

Supplementary Information

Compute-in-Memory Implementation of State Space Models for Event Sequence Processing

Xiaoyu Zhang^{1,§}, Mingtao Hu^{1,§}, Sen Lu¹, Soohyeon Kim¹, Eric Yeu-Jer Lee¹, Yuyang Liu¹

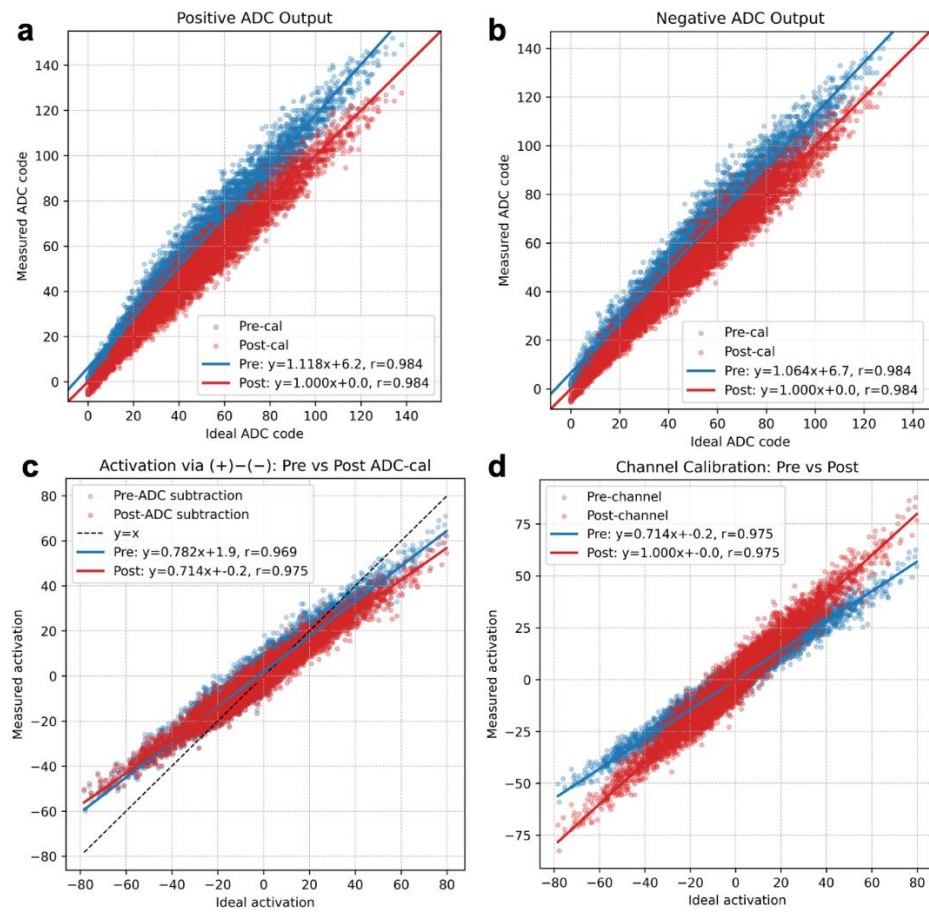
and Wei D. Lu^{1,*}

¹Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, USA.

