S1 Related work

S1.1 Gradient-based training methods for SNNs

The introduction of surrogate gradients has enabled the successful training of deep spiking neural networks (SNNs)[1]. Spatio-temporal backpropagation (STBP) serves as a fundamental framework for addressing the credit assignment problem in SNNs by propagating errors across both spatial and temporal dimensions (Supplementary Fig. S1b)[2]. Conceptually rooted in the backpropagation through time (BPTT) typically used for recurrent neural networks (RNNs), STBP adapts this mechanism to the SNN framework.

S1.1.1 Three types of gradient on SNNs

In this process, the gradient for weight updates is decomposed into three distinct backward paths to precisely assign credit to each synaptic contribution, as shown in Supplementary Fig. S1b. The spatial gradient represents inter-layer information through synaptic integration, while the temporal gradient arises from the internal membrane potential dynamics, such as leakage over time. The spatial-temporal gradient stems from the reset operations following a spike, which couples temporal and spatial dependencies. While STBP achieves high training precision by utilizing all three paths, the requirement to maintain and propagate these concurrent gradient components across every time step leads to significant computational complexity and memory overhead.

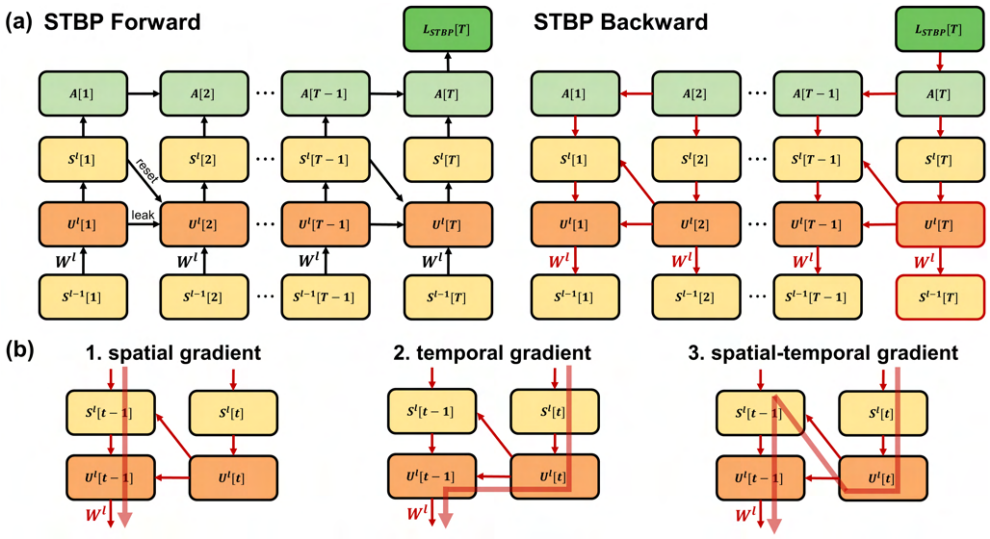


Fig. S1 (a) Roll-up diagram of the forward and backward passes in STBP. (b) Illustrations of the three distinct backward paths: spatial, temporal, and spatio-temporal gradients.

S1.1.2 Efficient Training Methods for SNNs

Recent efficient training methods for SNNs aim to reduce the computational and memory burden associated with complex temporal dependencies.

E-prop uses presynaptic activity to capture recurrent temporal dependencies, thereby approximating the temporal gradient chain and enabling biologically plausible weight updates[3].

Online training through time (OTTT) similarly extends this concept by incorporating presynaptic activity at each time step, enabling its application to deep layered architectures[4]. Eligibility traces transform temporal gradient information into spatial representations, providing an effective approximation that reduces both computational and memory requirements for credit assignment.

Spatial learning through time (SLTT) performs backpropagation independently at each time step while ignoring temporal gradients, based on the observation that temporal gradients are often negligible[5]. However, SLTT still processes all time steps, limiting its computational efficiency. SLTT-K further reduces the training cost by randomly sampling K time steps and computing spatial gradients only at the selected time steps.

Spatial online learning at once (SOLO) further streamlines training by using only the spatial gradient from the final time step, eliminating the need to maintain intermediate states throughout the sequence[6]. When combined with data-feeding strategies such as random sequential subset sampling (RSSS), SOLO achieves low computational overhead and high memory efficiency, making it a strong candidate for on-chip implementation and hardware-friendly neuromorphic systems.

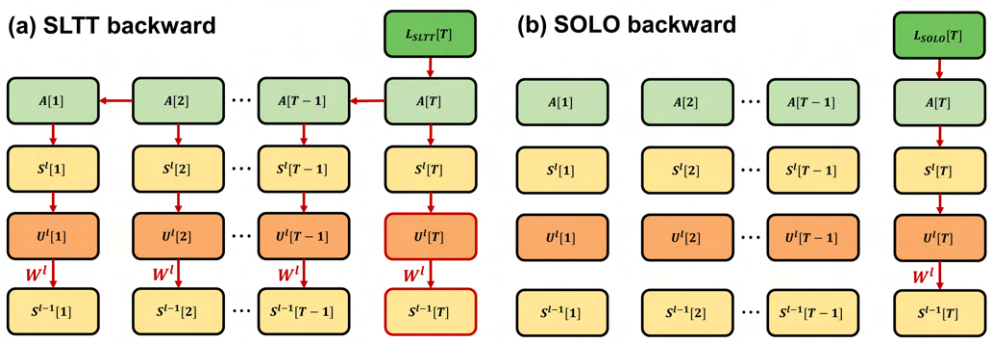


Fig. S2 (a) Roll-up diagram of the backward passes in SLTT. (b) Roll-up diagram of the backward passes in SOLO.

S1.2 Comparison of neuromorphic computing chips design

Table S1 compares representative neuromorphic computing (NC) chips in terms of neuron implementation, synapse memory, and membrane-potential (V_{mem}) readout capability. Most digital neuromorphic processors, such as TrueNorth, Tianjic, Loihi-1, and SpiNNaker-1, communicate neuronal activity mainly through 1-bit spike events and do not provide direct access to the membrane potential of individual neurons. Although Loihi-2, SpiNNaker-2, and BrainScaleS systems offer some form of V_{mem} readout, such functionality is either limited to internal registers, debug modes, or requires dedicated analog readout circuitry.

For example, BrainScaleS-2 must sustain up to 1.2 Gbit/s to stream membrane-potential readouts, incurring substantial area and power overhead for accurate membrane potential measurement[7].

Table S1 Comparison of Neuromorphic Computing Chips Design

NC Chip	Neuron (V_{mem} Resolution)	Synapse (States Resolution)	V_{mem} Readout?
TrueNorth[8]	Digital ASIC LIF (5 bit)	On-chip SRAM (4 bit)	**No**
Tianjic[9]	Digital ASIC LIF (8 bit)	On-chip SRAM (8 bit)	**No**
Loihi-1[10]	Digital ASIC LIF (8 bit)	On-chip SRAM (1–9 bit)	**No**
Loihi-2[11]	Digital ASIC LIF (8 bit)	On-chip SRAM (1–9 bit)	**Yes**[1]
SpiNNaker-1[12]	ARM + software LIF (FP32)	External SDRAM (8, 16 bit)	**No**
SpiNNaker-2[13]	ARM + software LIF (FP32)	External SDRAM (8, 16 bit)	**Partially**[2]
BrainScaleS-1[14]	Analog ASIC AdEx 180nm	On-chip SRAM (4 bit)	**Yes**
BrainScaleS-2[7]	Analog ASIC AdEx 65nm	On-chip SRAM (4 bit)	**Yes**

[1] Internal register readout via Loihi API.

[2] Approximate readout under debug mode.

S1.3 Eligibility representations and hardware implementation

S1.3.1 Hardware implementation of eligibility traces

Implementing leaky integration for eligibility traces on digital neuromorphic chips often necessitates additional hardware resources, such as dedicated neural cores. Moreover, while current-based LIF neuron models can encode these traces in analog NC chips, realizing their continuous dynamics on physical hardware remains a challenge. Recently, diffusive memristor devices have been proposed to directly emulate exponential decay, offering a promising analog solution for eligibility dynamics in neuromorphic circuits. However, integrating novel training operations like eligibility traces into neuromorphic architectures remains challenging. The requirement for direct accessibility to internal state variables, particularly membrane potentials, presents a primary hardware-level constraint.

Table S2 Comparison of Implementation of eligibility trace technique.

Method	Target module	Algorithm	Output	Hardware Implementation
SG [1]	Presynaptic potential	SS [15], DECOLLE [16]	Analog	DPI [17–19], memristor [20]
	Presynaptic trace	e-prop [3], OTTT [4]	Analog	Chip [21,22], PCM [23,24]
	eP2S	SOLO [6]	Binary	LI operation
	eLIFE (this work)	SLTT [5], SOLO [6]	Binary	LIF operation
EP [25]	IF Kernel	STD-ED [26]	Analog	IF operation

S1.3.2 Eligible potential-to-spike encoding (eP2S)

In previous research, the eligible potential to spike (eP2S) is proposed to implement eligibility traces in a hardware-friendly manner[6]. eP2S extends the spatial gradients of presynaptic spike occurrences by representing temporal gradients as an eligibility trace, which are typically computed through the leaky integration (LI) of presynaptic spike inputs over time. By applying a thresholding operation to the eligible potential, binary 'eligible spikes' are generated, thereby streamlining the gradient computation process. While beneficial for gradient computation, eP2S introduces a state-access problem on NC chips. Since the eligible potential requires retrieving a specific accumulated state prior to thresholding, its native implementation on the conventional NC architecture is challenging.

$$\tilde{U}_i^l[T] = \alpha \tilde{U}_i^l[T-1] + S_j^{l-1}[T] \tag{1}$$

$$\tilde{S}_i^l[T] = \Theta(\tilde{U}_i^l[T] - \tilde{\vartheta}_{\text{th}}) \tag{2}$$

where $\tilde{U}_i^l$ is the eligible potential of neuron i at layer l, $\alpha = 1/(1+\exp(-\tau_{ep}))$ is the eligible potential decaying constant, S_j^{l-1} represents the occurrence of the output spike of neuron j at layer $l-1$, $\tilde{S}_i^l[T]$ is the binary eligible spike at layer l, and $\tilde{\vartheta}_{\text{th}}$ is the eligible threshold.

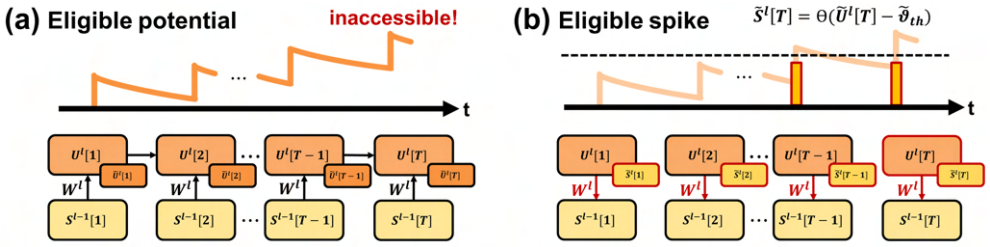


Fig. S3 Concept of the eligible potential to spike (eP2S). (a) During the forward path, the eligible potential traces the pre-synaptic spikes. (b) During the backward path, thresholding binarization converts eligible potentials into eligible spikes, which then replace the pre-spike computational components in the gradient chain.

S1.3.3 Network structure of the eLIFE module

Conceptually, the proposed eligible leaky integrate-and-fire (eLIFE) module can be modeled as a recurrent spiking neural network (RSNN) architecture composed of multiple cloned LIF neurons. Increasing the number of these neurons enhances the capability to approximate specific target functions, a principle consistent with the universal function approximation theory for spiking neurons [27]. However, to mitigate the potential for significant computational overhead, our eLIFE module utilizes only two cloned neurons to maintain a minimal hardware footprint.

To illustrate the structural representation of the eLIFE module, Supplementary Fig. S4 demonstrates how it expresses the eligibility trace utilizing this cluster of cloned LIF neurons. Specifically, the spatial diagram (Supplementary Fig. S4a) highlights the recurrent connections established between two cloned neurons. The temporal diagram (Supplementary Fig. S4b) depicts how a spike generated in the previous time step affects the membrane potential at the current time step.

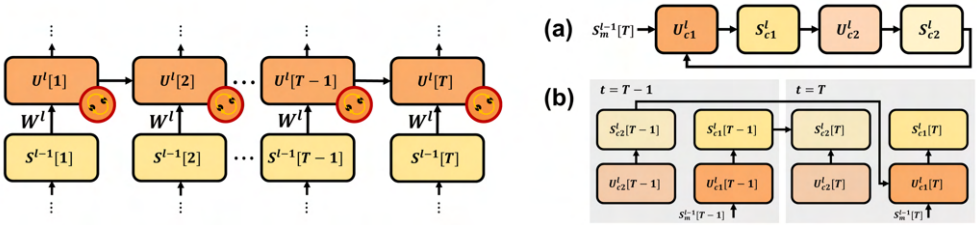


Fig. S4 Structural representation of the eLIFE module. The eLIFE module expresses the eligibility trace utilizing a cluster of clone LIF neurons. (a) Spatial diagram of eLIFE, demonstrating recurrent connections between two clone neurons. (b) Temporal diagram of eLIFE, depicting how temporal information is preserved as the spike from the previous time step affects the membrane potential at the current time step.

S2 Additional explanation of 1T-devices

S2.1 Device fabrication

The fabrication process flow, the resulting 8-inch wafer, and the scanning electron microscope (SEM) and transmission electron microscope (TEM) images are shown below.

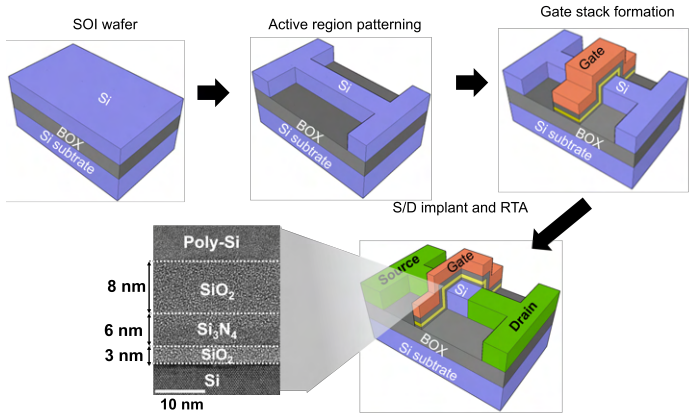


Fig. S5 Fabrication process flow of the homogeneous 1T-devices, together with a TEM image of the gate stack (see Methods, "Device fabrication").

