

Integration of Memristor Technology with CMOS: An AI-Based
Hybrid Architecture for Neuromorphic Computing Applications

<http://www.doi.org/10.62341/istj-vol39-2-fd04>

Received	2026/09/15	تم استلام الورقة العلمية في
Accepted	2026/10/04	تم قبول الورقة العلمية في
Published	2026/10/05	تم نشر الورقة العلمية في

Integration of Memristor Technology with CMOS: An AI-Based Hybrid Architecture for Neuromorphic Computing Applications

Ftimah Altahir Alhadi Ramadhan¹, Azdihar Alwuheeshi Ahmed²

1- <https://orcid.org/0009-0008-4717-9251>

2- <https://orcid.org/0009-0005-4342-2400>

¹Faculty of Information Technology, University of Zawia - Libya

²Sabratha Faculty of Engineering, Sabratha University- Libya

¹f.ramadhan@zu.edu.ly, ²ezdihar.alwheeshe@sabu.edu.ly

Abstract

Memristor-based neuromorphic computing is a promising approach for efficient AI processing. However, device variation and analog imperfections can affect computational reliability. This study proposes a hybrid architecture that combines an AI model with a conventional Complementary Metal-Oxide-Semiconductor (CMOS) fallback path to improve execution reliability. A lightweight AI model developed to classify four types of arithmetic and logical operations. The model trained and evaluated using a test set of 1,800 samples. The network architecture and required Multiply-Accumulate (MAC) operations were analysed to characterize its computational cost. A confidence-based mechanism then used to identify low-confidence decisions and route them to the CMOS fallback path. Energy and latency were analysed using published hardware references as analytical projections rather than direct hardware measurements. The AI model achieved 99.28% accuracy. The AI path processed 94.83% of the test samples, while 5.17% routed to the CMOS path. The hybrid system achieved 99.94% accuracy and corrected 12 of the 13 errors produced by the AI-only model. The results indicate that a CMOS fallback path can improve system reliability while limiting the use of conventional execution. Future work should implement the proposed architecture on physical

Integration of Memristor Technology with CMOS: An AI-Based
Hybrid Architecture for Neuromorphic Computing Applications
<http://www.doi.org/10.62341/istj-vol39-2-fd04>

hardware and directly measure its energy and latency. The effects of memristor variation and noise should also be investigated.

Keywords: memristor, CMOS, computing-in-memory, neuromorphic computing, neural network, hybrid architecture, hardware-aware analysis.

دمج تقنية الميمريستور مع تقنية CMOS: بنية هجينة قائمة على الذكاء الاصطناعي لتطبيقات الحوسبة العصبية

فاطمة الطاهر الهادي رمضان¹، ازدهار الوحيشي أحمد²

¹ كلية تقنية المعلومات - جامعة الزاوية - ليبيا

² كلية الهندسة - جامعة صبراتة - ليبيا

1- f.ramadhan@zu.edu.ly و 2- ezdihar.alwheshe@sabu.edu.ly

الملخص

تُعَدُّ الحوسبة العصبية القائمة على المقاومة الذاكرية نهجًا واعدًا لمعالجة الذكاء الاصطناعي بكفاءة. مع ذلك، قد يؤثر اختلاف الأجهزة وعيوب الدوائر التناظرية على موثوقية الحوسبة. تقترح هذه الدراسة بنية هجينة تجمع بين نموذج ذكاء اصطناعي ومسار احتياطي تقليدي بتقنية CMOS لتحسين موثوقية التنفيذ. تم تطوير نموذج ذكاء اصطناعي خفيف الوزن لتصنيف أربعة أنواع من العمليات الحسابية والمنطقية. دُرِبَ النموذج وقُيِّمَ باستخدام مجموعة اختبار مكونة من 1800 عينة. حُلِّتْ بنية الشبكة وعمليات MAC المطلوبة لتحديد تكلفتها الحسابية. ثم استُخدمت آلية قائمة على الثقة لتحديد القرارات ذات الثقة المنخفضة وتوجيهها إلى المسار الاحتياطي بتقنية CMOS. كما حُلِّتْ الطاقة وزمن الاستجابة باستخدام مراجع الأجهزة المنشورة كنماذج تحليلية. حقق نموذج الذكاء الاصطناعي دقة بلغت 99.28%. عالج مسار الذكاء الاصطناعي الهجين دقة بلغت 99.94%، وصحح 12 خطأً من أصل 13 خطأً أنتجها نموذج الذكاء الاصطناعي فقط. تشير النتائج إلى أن مسار النسخ الاحتياطي بتقنية CMOS يُحسِّن موثوقية النظام مع الحد من استخدام التنفيذ التقليدي. ينبغي أن تُطبَّق الأعمال

