ADVANCED BURSTPROP: AN ONLINE ENERGY-EFFICIENT LOCAL LEARNING PARADIGM FOR DEEP SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Backpropagation through time with surrogate gradients (SG-BPTT) is the state of the art for training deep spiking neural networks (SNNs), but it is offline, with an unrolled graph costing $O(T)$ activation memory, and its backward phase executes dense multiply-accumulate operations. Recent online local learning rules (E-prop, OSTL, OTTT, DECOLLE, ETLP) remove the unrolled graph, yet their weight updates remain dense floating-point multiply-accumulates at every timestep (or, for the event-triggered ETLP, at every triggered timestep), so the backward phase remains the energy sink. This raises a critical question for the field: **if training an SNN costs orders of magnitude more energy than its forward pass could ever save, in what sense is the SNN energy-efficient at all?** We answer this question with **Advanced Burstprop (ABP)**, an online local learning rule whose backward phase is as event-driven as the forward phase: signed ternary burst events $\{-1, 0, +1\}$, emitted at a $\sim 5\%$ rate, carry error transport and weight updates strictly in $O(1)$ memory, multiply-free and accumulate-only. On DVS-Gesture, under the 45 nm CMOS energy model, ABP matches the dense-backward baselines ($90.23 \pm 0.78\%$ vs. $92.45 \pm 0.60\%$ for the strongest baseline) at $\sim 31\times$ lower backward energy, and the deterministic variant pushes the reduction to $\sim 53\times$. Widening the burst alphabet beyond ternary to $\{-L, \ldots, +L\}$ multiplies the backward energy without accuracy gain, confirming ternary as the optimal code. In a controlled depth sweep, ABP's accuracy decays with depth, but widening the layers, rather than raising the burst rate, recovers it while maintaining the energy-efficiency advantage. On SHD, ABP attains the smallest gap to SG-BPTT among online methods on the temporal-convolutional LSTM architecture while being $\sim 25\times$ more backward-energy-efficient. We further discuss the feasibility of deploying ABP on existing neuromorphic hardware, including Intel Loihi 2 and SpiNNaker 2. To our knowledge, ABP is the first online learning rule to scale multiply-free, event-sparse burst learning to deep SNN architectures, and the first to quantify the resulting energy advantage, making SNN training, not merely SNN inference, energy-efficient.

1 INTRODUCTION

Deep spiking neural networks promise energy-efficient temporal computation: because activations are binary events, the forward phase replaces the multiply–accumulate (MAC) operations of analog networks with spike-triggered accumulations (AC), which cost roughly $5\times$ less energy per operation in 45 nm CMOS (Horowitz, 2014). This argument, however, covers only inference. Training an SNN is dominated by the backward phase, and the standard algorithm, backpropagation through time (BPTT) (Werbos, 1990) with surrogate gradients (Neftci et al., 2019; Shrestha & Orchard, 2018; Zenke & Ganguli, 2018), pays twice: it stores an unrolled computational graph whose memory grows linearly with the sequence length T, ruling out streaming and continual deployment for online learning; and it executes dense multi-bit MAC operations in the backward phase, erasing the significant energy advantage the SNN was built to provide.

A family of online local rules addresses the first cost. Eligibility-propagation (E-prop) (Bellec et al., 2020), online spatio-temporal learning (OSTL) (Bohnstingl et al., 2022), online training through

