Supplementary Information

Fully Integrated Memristive Spiking Neural Network with Analog Neurons for High-Speed Event-Based Data Processing

Zhu Wang[1,3,#], Song Wang[1,#], Zhiyuan Du[1,3], Ruibin Mao[1,3], Yu Xiao[2], Hayden Kwok-Hay So[1], Peng Lin[2]*, and Can Li[1,3]*

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong SAR, China

[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China

[3]Centre for Advanced Semiconductors and Integrated Circuits, The University of Hong Kong, Pokfulam, Hong Kong SAR, China

[#]These authors contributed equally

Email: penglin@zju.edu.cn; canl@hku.hk

Contents

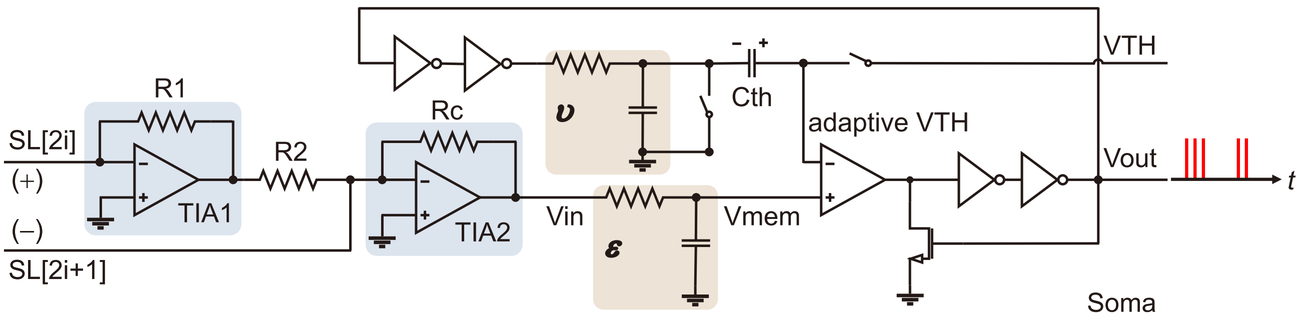

Figure S1. Detailed schematics of the neuron circuit. The design comprises a soma circuit and a differential TIA pair that converts incoming post-synaptic currents into a voltage. Resistors R1 and R2 are matched, while resistor Rc maps the post-synaptic voltage ("Vin") to the calculated post-synaptic vector-matrix multiplication result. The TIA2 confines Vin within the supply rails, thereby realizing a clipped activation function.

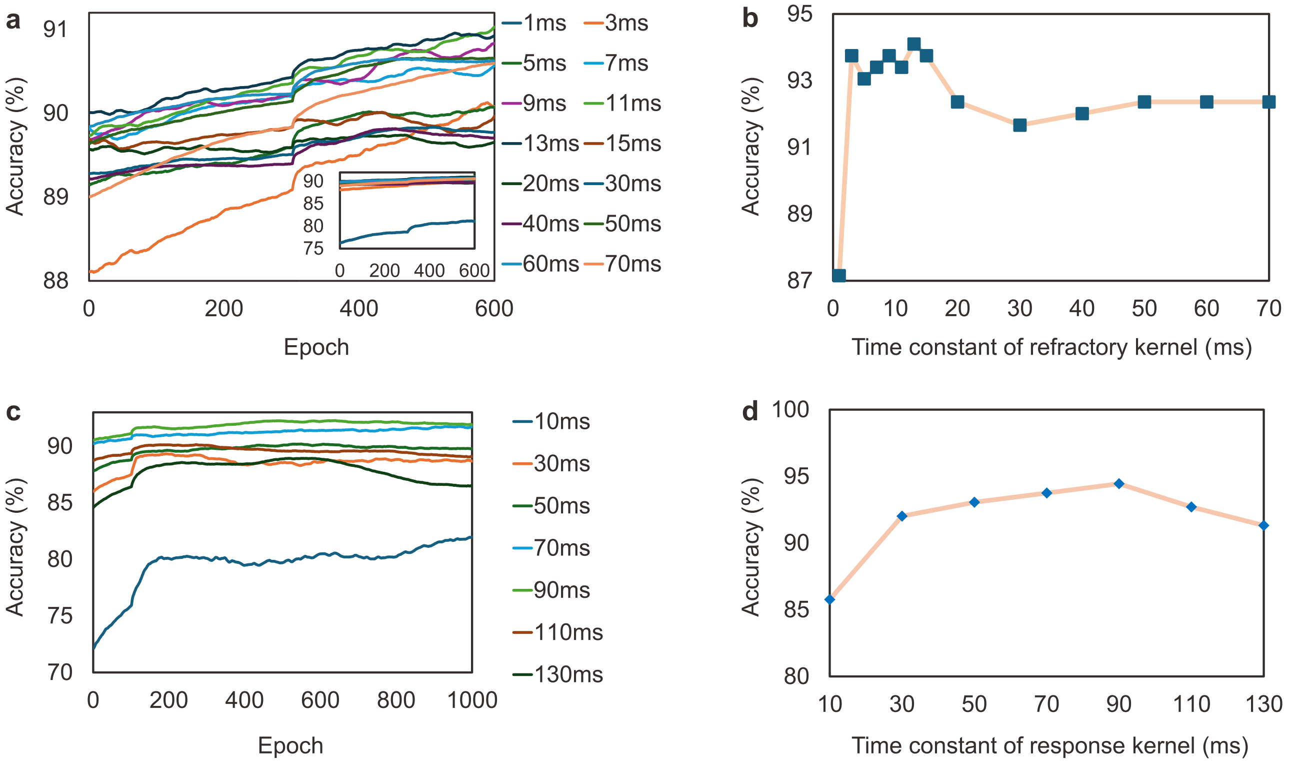

Figure S2. Impact of neuron time constants on SNN system inference accuracy. (a) Smoothed testing accuracy during training on the DVS128 Gesture dataset without data augmentation, varying the refractory kernel time constant (τ_r) from 1 ms to 70 ms with a fixed response kernel time constant (τ_s) of 50 ms. The inset shows a zoomed-out view of this plot. (b) Best testing accuracy from (a) versus τ_r. A τ_r of 5 ms was selected to maintain a compact neuron circuit, yielding a τ_s:τ_r ratio of 10:1. (c) Smoothed testing accuracy during training, varying τ_s from 10 ms to 130 ms while maintaining the τ_s:τ_r ratio at 10:1. (d) Peak testing accuracy from (c) versus τ_s. This panel, along with panel b, highlights the necessity of aligning neuronal time constants with the given task.

