Rebuilding a Reservoir-Computing Delta-Sigma Compute-in-Memory Classifier and Testing Ten Extensions in Simulation

Nithin Purushothama*
nithinpurushothama@gmail.com • nithinpuru.com

November 2026

Abstract

Gupta *et al.* [1] classify sensor signals inside the sensor with a switched-capacitor (SC) reservoir that extracts features and a delta-sigma-modulator (DSM) compute-in-memory (CIM) output layer whose precision is set at run time by the oversampling ratio (OSR). This report rebuilds that architecture as a block-level model with per-chip mismatch, adds two transistor-level SKY130 blocks (a 6T SRAM cell array and the neuron's amplifier and comparator, both with per-device Pelgrom mismatch), and uses the model to test ten extensions against the published flow. The public ECG200 dataset stands in for the paper's datasets, so only comparisons between methods are meaningful.

The baseline behaves as published: accuracy rises with OSR and flattens (78.9 % at OSR 10, 80.0 % at OSR 20), and a readout trained on one chip loses 1.8 points on another. The main results are:

- **Label-free per-chip normalisation** (new): one readout trained on a generic model plus each chip's own output statistics from unlabelled inputs reaches 79.0 %, the same as the paper's per-chip in-the-loop training (78.9 %), and lifts the worst chip from 77.0 % to 77.9 %.
- **SRAM power-up values as the random input weights**: accuracy unchanged from 27 to 85 °C (79.8 %); at −40 °C 17.2 % of cells change and accuracy drops to 66.8 %, which a three-temperature enrolment brings back to 77.8 %.
- **Per-chip digital twin** training matches in-the-loop training (79.0 % vs 78.9 %) without using the chip for training.
- **A bitstream (XNOR) input layer** that moves the paper's off-chip input multiply on chip costs ≈1 point at 32 bits per sample.
- **A safe early-exit rule** saves 5 to 13 % of output-DSM cycles with identical decisions.
- **SKY130 neuron mismatch** is far smaller than the hand-tuned value used in round 1; with it the cross-chip drop vanishes, so the paper's measured drop must come from elsewhere.

Two ideas did not win and are reported as such: a pure DSM neuron and a multi-timescale reservoir.

1 Introduction

1.1 Motivation

Wearable and remote sensors spend most of their energy moving raw samples off the node. Classifying in the sensor and sending only the label removes that cost, but only if the classifier itself costs a few nanojoules. Gupta *et al.* [1] (Sanyal group, ASU, with Liao and Seo, Cornell Tech) do this with two ideas that suit analog circuits:

- a *reservoir* of SC leaky integrators [2, 3], whose random, fixed dynamics turn analog imperfection into part of the feature map, and
- a *DSM-CIM* output layer, where every SRAM cell only multiplies by a 1-bit stream, so the multiply is linear, and precision is traded for time through the OSR.

They report 97.0 % (stress) and 95.8 % (seismic) mean accuracy at 1.3 nJ and 4.25 nJ per classification in 28 nm CMOS, with about 3 points lost when a readout trained on one chip is used on another, which is why each chip is trained in the loop.

*Simulation code, SKY130 testbenches and data are available on request. Every number in this report is generated from the simulation output files, not typed by hand.

1.2 Contributions

1. **A block-level rebuild** of the published signal chain with per-chip mismatch in every neuron and CIM column (Secs. 3 and 4).
2. **Two transistor-level SKY130 blocks** with per-device mismatch: 128 6T SRAM cells powered up at three temperatures, and 240 neuron amplifiers and comparators (Sec. 5).
3. **Ten extensions**, each run on the same model and data as the baseline (Sec. 6), including two honest negative results.

1.3 What this report is not

All results are simulation. The paper's WESAD [4] and seismic datasets are not used; the UCR ECG200 set [5, 6] (200 single-lead ECG beats, normal vs. myocardial ischaemia) stands in. SKY130 is not the paper's 28 nm process. The SKY130 numbers ground the *magnitude* of mismatch effects; they are not a claim about the paper's silicon.