Integration of Memristor Technology with CMOS: An AI-Based
Hybrid Architecture for Neuromorphic Computing Applications

<http://www.doi.org/10.62341/istj-vol39-2-fd04>

المستقبلية البنية المقترحة على أجهزة مادية، وأن تقيس استهلاكها للطاقة وزمن استجابتها بشكل مباشر. كما ينبغي دراسة تأثيرات تغيرات الميمريستور والتشويش.
الكلمات المفتاحية: الميمريستور، CMOS، ReRAM، الحوسبة في الذاكرة، الحوسبة العصبية الشكلية، الشبكة العصبية، البنية الهجينة، التحليل المدرك للعتاد.

1-Introduction

Artificial intelligence workloads require a large number of arithmetic operations and frequent data movement between memory and processing units. This movement can increase energy consumption and limit computational efficiency. Computing-in-memory (CIM) addresses this issue by performing computation close to the stored data. Memristive devices are particularly suitable for CIM because their conductance states can represent computational weights and support parallel multiply-accumulate operations (Aguirre et al., 2024; Huang et al., 2024).

Memristive CIM systems investigated as a way to improve the efficiency of AI acceleration. Chen et al. demonstrated a 65-nm Complementary Metal-Oxide-Semiconductor (CMOS)-integrated Resistive Random-Access Memory (ReRAM) CIM macro and reported 16.95 TOPS/W (tera operations per second per watt) with a 14.8-ns Multiply-Accumulate (MAC) computing time (Chen et al., 2019). These results demonstrate the potential of integrating memristive computation with conventional CMOS technology. Cai et al. developed an integrated memristor-CMOS system that combined a memristor crossbar with mixed-signal interfaces and digital processing (Cai et al., 2019). Their work showed that memristive computation could be combined with CMOS circuits to form a complete processing system.

CMOS remains important for digital control, data processing, and peripheral functions. Its integration with memristive devices can therefore provide complementary computational paths. However, memristive computing affected by device variation and analog non-idealities. These effects can reduce computational reliability (Huang et al., 2024; James, 2024). Several studies have investigated methods for handling such uncertainty. Bonnet et al. studied Bayesian inference with memristor-based neural networks

(Bonnet et al., 2023), while Xu et al. investigated fault-tolerant computation for imperfect analog hardware (Xu et al., 2026). These studies demonstrate the importance of considering hardware uncertainty when developing practical memristive AI systems. Despite these advances, most existing approaches focus on improving the memristive computing path or compensating for hardware imperfections. In the studies considered in this work, less attention given to using a conventional CMOS path selectively for uncertain AI decisions. A confidence-based routing mechanism can provide an additional reliability layer while allowing the primary AI path to process most inputs.

This study proposes a hybrid AI-CMOS architecture based on confidence-driven routing. Unlike previous approaches that mainly improve memristive computation or address hardware imperfections, the proposed approach uses AI confidence to selectively route uncertain predictions to a CMOS reference path. The AI controller serves as the primary computation path, while CMOS used only for low-confidence cases. The architecture evaluated on four digital-operation classes using classification accuracy, error correction, AI coverage, and CMOS fallback rate.

2- Proposed Hybrid Architecture

The proposed system contains two computational paths:

- (a) **Primary AI path:** a compact neural controller.
- (b) **Fallback CMOS path:** a conventional Multilayer Perceptron (MLP) reference model.

The AI path receives the original 24-dimensional input representation. It produces four class probabilities corresponding to Addition, Logical, Comparison and MAC. The highest-confidence prediction accepted when the confidence exceeds the selected threshold. Otherwise, the sample passed to the CMOS reference model.

The execution flow is:

Input → AI Controller → Confidence Decision → AI Result / CMOS Fallback

The CMOS model does not replace the AI model for all samples. It used only for samples that do not satisfy the confidence criterion.