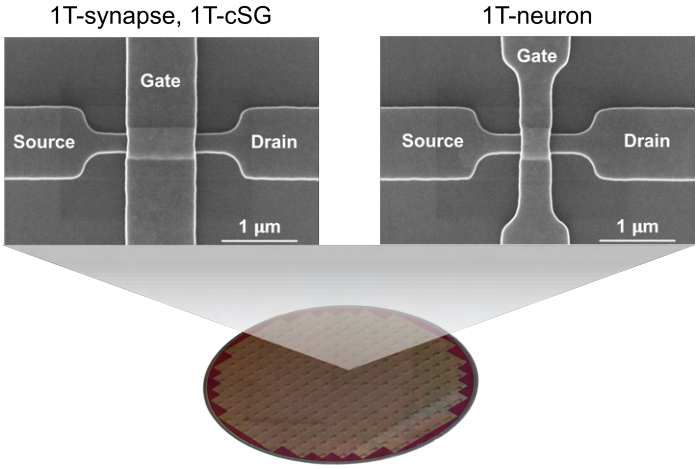


Fig. S6 The SEM images of the cointegrated 1T- synapse, 1T-cSG and 1T-neuron fabricated on a 8-inch silicon-on-insulator (SOI) wafer, along with a wafer photograph. Both devices have a channel width (W_{ch}) of 200 nm, with gate lengths (L_{G}) of 400 and 900 nm for the 1T-neuron and 1T-synapse, respectively. The 1T-cSG has the same dimensions as the 1T-synapse.

S2.2 Synaptic characteristics of 1T-synapse

The 1T-synapse stores synaptic weights through charge-trap-induced modulation of the channel conductance. Electron trapping shifts the threshold voltage positively and reduces the drain current, enabling nonvolatile conductance storage (Supplementary Fig. S7). Long-term potentiation (LTP) and depression (LTD) were induced using 10 V, 10 μs and +9.5 V, 100 ns gate pulses, respectively, while the conductance was read at $V_{\mathrm{G}} = 1$ V and $V_{\mathrm{D}} = 1$ V. Successive pulses gradually increased or decreased the conductance, and stable modulation was maintained over three cycles (Supplementary Fig. S7). The gradual LTP/LTD behavior was reproduced across eight devices, with comparable conductance ranges despite moderate device-to-device variation.

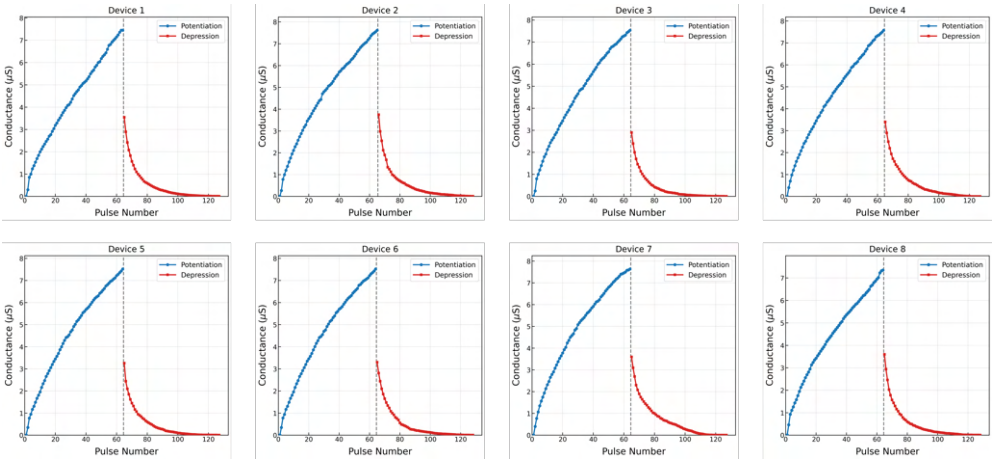


Fig. S7 LTP/D characteristics of 1T-synpase measured across eight devices.

S2.3 Leaky characteristics of 1T-neuron

The leaky characteristics of the 1T-neuron, where current pulses with an amplitude of 30 pA and a width of 350 ms are repeatedly applied and terminated once the membrane potential reaches the target range.

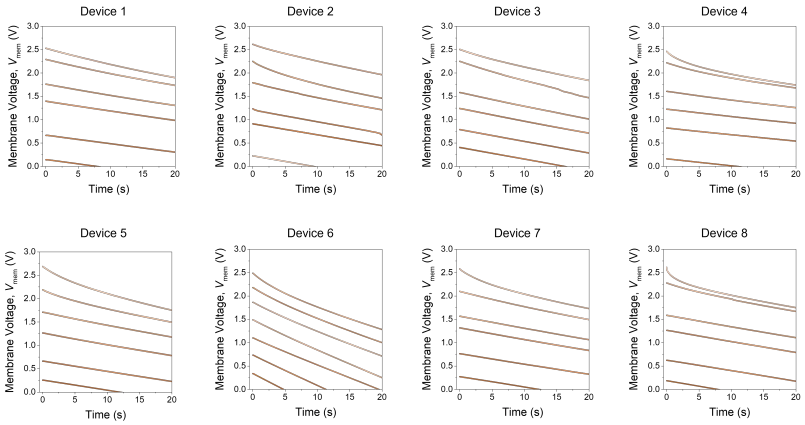


Fig. S8 Leaky characteristics of the 1T-neuron measured across eight devices.

S2.4 Operating principle of STL

Under a negative gate bias, the transistor channel remains effectively turned off, and the drain current stays low during the initial stage of the drain voltage (V_{D}) sweep. As V_{D} increases, the reverse bias at the drain-to-body junction becomes stronger, leading to a highly intensified electric field near the drain. Although the channel is not fully formed, a limited number of electrons injected from the source can still be accelerated into this high-field region. Once V_{D} reaches its critical threshold, referred as latch-up voltage (V_{latch}), source-injected electrons initiate the impact ionization (I.I) at the drain-to-body junction, generating electron–hole pairs. The electrons are swept toward the drain electrode while the generated holes accumulate in the FB. As the hole concentration in the FB increases, the body potential rises, which lowers the energy barrier between the source and the body. The reduced barrier promotes further electron injection from the source, enhancing I.I and generating even more holes in the FB. As a result, a regenerative positive-feedback loop is established, ultimately causing an abrupt transition of the 1T neuron from the high-resistance state (HRS) to the low-resistance state (LRS)[28].

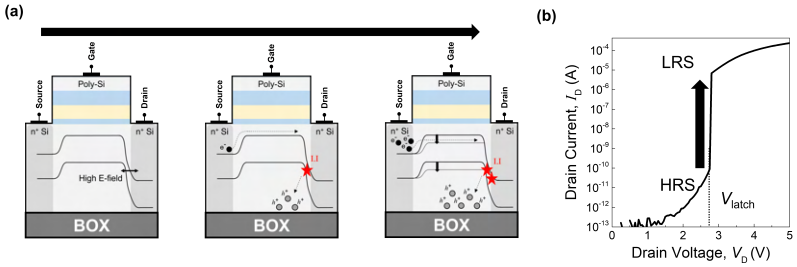


Fig. S9 (a) The procedure of single-transistor latch (STL) observed in floating body (FB) MOSFETs. (b) Output characteristics (I_{D}-V_{D}) of 1T-neuron under V_{G} = -3 V.

S2.5 Modulating V_{latch} of 1T-neuron

Trapping electrons in the CTL increases the barrier height between the source and the body, thereby suppressing electron injection from the source. Consequently, a stronger electric field is required to induce STL, leading to an increase in V_{latch}. Conversely, hole trapping lowers the source-to-body barrier and thus decreases V_{latch} [28,29].

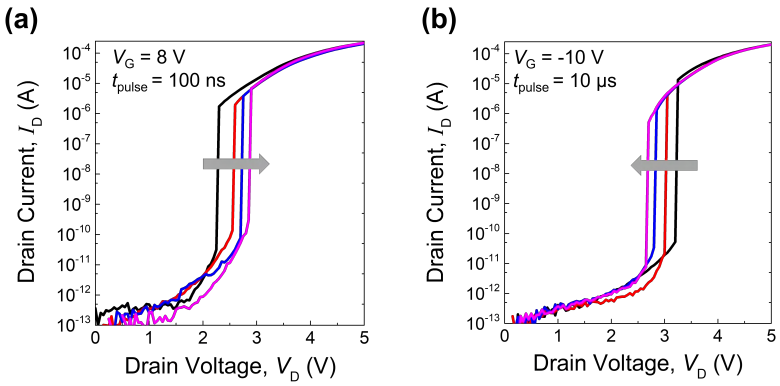


Fig. S10 (a) Stepwise increase in the latch-up voltage (V_{latch}) of the 1T-neuron through electron trapping using 8 V gate pulses with a pulse width of 100 ns. (b) Stepwise decrease in V_{latch} through hole trapping using -10 V gate pulses with a pulse width of 10 ms. Output characteristics of 1T-neuron was measured under V_G = -3 V

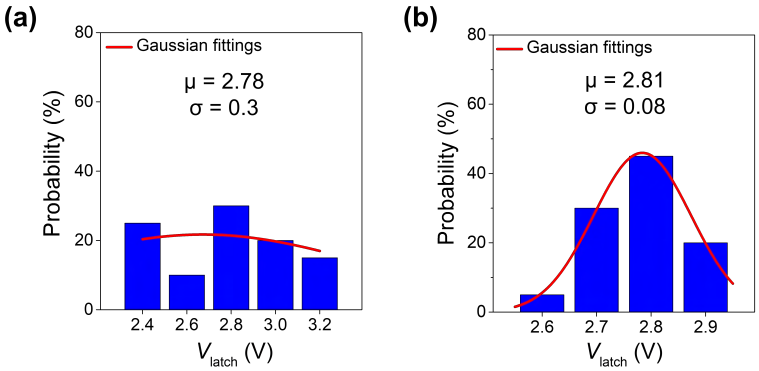


Fig. S11 (a) Distribution of V_{latch} before modulation, with a mean of 2.8 V and a standard deviation of 0.3 V measured from 20 different devices. (b) Distribution of V_{latch} after modulation, with the mean and standard deviation adjusted to 2.81 V and 0.08 V, respectively.

S2.6 Additional investigations of 1T-cSG

The 1T-cSG exhibited stable operation over 100 quasi-static I–V measurements and 10^5 AC cycles under the same gate pulse scheme employed for bidirectional latch switching. Specifically, forward and backward latch events were selectively induced by pulsing V_{G} from the standby bias of -2.5 V to -1.5 V and -3.5 V, respectively (main Fig. 2i). Device-to-device variation was also evaluated across 20 devices. The mean and standard deviation of $V_{\mathrm{th,\ backward}}$ were -2.97 V and 0.155 V, respectively, while those of $V_{\mathrm{th,\ forward}}$ were -1.49 V and 0.08 V, indicating reliable device operation.

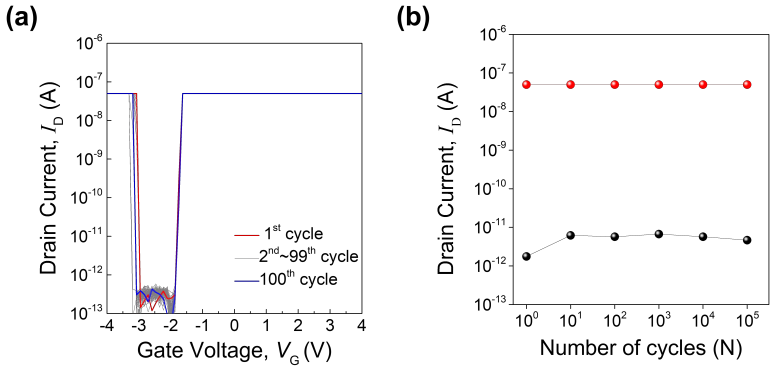


Fig. S12 (a) I_{D}-V_{G} characteristics of the 1T-cSG under repetitive 100 DC cycling. (b) Endurance characteristics of the 1T-cSG device under repetitive 10^5 cycling.

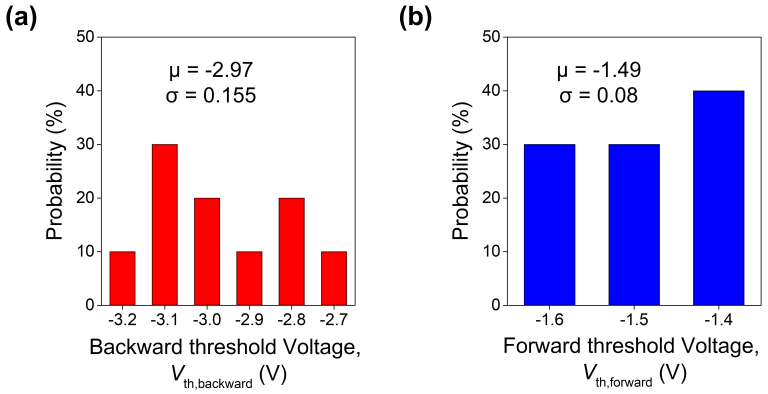


Fig. S13 Distribution of (a) Backward threshold voltage ($V_{\mathrm{th,\ backward}}$) and (b) forward threshold voltage ($V_{\mathrm{th,\ forward}}$) of the 1T-cSG measured across 20 devices.

S2.7 Detection of surrogate gradient signal

The bidirectional abrupt switching of the 1T-cSG can also be detected in the voltage domain, rather than in the current domain, thereby reducing the need for bulky amplifiers for signal conversion.