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

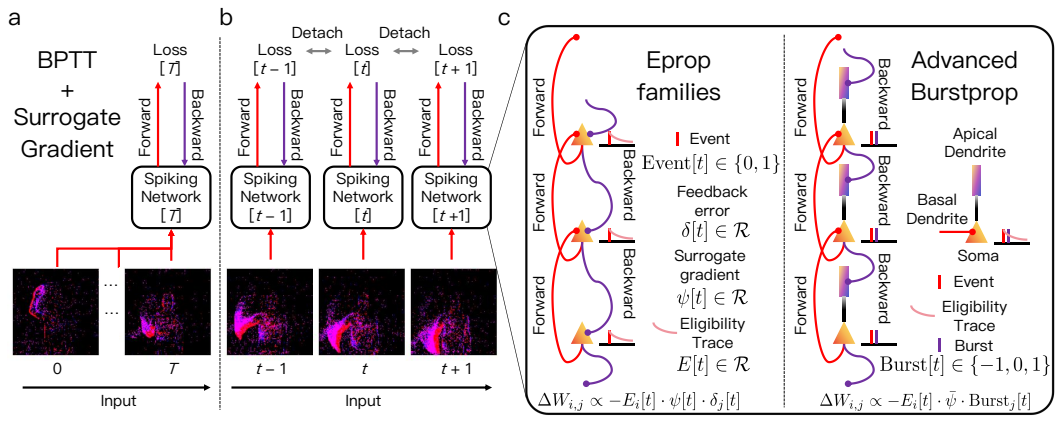


Figure 1: **Advanced Burstprop (ABP): an energy-efficient local online learning paradigm. (a)** SG-BPTT unrolls the network over the whole sequence: the loss is evaluated only at $t = T$ and errors propagate backward through all timesteps, costing $\mathcal{O}(T)$ memory and locking the model until the sequence ends. **(b)** The E-prop family executes forward and backward phases at every timestep, with eligibility traces approximating the BPTT recursion ($\mathcal{O}(1)$ memory); its error pathway, however, remains dense and analog, transported at full precision every timestep. **(c)** ABP keeps this online $\mathcal{O}(1)$ skeleton but makes the backward pass itself event-driven. Forward sensory input is integrated to emit singlet spikes (events) and low-pass filtered into an eligibility trace, while the error projected onto the apical dendrite is encoded by the soma as a digital *signed burst* $b \in \{-1, 0, +1\}$. The burst forms the three-factor weight update locally with the eligibility trace and a layer-specific scalar $\bar{\psi}$, and is relayed to the upstream layer to transport the error. Both local learning and error propagation are thus event-sparse and multiply-free, and hence energy-efficient.

time (OTTT) (Xiao et al., 2022), DECOLLE (Kaiser et al., 2020), and event-based three-factor local plasticity (ETLP) (Quintana et al., 2024) all detach the graph between timesteps, keep $O(1)$ memory in T, and factorize the update into a local eligibility trace multiplied by a broadcast learning signal, in the spirit of three-factor rules and feedback alignment (Lillicrap et al., 2016; Nøkland, 2016; Zenke & Ganguli, 2018; Murray, 2019). Yet they leave the second cost untouched. E-prop, OSTL, OTTT, and DECOLLE compute their weight updates as dense floating-point outer products at every time step and at every weight; ETLP is event-triggered rather than time step-dense, applying its updates only on a subset of time steps, but each triggered update remains a dense floating-point multiply-accumulate. Measured against the 45 nm energy table, the backward pass of these rules is one to two orders of magnitude more expensive than the SNN's own forward pass. If neuromorphic deployment is the motivation for SNNs, the backward pass, not the forward, is the remaining bottleneck.

We take the position that the backward pass of an SNN learning rule should obey the same physics as its forward pass: event-driven and multiply-free. Advanced Burstprop (ABP) realizes this by replacing the dense error vector with **sparse ternary burst events** $burst[t] \in \{-1, 0, +1\}$ that propagate back by the weight transpose. A two-compartment neuron model separates feedforward integration (basal dendrite and soma) from error integration (an apical dendrite), and a synapse updates only when a burst event coincides with its presynaptic eligibility trace at a measured $\nu \approx 5\%$ burst rate (the percentage of time steps that emits a nonzero burst), which means the training process no longer requires an $O(T)$ unrolled graph, a dense analog error transmission or multiplication.

2 Related Work

Surrogate-gradient BPTT. Training deep SNNs by unrolling the network in time and differentiating through a smooth surrogate of the spike nonlinearity (SG-BPTT) remains the state of the art (Neftci et al., 2019; Shrestha & Orchard, 2018; Zenke & Ganguli, 2018). Its costs are structural: unrolling requires $O(T)$ activation storage, the computation is offline since the model is locked until the sequence ends, and the backward pass executes dense multiply-accumulate operations.