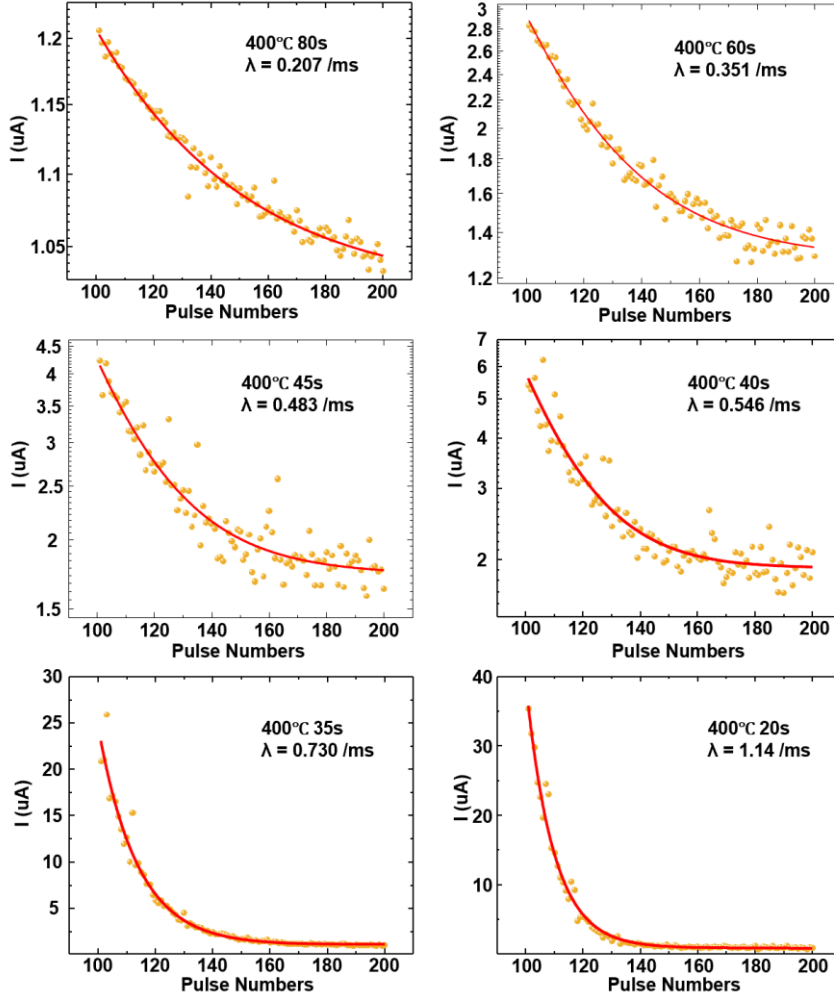
[§]These authors contributed equally.

* Correspondence to: wluee@umich.edu.