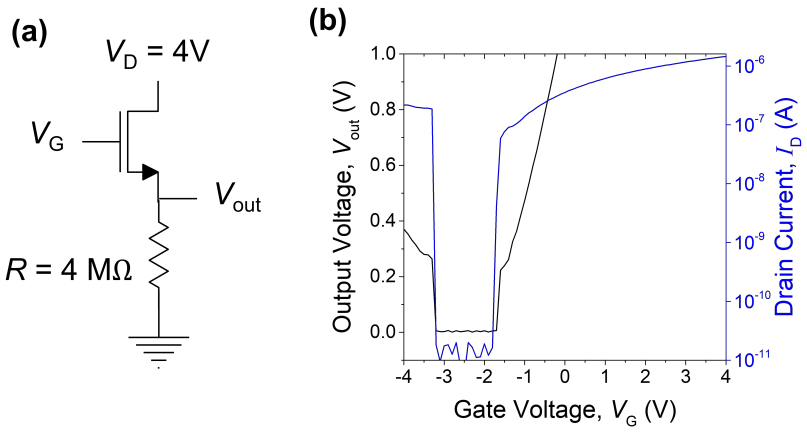


Fig. S14 (a) Schematic of the circuit used to sense the cSG signal in the voltage domain. (b) Measured output voltage (V_{out}) and drain current (I_{D}), showing that the cSG signal can be read out as a voltage signal.

S3 Circuit Implementation of Unit Architecture

S3.1 Current mirror interface

Interfacing between the 1T-based neurons and synapses and the current surrogate gradient (cSG) devices is achieved using external commercial PMOS transistors mounted on the breadboard.

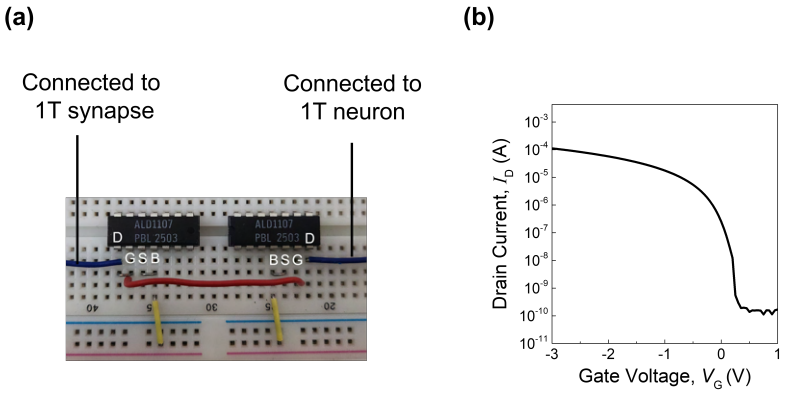


Fig. S15 (a) Photograph of the current-mirror-based interface circuit implemented on a breadboard. (b) I_{D}–V_{G} characteristics of the external commercial PMOS transistors measured under V_{D} = -3.3 V.

S3.2 LIF characteristics of 1T-neuron with reduced $C_{\mathbf{par}}$

In this configuration, C_{par} is reduced to under 10 pF by eliminating the large parasitic contribution introduced by the breadboard. Because the voltage drop is governed by discharge of the drain electrode, the reduced capacitance results in a larger voltage drop during the firing process. As a consequence, the reset voltage V_{reset} was pulled down to ground.

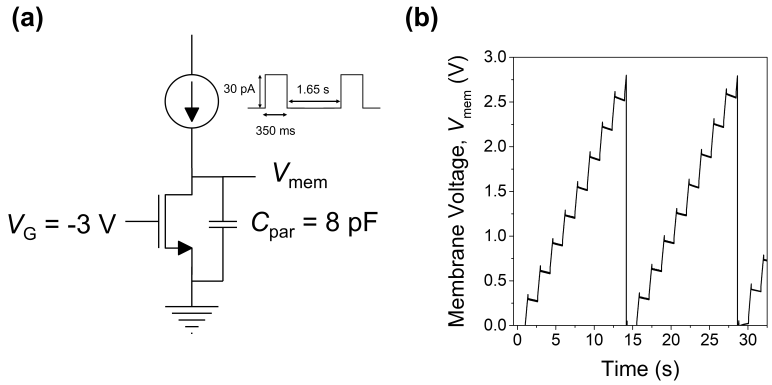


Fig. S16 (a) Schematic of the measurement setup for the 1T-neuron using 30 pA current pulses with an on-time of 350 ms and an off-time of 1.65 s. (b) Measured membrane potential (V_{mem}) as a function of time.

S3.3 Tile-scale SPICE simulation

To further evaluate scalability, the proposed architecture was simulated in Cadence Virtuoso based on 180 nm technology. The LIF behavior of the 1T-neuron was modeled using multiple switches and a capacitor. A large-scale system was constructed by connecting a 32 × 32 AND-type differential 2T-synapse array to 32 1T-neurons and 32 1T-cSGs following the same architecture. In addition, the negative-weight path was implemented using another set of 32 x 32 AND-type 1T-synapse array according to the conventional differential synaptic pair scheme, which is essential for representing signed synaptic weight updates during learning.

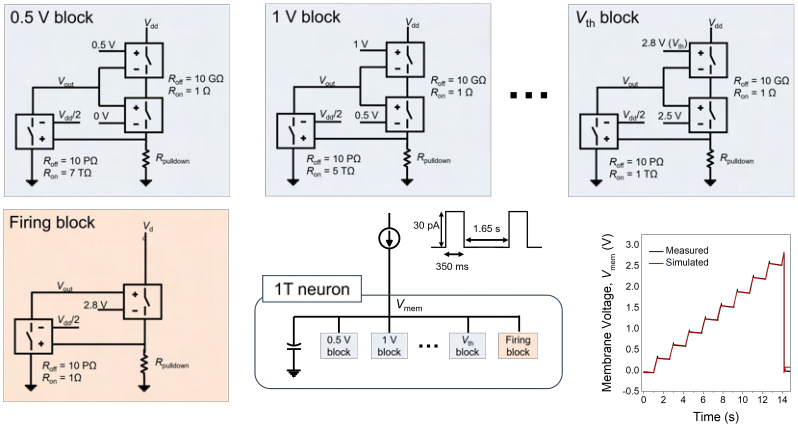


Fig. S17 Schematic illustration of the functional blocks used to model the 1T-neuron using switch-based elements and a capacitor. The overall model is divided into several sub-blocks, each representing a distinct operating regime of the device, in order to reproduce the LIF dynamics of the 1T-neuron.

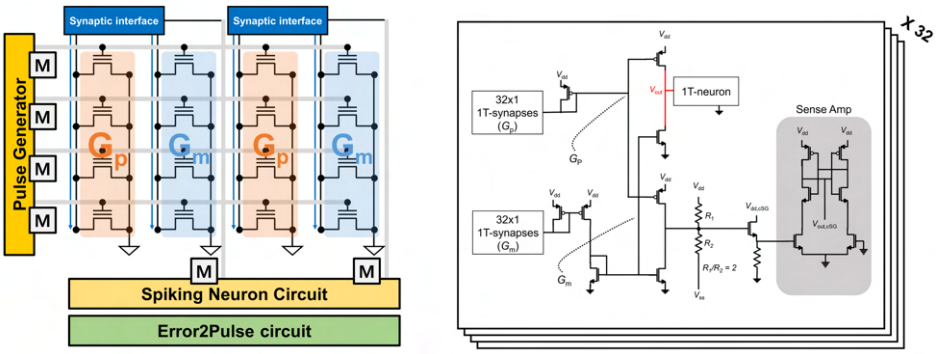


Fig. S18 Schematic of the complete tile architecture of the proposed system, comprising the 32x1 differential 1T-synapse arrays, 1T-neuron, cSG device, and the associated cSG sensing circuit.

S3.4 Results of tile-scale 32 neurons

To visually examine the neural activity across a layer, Supplementary Fig. S19 presents the membrane potential dynamics of 32 individual neurons over time, alongside their corresponding cSG signals.

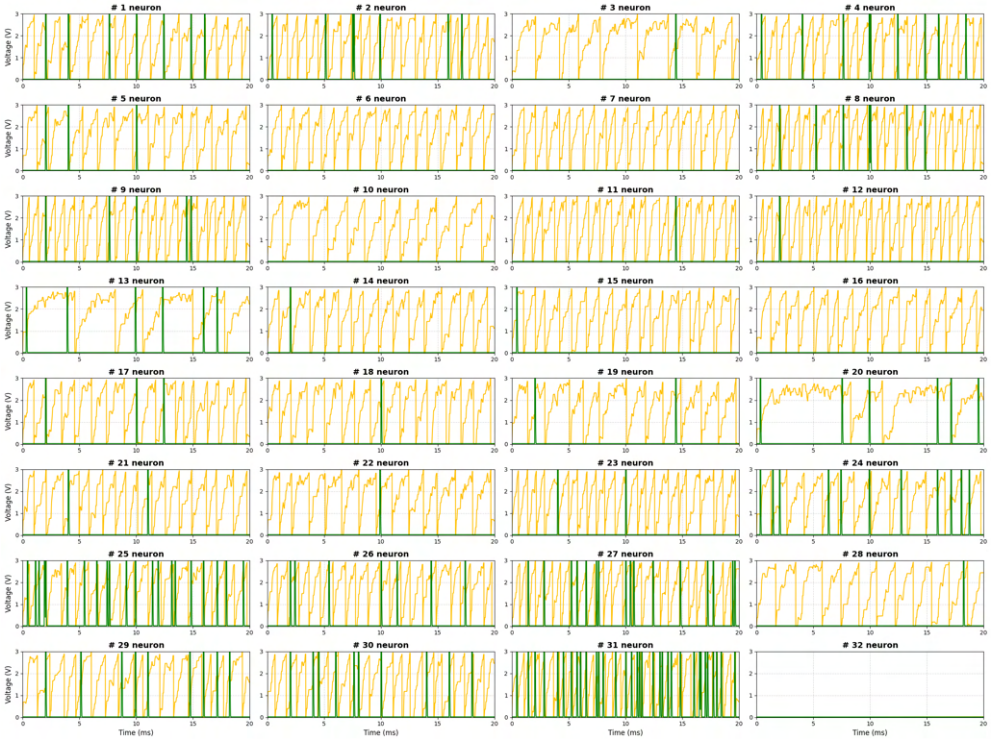


Fig. S19 Temporal dynamics of V_{mem} and cSG-device responses across 32 neurons. The waveforms show V_{mem} (orange) and the corresponding cSG responses (green) for each of the 32 neurons over a 20 ms time interval. The cSG device responds to the synaptic-current operating window rather than directly thresholding V_{mem}; the binary gradient follows the readout convention defined in the main-text Methods.

S3.5 1T-based implementation of the eLIFE module

An example implementation of the eLIFE module is illustrated using 1T-based spiking neurons integrated with spike-detector circuits. The detector outputs are active-low voltages, remaining high between spikes and falling low when a spike occurs. Let S_k denote a binary spike event and $D_k = 1 - S_k$ the corresponding normalized detector voltage for the main neuron or either clone neuron. The voltage-domain combination satisfies

$$D_{\text{elig}} = D_m \wedge D_{c1} \wedge D_{c2} = \neg\,(S_m \vee S_{c1} \vee S_{c2})\,. \tag{3}$$

Thus, AND combination of the active-low detector voltages produces an active-low representation of the algorithmic OR eligibility signal. An additional output inversion would produce a NAND function of the detector voltages and an active-high eligible signal, but is not required for the active-low output plotted here.

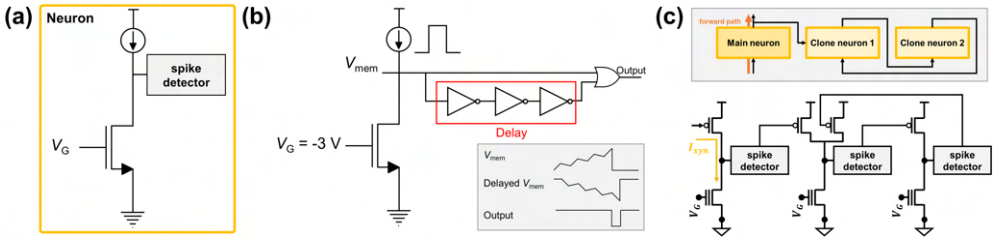


Fig. S20 (a) 1T-based spiking neuron with a spike detector. (b) Delay-line-based spike-detector circuit for generating spike pulses from membrane-potential dynamics. (c) eLIFE neural cluster composed of a main neuron and two recurrently connected clone neurons implemented using the 1T-based spiking-neuron structure.

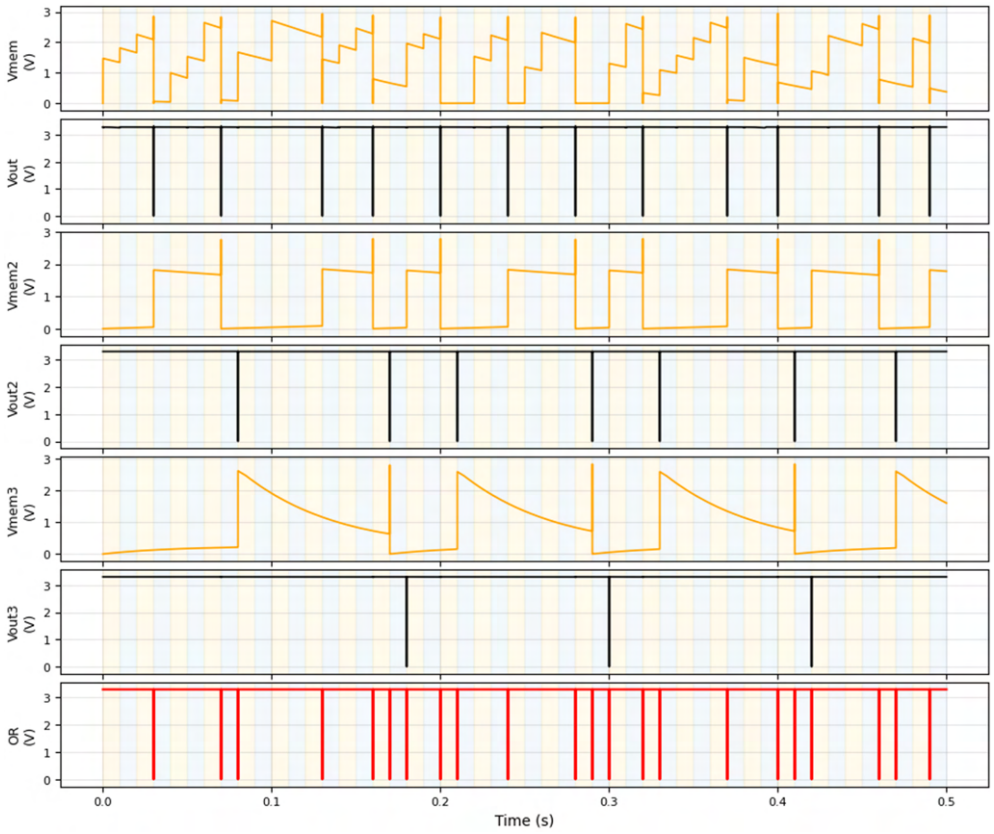


Fig. S21 HSPICE simulation of the eLIFE module implemented with 1T-based spiking neurons and spike-detector circuits. Membrane voltages and active-low detector outputs are shown for the main neuron and two clone neurons. The final trace is an active-low eligible output produced by AND combination of the detector voltages, equivalent to OR combination of spike events. A low output denotes an eligible event; the voltage polarity differs from the active-high binary spike convention used in the algorithmic equations.