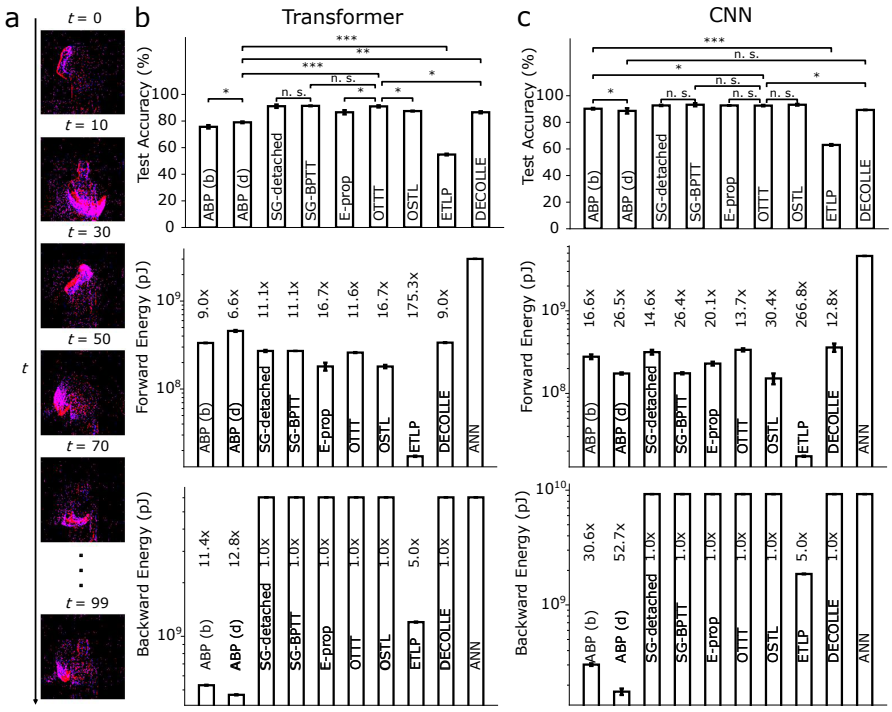


Figure 2: **Advanced Burstprop on the DVS-Gesture dataset. (a)** Illustration of the DVS-Gesture dataset: event streams of the 11 hand and arm gestures, recorded by a dynamic vision sensor and rendered as event frames. **(b)** Test accuracy, forward energy, and backward energy of ABP (Bernoulli group: ABP (b), deterministic group: ABP (d)) and all baseline methods on the Transformer architecture with learning rate $\mathrm{lr} = 2 \times 10^{-3}$ (over 3 seeds). **(c)** The same quantities on the CNN architecture. Accuracies are test accuracies and energies are per-sample means under the 45 nm accounting model (over 3 seeds). Two-sided Welch's t-test ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

Online and local three-factor rules. The three-factor family, in which a presynaptic eligibility trace is multiplied by a postsynaptic factor and a broadcast learning signal, includes E-prop (Bellec et al., 2020) and its recurrent LSNN substrate (Bellec et al., 2018), OSTL (Bohnstingl et al., 2022), OTTT (Xiao et al., 2022), DECOLLE (Kaiser et al., 2020), ETLP (Quintana et al., 2024), RFLO (Murray, 2019), and SuperSpike (Zenke & Ganguli, 2018). All achieve $O(1)$ memory in T. Where they differ is the temporal density of the backward: E-prop, OSTL, OTTT, DECOLLE, RFLO, and SuperSpike update densely at every timestep, whereas ETLP is event-triggered, updating on a subset of timesteps. Every one of them, however, computes each update as a floating-point multiply-accumulate; to our knowledge, none makes the backward pass itself event-sparse and multiply-free. ABP belongs to this family algebraically, since its expected update is the same three-factor product (Appendix A.1.4), but differs physically: the learning signal is a stream of ternary events, so the backward pass executes only accumulations.