AI Controller: The implemented AI model named Improved AI Controller. It contains three fully connected linear layers with 2,476 trainable parameters and approximately 0.0094 MB of FP32 parameter memory. The first layer maps 24 input features to 48 neurons and requires 1,152 MACs per inference. The second layer maps 48 neurons to 24 neurons also require 1,152 MACs. The third layer maps 24 neurons to the four output classes and requires 96 MACs. Therefore, the total computational cost of the AI controller is:

$$\text{MAC_AI} = 24(48) + 48(24) + 24(4) = 2,400 \text{ MACs/inference.}$$

The three linear layers correspond to three sequential matrix-vector multiplication (MVM) stages under the hardware mapping used in the analytical latency model.

CMOS Reference Model: The CMOS reference model is a conventional Multilayer Perceptron (MLP) classifier with an architecture of $24 \rightarrow 32 \rightarrow 16 \rightarrow 4$. The first layer maps 24 input features to 32 neurons and requires 768 MACs per inference. The second layer maps 32 neurons to 16 neurons and requires 512 MACs. The third layer maps 16 neurons to the four output classes and requires 64 MACs. Thus, the total computational cost of the CMOS reference model is:

$$\text{MAC_CMOS} = 24(32) + 32(16) + 16(4) = 1,344 \text{ MACs/fallback inference.}$$

The fallback energy calculation therefore uses 1,344 MACs per CMOS inference, rather than the 2,400 MACs required by the AI controller

3- Experimental Methodology

3.1- Dataset and Classification Task: The dataset contains 12,000 samples, with 3,000 samples generated for each of the four classes: Addition, Logical, Comparison, and MAC. The dataset was divided into training, validation, and test sets using a 70%/15%/15% split. This resulted in 8,400 training samples, 1,800

validation samples, and 1,800 test samples. The test set contains 450 samples per class.

3.2- AI Model and Training: The AI controller receives 24 memristor-transformed features and uses three fully connected layers with the structure $24 \rightarrow 48 \rightarrow 24 \rightarrow 4$. ReLU activation used in the hidden layers. The model contains 2,476 trainable parameters.

The model trained using cross-entropy loss and the Adam optimizer. The learning rate was set to 0.001 and the weight decay to 1×10^{-5} . Training used a batch size of 128 and a maximum of 100 epochs. Early stopping applied with a patience of 10 epochs. The model with the highest validation accuracy saved and used for the final test evaluation.

The memristor transformation used 5% conductance variation during training and validation feature generation. The test features were generated without additional variation for the main performance evaluation.

3.3- CMOS Reference Model: The CMOS reference model trained separately using the original input representation. It uses two hidden layers with 32 and 16 neurons, followed by the four output classes. The model uses ReLU activation and the Adam solver. The batch size was set to 128, the initial learning rate to 0.001, and the maximum number of iterations to 300. The regularization parameter was set to 1×10^{-4} . A random state of 42 used for reproducibility.

3.4- Confidence-Based Routing: For each test sample, the AI controller produces a class prediction and its corresponding confidence score. The maximum class probability is used as the confidence score. A threshold of **0.79** selected using the validation set. Predictions with confidence ≥ 0.79 accepted from the AI path. Predictions below this threshold routed to the CMOS reference model. The selected threshold resulted in 1,707 AI executions and 93 CMOS fallback executions on the 1,800-sample test set, corresponding to 94.83% AI coverage and a 5.17% fallback rate.

3.5- Confidence Threshold Selection: The AI confidence was defined as the highest class probability obtained from the softmax

output. The confidence threshold was selected using the validation set rather than the final test set.

Candidate thresholds from 0.50 to 0.99 evaluated with a step of 0.01. For each threshold, AI coverage and selective accuracy were calculated. Thresholds achieving 100% selective accuracy on the validation set retained. Among these thresholds, the one with the highest AI coverage selected.

The selected threshold was **0.79**. During the final test evaluation, samples with confidence ≥ 0.79 were accepted by the AI path, while samples with confidence < 0.79 were routed to the CMOS reference model.

This procedure resulted in 1,707 AI executions and 93 CMOS fallback executions, corresponding to 94.83% AI coverage and a 5.17% CMOS fallback rate.

3.6- Hybrid Decision: The hybrid system first evaluates every sample using the AI controller. The confidence value then compared with the selected threshold. Confident predictions retained, while low-confidence predictions replaced by the corresponding CMOS prediction.