S4 Algorithm-level simulation protocols and analyses

S4.1 Software simulation protocols

S4.1.1 Datasets and Data Augmentation

We utilize the datasets as follows:

The MNIST (for S-MNIST) feature images with a resolution of 28×28 and 1 channel, consisting of 55,000 training samples, 5,000 validation samples, and 10,000 test samples[30]. The F-MNIST feature images with a resolution of 28×28 and 1 channel, consisting of 55,000 training samples, 5,000 validation samples, and 10,000 test samples[31]. The spiking heidelberg digits (SHD) dataset, a neuromorphic audio benchmark, consists of 700 input channels representing spike timings, featuring 8,156 training samples and 1,026 test samples.

The CIFAR-10 and CIFAR-100 datasets include images with a resolution of 32×32 and 3 channels (RGB), comprising 45,000 training samples, 5,000 validation samples, and 10,000 test samples[32]. The CIFAR10DVS dataset includes images with a resolution of 128x128 and 2 channels (polarity), comprising 8,900 training samples, 100 validation samples, and 1,000 test samples[33]. The DVS128Gesture dataset features images with a resolution of 128x128 and 2 channels (polarity), including 1,176 training samples and 288 test samples[34].

The ImageNet-1K dataset features images with a resolution of 224 × 224 and 3 channels (RGB), containing approximately 1.28 million training samples and 50,000 validation samples[35].

For object detection tasks, the COCO 2017 dataset includes images with various resolutions (resized to 416 × 416) and 3 channels (RGB), comprising 118,287 training samples and 5,000 validation samples[36].

We employ data augmentation techniques such as normalization, random cropping, horizontal flipping, and cutout[37]. For static datasets (F-MNIST, CIFAR-10 and CIFAR-100 datasets), each image is fed into the model over 5 time steps. For sequential processing of S-MNIST, each image is fed into the model over 28 time steps, receiving a 28-dimensional flattened vector at each step. To process the raw spike timings of event-based neuromorphic datasets (SHD, CIFAR10DVS and DVS128Gesture), the data is formatted into a frame-based structure with 20 distinct time steps. For large-scale datasets (ImageNet-1K and COCO datasets), each image is fed into the model over 4 time steps.

S4.1.2 Training configuration on software-based simulation

For F-MNIST, S-MNIST, and SHD, we train SNNs for 100 epochs using the Adam optimizer. The initial learning rate for Adam is set to 0.001. The learning rate decays according to the cosine annealing learning rate scheduler.

For CIFAR-10, and CIFAR-100 datasets, we train SNNs for 300 epochs using the Adam optimizer. The initial learning rate for Adam is set to 0.0001. The learning rate decays according to the cosine annealing learning rate scheduler. The model is trained with a channel size of 128 and a batch size of 64. For CIFAR10DVS and DVS128Gesture datasets, the model is trained with a channel size of 64 and a batch size of 64.

For the ImageNet dataset, we train SNNs using the AdamW optimizer for 300 epochs in the SOLO configuration and 100 epochs in the SLTT configuration. The initial learning rate for Adam is set to 0.0001, a weight decay of 0.01. The learning rate decays according to the cosine annealing learning rate scheduler. The model is trained with a channel size of 128 and an effective batch size of 256. Additionally, 16-bit mixed precision training is employed to reduce memory overhead.

For object detection, we train SNNs on the COCO dataset for 300 epochs using the SGD optimizer. The initial learning rate is 0.01 with a momentum of 0.937 and a weight decay of 0.0005. After a 3-epoch warmup phase, the learning rate gradually decreases to 0.0001. The effective batch size is set to 64.

S4.1.3 Network Configuration

For the F-MNIST, S-MNIST, and SHD datasets, the network architecture is based on a 4-layer fully connected (4FC) network with a 1024-1024-1024-1024 configuration, incorporating both recurrent and feedback connections. An ablation study detailed below further analyzes the impact of the network configurations.

For the CIFAR-10, CIFAR-100, CIFAR10-DVS, and DVS128 Gesture datasets, the network architecture is based on PreActResNet18. To accommodate different input dimensions of neuromorphic datasets, the PreActResNet18-based architecture prepends a sequence of two 3×3 convolutional layers with 64 channels ($64C3-64C3$).

For the ImageNet-1000 datasets, the network architecture is based on spikingtransformer architecture, configured with 8 attention blocks, a patch size of 16×16, an embedding dimension of 512, and 8 heads[38].

For the object detection, the network architecture is based on EMS-YOLO, adopting the EMSResNet18 block structure[39]. The backbone consists of a sequence of EMS-Modules that gradually increase the number of channels from 32 to 512. In the head, two feature maps (P5 and P4) are extracted and passed to the detection head for the last predictions. P5 is obtained by passing additional EMS-Module in the head, while P4 is generated by concatenating the upsampled P5 feature with an intermediate backbone feature.

S4.2 Offline optimization and fixed implementation of eLIFE

The eLIFE module is optimized offline to approximate eligible spikes from eP2S using randomly generated input sequences of length $T = 28$. At time step t, a binary input $S_{\mathrm{rnd}}[t] \sim \mathrm{Bernoulli}(0.5)$ is multiplied by a random weight $W_{\mathrm{rnd}}[t] \sim \mathcal{U}(0,1)$ and injected into a main LIF neuron according to

$$I_{\mathrm{rnd}}[t] = W_{\mathrm{rnd}}[t] S_{\mathrm{rnd}}[t], \tag{4}$$
$$U_m[t] = \beta U_m[t-1](1 - S_m[t-1]) + I_{\mathrm{rnd}}[t], \tag{5}$$
$$S_m[t] = \Theta(U_m[t] - \vartheta_{\mathrm{th}}). \tag{6}$$

The main-neuron decay factor is $\beta = 0.78$ and the firing threshold is $\vartheta_{\mathrm{th}} = 1$. The generated spikes are supplied to both the reference eP2S module and the two-clone eLIFE module. The reference accumulates spikes with decay factor $\alpha = 0.78$ and produces an eligible spike $\tilde{S}[t]$ when the accumulated value exceeds 0.5. The decay factors correspond to a nominal 4 ms time constant for a simulation interval of approximately 1 ms.

For offline optimization, effective coefficients can be parameterized as $a = \sigma(\theta_a)$, $b = \sigma(\theta_b)$, $c = \sigma(\theta_c)$, and $\beta_k = \sigma(\theta_{\beta_k})$. The clone dynamics are expressed in terms of the effective coefficients as

$$U_{c1}[t] = \beta_1 U_{c1}[t-1](1 - S_{c1}[t-1]) + b S_{c2}[t-1] + (1-a) S_m[t], \tag{7}$$
$$S_{c1}[t] = \Theta(U_{c1}[t] - \vartheta_{\mathrm{th}}), \tag{8}$$
$$U_{c2}[t] = \beta_2 U_{c2}[t-1](1 - S_{c2}[t-1]) + c S_{c1}[t-1] + a S_m[t], \tag{9}$$
$$S_{c2}[t] = \Theta(U_{c2}[t] - \vartheta_{\mathrm{th}}). \tag{10}$$

The complementary main-spike input to $c1$ is $1 - a$, not $\sigma(1 - a)$. The eligible output is the OR of the binary main and clone spikes,

$$\hat{S}[t] = S_m[t] \vee S_{c1}[t] \vee S_{c2}[t] = \max(S_m[t], S_{c1}[t], S_{c2}[t]). \tag{11}$$

Offline fitting minimizes the mean-squared error

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{T} \sum_{t=1}^{T} (\tilde{S}[t] - \hat{S}[t])^2 \tag{12}$$

using Adam with learning rate 0.01 over 1,000,000 randomly generated sequences.

The deployed network uses fixed effective coefficients $a = 0$, $b = c = 1$, $\beta_1 = 0.5$, and $\beta_2 = 1$. Clone $c1$ receives the main spike and the delayed spike of $c2$, while clone $c2$ receives only the delayed spike of $c1$. Unit input and coupling weights are applied directly. In particular, $a = 0$ removes the direct main-spike input to $c2$; no sigmoid is reapplied to a or any other fixed coefficient at execution time. Endpoints 0 and 1 refer to fixed coefficients, not finite sigmoid logits. Algorithmic spike events are active-high; the active-low voltage representation used by the circuit is described in Supplementary Note S3.5.

S4.3 Architecture ablation studies

S4.3.1 Ablation study for recurrent connections and residual paths

For these exploratory architecture analyses, the original SHD training and test partitions were pooled and randomly split into 80% training and 20% validation subsets. These pooled-split results were used only to examine network properties and were not used for benchmark selection or held-out test reporting. We conduct an ablation study exploring recurrent connections and residual paths to effectively capture temporal information within a small network architecture on the SHD dataset. For SLTT, where gradients are computed across all time steps, the network configuration featuring recurrent and residual paths without feedback connections (Supplementary Fig. S22c) achieved the highest performance; introducing excessive recurrent or feedback connections resulted in performance degradation. For SOLO, where gradients are computed at the final time step, the network configuration featuring all available connections (Supplementary Fig. S22d) demonstrates the optimal approach. We observe the potential of the network architecture to compensate for the limited temporal information available to the training method.

Table S3 Performance of ablation for recurrent connections and residual paths. Values are reported as mean ± standard deviation (SD).

Algo.		Network (a)	Network (b)	Network (c)	Network (d)
SLTT	SG	92.37 ± 0.56%	96.33 ± 0.35%	**97.05 ± 0.31%**	93.79 ± 0.23%
	cSG	93.14 ± 0.33%	91.86 ± 1.67%	**97.26 ± 0.30%**	96.55 ± 0.83 %
SOLO	SG	83.84 ± 0.68%	91.62 ± 0.86%	94.24 ± 0.41%	**94.52 ± 0.40%**
	cSG	82.63 ± 1.32%	90.36 ± 0.74%	93.49 ± 0.80%	**93.79 ± 0.23%**

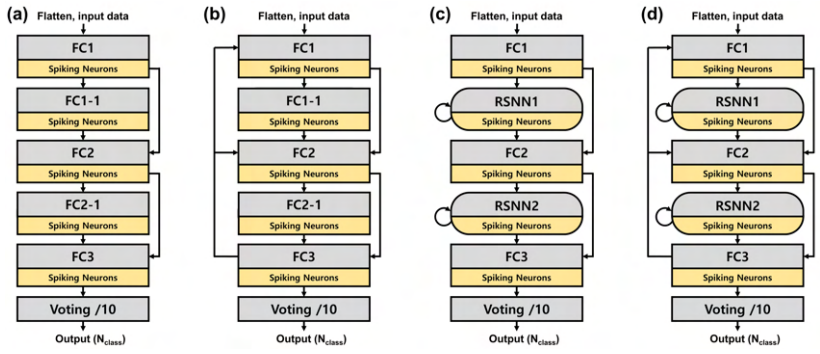


Fig. S22 Architectural variations for recurrent connections and residual paths. (a) Network with only residual paths. (b) Network with residual paths and feedback connections. (c) Network with recurrent connections and residual paths. (d) Network with recurrent connections, residual paths, and feedback connections.

S4.3.2 Ablation study for feedback connections

For these exploratory architecture analyses, the original SHD training and test partitions were pooled and randomly split into 80% training and 20% validation subsets. These pooled-split results were used only to examine network properties and were not used for benchmark selection or held-out test reporting. Next, we conduct an ablation study exploring various feedback connections. For SLTT, the configuration without feedback achieves the best results (Supplementary Fig. S23e). For SOLO, applying feedback connections to all layers except the one containing recurrent connections yields better performance than applying them to every layer (Supplementary Fig. S23c).

Table S4 Performance of ablation for feedback connections. Values are reported as mean ± standard deviation (SD).

Algo.		Feedback (a)	Feedback (b)	Feedback (c)	Feedback (d)	Feedback (e)
SLTT	cSG	95.53 ± 0.37%	95.49 ± 0.83%	96.55 ± 0.83%	96.14 ± 0.53%	97.04 ± 0.53%
SOLO	cSG	92.76 ± 0.69%	91.53 ± 0.56%	93.79 ± 0.23%	90.90 ± 0.37%	92.27 ± 0.43%

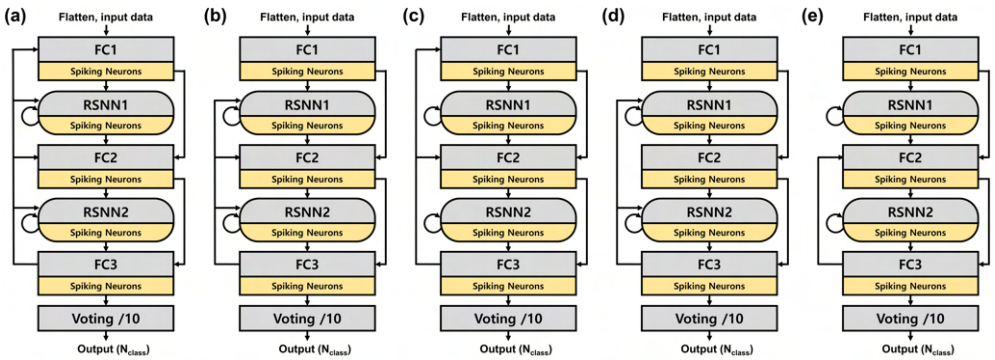


Fig. S23 Architectural variations for different feedback connections.

