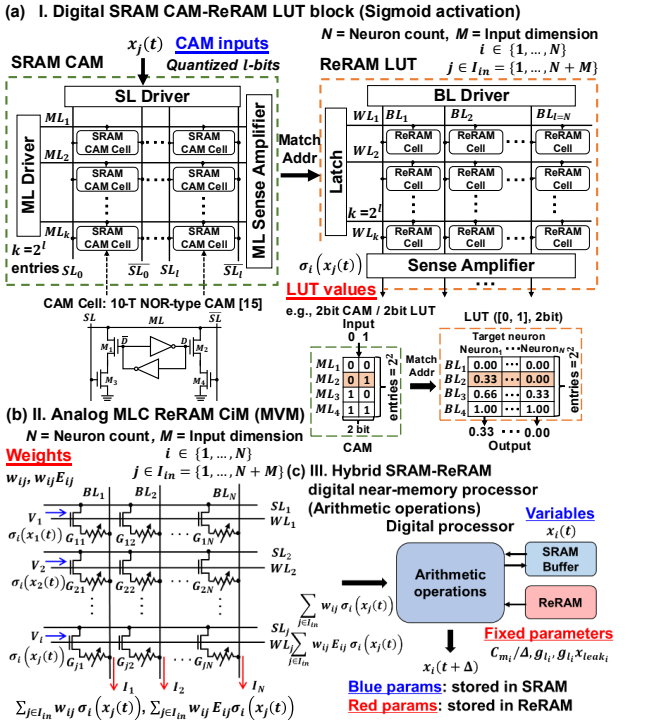


Fig. 3 Three hardware functional blocks of the proposed accelerator. (a) Prop. I: Digital SRAM CAM-ReRAM LUT block for sigmoid activation. (b) Prop. II: Analog MLC ReRAM CiM for MVM. (c) Prop. III: Hybrid SRAM-ReRAM digital near-memory processor for arithmetic operations.

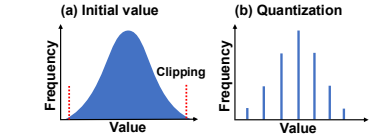
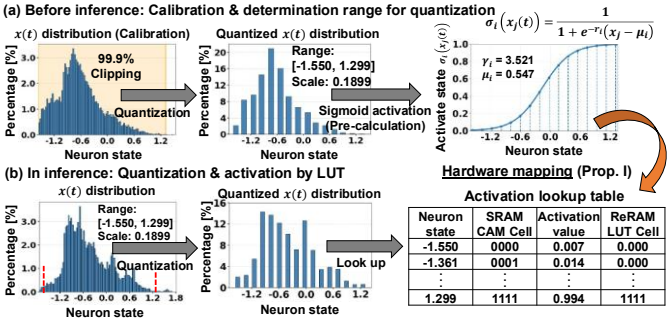


Fig. 4 Value distribution for hardware mapping. (a) Initial value with clipping. (b) Quantized values.



Neuron state	SRAM CAM Cell	Activation value	ReRAM LUT Cell
-1.550	0000	0.007	0.000
-1.361	0001	0.014	0.000
⋮	⋮	⋮	⋮
1.299	1111	0.994	1111

Fig. 5 Hardware mapping flow for digital SRAM CAM-ReRAM LUT block (Prop. I). (a) The distribution of neuron states $x(t)$ is analyzed to determine 99.9% clipping range and quantization scale. Quantized states are mapped to SRAM CAM, and the corresponding activation values are stored in ReRAM LUT. (b) During inference, neuron states are quantized using the predetermined range and scale, and the corresponding values are retrieved from lookup table.

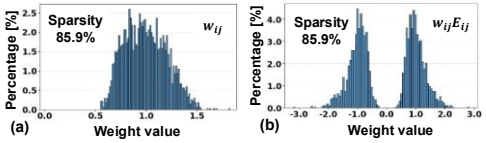


Fig. 6 Initial value distributions of weights (a) w_{ij} and (b) $w_{ij}E_{ij}$ of analog MLC ReRAM CiM (Prop. II) excluding zero-valued elements. Both distributions are highly sparse.