2 The Published Architecture

Fig. 1 shows the signal chain in the order the data flows. The neuron update, as modelled here, is

$$v_j[n+1] = a_j\, v_j[n] + G_{i,j}\, u_j[n] + G_{f,j}\big(H(v_{j-1}[n]) + H(v_{j+1}[n])\big) + \eta, \qquad a_j = 1 - \frac{1 + G_{i,j} + 2G_{f,j}}{A_j}, \quad (1)$$

where $u_j[n] = \frac{1}{\sqrt{D}} \sum_d W_{jd}\, x[nD + d]$ is the random binary input projection (off-chip in the paper), $A_j \approx 25$ is the finite amplifier gain that makes the integrator leak, $H(\cdot)$ is the neuron's soft saturation, and a neuron whose state crosses V_{th} is reset together with its two neighbours (lateral inhibition). The time-averaged neuron outputs are converted to bitstreams by an input DSM and classified by the DSM-CIM layer with 4-bit weights; only its output DSM is decimated.

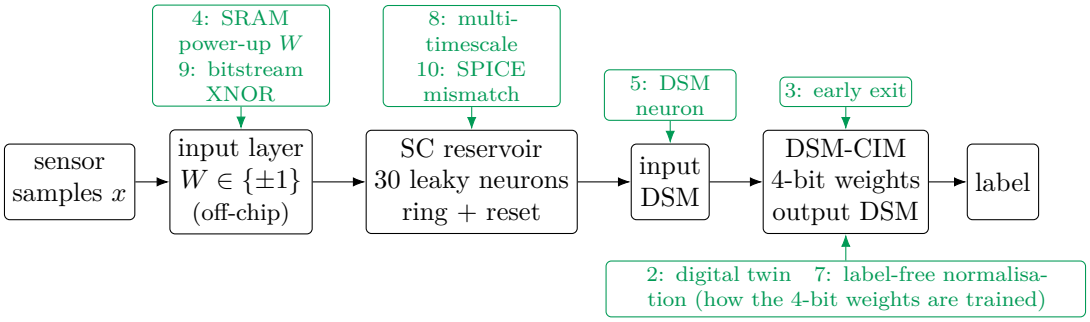

Figure 1: Signal chain of [1] (black) and where each simulated extension acts (green).

Table 1: Numbers reported in [1] that the model uses or compares against.

Process, area	28 nm CMOS, 0.1 mm^2
Reservoir	30 SC neurons, amplifier gain $\approx$ 25, 14.8 fF sampling capacitors, 7.2 µW at 31.25 kHz
Output layer	30×32 9T1C SRAM CIM, 4-bit weights, 1-bit DSM, 12.3 µW at 312.5 kHz (OSR 10)
Accuracy	97.0 % stress, 95.8 % seismic (3 chips); about 3 points lost across chips
Input layer	binary random W, multiplied off-chip

3 Methodology

- **Transistor level.** SKY130 `nfet_01v8/pfet_01v8` typical models [8] in ngspice [9]. The raw model files give ngspice no local mismatch, so the PDK's `vth0_slope` term was turned into an instance parameter: every transistor gets its own threshold shift $\sigma_{\Delta V_{th}} = A_{VT}/\sqrt{WL}$ [7].
- **Block level.** Equation (1) for the reservoir; a first-order 1-bit DSM per feature; a charge-domain MAC per cycle with per-column gain error and noise, followed by a first-order output DSM whose count is the decision. Every chip draws its own amplifier gains, capacitor ratios, offsets, thresholds, saturation levels and column gains. The random input matrix is the same on every chip, as in the paper.

- **Training.** Logistic regression on the time-averaged neuron outputs, then quantised to 4-bit signed weights; the MAC gain is set so the output DSM uses its full scale.
- **Data and statistics.** ECG200, $D = 32$ samples per reservoir step, 8 simulated chips, 10 random 60/40 train/test splits; every accuracy is the mean over chips and splits.
- **Round-1 mismatch scale.** In round 1 the neuron mismatch was scaled by 4× so that the cross-chip drop came close to the paper's; Sec. 5.2 replaces this with SKY130 numbers.