S4.4 Exploratory firing-rate analysis of EMS-YOLO

While the SLTT method demonstrates a faster initial convergence compared to the SOLO method, it results in a higher overall firing rate after stabilization. Conversely, the SOLO method exhibits a slower convergence trajectory but maintains a lower firing rate throughout the training process. Furthermore, applying the current-based SG consistently shows noticeably lower firing rates than standard SG. The SOLO configuration with cSG has the lowest firing rate among the evaluated conditions; total hardware energy is not quantified by this comparison.

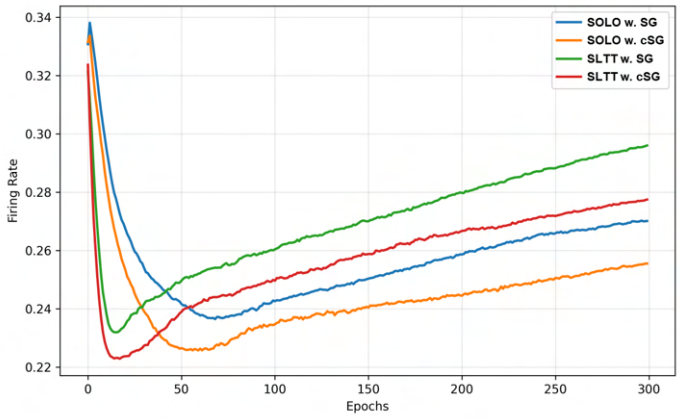


Fig. S24 Firing rate of Spiking YOLO trained on the COCO dataset. Applying the cSG consistently suppresses neuronal activity, with the SOLO configuration achieving the lowest firing rate.

S5 Analog on-chip continual-learning workflow

S5.1 Hardware-aware and analog-aware fine-tuning workflow

The experiments are conducted using a two-phase training methodology, consisting of a pre-training phase and a post-training phase[40].

During the hardware-aware pre-training phase, the networks are trained in a software-based simulation environment for 100 epochs using the Adam optimizer, with an initial learning rate of 0.001 that decays according to a cosine annealing schedule. At this stage, hardware-aware training (HAT) restricted to 64 states (6-bit precision) is implemented to constrain the synaptic weights, alongside device variation modeling for both the neurons and the surrogate gradients.

Following this, the analog-aware post-training phase is initiated by mapping the pre-trained network weights onto analog crossbar tiles within the aihwkit framework[41]. Analog-aware fine-tuning is then executed for 10 epochs, utilizing the AnalogAdam optimizer with a learning rate of 0.01. During this phase, the analog synapse model is actively employed to incorporate hardware non-idealities, alongside device variation modeling for both the neurons and the surrogate gradients.

For the spiking VGG9 comparisons in Supplementary Fig. S27a, CIFAR-10 uses SLTT-pretrained weights, whereas CIFAR10DVS uses STBP-pretrained weights. Post-training uses STBP or SOLO as indicated in the figure, and the learning curves in panel b start from STBP-pretrained weights. The 10-epoch AnalogAdam configuration applies to the batch fine-tuning workflow. Rehearsal-assisted adaptation from SLTT-pretrained Arch. 1 or SOLO-pretrained Arch. 2 weights is a separate protocol using sample-wise updates and the selective SGD policies described in Supplementary Note S5.3.1.

S5.1.1 Analog modeling for synapse, neuron and surrogate gradient

The synapse model captures weight updates based on experimentally measured conductance data, with its potentiation curve fitted using a one-sided unit configuration within a differential synaptic pair (Supplementary Fig. S25a). Additionally, the neuron model incorporates the time constant, which is extracted by analyzing voltage decay characteristics across varying numbers of applied pulses (Supplementary Fig. S25b). Using the relation $\beta = \exp(-t/\tau)$, the extracted decay factor is 0.8772 ± 0.0984 at $t = 10$, which is explicitly incorporated into our neuron model.

Furthermore, a boxcar function is applied to the surrogate gradient. By mapping the boxcar range to an instantaneous synaptic current via backward and forward latches, the threshold voltage variation inherent in the physical devices is integrated into our model (see Supplementary Fig. S13).

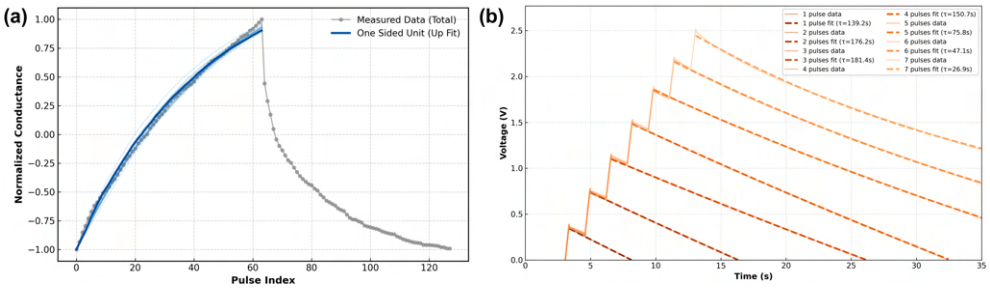


Fig. S25 (a) Synapse modeling based on measured normalized conductance data. The potentiation (up) curve is fitted using a one-sided unit configuration. (b) Neuron time constant (τ) fitting based on measured data. The plot demonstrates the exponential decay of voltage over time after applying consecutive pulses, used to extract and fit the time constant of the neuron.

S5.1.2 Analog spiking neural network

While the core multiply-accumulate (MAC) operations in analog neural networks offer significant energy efficiency, purely analog signals are highly susceptible to device mismatch, thermal noise, and signal attenuation. To prevent severe information loss caused by hardware non-idealities, the initial input layers and the final output layers in most analog neural networks are typically implemented as fully digital layers rather than analog blocks. Computing the boundary layers of the network in the digital domain preserves raw input features and final classification logits[42].

Despite these architectural mitigations, without any post-training to compensate for the mapped non-idealities, the accuracy of the analog spiking VGG-9 drastically collapses to near-random levels (10.86% and 11.84%). However, applying the 10-epoch post-training significantly recovers the performance.

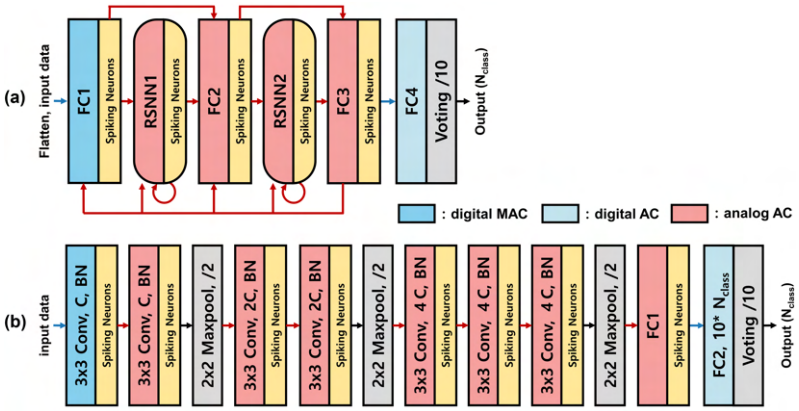


Fig. S26 (a) Analog network for Arch. 2. (b) Analog network for spiking VGG-9.

S5.2 Hardware-aware fine-tuning of spiking VGG-9

The hardware-aware fine-tuning results show that incorporating each device-level non-ideality individually produces only a limited accuracy change on CIFAR-10. CIFAR10DVS is more sensitive to the accumulated analog non-idealities, resulting in a larger accuracy reduction after weight mapping. Nevertheless, analog-aware fine-tuning recovers the performance in both cases. The displayed learning curves show accuracy increases of up to approximately 9 percentage points over the fine-tuning epochs.

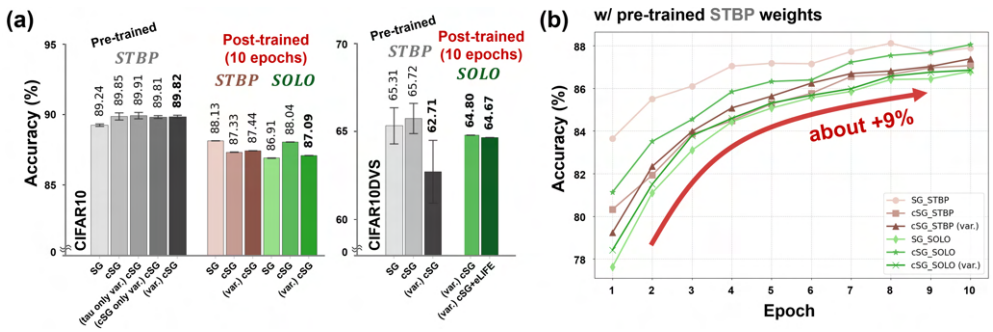


Fig. S27 Hardware-aware fine-tuning of spiking VGG-9. (a) Pretrained and post-trained classification accuracy under individual and combined hardware non-idealities. CIFAR-10 uses SLTT-pretrained weights, whereas CIFAR10DVS uses STBP-pretrained weights; post-training rules are indicated above the bars. (b) Accuracy evolution during 10-epoch fine-tuning from STBP-pretrained weights using STBP or SOLO, with gains of up to approximately 9 percentage points over the displayed epochs.

S5.3 Online continual-learning framework of analog SNNs

S5.3.1 Online learning and rehearsal policy

Rehearsal-assisted adaptation is evaluated independently for withheld SHD target Classes 0–9, with batch size 1 for sample-wise updates. Online therefore denotes updates over repeated class-data passes, not a single-pass stream or a sequence of newly arriving classes. Existing SLTT-pretrained Arch. 1 and SOLO-pretrained Arch. 2 checkpoints are reused from 100-epoch training with batch size 64, Adam, an initial learning rate of 0.001, and cosine annealing. Checkpoints follow the condition-specific pretraining configuration, with SG-pretrained weights for SG conditions and cSG-pretrained weights for cSG conditions. Shared weights are loaded into the feedback-connected adaptation network, and feedback layers absent from the checkpoint are initialized to zero. Supplementary Fig. S28 illustrates the target-Class-3 run. Replay is software-simulated rather than implemented in hardware-resident memory.

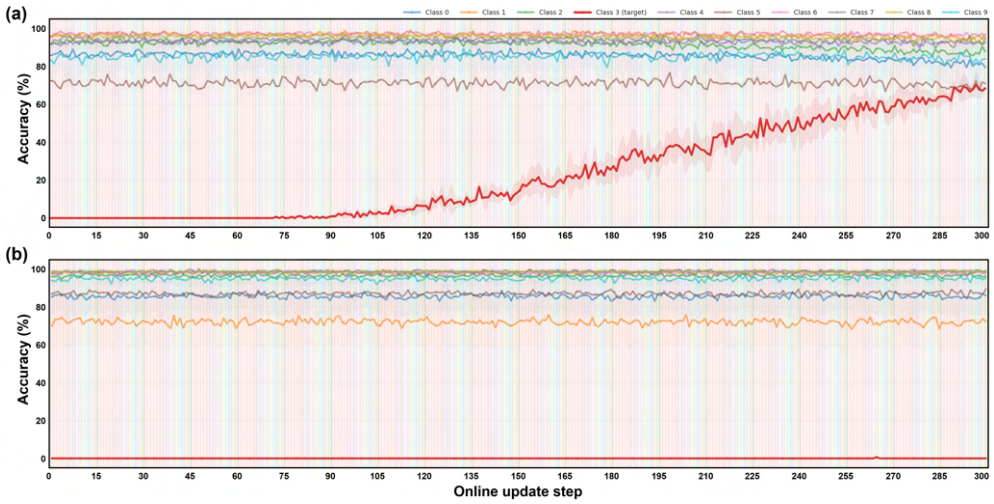


Fig. S28 Rehearsal-assisted online adaptation to independently withheld target Class 3 under modeled analog characteristics. (a) SOLO-initialized weights enable progressive target-class learning while maintaining comparatively stable accuracy for previously learned classes. (b) SLTT-initialized weights show limited target-class adaptation under the same update policy. Colored background bands indicate alternating target-learning and rehearsal phases; each phase contains sample-wise updates over the corresponding class dataset.

Each round contains five passes through the target-class training dataset followed by one rehearsal pass through each of the ten class datasets, including the target class. Rehearsal follows a cyclic class order from the target class to Class 9 and then from Class 0 to the class preceding the target. Samples are shuffled within each class pass. The 20 rounds comprise 300 class passes. The schedule applies the general replay principle of revisiting past data to limit forgetting[43,44]; the class order and selective-update rules are specific to the present experiment.

Only feedback weights FB0–FB3 and the output projection `outw` are made trainable; all remaining parameters are frozen. The bias-free output projection maps the final spiking layer to 200 readout units, and averaging each consecutive group of ten units produces 20 class scores. Reported comparisons evaluate Classes 0–9. Before adaptation, the target-class block is reinitialized using Xavier uniform initialization while the remaining output blocks retain their pretrained values. During target learning, a gradient mask retains only rows $10t$ through $10t + 9$ for target class t, leaving the other output blocks unchanged.

Feedback weights are updated using AnalogSGD at learning rate 0.001 during both target learning and rehearsal. The output projection is updated using SGD at learning rate 0.005 during target learning only; its gradient is removed and its optimizer step is omitted during rehearsal. The two learning rates apply to separate parameter groups rather than the whole network. Mean-squared error is computed against a 20-class one-hot target, and the gradient norm is clipped to 1.0 before each sample update.