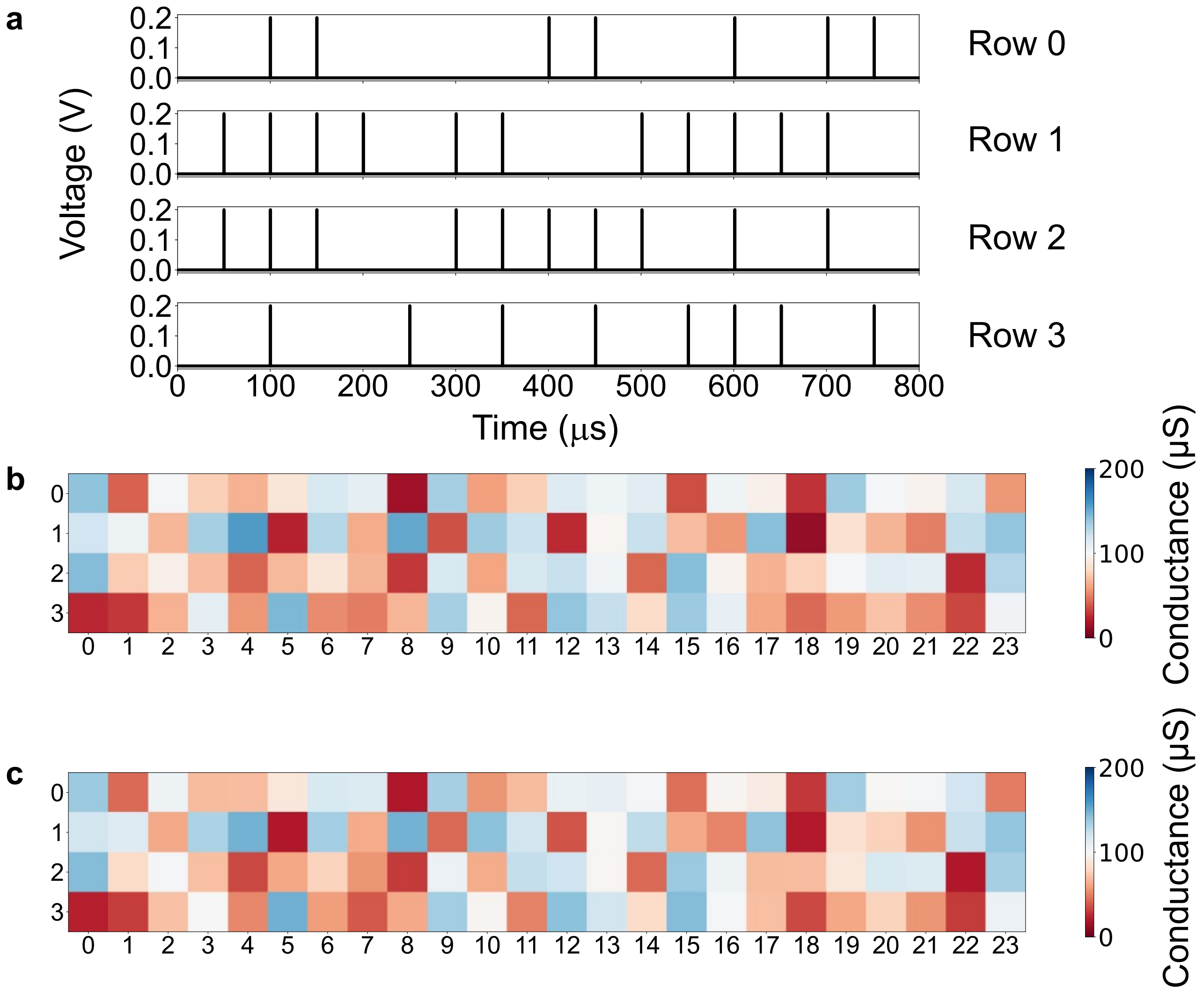

Figure S3. (a) Input spike trains used in the synaptic computation experiment in Fig. 3(e, f), (b) Corresponding readout conductances of the programmed array, and (c) Target conductance values. The post-synaptic voltage (Fig. 3f) was measured from column 0 of the array.

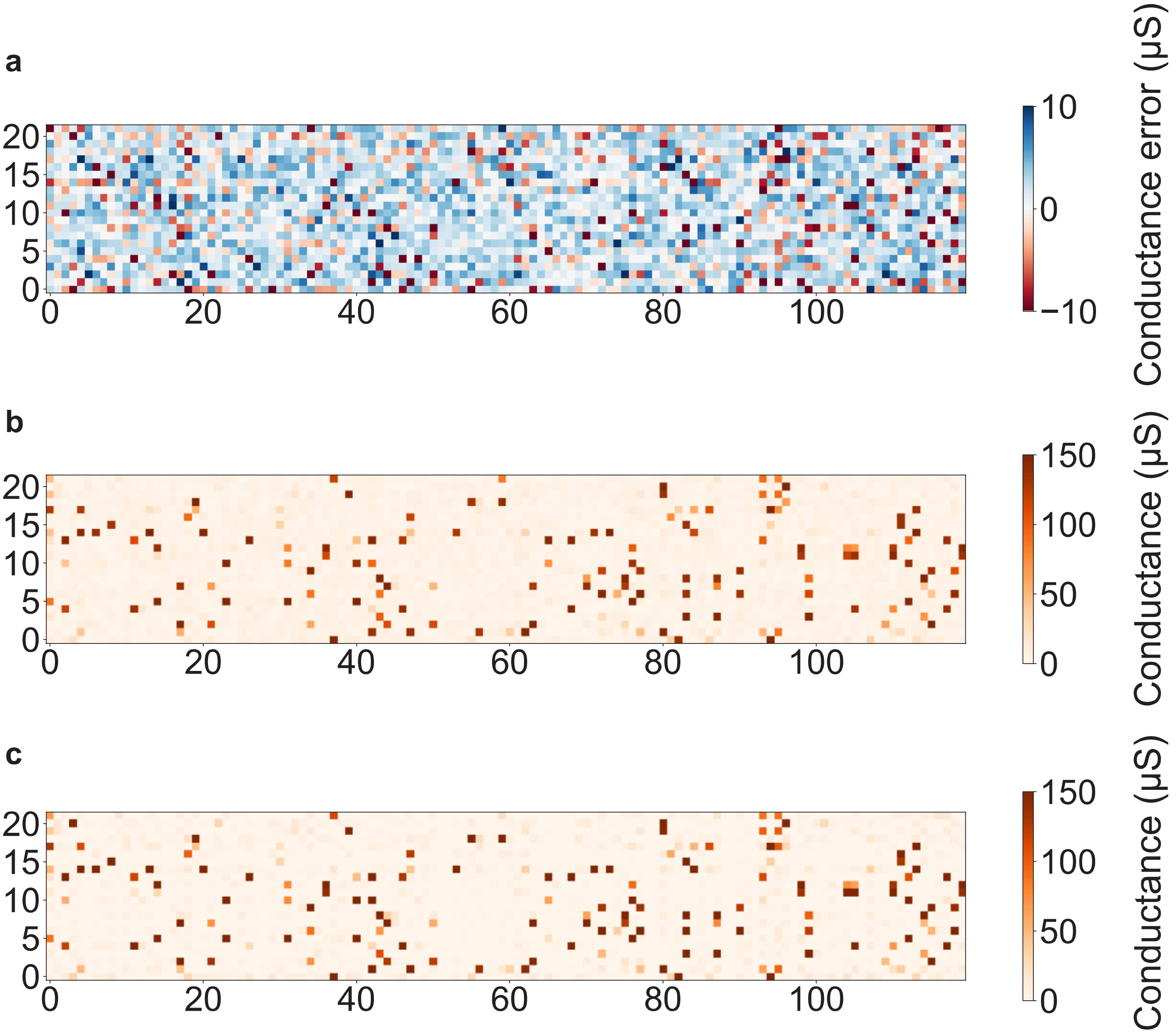

Figure S4. Readout conductance in the final layer for DVS128 Gesture classification. (a) Map of conductance errors across the 22×120 readout matrix. (b) The corresponding readout conductance values. (c) Target conductance values.