4 Baseline Reproduction

Table 2 and Fig. 2(a) show the published behaviour: accuracy climbs steeply up to OSR 6 and is flat from 10 to 20, which matches the paper's tables where many entries are identical across OSR 10, 15 and 20. The cross-chip drop (1.8 points) is of the same order as the paper's measured 3 points. The software reference (82.3 %) uses all 96 raw samples and bounds what a 30-feature readout can reach on this dataset.

Table 2: Baseline accuracy of the rebuilt architecture on ECG200 (8 chips × 10 splits).

OSR	same chip [%]	note
1	43.0	
2	66.0	
4	75.1	
6	77.4	
10	78.9	paper operating point
15	80.1	
20	80.0	
cross chip, OSR 10	77.0	drop 1.8 pts
software, 96 raw samples	82.3	4-bit logistic regression

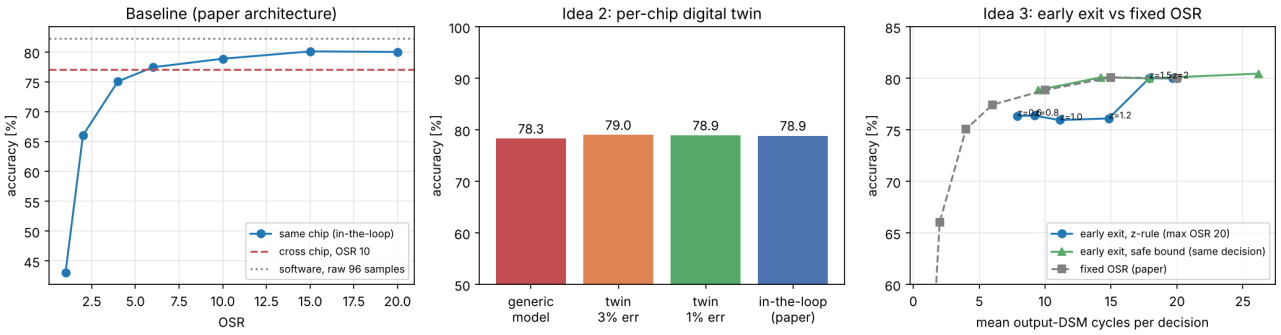

Figure 2: (a) Baseline accuracy vs. OSR, in the loop and across chips. (b) Idea 2: readout trained on a per-chip digital twin. (c) Idea 3: early exit of the output DSM.

5 Transistor-Level Circuits in SKY130

5.1 6T SRAM power-up array (for idea 4)

Idea 4 asks whether the power-up state of ordinary SRAM cells can supply the reservoir's random ±1 input weights on chip, removing the off-chip multiply's weight storage. 128 symmetric 6T cells (all devices 0.42 µm/0.15 µm) were powered up with V_{DD} ramping to 1.8 V in 1 µs, every transistor with its own mismatch (Fig. 3). A common offset source V_{off} in series with one gate of every cell was swept from −80 mV to 80 mV in 4 mV steps; the value at which a cell's power-up state flips is that cell's input-referred *skew*. A cell whose skew changes sign with temperature is temperature-unstable; with thermal noise σ_n its flip probability is $\Phi(-|\text{skew}|/\sigma_n)$.

Results. The power-up values are balanced (55.5 % ones at 27 °C). From 27 to 85 °C only 0.8 % of cells change, but at −40 °C 17.2 % change; 82 % of cells keep their value at all three temperatures (Fig. 4). The asymmetry is a real finding: at low temperature the skew distribution narrows towards zero in this cell, so cold is the hard corner.

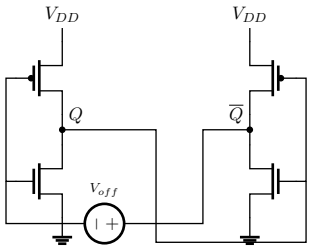

Figure 3: SKY130 6T cell used for the power-up study (access transistors off, not drawn). V_{off} is swept to measure each cell's skew; every transistor has its own Pelgrom threshold shift.