III. Quantization Tolerance and Optimization of Bit Precision for Proposed Accelerator

The proposed accelerator applies quantization to further reduce cell area and energy consumption across three functional blocks. To determine appropriate bit precision, quantization tolerance of each block is evaluated. The initial value distributions are quantized with clipping to remove outliers, enabling finer quantization steps (Fig. 4(a)). The quantized values are stored in ReRAM or SRAM (Fig. 4(b)).

To overcome the complexity of implementing input-dependent sigmoid activation, Proposal I utilizes a precomputation-based approach. Fig. 5 shows the mapping flow of sigmoid activation to digital SRAM CAM-ReRAM LUT block (Prop. I). Through calibration before inference, the distribution of neuron states $x_i(t)$ is analyzed, and the quantization range is determined by clipping at the 99.9th percentile to maximize the quantization resolution. Based on this range, quantized neuron states are stored in SRAM CAM, and precalculated sigmoid activations are stored in SLC ReRAM LUT (Fig. 5(a)). During inference, the input neuron states are quantized based on this range and the activations are retrieved from ReRAM LUT (Fig. 5(b)).

Fig. 6 shows the weight distributions (w_{ij}, $w_{ij}E_{ij}$) originally trained in 32-bit floating point (FP32) and mapped to analog MLC ReRAM CiM (Prop. II). Due to the high sparsity observed in both distributions, zero-valued weights are preserved as exact zeros and excluded from quantization to avoid unnecessary representation overhead.

To isolate quantization effects, Fig. 7 shows average success rates by varying only the target block's precision while all other blocks remain fixed at 8-bit precision. Three neuron models (N = 64, 128, 256) are evaluated.

A. Digital SRAM CAM-ReRAM LUT Block (Prop. I)

Inputs $x_j(t)$ in SRAM CAM (Fig. 7(a)) and precalculated activations $\sigma_i(x_j(t))$ in ReRAM LUT (Fig. 7(b)) are quantized. SRAM CAM maintains high average success rate up to 4-bit precision, followed by a sharp degradation below 4-bit precision. In contrast, ReRAM LUT exhibits higher tolerance

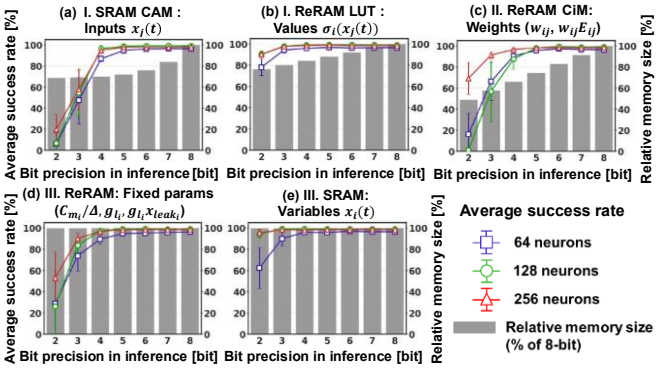


Fig. 7 Quantization tolerance of the proposed accelerator. Markers indicate average success rate (left) across 10 trials (N = 64, 128, 256), while bars show relative memory size (right) to 8-bit baseline. Quantization is applied to: (a) Inputs in SRAM CAM (Prop. I), (b) Values in ReRAM LUT (Prop. I), (c) Weights in analog MLC ReRAM CiM (Prop. II), (d) Fixed parameters in ReRAM (Prop. III), and (e) Variables in SRAM buffer (Prop. III).

to quantization with three neuron models. Since the number of entries scales exponentially with input bit precision, even a single-bit reduction halves model size of this block, leading to memory savings.

B. Analog MLC ReRAM CiM (Prop. II)

Synaptic weights (w_{ij}, $w_{ij}E_{ij}$) (Fig. 7 (c)) are quantized. Average success rate is high up to 5-bit precision with three different neurons, indicating low quantization tolerance. However, MVM operated by proposed analog MLC ReRAM CiM dominates the total memory footprint of the proposed accelerator. Thus, reducing the bit precision of weights significantly improves the compactness of the accelerator.