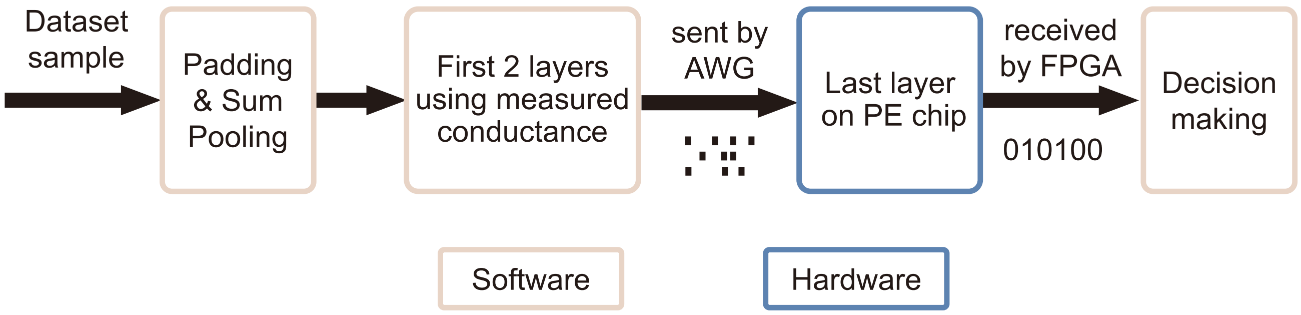

Figure S5. Detailed implementation of the classification experiments on the DVS128 Gesture and NMNIST datasets. The first two layers were performed in software using measured readout conductance, while the final layer was implemented with the PE macro chip.

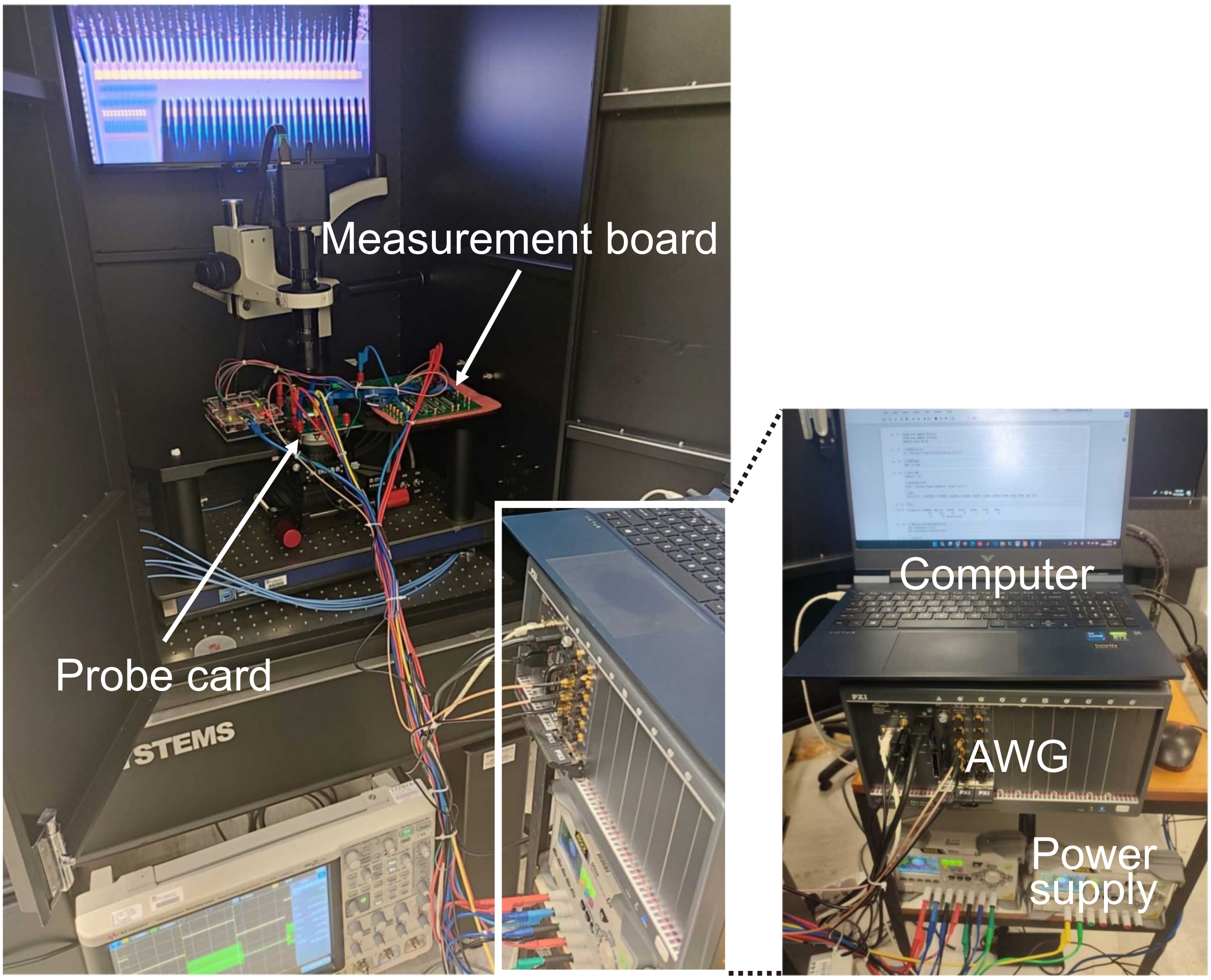

Figure S6. Experimental setup for measuring the system inference accuracy. The left image shows the experimental apparatus, including the probe station and measurement equipment. The inset shows the connected computer, AWG, and power supply.