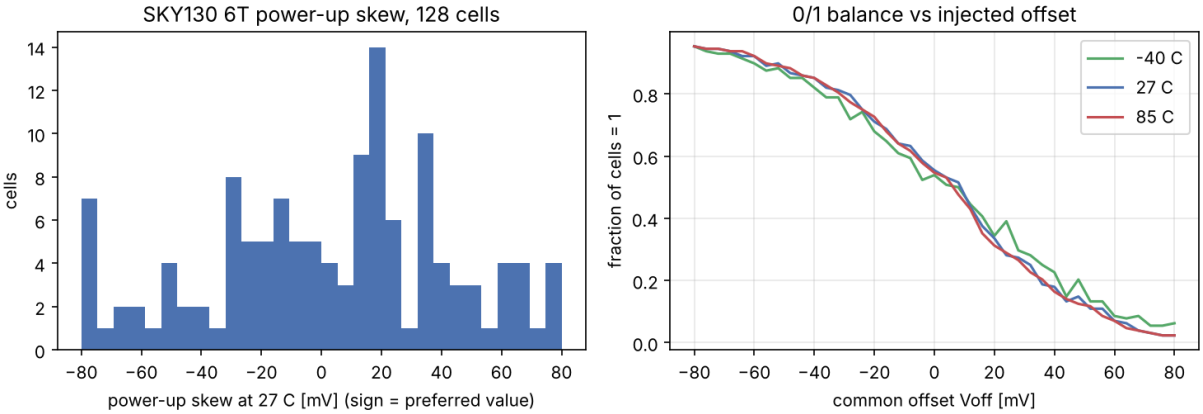

Figure 4: SKY130 SRAM power-up Monte Carlo: skew of each cell at three temperatures and the resulting flip statistics.

5.2 Neuron amplifier and comparator (for idea 10)

The paper's neuron needs a low-gain amplifier ($A \approx 25$) and a threshold comparator. The cheapest self-biased realisation of both is a CMOS inverter: biased at its trip point it is an amplifier whose trip point is the integrator's virtual ground; a second, minimum-size inverter acts as the comparator (Fig. 5). With $L = 0.18\,\mu\text{m}$ at 1.2 V the amplifier gain is 21.2, close to the paper's value. Over 8 chips $\times$ 30 neurons, each with its own device mismatch, the gain spread within a chip is only 0.7 %, the amplifier offset is 6.6 mV and the comparator offset 10.1 mV (1σ).

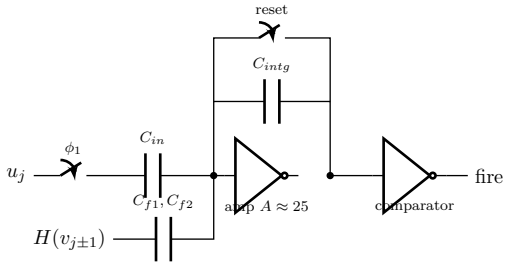

Figure 5: Simplified SC neuron with inverter-based amplifier and comparator (switch phases simplified). The two inverters are the transistor-level part of the Monte Carlo; capacitor ratios keep the round-1 1 % assumption.

Table 3: Round-1 neuron mismatch assumptions vs. SKY130 Monte Carlo, and the resulting cross-chip drop.

Neuron parameter	round-1 assumption	SKY130 Monte Carlo
amplifier gain spread (σ/μ)	40 %	0.7 %
amplifier offset (σ)	16 mV	6.6 mV
comparator offset (σ)	24 mV	10.1 mV
cross-chip drop	1.8 pts	0.0 pts

6 Proposed Extensions and Results

Table 4 summarises all ten ideas. Ideas 1 to 6 were proposed and tested in round 1; ideas 7 to 10 are new. Each is described below with its motivation, method, result and catch.

6.1 Idea 2: a per-chip digital twin instead of in-the-loop training

Motivation. In-the-loop training needs the chip, the labelled training set and a training run for every chip, which does not scale to production.

Table 4: Summary of the extensions (ECG200, 8 chips × 10 splits; in-the-loop baseline 78.9 % at OSR 10).