C. Hybrid SRAM-ReRAM Digital Near-memory Processor (Prop. III)

Quantization is applied to fixed parameters (C_{mi}/Δ, g_{li}, $g_{li}x_{leaki}$) stored in ReRAM (Fig. 7(d)) and variables $x_i(t)$ stored in SRAM (Fig. 7(e)). Fixed parameters exhibit stronger robustness to quantization, while state variables are more sensitive. Because the number of parameters in the digital near-memory processor is small, lowering the bit precision provides only marginal reduction in overall memory size.

Based on this analysis, bit precision reduction is prioritized for functional blocks that contribute most significantly to memory size. First, SRAM CAM and analog MLC ReRAM CiM are prioritized for bit precision reduction. Although inputs $x_j(t)$ in SRAM CAM and weights in analog MLC ReRAM CiM show relatively high sensitivity to quantization, they dominate the overall memory size; thus, precision scaling in SRAM CAM and analog MLC ReRAM CiM yields the largest system-level compression benefit. Next, ReRAM LUT precision is optimized, as its activation values are more robust to quantization while still providing moderate memory savings, making it suitable for secondary precision scaling. In contrast, ReRAM and SRAM buffer in the digital near-memory processor contribute minimally to total memory size while risking numerical instability. Therefore, their precision is maintained. Thus, the proposed accelerator is configured as follows: I) 6-bit for SRAM CAM and 6-bit for ReRAM LUT, II) 6-bit for analog MLC ReRAM CiM, and III) 8-bit for ReRAM and 8-bit for SRAM buffers in the digital near-memory processor.

IV. Error Tolerance of Proposed Accelerator

Analog MLC ReRAM CiM in the proposed accelerator operates MVM faster with compact memory size. Fig. 8 shows the structure of TaO_X-based ReRAM and the switching mechanism between high-resistance states (HRS) and low-resistance state (LRS) [17 - 19]. ReRAM stores data by set/reset switching operations. However, analog MLC ReRAM, which stores weights (w_{ij}, $w_{ij}E_{ij}$), inherently suffers from device non-idealities. Thus, the error tolerance of the

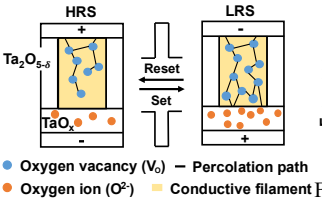


Fig. 8 Structure and switching mechanism of ReRAM between HRS and LRS [17].

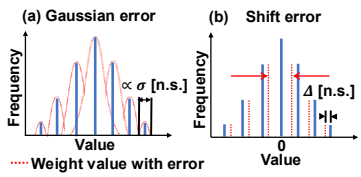


Fig. 9 Values corresponding to weights w_{ij} of analog MLC ReRAM CiM (Prop. II). (a) Gaussian error injected and (b) shift error injected into quantized values.

proposed accelerator is evaluated. In ReRAM devices, write variation and conductance shifts due to data-retention (DR) are represented as Gaussian errors (Fig. 9(a)) and shift errors (Fig. 9(b)), respectively. The error magnitude is expressed as a normalized step (n.s.), defined relative to the weight range normalized between -1 and 1.

Fig. 10 shows the measured current distribution of 3-bit/cell MLC ReRAM at a read voltage (V_{read}) of 0.2 V [20 - 22]. The distributions are shown for DR time t = 0 at room temperature (Figs. 10(a) and (c)) and DR t = 22.44 hours at 150 degC (equivalent time 12-month DR time at 85 degC) (Figs. 10(b) and (d)). To examine the impact of different memory windows, Figs. 10(c) and (d) utilize a narrower memory window than Figs. 10(a) and (b), reducing energy consumption by limiting the maximum cell current. Due to data-retention, cell currents in LRS shift towards HRS. Cells with lower current tend to stick to 0 µA.

Fig. 11 shows average success rates when Gaussian errors (Fig. 11(a)) and shift errors (Fig. 11(b)) are injected into weights in analog MLC ReRAM CiM before applying quantization. These error models are derived from the measured non-idealities of ReRAM. Using 95% average task success rate as the robustness criterion, acceptable Gaussian error is up to 0.05 n.s. and acceptable shift error is up to 0.07 n.s. with N = 256.