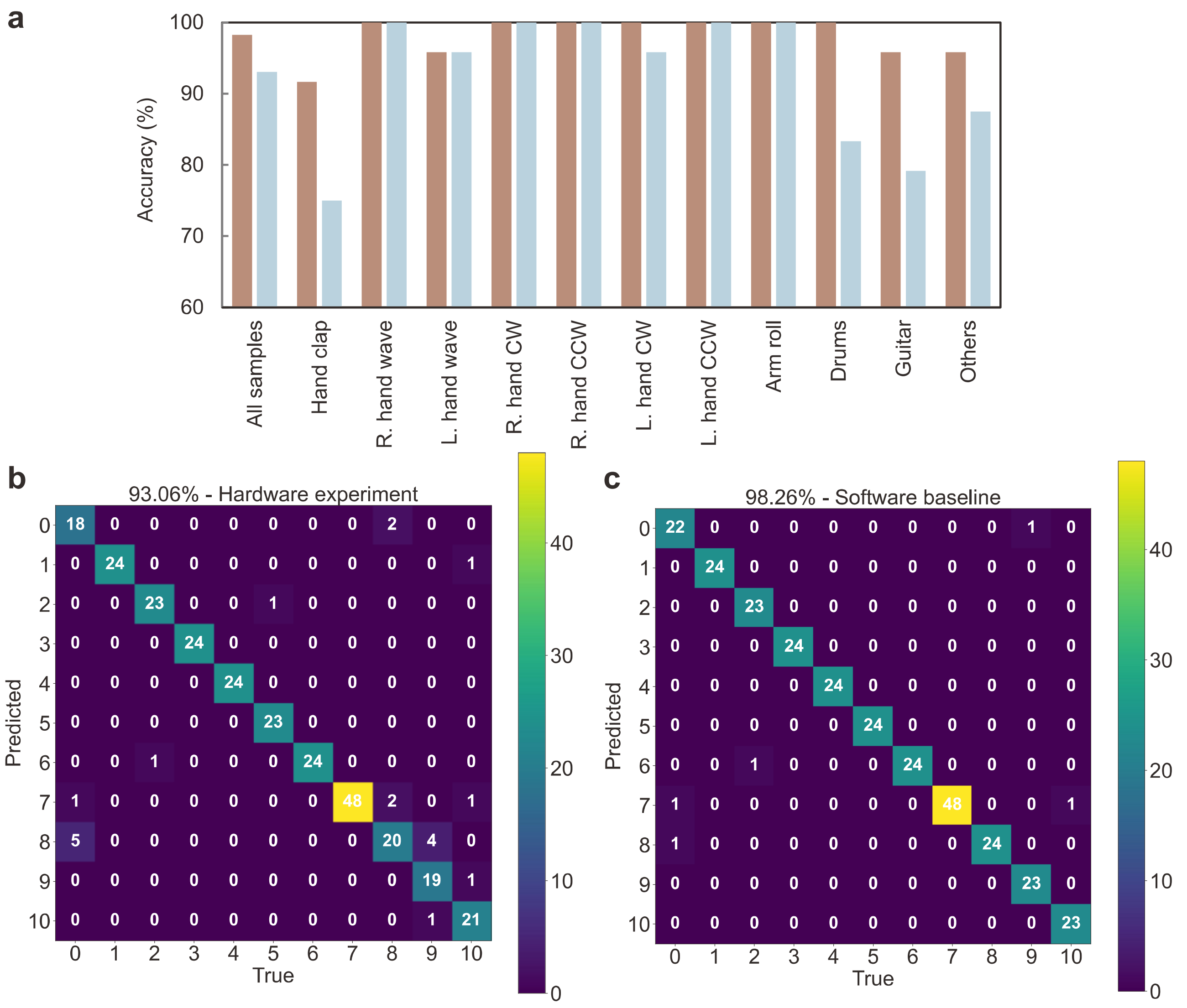

Figure S7. Comparison of experimental classification accuracy to the software baseline on the DVS128 Gesture dataset. (a) Experimental inference accuracy for each class (blue bars) versus the algorithm baseline (brown bars). The experimental accuracy (93.06%) decreases by 5.2% from the baseline (98.26%). (b) Confusion matrix for the experimental result, showing detailed classification, where gestures are labeled 0-10 (Hand clap, R. hand wave, L. hand wave, R. hand CW, R. hand CCW, L. hand CW, L. hand CCW, Arm roll, Drums, Guitar, Others). (c) Confusion matrix for the software baseline.

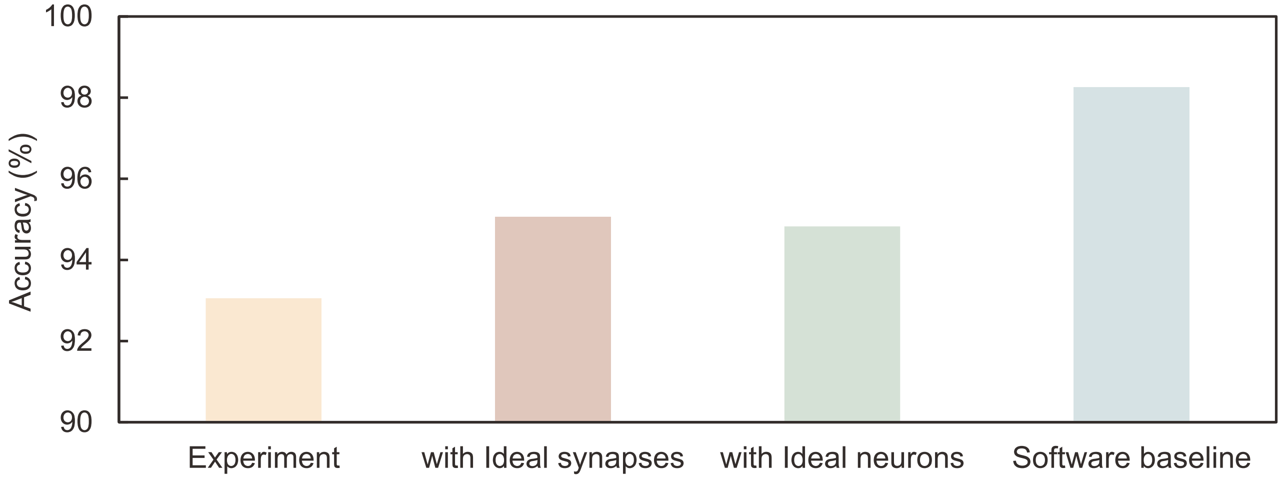

Figure S8. RRAM synapse variation and neuron circuit non-idealities reduce system inference accuracy. Benchmarked against the software baseline on the DVS128 Gesture dataset over ten inferences, a 3.19% average accuracy drop is caused by non-ideal neuron circuits (“with Ideal synapses”), while a 3.43% average accuracy drop is due to RRAM synapse variation (“with Ideal neurons”), compared with the experimental accuracy drop of 5.20% when both RRAM synapse variation and neuron circuit non-idealities are present.