Burst coding and two-compartment neurons in biology. Two-compartment neuron models (Guerguiev et al., 2017) approximate backpropagation locally by multiplexing feedforward sensory signals with top-down error signals across segregated somatic and dendritic compartments. Burst-dependent plasticity (Burstprop) (Payeur et al., 2021) realizes this principle in SNNs, encoding errors as discrete burst events and confining learning to sparse integer accumulate (AC) operations, but its spiking demonstration was confined to XOR. BurstCCN (Greedy et al., 2022) systematized the single-phase variant in cortico-cortical network models, and the recent BurstMLP study (Stuck et al., 2025) reached MNIST-family benchmarks with fully-connected networks. None of these works, however, scales burst learning to modern deep-learning architectures, and none quantifies the energy efficiency of the burst-coded error channel: the architectural scale up and the energy evidence are both left unaddressed. **Advanced Burstprop** bridges this scaling gap. By encapsulating

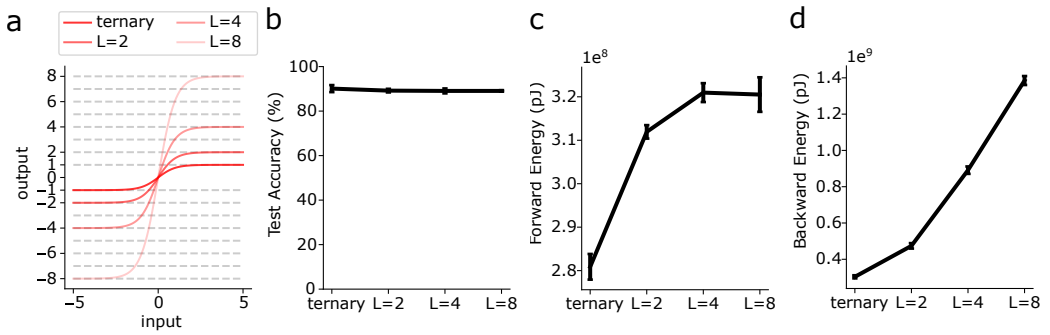


Figure 3: **The burst-payload alphabet of Advanced Burstprop. (a)** Schematic of alphabet coding in the deterministic group: the processed dendritic-error drive d is scaled by the alphabet half-width L and rounded to the nearest integer magnitude, $\hat{m} = \min\big(\lfloor L \cdot d \rceil, L\big)$ — e.g., $6.4 \to 6$, $6.7 \to 7$ — and emitted with the sign of the dendritic error, giving the signed burst $b = \hat{m} \cdot \mathrm{sign}(V_{\mathrm{d}}) \in \{-L, \ldots, +L\}$ within the bAP permission window. $L = 1$ recovers the default ternary code $\{-1, 0, +1\}$; larger L yields a finer-grained, wider-payload code. **(b)** Test accuracy for each alphabet on DVS-Gesture, CNN architecture (mean $\pm$ std over 3 seeds). **(c)** Forward energy per sample for each alphabet. **(d)** Backward energy per sample for each alphabet.

the two-compartment dynamics in a modular `BurstLIF` primitive and by designing a generalized basal-dendrite abstraction, ABP scales AC-only burst learning to modern SNN building blocks. In our PyTorch-based implementation, for each timestep, the forward pass communicates sensory information via discrete spike events while the backward pass propagates errors via discrete ternary bursts, and on this substrate we construct a suite of modern architectural primitives: Convolutional, Transformer, LSTM, and Mamba blocks with specific designed two control loops (see Appendix A.4) to make the deep complex SNN training feasible.

Neuromorphic hardware and on-chip learning. Digital manycores with on-chip plasticity (Loihi (Davies et al., 2018), Loihi 2 (Davies et al., 2021; Intel Corporation, 2021), SpiNNaker/2 (Furber et al., 2014; Höppner et al., 2021)), research ASICs with silicon-proven online three-factor learning (ODIN (Frenkel et al., 2019), ReckOn (Frenkel & Indiveri, 2022)), and an E-prop deployment on SpiNNaker 2 (Rostami et al., 2022) demonstrate that $O(1)$-memory online learning is silicon-feasible. Section 5 discusses the feasibility of deploying ABP on these platforms.