The four conditions in main Fig. 5 use matched seed identifiers 1, 2, and 42 and the same schedule and update policy. Baseline accuracy is tested after analog weight mapping, or at the corresponding initialized digital stage, before adaptation. Final accuracy is tested after all 20 rounds, not at an intermediate best-accuracy checkpoint, and each change is calculated relative to the corresponding condition's baseline. Each point is a three-seed mean for one target class; connections pair the same target across implementations, and diamonds and horizontal bars average over the ten target conditions. Target conditions are not independent seed replicates. Aggregate-only records for some conditions prevent reconstruction of all paired-run variances, so proximity of the displayed means is not interpreted as statistical equivalence or significance.

S5.3.2 Comparison of pretrained initializations

Supplementary Fig. S29b,c supports selecting SOLO-pretrained Arch. 2 for the tested adaptation setting. SLTT-pretrained initialization fails to achieve meaningful target-class learning under the tested conditions, despite small nonzero accuracy changes in some cells, whereas SOLO-pretrained initialization supports clear target-class gains. The adaptation result does not imply failure of SLTT pretraining. Target-class gains accompanied by comparatively limited changes in previously seen classes suggest more controlled adaptation in the selected SOLO configuration. The comparison does not isolate an intrinsic class-control advantage of SOLO because architecture and pretraining rule both differ, and selective updates, rehearsal, and eLIFE also contribute to retention.

Computational cost provides an additional reason to favor SOLO for on-chip learning. For a sequence of T time steps, SLTT evaluates spatial gradients at every step, so the number of backward evaluations grows with T. SOLO evaluates a spatial gradient only at the final selected step, reducing the number of backward evaluations from T to one. Both methods still process the forward sequence; the reduction concerns backward computation and does not make total sequence-processing cost independent of T. Fewer backward evaluations are advantageous when the computation available for on-chip updates is constrained. Both pretrained initializations in the adaptation comparison nevertheless use SOLO during adaptation, so the initialization results are not attributed to different adaptation rules or interpreted as a measured runtime or energy comparison.

The error bars in Supplementary Fig. S29b are propagated dispersion estimates calculated from baseline and post-adaptation sample SDs, each reported over three training seeds. Under a zero-covariance approximation across endpoints and evaluation classes, the estimate for target t is

$$\sigma_{\mathrm{prop},t} = \frac{1}{10}\sqrt{\sum_{j=0}^{9}\left[(\sigma_{t,j}^{\mathrm{post}})^2 + (\sigma_{t,j}^{\mathrm{base}})^2\right]}, \tag{13}$$

These bars describe dispersion inferred from marginal summary SDs, not empirical SDs of paired seed-wise accuracy changes, and are not used for significance testing.

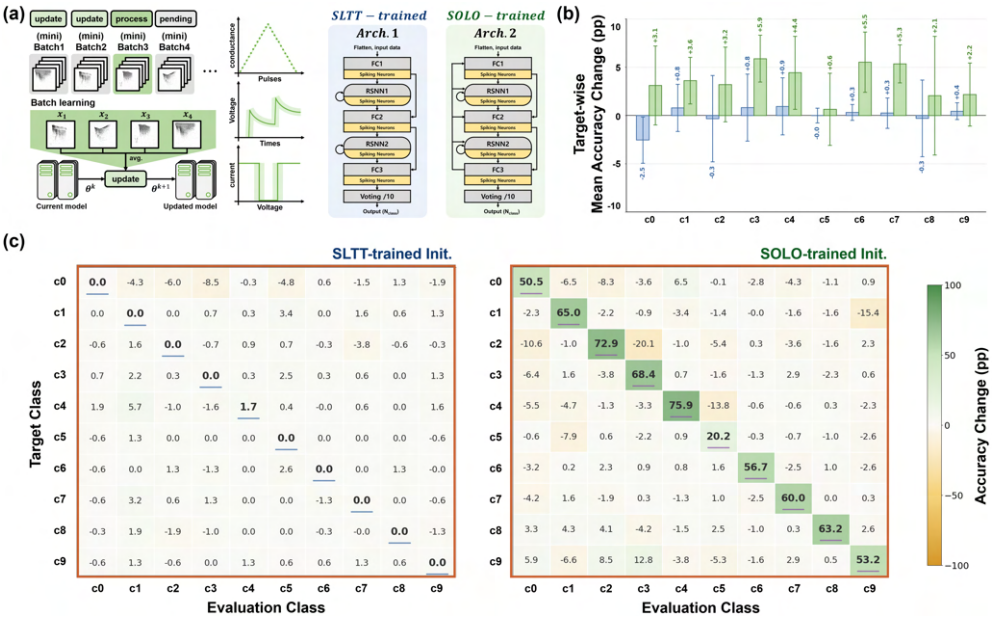


Fig. S29 Adaptation from existing SLTT- and SOLO-pretrained initializations on SHD. (a) Overview of the previously used hardware-aware batch pretraining and the two network structures. SLTT computes spatial gradients at each time step with Arch. 1, whereas SOLO uses the final-time-step spatial gradient with the feedback-connected Arch. 2. These are reused pretrained weights, not newly pretrained models for this comparison. (b) For each withheld target class, the mean accuracy change across all ten evaluation classes after online adaptation. Error bars show propagated dispersion from three-seed baseline and post-adaptation summary SDs, not empirical paired-run SDs (Supplementary Note S5.3.2). (c) Class-wise post-minus-baseline accuracy-change matrices for the two initializations. Rows indicate target classes and columns indicate evaluation classes; diagonal values indicate target-class gains and off-diagonal values indicate changes in previously seen classes. Shared weights from Arch. 1 are transferred to Arch. 2 with newly introduced feedback weights initialized to zero; SOLO-pretrained Arch. 2 weights are mapped directly. Subsequent adaptation uses SOLO, updating feedback weights FB0–FB3 and only the target-class ten-row output-weight block during target learning, with feedback-only updates during output-frozen rehearsal (Supplementary Note S5.3.1). The compared initializations differ in architecture as well as pretraining rule.

S5.4 Digital SG with and without eLIFE

In addition to the cSG conditions in main Fig. 5, Supplementary Fig. S30 compares digital SG alone and with eLIFE at output-projection learning rates of 0.005 and 0.01. The rates apply only to the output weights, not to the whole network, and the comparison does not establish an independent feedback-rate effect. Diagonal elements show target-class accuracy gains, while off-diagonal elements show accuracy changes in previously seen classes. All accuracy changes are differences in percentage points, not relative percentage changes. At both output-weight rates, including eLIFE reduces average old-class accuracy losses, although target-class gains can also decrease and individual off-diagonal entries do not all improve. Precise off-diagonal means are not inferred from rounded image annotations because the source matrices are not included in the manuscript repository.

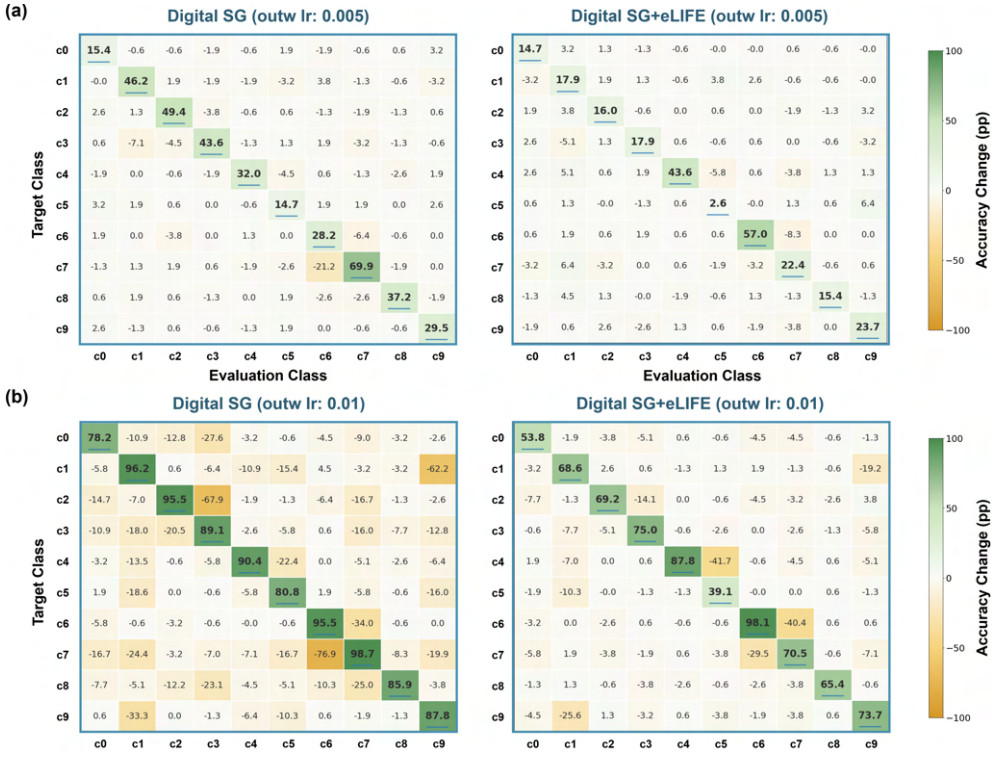


Fig. S30 Digital conventional surrogate gradient (SG) versus digital SG+eLIFE during target-class adaptation on SHD. (a) Output-weight (outW, the weights from the final spiking layer to the class readout) learning rate 0.005. (b) Output-weight learning rate 0.01. Each panel shows post-minus-baseline accuracy change in percentage points with the target class on the vertical axis and evaluation class on the horizontal axis. Diagonal cells measure target-class gain; off-diagonal cells measure changes in previously seen classes. These rates apply to output weights, not to the entire network. eLIFE reduces mean old-class losses in these comparisons, though target-class gains can also decrease; the panels do not establish an advantage in every cell.

S5.5 Representation and output-state analyses

Exploratory analyses examine Arch. 2 with all four feedback connections (FB0–FB3) enabled after pretraining with SOLO or SLTT. Layer-wise t-SNE visualizations compare feature representations before and after fine-tuning (Supplementary Figs. S31 and S32), while accumulate-neuron traces illustrate output-state dynamics over the input sequence (Supplementary Figs. S33 and S34). The representation analyses are separate from the initialization comparison in Supplementary Fig. S29 and do not independently establish a training-rule-specific retention advantage.

S5.5.1 t-SNE visualization of feature representations

Before fine-tuning.

The original SHD training and test partitions are pooled and randomly split into 80% training and 20% validation subsets for the exploratory representation analysis, not for benchmark selection or held-out test reporting. Supplementary Fig. S31 shows layer-wise features before fine-tuning for (a) SOLO-pretrained and (b) SLTT-pretrained weights. Red points denote Class 0 and remain dispersed across parts of the projected feature space. The two-dimensional projections provide a qualitative view of class separation rather than a quantitative measure of recognition or retention.

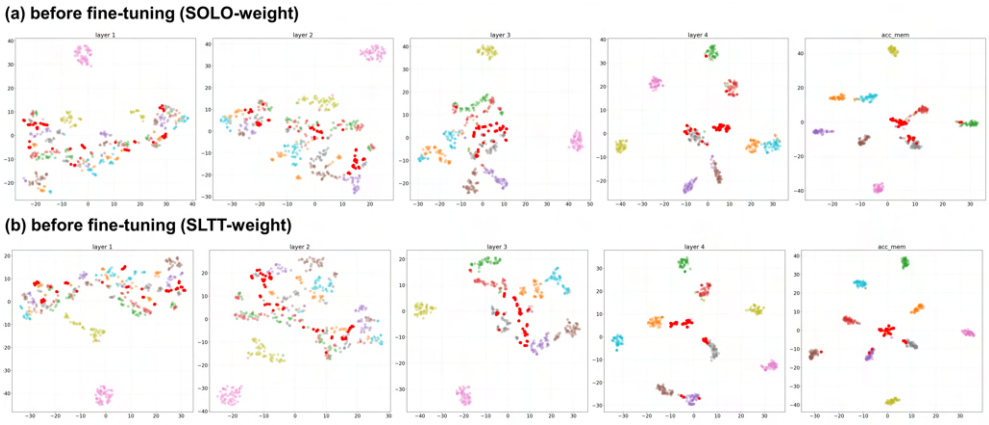


Fig. S31 t-SNE visualization before fine-tuning based on (a) SOLO-weight and (b) SLTT-weight. Red dots representing Class 0 appear somewhat scattered.

After fine-tuning.

The original SHD training and test partitions were pooled and randomly split into 80% training and 20% validation subsets for this exploratory representation analysis. These pooled-split results are not used for benchmark selection or held-out test reporting. Visualizations depict feature distributions after fine-tuning. For configurations with a frozen network based on SOLO-weight (Supplementary Fig. S32a) and SLTT-weight (Supplementary Fig. S32b), red dots corresponding to Class 0 gather into clusters. In the unfrozen configurations (Supplementary Fig. S32c,d), these dots appear more scattered. These visualizations alone do not quantify class-wise forgetting.