#	Idea	Result	Verdict
1	R_{on}/g_m sizing of the CIM buffers	not simulated here	not simulated
2	per-chip digital twin	79.0 % vs. 78.9 % in the loop, chip not used for training	win
3	safe early exit of the output DSM	5 to 13 % fewer cycles, same decisions	win
4	SRAM power-up values as input weights	79.8 % at 85 °C; 66.8 % at −40 °C, 77.8 % with 3-temperature enrolment	win
5	pure DSM neuron (no input DSM)	6 to 7 points worse at equal bits	no win
6	PUF-authenticated labels	not simulated here	not simulated
7	label-free per-chip normalisation	79.0 % from a generic model, worst chip 77.9 % vs. 77.0 %	win
8	multi-timescale reservoir	73.4 % vs. 78.9 %; ECG200 too short to test	no win
9	bitstream XNOR input layer on chip	78.1 % at 32 bits/sample vs. 78.9 % off-chip	partial
10	SKY130-grounded neuron mismatch	cross-chip drop 0.0 pts: neuron mismatch is not the cause	win

Method. Each chip is characterised once with impulse, step and saturation sweeps on each neuron. The fitted leak, gains, threshold, saturation and offset (with a stated characterisation error of 1 or 3 %) define a twin; the readout is trained on the twin's features and loaded into the real chip.

Result. 79.0 % with 3 % characterisation error and 78.9 % with 1 %, the same as in-the-loop training (78.9 %) and better than a generic model (78.3 %) or another chip's readout (77.0 %), Fig. 2(b).

Catch. It needs per-neuron test access and a characterisation routine; idea 7 removes even that.

6.2 Idea 3: safe early exit of the output DSM

Motivation. Accuracy is flat from OSR 10 to 20, so many decisions are settled long before the last cycle.

Method. After k of K cycles the remaining cycles can move the output count by at most $K - k$, plus one for the integrator residue. If $|\text{count}| > K - k + 1$ the sign of the final count cannot change, so the macro stops. This gives *exactly* the decision of the full run (noise aside). Faster, approximate z-score rules were also tested.

Result. 5 to 13 % fewer cycles at identical accuracy (Fig. 2(c)); the approximate rules lose accuracy on this dataset.

Catch. The saving depends on how confident the classifier is; on the paper's tasks, where accuracy is 96 to 97 %, margins are larger and the saving should be larger, but this is not shown here.

6.3 Idea 4: SRAM power-up values as the random input weights

Motivation. The paper performs the input projection off-chip. A random ±1 matrix is exactly what SRAM power-up mismatch provides [10, 11], for free.

Method. Cells are drawn from the SKY130 population of Sec. 5.1. Weights are enrolled at 27 °C and the readout is trained on them; at inference the cells are read again at another temperature with 1 mV thermal noise.

Result. Table 5: power-up weights work as well as ideal random ones at 27 and 85 °C. At −40 °C the readout no longer matches the changed weights. Masking weak cells by their skew does not fix it (71.2 %), but enrolling at three temperatures and keeping only cells that agree (82 % of cells) restores 77.8 % at both extremes.

Catch. Enrolment at three temperatures is a test-time cost; the cell here is a generic symmetric 6T cell, not a foundry bit cell.

Table 5: Idea 4: accuracy with reservoir input weights taken from SKY130 SRAM power-up states.

Configuration	accuracy [%]	cells changed vs enrolment [%]
ideal random W (paper, off-chip)	78.9	0.0
power-up W, read at 27 °C	79.8	0.0
power-up W, read at 85 °C	79.8	0.6
power-up W, read at −40 °C	66.8	17.6
85 °C, skew mask 4 mV	78.4	0.4
−40 °C, skew mask 4 mV	71.2	16.4
−40 °C, 3-temperature enrolment	77.8	0.0
85 °C, 3-temperature enrolment	77.8	0.0