3 The Advanced Burstprop Paradigm

As shown in Fig. 1a, backpropagation through time (BPTT) (Werbos, 1990) with surrogate gradients (Neftci et al., 2019; Shrestha & Orchard, 2018; Zenke & Ganguli, 2018) stores an unrolled computational graph whose memory grows linearly with the sequence length T, locking the model until the sequence ends and making online learning impossible, and its backward phase executes dense multi-bit MAC operations, erasing the energy advantage that SNNs were intended to provide. As shown in Fig. 1b, the E-prop family executes the forward and backward phases at every timestep with a detached per-timestep loss, and eligibility traces approximate the temporal-recursion term of BPTT, yielding online learning with $\mathcal{O}(1)$ memory (see Appendix A.1.3). The error pathway, however, remains dense and analog, with a full-precision signal transported at every timestep. To make the backward pass as energy-efficient as the forward pass and achieve the efficiency SNNs were built to provide, we develop Advanced Burstprop (ABP). For the derivation from SG-BPTT to ABP by using E-prop as the ladder, please check Appendix A.1.

ABP relies on a multi-compartment neuron model comprising a basal dendrite, a soma, and an apical dendrite (Fig. 1c). The basal dendritic compartment receives and processes feedforward sensory information, accepting either discrete spikes (digital input) or continuous analog currents (analog input). We define the basal dendrite as a generalized functional class, encapsulating complex spatial operations such as linear projections, convolutions, and pooling. Following dendritic processing,

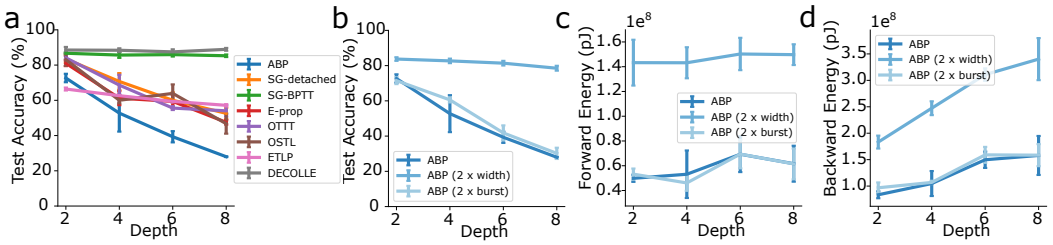


Figure 4: **Depth scaling of Advanced Burstprop (DVS-Gesture, Bernoulli group).** A single spiking convolutional block is held fixed as the input stem and followed by a spiking MLP of depth $\in \{2, 4, 6, 8\}$, so the depth sweep is attributable to the MLP depth alone. Accuracies are test accuracies; energies are per-sample means under the 45 nm model. **(a)** Test accuracy of ABP and all baseline methods versus depth. Rules that transport a global error signal degrade with depth, and the degradation grows with the sparsity of the transport — as predicted by the hop law — while layer-local (DECOLLE) and dense-transport (SG-BPTT) rules are depth-robust. **(b)** Test accuracy of ABP at the default setting (burst rate ν, layer width w) and of the two compensation methods that double the bursts-per-layer budget: doubled burst rate ($2 \times \nu$) and doubled layer width ($2 \times w$). **(c)** Forward energy of the same arms; doubling the width raises the forward cost. **(d)** Backward energy of the same arms; doubling the width raises the backward cost.

the somatic compartment implements a Leaky Integrate-and-Fire (LIF) mechanism, integrating the continuous dendritic currents to emit discrete spikes (events). Conversely, the apical dendritic compartment integrates top-down feedback error signals. Upon sufficient error accumulation, the apical dendrite triggers "bursts" that instruct local synaptic weight updates and propagate the sparse error signal backward through the network layers.

3.1 Basal Dendrite Dynamics (Synaptic Integration and Traces)

In biological pyramidal neurons, feedforward sensory information is received by the basal dendrites. Mathematically, the basal dendritic compartment computes the spatial synaptic projections (e.g., linear or convolutional operations) while simultaneously managing an $\mathcal{O}(1)$ presynaptic eligibility trace. Given an input spike train $\mathbf{s}_{pre}[t]$, the basal dendrite computes the instantaneous synaptic current $I_{syn}[t]$ via the weights $\mathbf{w}$:

$$I_{syn}[t] = \mathbf{w}^T \mathbf{s}_{pre}[t]. \tag{1}$$

To decouple the weight update from the temporal unrolling of BPTT, the basal dendrite maintains a continuous presynaptic eligibility trace $\mathbf{E}[t]$, a low-pass filter of the spike history with decay constant τ_{pre}:

$$\mathbf{E}[t] = \mathbf{E}[t-1] \cdot \left(1 - \frac{dt}{\tau_{pre}}\right) + \mathbf{s}_{pre}[t]. \tag{2}$$

During the backward pass, the instantaneous input $\mathbf{s}_{pre}[t]$ is systematically substituted with this trace $\mathbf{E}[t]$ using autograd hooks. The decay constant is not shared across the layer: within each layer, neuron i is assigned its own time constant $\tau_{pre,i}$, drawn log-uniformly from $[\tau_{\min}, \tau_{\max}] = [2\,\text{ms}, 1\,\text{s}]$, providing richer temporal credit coverage than any single shared constant.