4- Experimental Results:

4.1- AI-Only Performance

The AI-only model achieved an accuracy of 99.28%. For the 1,800 test samples, this corresponds to 13 incorrect predictions. The model provides high classification accuracy before the confidence-based CMOS fallback mechanism activated.

4.2- CMOS Fallback Performance

The CMOS reference model evaluated on the 93 samples selected by the confidence-based routing mechanism. It achieved an accuracy of 98.92% on these samples.

This result indicates that the reference model can provide an effective alternative for low-confidence predictions. The reported value represents the classification accuracy of the CMOS reference model and does not represent a measured CMOS hardware accuracy.

4.3- Hybrid Performance

The AI predictions for the 93 low-confidence samples replaced with the corresponding predictions from the CMOS reference model. The resulting hybrid system achieved an accuracy of 99.94%. As shown in Table 1, the hybrid architecture reduced the number of errors from 13 to 1. It corrected 12 of the 13 errors produced by the AI-only model. The AI path processed 94.83% of the test samples, while 5.17% directed to the CMOS fallback path. The accuracy improvement was 0.66 percentage points:

$$99.94 - 99.28 = 0.66$$

Thus, the confidence-based routing mechanism improved the overall classification accuracy while using the CMOS path for only a small portion of the test samples.

Table 1. AI-Only and Hybrid Results.

Metric	AI-only	Hybrid
Accuracy	99.28%	99.94%
Errors	13	1
Corrected errors	—	12
AI coverage	100%	94.83%
CMOS fallback	0%	5.17%

4.4- Class-Level Results

The class-level performance of the hybrid architecture presented in Table2. The model achieved perfect precision, recall, and F1-score for the Addition and Logical classes. The Comparison class achieved a precision of 0.9978 and a recall of 1.0000. The MAC class achieved a precision of 1.0000 and a recall of 0.9978. The macro and weighted averages were both 0.9994, indicating consistent performance across the four classes. The support balanced, with 450 test samples for each class.

Table 2. Class-level Classification Results.

Class	Precision	Recall	F1-score	Support
Addition	1.0000	1.0000	1.0000	450
Logical	1.0000	1.0000	1.0000	450

Class	Precision	Recall	F1-score	Support
Comparison	0.9978	1.0000	0.9989	450
MAC	1.0000	0.9978	0.9989	450
Macro average	0.9994	0.9994	0.9994	1800

The lowest recall was 0.9978 for the MAC class, while the lowest precision was 0.9978 for the Comparison class. The remaining class-level measures were 1.0000.

5- Hardware-Aware Compute Analysis

The hardware analysis uses the implemented network dimensions and published hardware references. All energy and latency values reported in this section are analytical estimates based on external hardware references. They are not measured values from the proposed architecture or a fabricated chip. The AI path requires 2,400 MACs per inference, or 4.32 million MACs for the 1,800-sample test set. The CMOS reference requires 1,344 MACs per fallback. For 93 fallback samples, this corresponds to 124,992 MACs.

6- Hardware Reference Parameters

The hardware analysis uses published results from memristive and CMOS-based neural accelerators. These values used as external references for the analytical energy and latency estimates.

Table 3. Hardware Reference Parameters Used in the Analytical Model.

Parameter	Value	Reference
ReRAM-CMOS energy efficiency	16.95 TOPS/W	Chen et al. (2019)
ReRAM-CMOS MAC computing time	14.8 ns	Chen et al. (2019)
CMOS reference energy	19.1 fJ/MAC	Everson et al. (2019)

The memristive reference is the 65-nm CMOS-integrated ReRAM CIM work reported by Chen et al. (2019), which reported an energy

efficiency of 16.95 TOPS/W and a MAC computing time of 14.8 ns. The CMOS energy reference taken from the 65-nm neural-network accelerator reported by Everson et al. (2019), which reported an energy of 19.1 fJ/MAC at 3-bit resolution. These values used only as external hardware references for analytical estimation. They are not measurements of the proposed architecture, its AI controller, or its CMOS fallback path.

7- Analytical Energy Model

The energy values in this section are analytical estimates derived from published hardware efficiency and the MAC counts of the implemented and reference networks. They do not represent measured energy consumption of the proposed system.

7.1- AI Energy: The published reference efficiency of 16.95 TOPS/W corresponds to approximately 58.997 fJ/op. Since TOPS/W conventions may differ in how a MAC is counted, two cases are considered. The resulting AI energy estimates presented in Table 4.

Table 4. AI Energy Estimates.