Fig. 12 shows the evaluation of the proposed accelerator, where both quantization and analog non-idealities are jointly applied. The proposed accelerator maintains valid task performance under Gaussian errors up to 0.02 n.s. and shift errors up to 0.08 n.s., with tolerance depending on the network size.

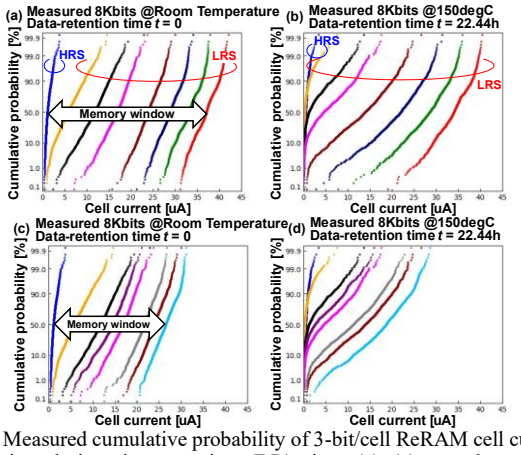


Fig. 10 Measured cumulative probability of 3-bit/cell ReRAM cell current distribution during data-retention (DR) time (a) (c) t = 0 at room temperature and (b) (d) t = 22.44 h at 150 degC (Equivalent time 12-month DR time at 85 degC) [21, 22]. (c) and (d) use a smaller memory window than (a) and (b) to reduce energy consumption.

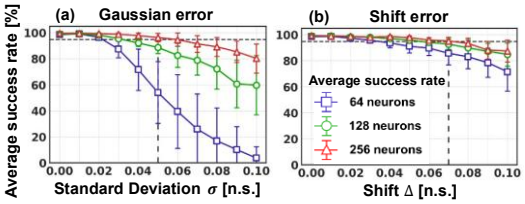


Fig. 11 Error tolerance of NCP against Gaussian error and shift error across 10 trials (N = 64, 128, 256). Errors are injected into weights ($w_{ij}, w_{ij}E_{ij}$).

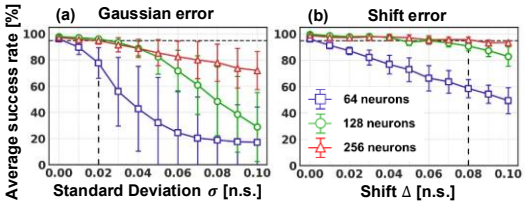


Fig. 12 Error tolerance of the proposed accelerator against Gaussian error and shift error in 10 trials (N = 64, 128, 256). Errors are injected into the quantized weights ($w_{ij}, w_{ij}E_{ij}$) in analog MLC ReRAM CiM (Prop. II).

V. Conclusion

In this paper, an NV-memory centric LNN accelerator for NCP is proposed to enable efficient reflexive robotic control. Table III shows a summary of this work. In this evaluation, the energy consumption of the read circuitry is assumed to be negligible by employing an ADC-less sensing scheme [23, 24]. By decomposing NCP dynamics into computational blocks and mapping the optimal memory and hardware, the proposed accelerator reduces iterative data movement and maximizes hardware density. Optimized bit precision based on quantization tolerance further minimizes memory footprint and computational overhead without degrading performance. The proposed accelerator achieves 1/12.4 energy consumption and an 88.1% reduction in memory area compared to the SRAM-only configuration, and 1/688 energy consumption relative to Jetson Orin Nano 8 GB [25], demonstrating high efficiency for edge robotic applications. Furthermore, the proposed accelerator exhibits robustness against device non-idealities, tolerating a maximum acceptable Gaussian error of 0.02 n.s. and shift error of 0.08 n.s. (Table IV). These results indicate that the proposed accelerator is a promising solution for realizing reflexive robotic control on edge devices.