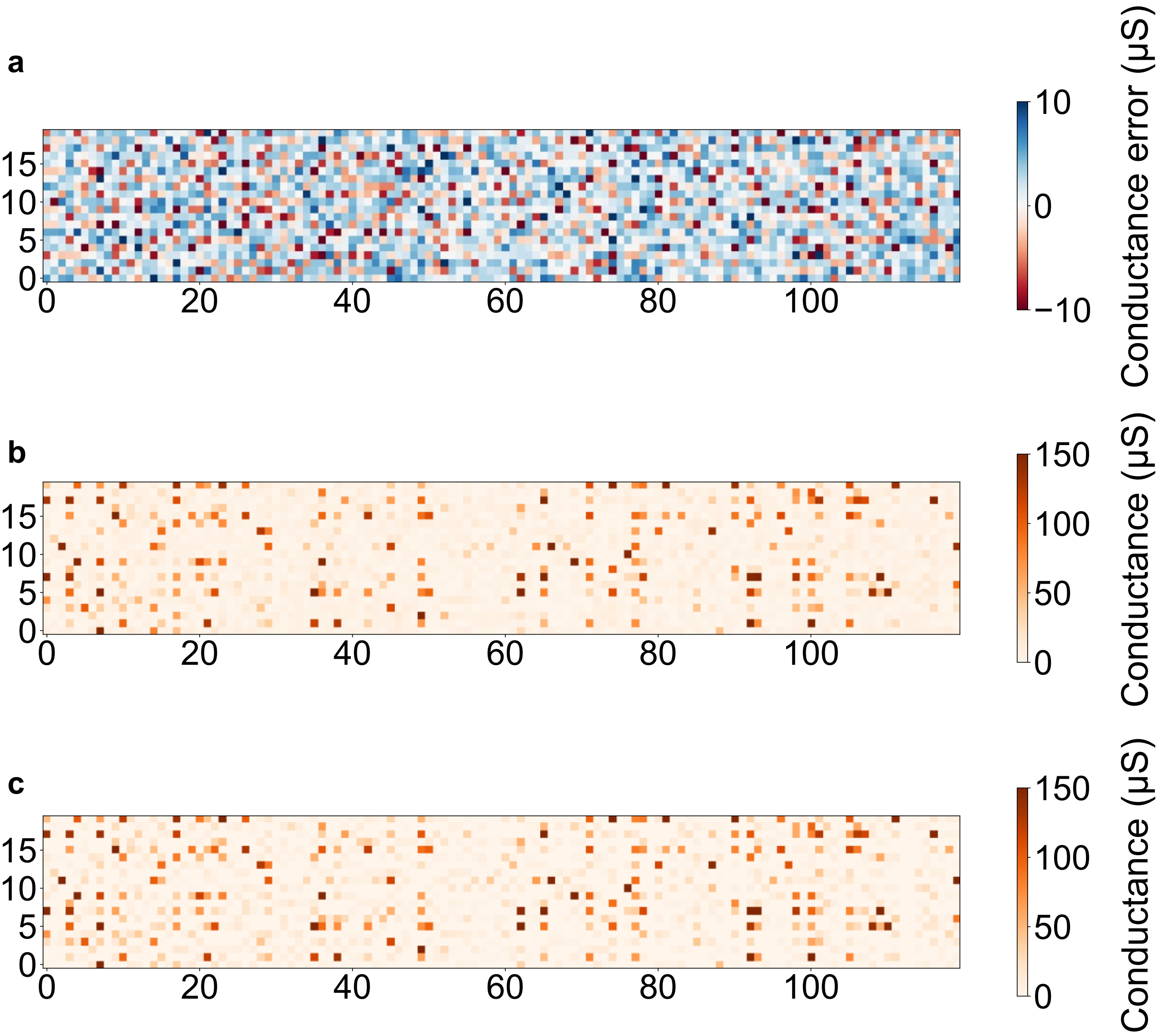

Figure S9. Readout conductance in the final layer for NMNIST classification. (a) Map of conductance errors across the 20×120 readout matrix. (b) The corresponding readout conductance values. (c) Target conductance values.

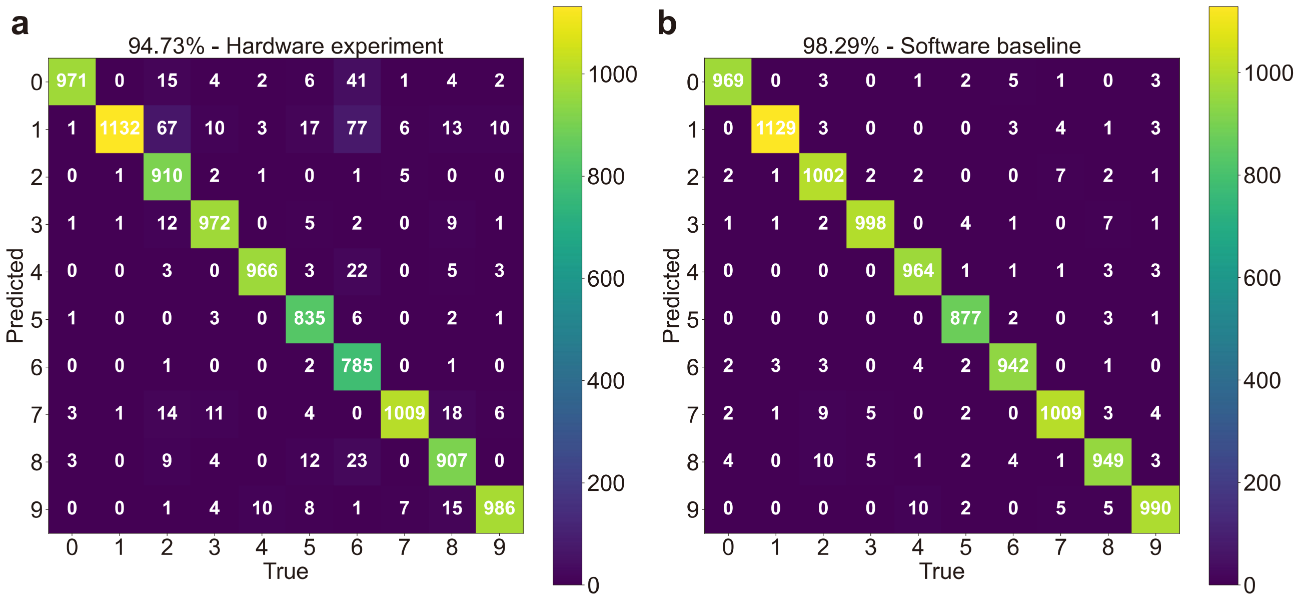

Figure S10. Experimental accuracy compared to the software baseline for classifying digits on the NMNIST dataset. (a) Confusion matrix for the experimental classification result. (b) Confusion matrix of the software baseline.