Case	Energy/MAC	AI energy/inference
A: 1 op = 1 MAC-equivalent	58.997 fJ/MAC	141.5929 pJ
B: 1 MAC = 2 elementary operations	117.994 fJ/MAC	283.1858 pJ

Both cases reported to avoid relying on a single TOPS convention.

7.2- CMOS Fallback Energy: The published CMOS reference reports 19.1 fJ/MAC (Everson et al., 2019). The CMOS reference architecture requires 1,344 MACs per inference. Therefore, the estimated energy for one fallback inference is 25.6704 pJ.

With a fallback rate of 5.17%, the expected CMOS contribution is 1.3263 pJ per original test sample. For the 93 fallback samples, the estimated CMOS energy is 2,387.3472 pJ.

These values are reference-based estimates and do not represent-measured energy from the proposed system.

7.3- Hybrid Energy: The resulting hybrid energy estimates summarized in Table 5.

Table 5. Hybrid Energy Estimates.

Metric	Case A	Case B
AI energy (pJ/sample)	141.5929	283.1858
Expected CMOS contribution (pJ/sample)	1.3263	1.3263
Hybrid energy (pJ/sample)	142.9192	284.5121
Hybrid energy overhead	0.9367%	0.4684%

As shown in Table 5, the CMOS path activated after the AI path. Therefore, fallback adds energy rather than replacing the AI computation. The main measured benefit of the hybrid approach is improved classification reliability.

8- Analytical Latency Model

The published ReRAM-CIM reference reports a 14.8-ns computing time for the reported MAC operation (Chen et al., 2019). The implemented AI controller contains three Matrix-Vector Multiplication (MVM) stages. Assuming that each network layer mapped to one sequential MVM stage, the projected AI latency is:

$$T_{AI} = 3 \times 14.8 = 44.4 \text{ ns/inference.}$$

This value is a reference-based analytical projection, not a measured system latency.

8.1- CMOS Fallback Latency Sensitivity: Because the CMOS fallback implementation not measured in hardware, no single CMOS latency value assumed. Instead, the effect of different hypothetical CMOS latencies evaluated. The results presented in Table 6.

Table 6. AI Latency and CMOS Fallback Sensitivity.

CMOS fallback (ns)	AI Latency (ns)	Expected fallback contribution (ns)	Hybrid latency (ns)	Overhead
10	44.40	0.5167	44.9167	1.16%
25	44.40	1.2917	45.6917	2.91%
50	44.40	2.5833	46.9833	5.82%
100	44.40	5.1667	49.5667	11.64%
250	44.40	12.9167	57.3167	29.09%
500	44.40	25.8333	70.2333	58.18%

As shown in Table 6, the 5.17% fallback rate limits its average latency contribution when the CMOS path is relatively fast. However, the actual latency impact can only be established after hardware implementation and measurement.

9- Discussion

The experimental results support the main system-level objective of the proposed architecture. The AI-only model achieved 99.28% accuracy, while the hybrid system achieved 99.94%. Only 5.17% of the test samples routed to the CMOS path, and 12 of the 13 AI errors corrected.

These results show that the confidence-based fallback can improve classification reliability while keeping the secondary path limited to a small part of the workload. The AI controller remains the primary computation path, while the CMOS model used selectively for low-confidence cases.

The hardware analysis should be interpreted separately from the experimental results. The reported energy values derived from published hardware efficiency and the MAC count of the implemented network. They do not include peripheral circuits, Analog-to-Digital Converter (ADC)/ Digital-to-Analog Converter (DAC) conversion, routing, control, memory, or communication costs. These factors can affect the practical efficiency of memristive-CMOS systems (Huang et al., 2024; Aguirre et al., 2024).

The same limitation applies to latency. The 44.4-ns value based on a three-stage MVM mapping and the published 14.8-ns reference. Actual latency will also depend on data transfer, sensing, conversion, control, communication, and the CMOS fallback implementation.

Therefore, the hardware values should consider reference-based analytical projections, while the accuracy, error correction, AI coverage, and fallback rate experimentally obtained results.

10- Limitations

The proposed hardware not fabricated. Therefore, none of the reported energy or latency values should interpreted as chip-level measurements.

The energy estimates derived from published hardware references and the MAC counts of the implemented and reference architectures. The memristive energy estimate also depends on the interpretation of the published TOPS/W value. Two cases therefore reported to make this assumption explicit.