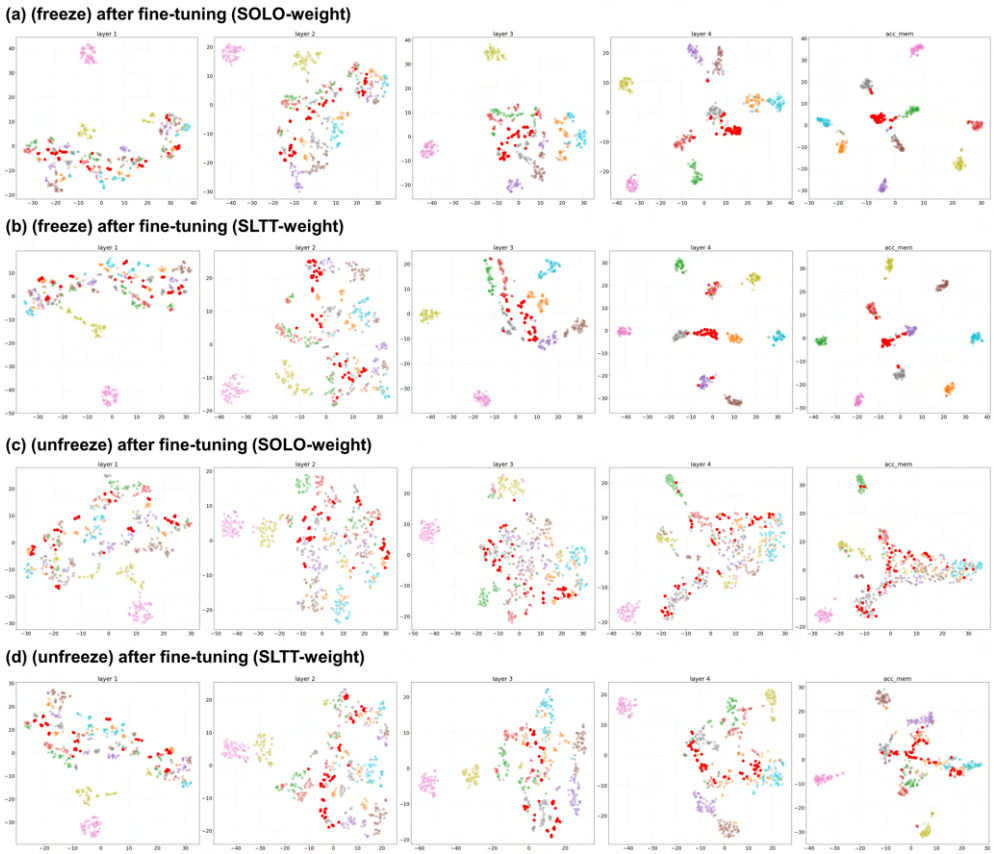


Fig. S32 Layer-wise t-SNE visualizations after fine-tuning using (a,c) SOLO-pretrained and (b,d) SLTT-pretrained weights. Panels (a,b) retain a frozen pretrained backbone during selective fine-tuning, whereas panels (c,d) allow backbone updates. Red points denote Class 0. Class 0 points form more compact groups in the displayed late-layer projections with the backbone frozen and appear more dispersed with backbone updates. The projections do not quantify forgetting rates.

S5.5.2 Accumulate-neuron responses

Before fine-tuning.

The original SHD training and test partitions were pooled and randomly split into 80% training and 20% validation subsets for this exploratory representation analysis. These pooled-split results are not used for benchmark selection or held-out test reporting. Network outputs represent logits calculated as the average membrane potential of accumulate neurons across time steps. Membrane potentials for other classes show an upward trend over time, whereas the average membrane potential for Class 0 does not rise in these examples. Visualizations use SOLO-weight (Supplementary Fig. S33a) and SLTT-weight (Supplementary Fig. S33b); these traces alone do not establish a forgetting rate.

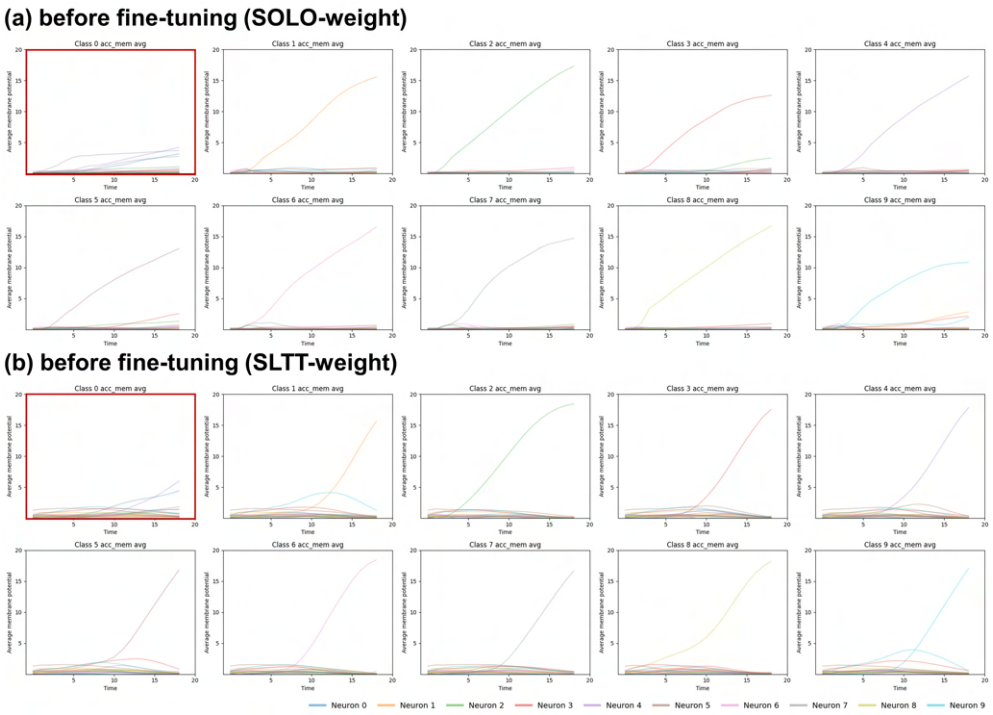


Fig. S33 Average membrane potential of accumulate neurons across time steps before fine-tuning based on (a) SOLO-weight and (b) SLTT-weight. The membrane potential for Class 0 fails to increase, whereas values for other classes show distinct upward trends.

After fine-tuning.

The after-fine-tuning traces use the same exploratory pooled-split protocol and are not used for benchmark selection or held-out test reporting. Supplementary Fig. S34 shows an increasing Class 0 output response with (a) SOLO-pretrained and (b) SLTT-pretrained weights under the displayed selective-fine-tuning conditions. Responses associated with other classes also change. The traces illustrate output-state adaptation and are not evidence that the separate constrained adaptation comparison in Supplementary Fig. S29 yields equivalent target-class learning from both initializations.

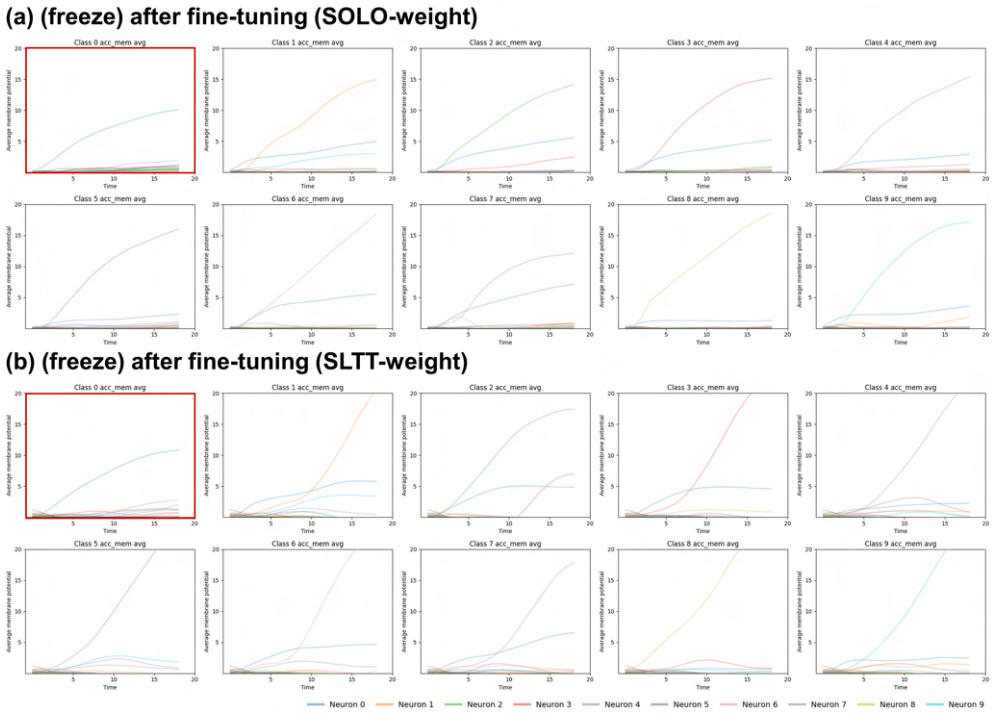


Fig. S34 Average membrane potential of accumulate neurons across time steps after fine-tuning based on (a) SOLO-weight and (b) SLTT-weight. The membrane potential for Class 0 successfully increases along with other classes.

S6 Device benchmarks and energy analysis

S6.1 Benchmarks of 1T-neuron

S6.1.1 Comparison with analog circuit-based neurons

The 1T-neuron combines a compact footprint with LIF neuronal behavior. The measured energy consumption includes the effect of a relatively large parasitic capacitance (on the order of a few pF) introduced by the measurement setup. Therefore, the energy consumption is expected to be further reduced as the C_{par} decreases to approximately 80 fF.

	Architecture	Target model	Energy (pJ/spike)	Area (F^2)
[45]	350 nm CMOS	AIF neuron	900	21,000
[46]	65 nm CMOS	LIF neuron	41.3	127,000
[47]	65 nm CMOS	AdEx-IF neuron	200	796,000
[48]	65 nm CMOS	LIF neuron	4	30,000
[49]	32 nm SOI MOSFET	LIF neuron	35	1,760
Our work	200 nm SOI MOSFET	LIF neuron	31	6

Table S5 Comparison of the 1T-neuron with analog CMOS circuit-implemented neurons

S6.1.2 Comparison with device-based neurons

The proposed 1T-neuron combines a compact footprint, LIF dynamics, and CMOS compatibility compared with representative device-based neurons. Leaky integration supports the LIF dynamics used in the SNN model, while nonvolatile V_{firing} calibration enables mitigation of device-to-device variation.

Ref.	Structure	Target model	Energy (pJ/spike)	Footprint (F^2)	CMOS comp.	Additional func.
[50]	Biristor (InGaAs)	Oscillatory neuron	270	4	X	Photoresponsivity
[51]	SOI	LIF neuron	45	6	O	Excitation/ inhibition
[52]	Bulk-Si	Oscillatory neuron	75	12	O	Adaptation/ bursting
[53]	SOI (1B2T)	Oscillatory neuron	105	14	O	None
[54]	Pt/Nb/NbO_x/W	LIF neuron	20	4	X	None
[55]	Au/Ti/VO_2/Al_2O_3	Oscillatory neuron	0.0029	4	X	V_{firing} calibration
[56]	Mo/$Ge_{0.6}Se_{0.4}$/Mo	Oscillatory neuron	0.001	4	X	Adaptation/ chaotic activity
Our work	**SOI (1T)**	**LIF neuron**	**31**	**6**	**O**	**V_{firing} calibration**

Table S6 Comparison of the 1T-neuron with representative device-based neurons. CMOS compatibility is indicated by O (compatible) and X (not compatible). For this work, the measured 31 pJ/spike includes measurement-setup parasitics and is distinct from the scaled-capacitance estimate used in the inference-energy model.

S6.2 Additional explanation of power analysis

S6.2.1 Interface circuit-aware power consumption of 1T-synapse

The power consumption of the 1T-synapse is analyzed by taking into account not only the single device, but also the current-mirror-based interface circuit. This confirms that the state-dependent current is effectively transferred and maintained after circuit-level interfacing, enabling a more reliable assessment of the overall system-level power consumption.

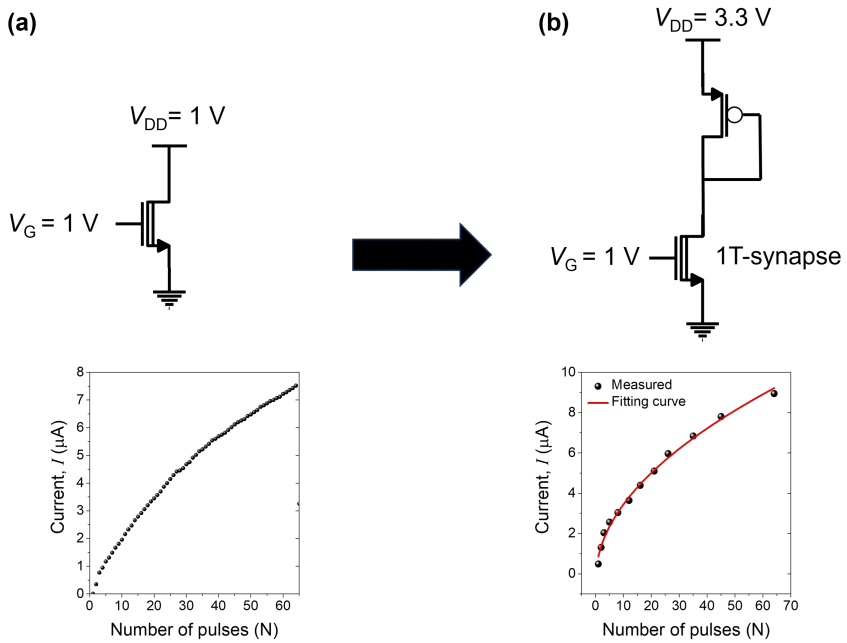


Fig. S35 (a) Measured current as a function of the synaptic state. (b) Measured current obtained through the interface circuit.

S6.2.2 Power consumption of 1T-neuron

The power consumption of the 1T neuron during LIF operation is quantified using the measurement setup shown in Supplementary Fig. S36a.

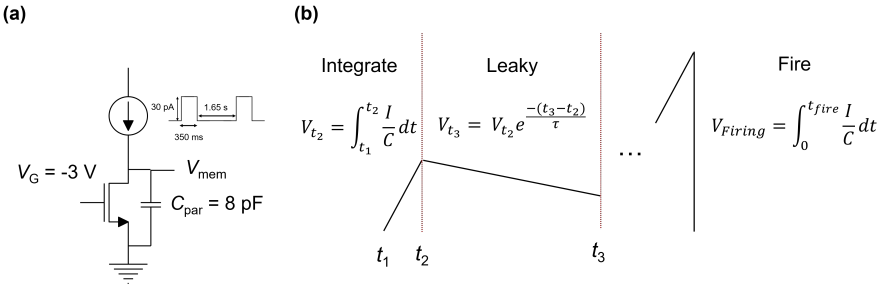


Fig. S36 (a) Measurement scheme used to estimate the power consumption of the 1T-neuron. (b) Theoretical framework for estimating the power consumption of the 1T-neuron during the integration, leakage, and firing processes.

S6.2.3 Estimation of unit energy parameters

The baseline component accounts for periodic membrane potential updates and leakage dynamics, whereas the active component reflects the dynamic energy dissipated during synaptic accumulation and spike generation. The unit parameters used in the inference-energy model are summarized in Supplementary Table S7.