3.2 Somatic Dynamics (Forward Pass)

The somatic compartment is governed by discrete-time LIF dynamics. At timestep t, the membrane potential $V_s[t]$ integrates the incoming sparse synaptic input $I_{syn}[t]$ with decay constant τ_s:

$$V_s[t] = V_s[t-1] \cdot \left(1 - \frac{dt}{\tau_s}\right) + I_{syn}[t] \tag{3}$$

A sparse binary spike $s[t] \in \{0, 1\}$ is emitted when the potential crosses the threshold V_{th}:

$$s[t] = \Theta(V_s[t] - V_{th}), \tag{4}$$

where $\Theta(\cdot)$ is the Heaviside step function. After firing, the potential is reset.

Table 1: DVS-Gesture comparison.

	BurstCNN			BurstFormer		
Method	Acc (%)	E_{fwd}	E_{bwd}	Acc (%)	E_{fwd}	E_{bwd}
Bernoulli group (all methods, lr $= 10^{-4}$)						
ABP (b)	90.23 ± 0.78	278.3	**302.1**	75.65 ± 1.58	334.0	**531.3**
SG-detached	91.28 ± 0.60	236.4	9,245.6	87.89 ± 1.35	286.3	6,030.6
SG-BPTT	89.97 ± 0.60	206.1	9,245.6	82.94 ± 1.80	266.8	6,030.6
E-prop	91.41 ± 0.68	209.8	9,245.6	83.46 ± 1.37	265.7	6,030.6
OSTL	$\mathbf{92.45 \pm 0.60}$	232.7	9,245.7	85.68 ± 0.23	236.6	6,031.2
OTTT	91.80 ± 0.39	234.7	9,245.6	$\mathbf{88.41 \pm 0.98}$	284.3	6,030.6
ETLP	72.14 ± 0.90	108.5	1,853.7	59.77 ± 0.39	180.0	1,199.7
DECOLLE	89.71 ± 2.00	291.4	9,245.6	83.59 ± 1.41	335.5	6,030.6
Deterministic group (all methods, lr $= 2 \times 10^{-3}$)						
ABP (d)	87.11 ± 1.41	174.3	**175.5**	79.04 ± 0.81	457.5	**469.3**
SG-detached	92.71 ± 0.60	316.2	9,245.6	91.15 ± 1.58	270.9	6,030.6
SG-BPTT	$\mathbf{93.23 \pm 1.48}$	175.3	9,245.6	$\mathbf{91.41 \pm 0.39}$	270.9	6,030.6
E-prop	92.71 ± 0.23	229.9	9,245.6	86.59 ± 1.97	180.6	6,030.6
OSTL	$\mathbf{93.23 \pm 0.81}$	151.8	9,245.7	87.50 ± 0.39	180.3	6,031.2
OTTT	92.58 ± 0.68	337.2	9,245.6	91.02 ± 1.03	259.7	6,030.6
ETLP	63.02 ± 0.81	17.3	1,861.4	54.82 ± 0.81	17.2	1,201.7
DECOLLE	89.32 ± 0.23	360.3	9,245.6	86.59 ± 0.98	336.9	6,030.6

3.3 Apical Dendrite Dynamics (Backward Pass)

Unlike BPTT (Fig. 1a), which unrolls gradients globally, Advanced Burstprop (Fig. 1c) integrates a top-down feedback signal $B_{fb}[t]$ into an apical dendritic potential $V_d[t]$:

$$V_d[t] = V_d[t-1] \cdot \left(1 - \frac{dt}{\tau_d}\right) + B_{fb}[t]. \tag{5}$$