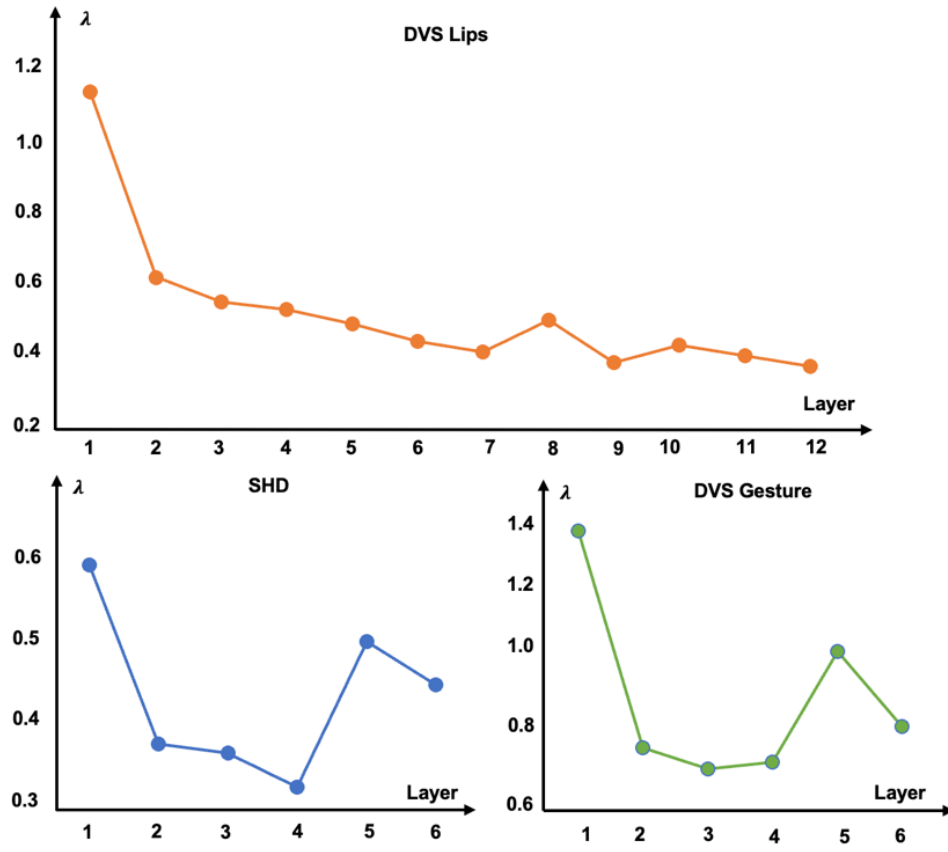
Supplementary Figures



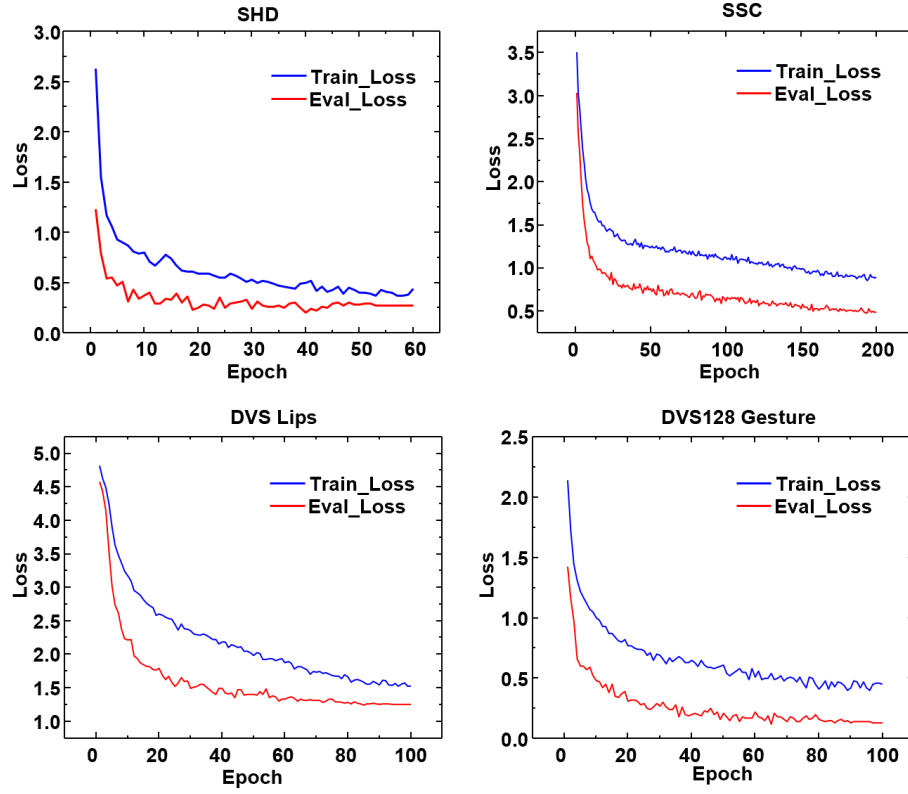
Supplementary Figure 1. Minimizing systematic errors through a two-stage calibration process. **a**, **b** Pre-calibration (blue) and post-calibration (red) outputs for the positive and negative columns through ADC, respectively, demonstrating the correction at the individual ADC-level (Stage I). **c** The combined activation after Stage I (ADC-level calibration and subtraction). While the correlation is much improved, some residual deviation from the ideal identity line (dashed black) remains. **d** The final output after Stage II (channel level calibration), which aligns almost perfectly with the ideal $y=x$ relationship.