VI. Acknowledgement

Table III Summary of this work with hardware costs, memory access energy, and performance for NCP (N = 256) on Lift task. The accelerator is compared with Jetson Orin Nano (8 GB) [25] and the SRAM-only configuration. For analog MLC ReRAM CiM (Prop. II), 3-bit/cell MLC is assumed.

Inference		Jetson Orin Nano (8GB) [25]	SRAM-only configuration		Proposed accelerator	
Sigmoid activation (I. Digital SRAM CAM-ReRAM LUT block)	Bit precision	32 bits	SRAM CAM	SRAM LUT	SRAM CAM	SLC ReRAM LUT
			6 bits	6 bits	6 bits	6 bits
	Memory bits (Area)	4.51 M (-)	384 (102 KF²)	98.3 K (15.7 MF²)	384 (102 KF²)	98.3 K (688 KF²)
	Memory energy (per timestep)	84.8 µJ**	4.47 nJ*	1.27 µJ	4.47 nJ*	127 nJ
MVM (II. Analog MLC ReRAM CiM)	Bit precision	32 bits	SRAM CiM		MLC ReRAM CiM	
			6 bits		6 bits	
	Memory bits (Area*)	6.76 M (-)	845 K (135 MF²)		845 K (16.9 MF²)	
	Memory energy (per timestep)	127 µJ**	2.53 µJ		169 nJ	
Arithmetic operations (III. Hybrid SRAM-ReRAM digital near-memory processor)	Bit precision	32 bits	SRAM		SLC ReRAM	SRAM
			8 bits		8 bits	8 bits
	Memory bits (Area)	24.6 K (-)	8.19 K (1.31MF²)		6.14 K (43.0 KF²)	2.05K (328KF²)
	Memory energy (per timestep)	463 nJ**	24.6 nJ		7.99 nJ	
Total	Memory bits (Area)	11.3 M (-)	935 K (152 MF²)		935 K (18.1 MF²)	
	Memory energy (per timestep)	212 µJ**	3.83 µJ		308 nJ	
Average success rate		99.5 %	98.8 %		98.8 %	

- 95.6 %, × 1/10, - 87.5 %, × 1/15, - 67.5 %, - 88.1 %, × 1/12.4, × 1/688, - 0.7 %

Area = Memory bits × Cell size (Table II)
Area* (Analog MLC ReRAM CiM) = Memory bits / 3 (3-bit MLC ReRAM) × 2 (a differential pair) × Cell size (Table II)
Memory energy = (SRAM bits × 1.0 pJ + ReRAM bits × 0.1 pJ) × 3 (iterations per timestep) [12]
Memory energy* (SRAM CAM) = {9.5 fJ (ML) + 4.6 fJ (SL)} [15] × Operations (CAM) × 6 (Bit Precision)
Memory energy** (Jetson Orin Nano [25]) = 11 W (Average power) × Total memory bits / 68 GB/s (Memory access)

Table IV Error tolerance of the proposed accelerator

Error type	The number of NCP neurons (N)		
	64 neurons	128 neurons	256 neurons
Gaussian σ [n.s.]	< 0.005	< 0.02	< 0.02
Shift Δ [n.s.]	< 0.005	< 0.04	< 0.08

REFERENCES

[1] R. Hasani et al., “Liquid Time-constant Networks”, *in Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no.9, pp. 7657–7666, 2021.

[2] R. Hasani et al., “Closed-form continuous-time neural networks”, *Nature Machine Intelligence*, vol. 4, no. 11, pp. 992–1003, Nov. 2022.

[3] R. Hasani et al., “Liquid structural state-space models”, *in Proc. International Conference on Learning Representations (ICLR)*, 2023.

[4] M. Lechner et al., “Neural circuit policies enabling auditable autonomy”, *Nature Machine Intelligence.*, vol.2, pp. 642–652, 2020.

[5] W. H. Press et al., “Numerical Recipes: The Art of Scientific Computing 3rd edition”, *Cambridge University Press*, Chapter. 17, 2007.

[6] D. Zhao et al., “Compute-in-memory for numerical computations”, *Micromachines*, vol. 13, no. 5, p. 731, May 2022.