The apical dendrite converts this continuous error into an event drive $P_{raw}[t]$, modulated by a sensitivity scalar γ:

$$P_{raw}[t] = |\tanh(V_d[t] \cdot \gamma)|. \tag{6}$$

Crucially, learning only occurs if the soma was recently active. This is modeled by a back-propagating action potential (bAP) trace ($\text{bAP}[t] = (1 - dt/\tau_{\text{bap}})^{k(t)}$, where $k(t)$ is the number of time steps since the last somatic spike), which creates a temporally gated burst drive:

$$P_{burst}[t] = P_{raw}[t] \cdot \text{bAP}[t]. \tag{7}$$

3.4 Stochastic Burst Generation (Bernoulli Group)

Unlike the E-prop family, which executes its updates densely in floating point at every timestep (Fig. 1b), ABP encodes the error as ternary burst events (Fig. 1c). In the stochastic group, the neuron samples a random variable $r \sim \mathcal{U}(0,1)$ at each timestep, and the burst event inherits the exact sign of the dendritic error, creating a sparse ternary signal:

$$\text{Burst}[t] = \text{sign}(V_d[t]) \cdot \Theta(P_{burst}[t] - r) \quad \in \{-1, 0, 1\}. \tag{8}$$

This stochastic process is an unbiased estimator of the gated drive,

$$\mathbb{E}[\text{Burst}[t] \mid V_d[t]] = \text{sign}(V_d[t]) \cdot P_{burst}[t] \tag{9}$$

3.5 Deterministic Burst Generation

The deterministic group replaces the sampled comparison with a per-layer threshold θ_l: a burst is emitted whenever the gated drive crosses it,

$$\text{Burst}[t] = \text{sign}(V_d[t]) \cdot \Theta(P_{burst}[t] - \theta_l) \quad \in \{-1, 0, 1\}, \tag{10}$$

where $P_{\text{burst}}[t]$ is the bAP-gated drive of Equation (7), so the same somatic gating applies as in the stochastic group. The threshold θ_l is adapted per layer by a rate controller (Loop A, Appendix A.4) so that the realized burst rate tracks the budget ν^*; this group therefore contains no sampling noise.

Table 2: SHD comparison.

	LSTM			Mamba			TCNN-LSTM			TCNN-Mamba		
Method	Acc (%)	E_{fwd}	E_{bwd}	Acc (%)	E_{fwd}	E_{bwd}	Acc (%)	E_{fwd}	E_{bwd}	Acc (%)	E_{fwd}	E_{bwd}
ABP (b)	**17.20 ± 1.97**	2.9	**8.2**	9.92 ± 0.63	1.1	**2.6**	26.92 ± 0.65	789.4	**220.1**	19.04 ± 2.39	787.0	**198.5**
ABP (d)	16.76 ± 1.51	2.5	**5.1**	11.12 ± 0.31	1.0	**1.8**	24.24 ± 0.52	788.0	**186.2**	22.71 ± 2.29	785.5	**162.5**
SG-detached	14.03 ± 1.32	6.1	326.0	13.10 ± 1.92	1.8	122.5	22.36 ± 3.19	799.3	5,168.8	23.84 ± 3.59	794.5	4,965.4
SG-BPTT	16.67 ± 3.54	5.7	326.0	7.36 ± 1.45	1.6	122.5	**34.90 ± 2.10**	802.6	5,168.8	**29.62 ± 6.41**	800.1	4,965.4
E-prop	16.78 ± 2.54	7.8	326.0	14.61 ± 2.55	1.9	122.5	25.55 ± 1.10	796.1	5,168.8	20.69 ± 2.27	797.7	4,965.4
OSTL	16.31 ± 1.35	8.6	326.5	12.93 ± 1.62	2.2	122.7	26.09 ± 2.84	795.6	5,171.3	20.05 ± 1.75	799.5	4,967.5
OTTT	13.90 ± 0.87	6.2	326.0	11.87 ± 1.24	1.9	122.5	21.50 ± 3.32	800.3	5,168.8	25.13 ± 2.42	795.1	4,965.4
ETLP	13.35 ± 1.25	2.1	64.9	**16.48 ± 1.03**	0.9	24.5	18.47 ± 0.57	785.0	1,033.4	18.38 ± 0.89	783.5	987.4
DECOLLE	11.33 ± 1.73	4.1	326.0	11.53 ± 1.79	1.7	122.5	25.57 ± 2.21	792.2	5,168.8	23.39 ± 1.89	793.9	4,965.4