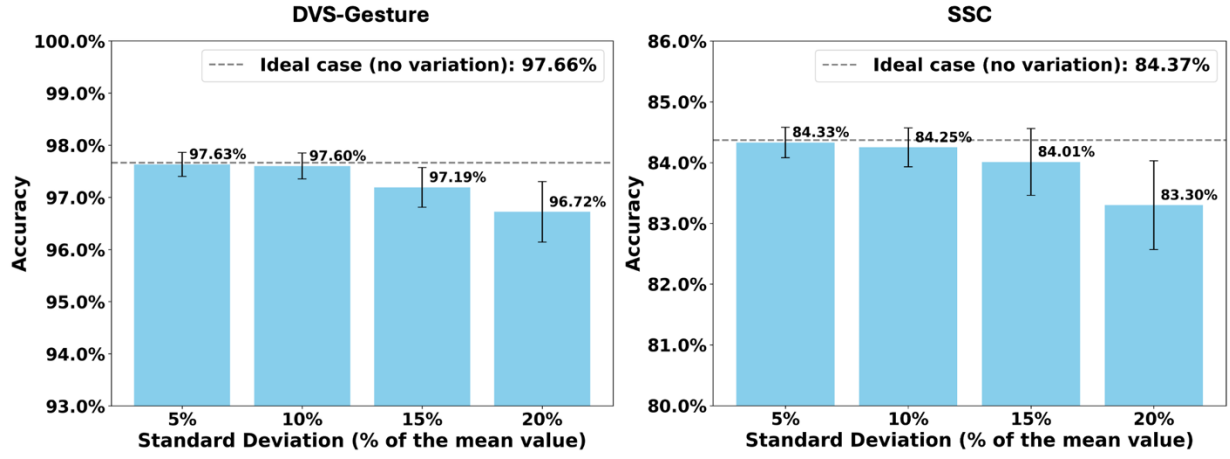
Supplementary Figure 2. Fabricated STM memristors with different decay rates The STM memristor's decay behavior can be tuned during the rapid thermal processing (RTP) process. The decay rate (λ) can be effectively tailored by varying the RTP time while keeping the annealing temperature at 400°C . This effect is attributed to changes in the oxygen vacancy density within the oxide layer: longer RTP durations reduce vacancy density, resulting in slower decay rates, as reflected in smaller λ values. The extracted decay constants span a broad range ($\lambda = 0.207 \text{ ms}^{-1}$ to 1.14 ms^{-1}), confirming that RTP duration serves as a viable process-level knob for engineering memristor temporal dynamics. The λ values were obtained by fitting the experimental data (yellow dots) to an exponential function (red curves).



Supplementary Figure 3. Learned λ values across layers when trained on DVS Lips, DVS Gesture and SHD dataset



Supplementary Figure 4. Training loss (blue line) and evaluation loss (red line) versus training epochs for the model for the four event-driven datasets (Spiking audio tasks SHD and SSC, and the event-based vision tasks DVS Lips and DVS128 Gesture).



Supplementary Figure 5. Model accuracy as a function of device variations in the decay constant λ . The decay constant λ of the STM memristor devices are modeled to follow a Gaussian distribution centered on the target mean value, and the standard deviation of the distribution is increased from 5% to 20% of the mean. Error bars represent results from 40 independent test runs at each variation level. The dotted line represents the ideal baseline accuracy without variation from the quantized-weight model.

Supplementary Tables

Dataset $\lambda(ms^{-1})$	SHD	SSC	DVS-Gesture	DVS-lips
First layer	0.55	0.35	1.15	1.15
Other layers	0.35	0.20	0.70	0.50

Supplementary Table 1 λ s selected for each dataset.

Supplementary Notes

Supplementary Notes 1: Selection of Fixed λ Values

To determine the fixed λ values, we examine λ learned from unconstrained models on various event-driven datasets. Notably, the first SSM layer tends to converge to a distinctly larger λ than subsequent layers, reflecting its role in rapidly filtering or dampening high-frequency input components. Deeper layers, by contrast, stabilize around smaller λ values that favor the accumulation of longer-term information. A special case is the SSC dataset, where unconstrained λ updates lead to divergence before the model fully converges. In this scenario, we extract λ values from the last stable iteration before encountering numerical errors. Based on these observations, our fixed- λ design adopts a two-tier approach: For the first layer, we assign a larger λ , mirroring the heightened filtering observed in the unconstrained models. This ensures the network can still capture the fast-decay dynamics critical for early-stage feature extraction. Then, we use a single shared λ for all remaining layers. Empirically, we set this to the approximate average of the layer-wise λ s found in the unconstrained model (excluding the first layer). The training then continues with the fixed λ s until final convergence.