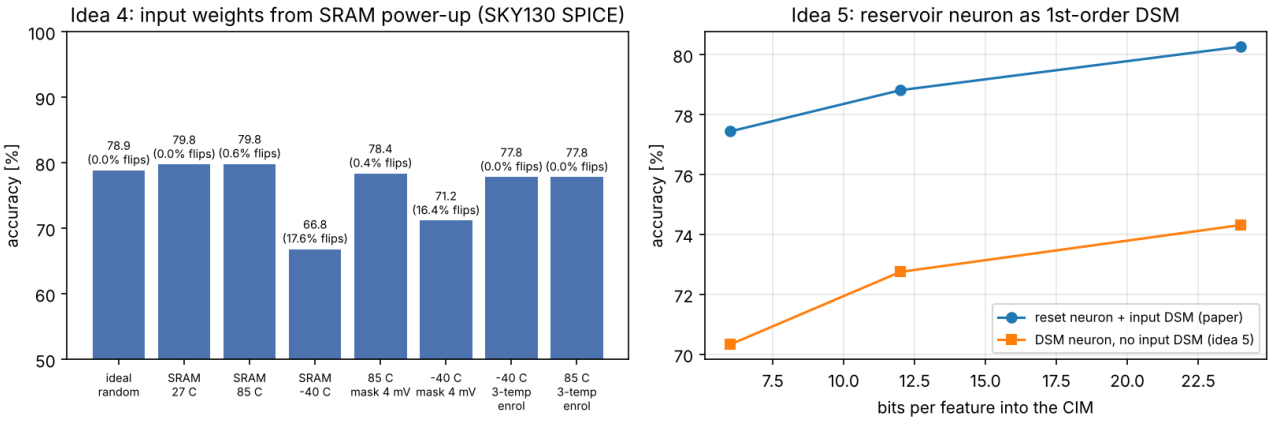

Figure 6: Idea 4 (power-up weights vs. temperature) and idea 5 (DSM neuron vs. paper neuron).

6.4 Idea 5: a pure delta-sigma neuron (negative result)

Motivation. If each neuron were itself a first-order DSM (integrate and subtract instead of reset), its output would already be a bitstream and the input DSM could be removed.

Result. At an equal number of bits the DSM neuron is 6 to 7 points worse (Fig. 6). The paper's reset neuron with lateral inhibition carries transient information that a linear DSM neuron averages away. Not worth pitching without a better neuron.

6.5 Idea 7 (new): label-free per-chip normalisation

Motivation. The cross-chip loss means the readout is tuned to one chip's feature statistics. If the differences between chips are mostly per-neuron offsets and gains, a chip can measure them itself without labels.

Method. One readout is trained on a generic, nominal model. Each chip runs its unlabelled training inputs once and records the mean μ_j and spread σ_j of every neuron's output (a running sum and sum of squares, or simply firmware). The firmware then folds $1/\sigma_j$ into the 4-bit weights and $-\sum_j w_j \mu_j / \sigma_j$ into the bias. No labels, no characterisation and no training run on the chip. In machine learning the same principle is known as test-time adaptation of batch-normalisation statistics [12]; using it to replace per-chip in-the-loop training of an analog reservoir is the new part here.

Result. Table 6 and Fig. 7(a): 79.0 % from the generic model, the same as in-the-loop training, and the worst chip improves from 77.0 % (in the loop) and 73.2 % (another chip's readout) to 77.9 %. Training on an ensemble of simulated chips, the obvious alternative, does not help (77.0 %).

Catch. It corrects offset and gain per neuron, not changes in the shape of the dynamics; it needs a short unlabelled warm-up per chip.

6.6 Idea 8 (new): a multi-timescale reservoir (inconclusive here)

Motivation. All neurons share one leak ($a \approx 0.954$). Giving groups of neurons different time constants, with one extra switched leak capacitor per neuron, should let the reservoir remember both fast and slow structure.

Table 6: Idea 7: where the readout comes from vs. accuracy on each real chip.