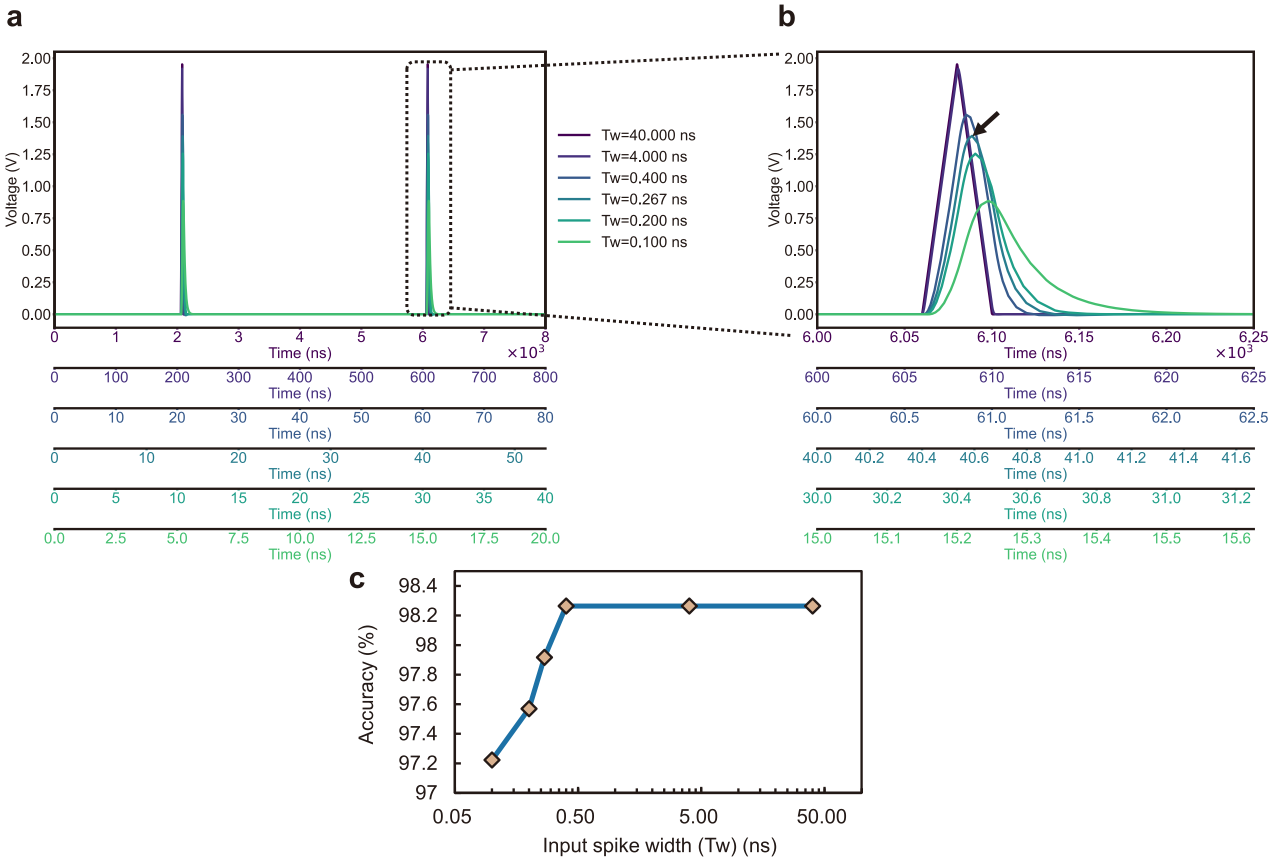

Figure S11. Response of the RRAM crossbar array to input spikes of varying widths. (a) Post-layout simulation results demonstrating the array's capability to process input spikes with widths ("Tw") down to 267 ps (arrow). (b) Zoomed-in view of the resultant output spike (Vin) attenuation with decreasing input spike width (Tw). (c) Impact of reduced input spike width (Tw) on the inference accuracy, decreasing from 98.26% to 97.92% at the narrowest supported spike width of 267 ps.

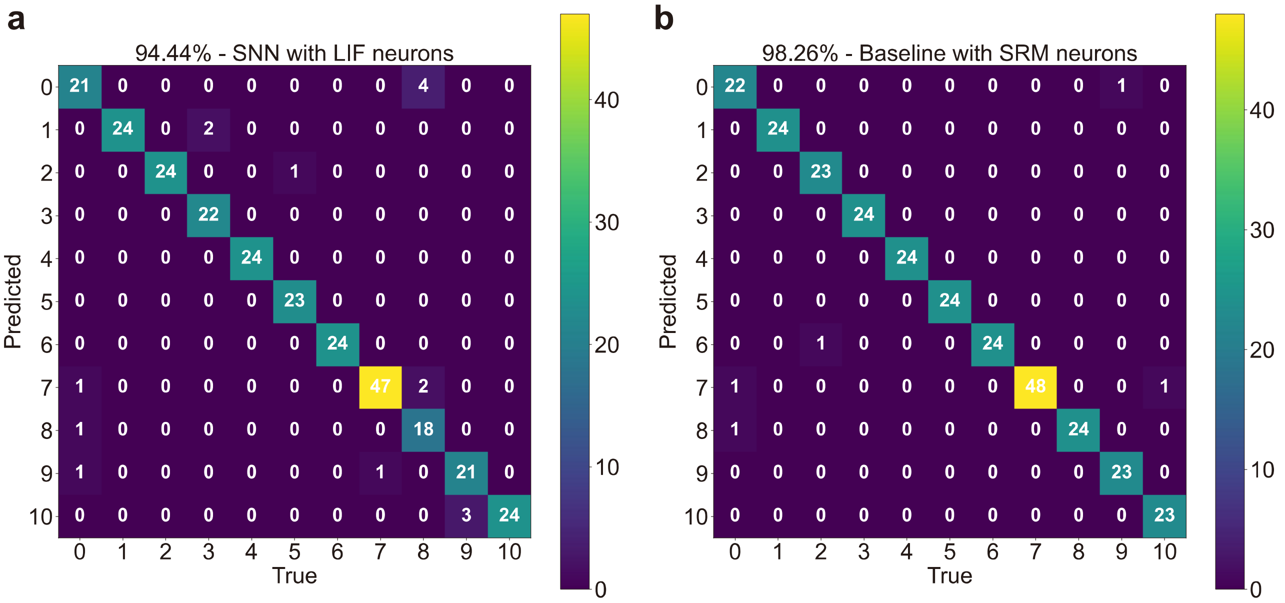

Figure S12. Comparative analysis of algorithmic performance. (a) Replacing SRM neurons with LIF neurons in an otherwise unchanged network structure results in a peak accuracy of 94.44%. (b) The baseline SNN with SRM neurons achieves a higher peak accuracy of 98.26%.