[7] Y. Chen et al., “ReLOPE Resistive RAM-based linear first-order partial differential equation solver”, *in Proc. IEEE 40th International Conference Computer Design (ICCD)*, pp. 608–615, 2022.

[8] A. Mandlekar et al., “What Matters in Learning from Offline Human Demonstrations for Robot Manipulation”, *in Proc. Conference on Robot Learning (CoRL)*, vol. 164, pp. 1678-1690, 2021.

[9] Y. Zhu et al., “robosuite: A Modular Simulation Framework and Benchmark for Robot Learning”, *arXiv:2009.12293*, 2020.

[10] J. Boukhobza et al., “Emerging NVM: A Survey on Architectural Integration and Research Challenges”, *in Proc. ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 2, 2017.

[11] Y. -C. Luo et al., “Correlated Oxide Selector for Cross-Point Embedded Non-Volatile Memory”, *IEEE Transactions on Electronic Devices*, vol. 71, no. 1, pp. 916-921, 2024.

[12] J. Zhao et al., “Memory and Storage System Design with Nonvolatile Memory Technologies”, *IPSJ Transactions on System LSI Design Methodology*, pp. 2-11, 2015.

[13] Y. Zhai et al., “STAR: An Efficient Softmax Engine for Attention Model with RRAM Crossbar”, in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-2, 2023.

[14] X. Yuemaier et al., “A streaming data processing architecture based on lookup tables”, *Electronics*, vol. 12, no. 12, p. 2725, Jun 2023.

[15] K. Pagiamtzis et al., “Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey”, *in IEEE Journal of Solid-State Circuits (JSSC)*, vol. 41, no. 3, pp. 712-727, 2006.

[16] C. Wolters et al., “Memory Is All You Need: An Overview of Compute-in-Memory Architectures for Accelerating Large Language Model Inference”, *arXiv:2406.08413*, 2024.

[17] K. Kawai et al., “Highly-reliable TaOx ReRAM Technology using Automatic Forming Circuit”, *in Proc. International Conference on IC Design & Technology (ICICDT)*, pp. 1-4, 2014.

[18] R. Mochida et al., “A 4M Synapses integrated Analog ReRAM based 66.5 TOPS/W Neural-Network Processor with Cell Current Controlled Writing and Flexible Network Architecture”, *in Proc. IEEE Symposium on VLSI Technology*, pp. 175-176, 2018.

[19] T. Mikawa et al., “Neuromorphic computing based on Analog ReRAM as low power solution for edge application", *in Proc. IEEE International Memory Workshop (IMW)*, 2019.

[20] T. Ninomiya et al., “Conductive filament expansion in TaOx bipolar resistive random access memory during pulse cycling”, *Japanese Journal of Applied Physics 52.11R*, 2013.

[21] Y. Izume et al., “Column-wise Weight Value Inversion (Colwin) of ReRAM CiM to Reduce MAC Error during Data-retention”, *in Proc. International Conference on Solid State Devices and Materials (SSDM)*, pp. 103-104, 2025.

[22] Y. Hirata et al., “Adaptive Oxygen Vacancy Diffusion Compensation in MLC Intermediate States for over 10-year Data-retention of TaOx ReRAM Analog CiM Array”, *in Proc. European Solid-State Electronics and Computing Conference (ESSERC)*, pp. 605-608, 2025.

[23] Q. Zheng et al., “Lattice An ADCDAC-less ReRAM-based Processing-In-Memory Architecture for Accelerating Deep Convolutional Neural Networks”, *in Proc. 57th ACMIEEE Design Automation Conference (DAC)*, pp. 1-6, 2020.

[24] H. Jiang et al., “ENNA An Efficient Neural Network Accelerator Design Based on ADC-Free Compute-In-Memory Subarrays”, *in Proc. IEEE Transactions on Circuits and Systems I Regular Papers*, vol. 70, no. 1, 2023.

[25] NVIDIA, “Jetson Orin Nano Series Data Sheet DS-11105-001_v1.1”, 2023. [Online]. Available: https://www.mouser.com/pdfDocs/Jetson_Orin_Nano_Series_DS-11105-001_v11.pdf