Readout trained on	mean [%]	worst chip [%]
the same chip, in the loop (paper)	78.9	77.0
another chip	77.0	73.2
a generic nominal model	78.3	75.5
an ensemble of 6 simulated chips	77.0	74.8
generic model + label-free normalisation (idea 7)	79.0	77.9
another chip + label-free normalisation (idea 7)	79.1	77.9

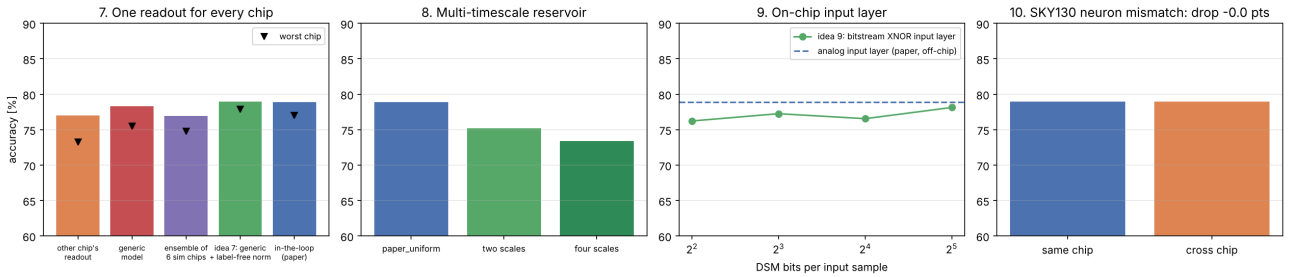

Figure 7: Round-2 ideas: (a) label-free normalisation, (b) multi-timescale reservoir, (c) bitstream input layer, (d) cross-chip test with SKY130 neuron mismatch.

Result. 75.2 % with two time constants and 73.4 % with four, below the uniform 78.9 %. With $D = 32$ an ECG200 beat gives only three reservoir steps, so there is no long history for slow neurons to use. The idea needs the paper's longer stress and seismic sequences before it can be judged.

6.7 Idea 9 (new): a bitstream XNOR input layer on chip

Motivation. The input projection $u_j = \sum_d W_{jd} x_d$ is the one part of the network that the paper computes off-chip.

Method. Each input sample is converted by a first-order DSM into L bits. Each neuron forms $\mathrm{XNOR}(W_{jd}, b_d)$ over its D taps and sums them on a charge-sharing capacitor bank; the average over the L bits is the projection. The hardware is 30×32 XNOR gates and one capacitor bank per neuron; there is no multiplier.

Result. 78.1 % at $L = 32$ against 78.9 % for the ideal analog projection, and 77.2 % at $L = 8$ (Fig. 7(c)). With idea 4 supplying W from SRAM power-up, the whole network becomes on-chip.

Catch. About one point is lost at 32 bits per sample, and the input DSM must run L times faster than the sample rate.

6.8 Idea 10 (new): grounding neuron mismatch in SKY130

Motivation. Round 1 scaled the neuron mismatch by 4× to reproduce the paper's cross-chip drop. That is a fit, not a prediction.

Method. Each neuron's amplifier gain, amplifier offset and comparator offset are taken from the SKY130 Monte Carlo of Sec. 5.2; capacitor ratios, saturation and column gains keep their round-1 base values without the 4× scale.

Result. Table 3: the transistor-level spreads are much smaller than assumed, and with them the cross-chip drop disappears (79.0 % same chip, 79.0 % cross chip). Local device mismatch in the neuron core therefore cannot explain the paper's measured 3 points. The likely sources are die-to-die (global) variation, which shifts every neuron of a chip together, the input capacitor matching, or the CIM columns. This is a useful question for the authors, and it tells a designer where calibration effort should go: idea 7 corrects exactly such per-chip shifts.

Catch. Only local mismatch at the typical corner was simulated; the inverter amplifier is a stand-in for the paper's amplifier.

7 Further Ideas Not Yet Simulated

- **Idea 1, R_{on}/g_m sizing of the CIM unity-gain buffers** (my own sizing metric for dynamic settling): size the buffers that drive the DSM input capacitors for a target settling error rather than by margin.
- **Idea 6, PUF-authenticated labels**: the same SRAM power-up fingerprint that supplies W (idea 4) can sign each transmitted label, so a receiver knows which sensor produced it.
- **On-chip readout adaptation from bitstreams** (ambitious): a sign-sign LMS update of the 4-bit weights driven by the output DSM's bitstream, so the chip tracks slow drift after deployment without a host.
- **Temperature from the power-up array** (ambitious): the fraction of cells that differ from enrolment rises steeply below room temperature (Sec. 5.1), so the same array is a coarse on-chip thermometer that could trigger re-enrolment.