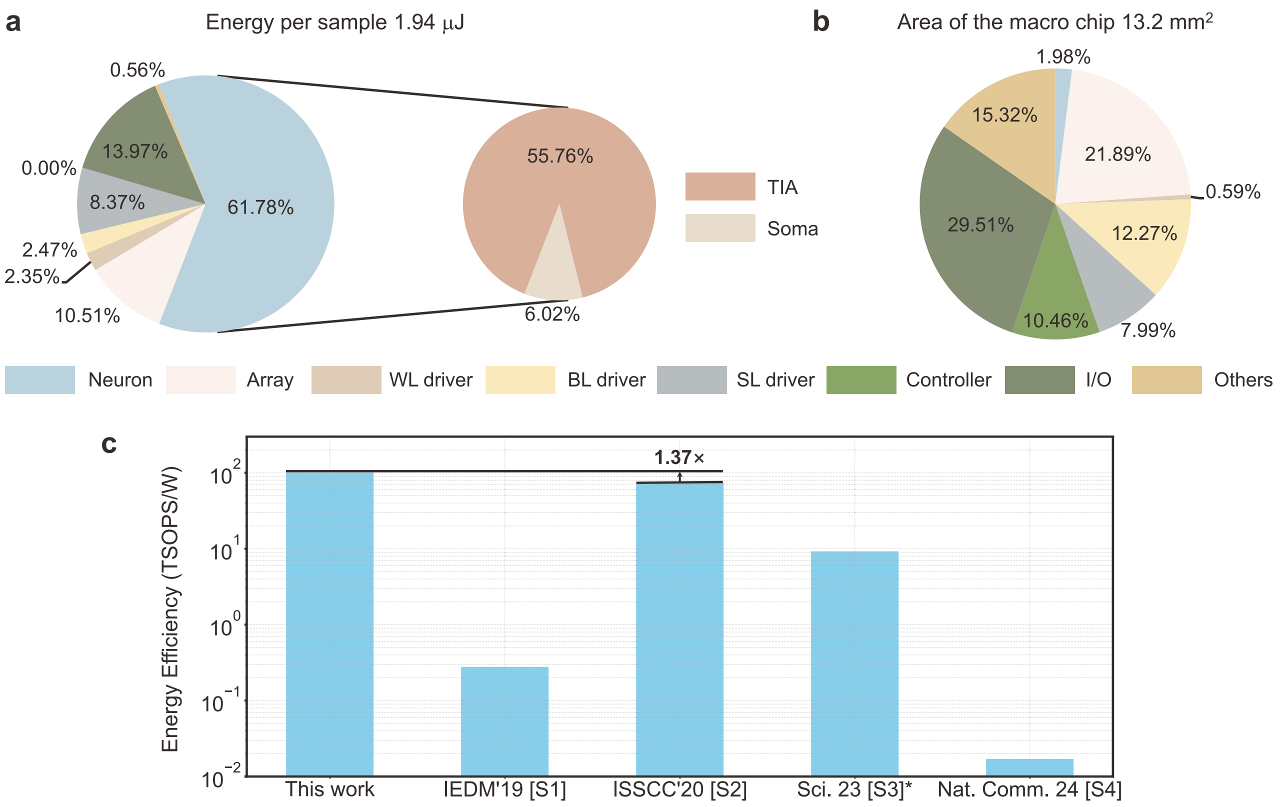

Figure S13. Energy/area breakdown of the PE macro and energy-efficiency benchmarking. (a) Energy breakdown during inference on the DVS128 Gesture dataset: TIAs within the neuron circuits dominate energy consumption (55.76%). (b) Area breakdown of the PE macro: neuron circuits occupy a relatively small footprint (1.98%) of the die, compared to the RRAM array (21.89%). (c) Energy-efficiency comparison with recent neuromorphic systems that integrate RRAM synapses with CMOS neurons [S1-4]. Our design achieves a 1.37× enhancement in energy efficiency compared to the prior SOTA. For consistency, one multiply-accumulate (MAC) counts as one synaptic operation, and each synaptic operation counts as two operations. * Neuron type: analog ReLU; included for the energy-efficiency benchmarking.

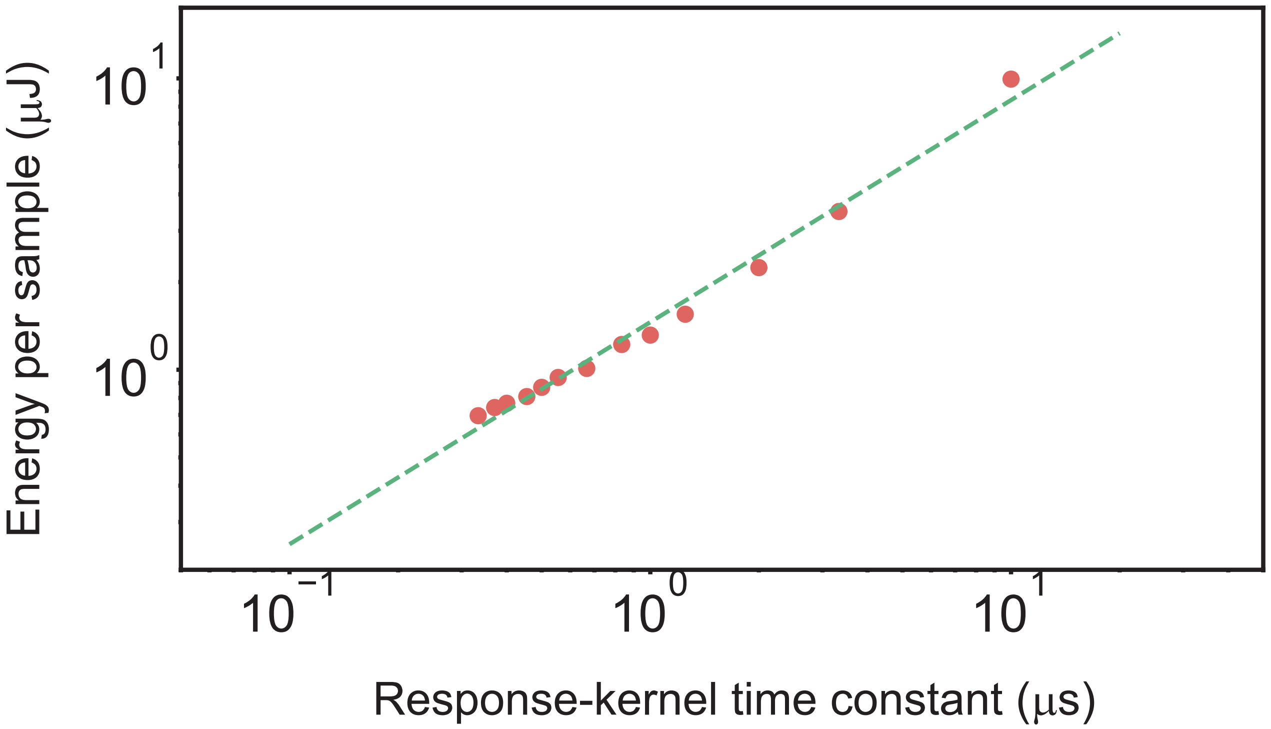

Figure S14. Scalability analysis of the SNN system. Energy consumed per inference sample in the PE macros falls as the response-kernel time constant (τ_s) is further downscaled, while the ratio of response to refractory kernel time constants (τ_s:τ_r) is maintained at 10:1.

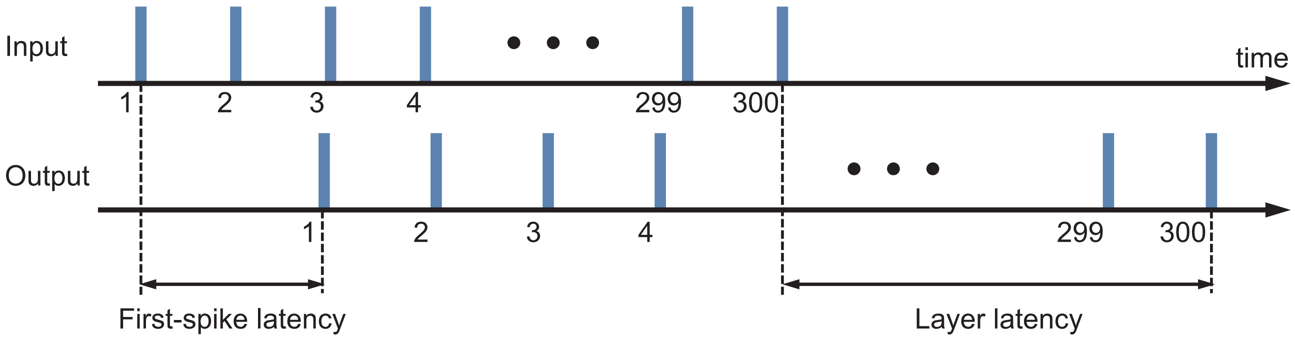

Figure S15. Schematic illustrating the latency incurred by a single layer within the SNN system.

Supplementary Table S1. Benchmarking of algorithm performance on the DVS128 Gesture dataset.