Supplementary Notes 2: Calibrating the output of RRAM CIM chip

The analog nature of compute-in-memory (CIM) systems introduces several sources of error. Non-idealities in the RRAM devices and readout circuitry lead to a discrepancy between the ideal and measured VMM outputs. To address these cumulative errors, we employ a two-stage linear calibration procedure: an ADC-level calibration followed by a channel-level activation calibration.

ADC-level Calibration: The first stage corrects errors at the individual ADC. Since our system uses a differential mapping scheme, where positive (W^+) and negative (W^-) components of a weight are mapped to separate physical columns, the corresponding positive and negative ADC channels are calibrated independently. For each column c (either positive or negative), we perform a linear regression to model the relationship between the measured raw ADC codes (y_c) and the ideal codes (x_c), such that $y_c \approx s_c x_c + b_c$. The transformation is then applied to correct the raw output, yielding the ADC-calibrated code $\hat{y}_c$: $\hat{y}_c = a_c y_c + d_c$ where $a_c = \frac{1}{s_c}$ and $d_c = -\frac{b_c}{s_c}$. This step aligns the output of each physical channel to its ideal value as shown in Supplementary Figure 1a and b. The ADC-calibrated activation is then obtained by subtracting the calibrated negative column's output from the calibrated positive column's output (Supplementary Figure 1c).

Channel-level Calibration: The second stage corrects remaining channel-level (the difference of the calibrated positive column and the corresponding negative column) errors. We regress the ADC-calibrated channel activation ($\hat{y}_k$) for a given channel k on the final ideal activation (x_k), modeling the relationship as $\hat{y}_k \approx \alpha_k x_k + \beta_k$. Following the same principle, we get calibrated output $\tilde{y}_k = A_k \hat{y}_k + B_k$ where $A_k = \frac{1}{\alpha_k}$ and $B_k = -\frac{\beta_k}{\alpha_k}$.

These two sequential linear corrections can be mathematically fused into a single operation for direct physical hardware calibration. By substituting the Stage 1 equations into the Stage 2 equation, we derive a single transformation from the raw ADC codes ($y_{c,pos}$, $y_{c,neg}$) to the final output ($\tilde{y}_k$):

$$\begin{aligned} \tilde{y}_k &= A_k \hat{y}_k + B_k = A_k (\widehat{y_{c,pos}} - \widehat{y_{c,neg}}) + B_k = A_k [(a_{c,pos} y_{c,pos} + d_{c,pos}) + \\ & (a_{c,neg} y_{c,neg} + d_{c,neg})] + B_k = (A_k a_{c,pos}) y_{c,pos} - (A_k a_{c,neg}) y_{c,neg} + [A_k(d_{c,pos} - \\ & d_{c,neg}) + B_k] \end{aligned}$$

This fused equation provides the exact target parameters for physically tuning the ADC hardware: The W^+ ADC's gain is set to $G_{pos} = A_k a_{c,pos}$, and the W^- ADC's gain is set to $G_{neg} = A_k a_{c,neg}$ and The total differential offset is set to $O_{diff} = A_k(d_{c,pos} - d_{c,neg}) + B_k$. In practice, this allows us to program these fused coefficients directly into the circuitry (e.g., by adjusting ADC reference voltages for gain and comparator biases for offset). This corrects all linear errors at the point of conversion, achieving the final calibrated result shown in Supplementary Figure 1d and Fig. 2b without the overhead of sequential digital post-processing.

Supplementary Notes 3: Model's Performance against Decay Constant (λ) Variations.

Physical device fabrication is subject to inherent process variability, meaning the actual decay constant λ of a fabricated STM memristor device will likely deviate from its intended target value. To analyze the model's sensitivity to device-to-device variations, we modeled the device decay constant λ as following a Gaussian distribution centered on the layer's target mean λ . The standard deviation of the distribution was increased from 5% to 20% of the target mean λ . The model's accuracy was then evaluated by randomly assigning λ values to the STM memristor devices following the distribution, and repeated 40 times for each distribution to obtain statistically significant results.

The results, plotted in Supplementary Figure 5, show that the model maintains high accuracy even with substantial λ variations up to 10% of the target mean λ . This result suggests that the proposed network can tolerate inherent variability of physical devices, confirming the feasibility of our hardware co-design approach.