8 Conclusion

The reservoir and DSM-CIM classifier of [1] was rebuilt at block level with per-chip mismatch, extended with two SKY130 transistor-level studies, and used to test ten extensions under identical conditions.

- **Per-chip training can be replaced.** A digital twin matches in-the-loop accuracy, and label-free per-chip normalisation does so from a generic model with no characterisation at all, while improving the worst chip.
- **The input layer can come on chip.** SRAM power-up values supply W (with a three-temperature enrolment for cold operation), and a bitstream XNOR layer computes the projection at a cost of about one point.
- **The output DSM can stop early** with no change in decisions.
- **Transistor-level grounding matters.** SKY130 neuron mismatch is too small to explain the paper's cross-chip drop, which points at global variation or the CIM as the place to look.
- **Two ideas were negative** (pure DSM neuron) or untestable on this dataset (multi-timescale reservoir), and are reported as such.

Acknowledgments. The architecture and all reported numbers of the original design are from Gupta, Damodaran, Liao, Seo and Sanyal [1]. Any error in the rebuild is mine.

A Limitations

- **Dataset.** ECG200 replaces WESAD and the seismic set; absolute accuracies are not comparable with the paper's.
- **Block-level model.** The reservoir and CIM are behavioural; switch charge injection, kT/C noise of each capacitor and the CIM bit cells are not transistor-level.
- **Process.** SKY130 typical corner, local mismatch only; no global corners, no layout.
- **SRAM cell.** Generic symmetric 6T sizing, not a foundry bit cell; thermal noise is a parameter.
- **Statistics.** 8 chips × 10 splits of a 200-sample dataset; differences below about one point are within noise.

B Reproducing the Results

Step	Command (in `demos/Seo`)
SKY130 models + mismatch hook	`bash ../common/setup_sky130.sh`
SRAM power-up Monte Carlo	`python3 spice/sram_powerup_mc.py` (about 25 min on 4 cores)
Neuron amplifier Monte Carlo	`python3 spice/neuron_amp_mc.py` (about 30 s)
Round-1 and round-2 experiments	`python3 run_experiments.py`; `python3 run_new_ideas.py`
This report	`cd report && python3 make_tex.py && latexmk -pdf main.tex`

References

[1] T. Gupta, V. Damodaran, Y. Liao, J.-s. Seo and A. Sanyal, "Delta-sigma modulator-based compute-in-memory neural network with analog feature extraction and classification for edge sensors," in *Proc. IEEE ISCAS*, 2026, doi:10.1109/ISCAS66217.2026.11562729.

[2] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," GMD Report 148, 2001.

[3] L. Appeltant *et al.*, "Information processing using a single dynamical node as complex system," *Nature Communications*, vol. 2, 468, 2011.

[4] P. Schmidt *et al.*, "Introducing WESAD, a multimodal dataset for wearable stress and affect detection," in *Proc. ACM ICMI*, 2018.

[5] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Ph.D. thesis, Carnegie Mellon University, 2001.

[6] H. A. Dau *et al.*, "The UCR time series archive," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, 2019.

[7] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE JSSC*, vol. 24, no. 5, 1989.

[8] SkyWater Technology and Google, "SkyWater SKY130 open source PDK," `https://github.com/google/skywater-pdk`.

[9] H. Vogt *et al.*, "Ngspice user's manual," version 42, 2024.

[10] J. Guajardo, S. S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. CHES*, 2007.

[11] D. E. Holcomb, W. P. Burleson and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Computers*, vol. 58, no. 9, 2009.

[12] Y. Li, N. Wang, J. Shi, J. Liu and X. Hou, "Revisiting batch normalization for practical domain adaptation," arXiv:1603.04779, 2016.