Table S7 Unit energy parameters used in the inference-energy model. The analog AC value is specified per individual device; a differential G_P/G_M pair therefore costs 0.10 pJ per logical signed synaptic operation. Analog neuronal values are model estimates.

Parameter	Value	Basis
E^d_{MAC}	4.6 pJ per MAC	Reference FP32 MAC [57]
E^d_{AC}	0.9 pJ per AC	Reference FP32 accumulation [57]
E^a_{AC}	0.05 pJ per device	5 μW $\times$ 10 ns read operation
E^d_{up}	9.736 pJ per neuron per time step	Digital membrane update
E^d_{fire}	1.0 pJ per spike	Digital spike generation
E^a_{up}	0.0072 pJ per neuron per time step	Estimated leak energy
E^a_{fire}	0.3136 pJ per spike	Estimated capacitor discharge

The energy for a digital Multiply-Accumulate (MAC) operation, E^d_{MAC}, is set to 4.6 pJ, and the digital accumulation (AC) energy, E^d_{AC}, is estimated at 0.9 pJ [57]. These reference costs use 32-bit floating-point arithmetic in 45 nm CMOS at 0.9 V: the MAC cost is the sum of a 3.7 pJ multiplication and a 0.9 pJ addition. They are adopted model parameters, not measurements of a digital implementation matched to the fabricated platform.

The energy for analog accumulation, E^a_{AC}, is assumed to be 0.05 pJ per individual 1T synaptic device. This value is derived from experimental measurements indicating that an individual synaptic device with a current-mirror interface consumes approximately 5 μW of power during operation with a read pulse duration of 10 ns. Signed weights are represented by a differential G_P/G_M pair; accordingly, the analog accumulation term in the main-text energy equation applies an explicit factor of two, resulting in 0.10 pJ per logical signed synaptic operation.

The digital basal energy, E^d_{up}, is assumed to be 9.736 pJ per time step and the digital firing energy, E^d_{fire}, is assumed to be 1.0 pJ per spike [58]. The basal energy for maintaining and updating the digital registers that store the membrane potential includes memory access, leakage in the register file, and the comparison logic executed at every time step. The firing energy covers the dynamic energy dissipated during the toggle of the output spike and the subsequent reset of the digital membrane potential register to its initial state.

The analog neuronal energy is estimated using a baseline capacitance of 0.08 pF, excluding the measurement-setup parasitic capacitance. The basal energy, E^a_{up}, is set to 0.0072 pJ per neuron per time step and represents the decrease in stored capacitive energy during input-free leakage. Leakage is evaluated over one energy-model time step using the assumed exponential decay, starting from the firing-threshold voltage to obtain an upper estimate for nonnegative subthreshold voltages. Charging energy

is excluded from this basal term. The firing energy, E^a_{fire}, is set to 0.3136 pJ per spike and is estimated from the decrease in stored capacitive energy as the membrane voltage falls from the firing threshold to the reset voltage. These parameters describe capacitive leak and firing losses rather than a full circuit-level supply-energy measurement.

S6.2.4 Estimation of operational counts

In the main text, high-level operational counts ($OP_{MAC}^{(l)}$ and $OP_{AC}^{(l)}$) are utilized to maintain the clarity of the total energy equation, fundamentally encapsulating the hardware-specific mapping onto $N_{xbar} \times N_{xbar}$ crossbar arrays[58]. The total number of operations for a given layer l is strictly determined by the required Processing Element (PE) cycles, denoted as $C_{pe}^{(l)}$, multiplied by the crossbar array size. The conversion from hardware parameters to the high-level operational counts is formulated as follows:

$$OP_{MAC}^{(l)} = C_{pe}^{(l)} \cdot N_{xbar}^2 \tag{14}$$

$$OP_{AC}^{(l)} = C_{pe}^{(l)} \cdot N_{xbar}^2 \tag{15}$$

The number of PE cycles $C_{pe}^{(l)}$ inherently depends on the network architecture and the crossbar dimensions ($N_{xbar} = 32$). For a fully connected (FC) layer with N_{in} inputs and N_{out} outputs, $C_{pe}^{(l)}$ is calculated by partitioning the weight matrix into crossbar-sized blocks:

$$C_{pe}^{(l)}(\text{FC}) = \left\lceil \frac{N_{in}}{N_{xbar}} \right\rceil \times \left\lceil \frac{N_{out}}{N_{xbar}} \right\rceil \tag{16}$$

For a convolutional layer with a kernel size of K, input channels C_{in}, output channels C_{out}, and an output feature map size of $H_{out} \times W_{out}$, the mapping requires unrolling the sliding window operations:

$$C_{pe}^{(l)}(\text{Conv}) = \left\lceil \frac{K^2 \cdot C_{in}}{N_{xbar}} \right\rceil \times \left\lceil \frac{C_{out}}{N_{xbar}} \right\rceil \times H_{out} \times W_{out} \tag{17}$$

By defining OP_{MAC} and OP_{AC} through $C_{pe}^{(l)}$, our energy model accurately reflects both the spatial dimension of the crossbar arrays and the temporal dynamics of the spiking neural network.

References

1. Neftci, E. O., Mostafa, H. & Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* **36**, 51–63 (2019).

2. Wu, Y., Deng, L., Li, G., Zhu, J. & Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* **12**, 331 (2018).

3. Bellec, G. *et al.* A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* **11**, 3625 (2020).

4. Xiao, M., Meng, Q., Zhang, Z., He, D. & Lin, Z. Online training through time for spiking neural networks. In *Advances in Neural Information Processing Systems*, Vol. 35, 20717–20730 (Curran Associates, Inc., 2022). URL https://proceedings.neurips.cc/paper_files/paper/2022/file/82846e19e6d42ebfd4ace4361def29ae-Paper-Conference.pdf.

5. Meng, Q. *et al.* Towards memory- and time-efficient backpropagation for training spiking neural networks. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6143–6153 (IEEE, Paris, France, 2023).

6. Lee, S., Moon, H., Yoon, M., Heo, W. & Kim, S. SOLO: Spatial online learning at once for spiking neural networks. *Neurocomputing* **663**, 131940 (2026).

7. Pehle, C. *et al.* The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci.* **16**, 795876 (2022).

8. Merolla, P. A. *et al.* A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).

9. Pei, J. *et al.* Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572**, 106–111 (2019).

10. Davies, M. *et al.* Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).

11. Orchard, G. *et al.* Efficient neuromorphic signal processing with Loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 254–259 (IEEE, Coimbra, Portugal, 2021).

12. Furber, S. B., Galluppi, F., Temple, S. & Plana, L. A. The SpiNNaker project. *Proc. IEEE* **102**, 652–665 (2014).

13. Gonzalez, H. A. *et al.* SpiNNaker2: A large-scale neuromorphic system for event-based and asynchronous machine learning. Preprint at https://arxiv.org/abs/2401.04491 (2024).

14. Schemmel, J. *et al.* A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 1947–1950 (IEEE, Paris, France, 2010).

15. Zenke, F. & Ganguli, S. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Comput.* **30**, 1514–1541 (2018).

16. Kaiser, J., Mostafa, H. & Neftci, E. Synaptic plasticity dynamics for deep continuous local learning (DECOLLE). *Front. Neurosci.* **14**, 424 (2020).

17. Müller, E. *et al.* A scalable approach to modeling on accelerated neuromorphic hardware. *Front. Neurosci.* **16**, 884128 (2022).

18. Greatorex, H. *et al.* A neuromorphic processor with on-chip learning for beyond-CMOS device integration. *Nat. Commun.* **16**, 6424 (2025).

19. Dalgaty, T. *et al.* Mosaic: in-memory computing and routing for small-world spike-based neuromorphic systems. *Nat. Commun.* **15**, 142 (2024).

20. Lim, J. G. *et al.* Design of CMOS-memristor hybrid synapse and its application for noise-tolerant memristive spiking neural network. *Front. Neurosci.* **19**, 1516971 (2025).

21. Hajizada, E. *et al.* Real-time continual learning on Intel Loihi 2. Preprint at https://arxiv.org/abs/2511.01553v1 (2025).

22. Gao, T., Deng, B., Wang, J. & Yi, G. Presynaptic spike-driven plasticity based on eligibility trace for on-chip learning system. *Front. Neurosci.* **17**, 1107089 (2023).

23. Weilenmann, C. *et al.* Single neuromorphic memristor closely emulates multiple synaptic mechanisms for energy efficient neural networks. *Nat. Commun.* **15**, 6898 (2024).

24. Lu, Y. *et al.* In-memory realization of eligibility traces based on conductance drift of phase change memory for energy-efficient reinforcement learning. *Adv. Mater.* **34**, 2107811 (2022).

25. Wunderlich, T. C. & Pehle, C. Event-based backpropagation can compute exact gradients for spiking neural networks. *Sci. Rep.* **11**, 12829 (2021).

26. Wei, W. *et al.* Event-driven learning for spiking neural networks. Preprint at https://arxiv.org/abs/2403.00270 (2024).

27. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw.* **10**, 1659–1671 (1997).

28. Chen, C.-D. *et al.* Single-transistor latch in SOI MOSFETs. *IEEE Electron Device Lett.* **9**, 636–638 (1988).

29. Han, J.-K. *et al.* Cointegration of single-transistor neurons and synapses by nanoscale CMOS fabrication for highly scalable neuromorphic hardware. *Sci. Adv.* **7**, eabg8836 (2021).

30. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).

31. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Preprint at https://arxiv.org/abs/1708.07747 (2017).

32. Krizhevsky, A. Learning multiple layers of features from tiny images. Tech. Rep., University of Toronto, Toronto, ON, Canada (2009).

33. Li, H., Liu, H., Ji, X., Li, G. & Shi, L. CIFAR10-DVS: An event-stream dataset for object classification. *Front. Neurosci.* **11**, 309 (2017).

34. Amir, A. *et al.* A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7388–7397 (IEEE, Honolulu, HI, 2017).

35. Deng, J. *et al.* ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255 (IEEE, Miami, FL, 2009).

36. Lin, T.-Y. *et al.* Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, 740–755 (Springer International Publishing, Cham, 2014).

37. DeVries, T. & Taylor, G. W. Improved regularization of convolutional neural networks with Cutout. Preprint at https://arxiv.org/abs/1708.04552 (2017).

38. Zhou, Z. *et al.* Spikformer: When spiking neural network meets transformer. Preprint at https://arxiv.org/abs/2209.15425 (2022).

39. Su, Q. *et al.* Deep directly-trained spiking neural networks for object detection. Preprint at https://arxiv.org/abs/2307.11411 (2023).

40. Kang, J. *et al.* Achieving low-power analog resistive switching in filamentary memristive devices for energy-efficient analog in-memory computing. *Small* **21**, e05708 (2025).

41. Le Gallo, M. *et al.* Using the IBM analog in-memory hardware acceleration kit for neural network training and inference. *APL Mach. Learn.* **1**, 041102 (2023).

42. Ueyoshi, K. *et al.* DIANA: An end-to-end energy-efficient digital and ANAlog hybrid neural network SoC. In *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, 1–3 (IEEE, San Francisco, CA, USA, 2022).

43. Rebuffi, S.-A., Kolesnikov, A., Sperl, G. & Lampert, C. H. iCaRL: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5533–5542 (IEEE, Honolulu, HI, 2017).

44. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. & Wayne, G. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, Vol. 32 (Curran Associates, Inc., 2019). URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

45. Indiveri, G., Chicca, E. & Douglas, R. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.* **17**, 211–221 (2006).

46. Joubert, A., Belhadj, B., Temam, O. & Heliot, R. Hardware spiking neurons design: Analog or digital? In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 1–5 (IEEE, Brisbane, Australia, 2012).

47. Aamir, S. A. *et al.* A mixed-signal structured AdEx neuron for accelerated neuromorphic cores. *IEEE Trans. Biomed. Circuits Syst.* **12**, 1027–1037 (2018).

48. Chen, X. *et al.* CMOS-based area-and-power-efficient neuron and synapse circuits for time-domain analog spiking neural networks. *Appl. Phys. Lett.* **122**, 074102 (2023).

49. Dutta, S., Kumar, V., Shukla, A., Mohapatra, N. R. & Ganguly, U. Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET. *Sci. Rep.* **7**, 8257 (2017).

50. Han, J.-K. *et al.* 3D stackable broadband photoresponsive InGaAs biristor neuron for a neuromorphic visual system with near 1 V operation. In *2021 IEEE International Electron Devices Meeting (IEDM)*, 1–4 (IEEE, San Francisco, CA, USA, 2021).

51. Han, J.-K. *et al.* Mimicry of excitatory and inhibitory artificial neuron with leaky integrate-and-fire function by a single MOSFET. *IEEE Electron Device Lett.* **41**, 208–211 (2020).

52. Pazos, S. *et al.* Synaptic and neural behaviours in a standard silicon transistor. *Nature* **640**, 69–76 (2025).

53. Han, J.-K., Yun, G.-J., Han, S.-J., Yu, J.-M. & Choi, Y.-K. One biristor-two transistor (1B2T) neuron with reduced output voltage and pulsewidth for energy-efficient neuromorphic hardware. *IEEE Trans. Electron Devices* **68**, 430–433 (2021).

54. Han, C. Y. *et al.* Configurable NbO_x memristors as artificial synapses or neurons achieved by regulating the forming compliance current for the spiking neural

network. *Adv. Electron. Mater.* **9**, 2300018 (2023).

55. Yuan, R. *et al.* A calibratable sensory neuron based on epitaxial VO2 for spike-based neuromorphic multisensory system. *Nat. Commun.* **13**, 3973 (2022).

56. Lee, M. *et al.* Simple artificial neuron using an ovonic threshold switch featuring spike-frequency adaptation and chaotic activity. *Phys. Rev. Applied* **13**, 064056 (2020).

57. Horowitz, M. 1.1 Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14 (IEEE, San Francisco, CA, USA, 2014).

58. Moitra, A. *et al.* SpikeSim: An end-to-end compute-in-memory hardware evaluation tool for benchmarking spiking neural networks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **42**, 3815–3828 (2023).