The AI latency projected from the published ReRAM-CIM computing time under a three-stage sequential MVM assumption. Actual hardware scheduling and data movement may produce a different latency. CMOS fallback latency has not been measured and therefore evaluated through sensitivity analysis rather than reported as a single hardware value.

The CMOS energy estimate based on a published accelerator reference. It should not interpreted as the measured energy of the software CMOS model used in the experiment.

Another limitation is the absence of physical memristor non-idealities in the current implementation. Device-to-device variation, conductance variation, programming errors, sensing noise, and other analog imperfections not measured on fabricated devices. ADC/DAC overhead, interconnect effects, and thermal effects not included in the analytical model.

Finally, the experimental evaluation is limited to a four-class classification task with 1,800 test samples. Larger datasets, more diverse workloads, and hardware experiments are required to assess the scalability and robustness of the proposed approach.

12- Future Work

Future work should focus on hardware validation. The AI controller, confidence mechanism, and CMOS fallback should implemented and measured together so that controller, sensing, conversion, and communication costs are included.

Device variation, conductance drift, analog noise, and other hardware non-idealities should evaluated experimentally. Hardware-aware training can then investigated to improve robustness.

Adaptive confidence thresholds should also studied as a trade-off among accuracy, energy, and latency. Further evaluation should include larger neural networks and more diverse workloads to determine whether the proposed hybrid strategy can scale beyond the current four-class task.

13- Conclusion

This work presented a hybrid AI-CMOS architecture in which a compact neural controller performs the primary classification and a CMOS reference model handles low-confidence cases. The AI controller requires 2,400 MACs per inference, while the CMOS reference requires 1,344 MACs.

On 1,800 test samples, the AI-only model achieved 99.28% accuracy. The confidence-based mechanism routed 93 samples to the CMOS path and increased the final accuracy to 99.94%. Twelve of the 13 AI errors corrected, leaving only one error.

The hardware analysis provides reference-based analytical projections rather than measured chip-level results. The AI energy estimated at 141.59 pJ/inference in Case A and 283.19 pJ/inference in Case B. The three-stage AI latency projection is 44.4 ns/inference, while the expected CMOS fallback contribution is 1.3263 pJ per original test sample. These values derived from published hardware references and architectural MAC counts and should not interpreted as measurements of a fabricated implementation. The main experimental contribution is the demonstrated improvement in classification reliability with limited fallback usage.

References:

- F. Aguirre, A. Sebastian, M. Le Gallo, et al., "Hardware implementation of memristor-based artificial neural networks," *Nature Communications*, 15, 1974, 2024.
- D. Bonnet, T. Hirtzlin, A. Majumdar, et al., "Bringing uncertainty quantification to the extreme-edge with memristor-based Bayesian neural networks," *Nature Communications*, 14, 7530, 2023.
- F. Cai, J. M. Correll, S. H. Lee, et al., "A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations," *Nature Electronics*, 2, 290–299, 2019.
- W.-H. Chen, C. Dou, K.-X. Li, et al., "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors," *Nature Electronics*, 2, 420–428, 2019.
- T. Dalgaty, F. Moro, Y. Demirağ, et al., "Mosaic: in-memory computing and routing for small-world spike-based neuromorphic systems," *Nature Communications*, 15, 142, 2024.
- L. R. Everson, M. Liu, N. Pande, and C. H. Kim, "An Energy-Efficient One-Shot Time-Based Neural Network Accelerator Employing Dynamic Threshold Error Correction in 65 nm," *IEEE Journal of Solid-State Circuits*, 54(10), 2777–2785, 2019.
- Y. Huang, T. Ando, A. Sebastian, M.-F. Chang, J. J. Yang, and Q. Xia, "Memristor-based hardware accelerators for artificial intelligence," *Nature Reviews Electrical Engineering*, 1, 286–299, 2024.
- A. James, "Energy efficiency and design challenges in analogue memristive chips," *Nature Reviews Electrical Engineering*, 1, 6–7, 2024.
- P. Mannocci, G. Larelli, M. Bonomi, and D. Ielmini, "Achieving high precision in analog in-memory computing systems," *npj Unconventional Computing*, 3, 1, 2026.
- Z. Xu, J. Liu, S. Huang, et al., "Fault-free analogue computing with imperfect hardware," *Nature Electronics*, 9, 811–824, 2026.