Method	Network	Structure	Neuron	Parameters	Accuracy (%)
mMND (BPTT) [S5]	SNN	CONV	LIF	1.1M	98.0
BPTT+PLIF [S6]	SNN	CONV	Parametric LIF	6.7M	97.57
FPTT+LTC [S7]	SNN	CONV	Liquid Time-Constant Spiking Neuron	6.7M	97.22
OTTT [S8]	SNN	CONV	LIF	9.2M	96.88
mMND (STDP) [S5]	SNN	CONV	LIF	0.81M	96.6
DECOLLE [S9]	SNN	CONV	LIF	1.6M	95.54±0.16
DVSNet [S10]	SNN	CONV	LIF	94K	95.15 [a]
SLAYER [S11]	SNN	CONV	SRM	1.1M	93.64±0.49
EGRU [S12]	RNN	GRU	–	4.8M	97.8
LSTM [S13]	RNN	MLP	–	4.2M	86.81
This work	SNN	MLP	SRM	443K	97.71±0.47

[a] Hardware-aware training

Our SNN model attained a peak accuracy of 98.26%, with an average accuracy of 97.71±0.47% over five trials.

Supplementary Note 1. Rationale for employing backpropagation with surrogate gradients in SNN training.

The training paradigm chosen for a SNN determines the network's achievable accuracy and latency. While several methods exist, our work adopts direct training with backpropagation through surrogate gradients. This choice is predicated on an analysis of the primary alternatives, biologically-inspired local learning and ANN-to-SNN conversion, whose respective limitations make them less suitable for achieving the SOTA performance required by our objectives in accuracy and latency.

Biologically-inspired pure Hebbian rules like spike-timing-dependent plasticity are efficient, highly valuable for on-chip learning. However, being fundamentally local and unsupervised, they lack a mechanism for global error correction. This locality makes them difficult to optimize the entire network for precise, complex tasks, resulting in a performance ceiling where their accuracy still lags behind supervised methods.

ANN-to-SNN conversion, another popular approach, leverages mature ANN training by translating a pre-trained ANN into an SNN [S14, 15]. While this can yield high accuracy, to accurately approximate the continuous activations of an ANN, the converted SNN has traditionally required long inference times (many time steps), which diminishes the latency and energy advantages inherent to spiking computation. Although recent work aims to reduce this latency, the paradigm is fundamentally designed to replicate ANN behavior through rate coding, rather than to natively exploit the rich temporal dynamics SNNs can offer. This makes conversion suboptimal for tasks where information is encoded in the precise timing of individual spikes.

To overcome these limitations, we employ backpropagation with surrogate gradients [S16-18]. This approach adapts the cornerstone of modern deep learning—global optimization via backpropagation—for direct SNN training. By doing so, it directly addresses the shortcomings of the other methods:

(1) Overcomes local learning limits

Unlike Hebbian rules, it provides global credit assignment that optimizes the entire network, closing the accuracy gap to ANN baselines.

(2) Enables native temporal learning

In contrast to ANN-to-SNN conversion, direct training allows the SNNs to learn features directly from temporal spike streams. This makes the SNNs effective to process event-based data and discover novel temporal patterns, fully leveraging the inherent computational strengths of SNNs [S6].

(3) Achieves low-latency performance

By training the SNNs end-to-end in the spiking domain, this method is not constrained by the inference time of conversion. It can achieve high accuracy with short inference time, leading to the low latency and high efficiency that SNNs promise.

While challenges remain in matching the performance of deep ANNs, direct backpropagation-based training represents an effective and direct path to training high-performance SNNs that are optimized for low-latency inference.

Supplementary Note 2. On-chip integration for enhanced energy efficiency of PE macros.

A detailed examination of the energy distribution within the PE macros (Fig. S13a) reveals that neuronal TIAs are the primary energy consumers, followed by the I/O elements. Currently, spike outputs are measured off-chip, and the associated parasitic limits how much neuronal time constants can be scaled down.

However, integrating spike capture on-chip can bypass these external limitations and enable more direct neuronal connections to the measurement. This integration offers the potential for a further reduction in neuronal time constants, thus enabling faster input stream processing and decreasing the per-sample energy consumption of PE macros (Fig. S14), although the overall energy per sample inference would depend on the efficiency of on-chip data processing and power management. Additionally, the area breakdown of our PE chip (Fig. S13b) underscores the same implication: neurons occupy a relatively small portion of the chip (1.98%), whereas I/O elements constitute the majority (29.51%)

Supplementary References

1. Valentian, A., et al. *Fully integrated spiking neural network with analog neurons and RRAM synapses*. in *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019. IEEE.
2. Wan, W., et al. *33.1 A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models*. in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. 2020. IEEE.
3. Zhang, W., et al., *Edge learning using a fully integrated neuro-inspired memristor chip.* Science, 2023. **381**(6663): p. 1205-1211.
4. D'agostino, S., et al., *DenRAM: neuromorphic dendritic architecture with RRAM for efficient temporal processing with delays.* Nature Communications, 2024. **15**(1): p. 3446.
5. She, X., S. Dash, and S. Mukhopadhyay. *Sequence approximation using feedforward spiking neural network for spatiotemporal learning: Theory and optimization methods*. in *International Conference on Learning Representations*. 2021.
6. Fang, W., et al. *Incorporating learnable membrane time constant to enhance learning of spiking neural networks*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
7. Yin, B., F. Corradi, and S.M. Bohté, *Accurate online training of dynamical spiking neural networks through Forward Propagation Through Time.* Nature Machine Intelligence, 2023: p. 1-10.
8. Xiao, M., et al., *Online training through time for spiking neural networks.* Advances in Neural Information Processing Systems, 2022. **35**: p. 20717-20730.
9. Kaiser, J., H. Mostafa, and E. Neftci, *Synaptic plasticity dynamics for deep continuous local learning (DECOLLE).* Frontiers in Neuroscience, 2020. **14**: p. 424.
10. Apolinario, M.P., et al., *Hardware/software co-design with adc-less in-memory computing hardware for spiking neural networks.* IEEE Transactions on Emerging Topics in Computing, 2023.
11. Shrestha, S.B. and G. Orchard. *SLAYER: spike layer error reassignment in time*. in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018.
12. Subramoney, A., et al. *Efficient recurrent architectures through activity sparsity and sparse back-propagation through time*. in *The Eleventh International Conference on Learning Representations*. 2023.
13. He, W., et al., *Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences.* Neural Networks, 2020. **132**: p. 108-120.
14. Rueckauer, B., et al., *Conversion of continuous-valued deep networks to efficient event-driven networks for image classification.* Frontiers in Neuroscience, 2017. **11**: p. 682.
15. Sengupta, A., et al., *Going deeper in spiking neural networks: VGG and residual architectures.* Frontiers in Neuroscience, 2019. **13**: p. 95.
16. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* Nature, 1986. **323**(6088): p. 533-536.
17. Neftci, E.O., H. Mostafa, and F. Zenke, *Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks.* IEEE Signal Processing Magazine, 2019. **36**(6): p. 51-63.

18. Wu, Y., et al., *Spatio-temporal backpropagation for training high-performance spiking neural networks.* Frontiers in Neuroscience, 2018. **12**: p. 331.