3.6 Burst alphabet

For the deterministic group, the ternary code generalizes to an alphabet $\{-L, \ldots, +L\}$: the drive magnitude is scaled by the alphabet half-width L, rounded to the nearest integer, and clipped to the alphabet bound,

$$\hat{m}[t] = \min\big(\text{round}(L \cdot P_{\text{raw}}[t]),\, L\big), \quad \text{Burst}[t] = \hat{m}[t] \cdot \text{sign}(V_d[t]) \cdot \mathbb{1}[\text{bAP}[t] > \epsilon], \tag{11}$$

where $\text{Burst}[t] \in \{-L, \ldots, +L\}$. Event sparsity emerges naturally because small drives round to zero. $L = 1$ recovers the ternary code; wider alphabets carry finer error magnitude at the cost of additional shift-add operations per event (see Appendix A.3).

3.7 Local Learning and Error Transport

Because the error is physicalized as a sparse signed burst in both groups, the local weight update $\Delta\mathbf{W}$ is computed entirely online by interacting the somatic error burst with the basal dendrite's presynaptic eligibility trace $\mathbf{E}[t]$ and a layer-specific constant $\bar{\psi}$ that stands in for the derivative of the spiking nonlinearity:

$$\Delta\mathbf{W} \propto -\eta\bar{\psi} \cdot \mathbf{Burst}[t] \cdot \mathbf{E}[t]^T, \tag{12}$$

and the feedback error relayed to the previous layer is

$$\mathbf{B}_{fb}[t] = \mathbf{W}^T \cdot \bar{\psi} \cdot \mathbf{Burst}[t]. \tag{13}$$

The layer-specific constant $\bar{\psi}$ can be absorbed into the hardware (see Appendix A.1.5), which makes both operations consist strictly of accumulate (AC) operations, so that, for the first time, the backward phase is event-driven and multiply-free. The difference between E-prop family and ABP can be found in Appendix A.1.4.

3.8 The Advanced Burstprop Meets Deep SNNs

The Advanced Burstprop framework seamlessly maps onto state-of-the-art Deep Learning architectures without breaking the strict local $\mathcal{O}(1)$ learning constraint. We detail the mathematical mapping of four primary architectures: CNN, LSTM, Transformer, and Mamba in Supp Fig. 5b and (Appendix A.2). For PyTorch implementation, the $tanh$ linearity control method for the apical dendrite can be found in Appendix A.4.

4 Experiments and Results

4.1 On DVS-Gesture, ABP matches the E-prop family at a small fraction of the backward energy

We evaluate ABP against all comparison methods on the DVS-Gesture dataset, whose recordings carry rich spatio-temporal structure encoded as sparse events (Fig. 2a). ABP is tested in two burst-generation regimes: stochastic Bernoulli (ABP (b)) and deterministic threshold (ABP (d)). The comparison set covers four families: SG-BPTT and SG-detached (which detaches the computational graph between consecutive steps) as control groups; E-prop, OTTT, and OSTL from the E-prop

family; ETLP as an event-triggered baseline, temporally sparse (a stochastic trigger $\theta_{\text{trig}} \approx 0.2$) but with each triggered update still a dense floating-point MAC; and DECOLLE as a local-error baseline. All methods are evaluated on two spiking architectures, a spiking Transformer and a spiking CNN (Supp. Fig. 5b, c; Appendix A.2). Energies are estimated under the 45 nm CMOS model of Horowitz (2014). Please see Appendix Table 3, for all hyperparameter settings.

Accuracy On the spiking Transformer (Fig. 2b, top; Table 1), SG-BPTT achieves the best accuracy (91.41 ± 0.39), with OTTT statistically indistinguishable from it ($p > 0.05$). The remaining E-prop family methods (E-prop, OSTL) are slightly but significantly below OTTT, yet remain close to SG-BPTT. ABP (d) (79.04 ± 0.81) trails the E-prop family and DECOLLE significantly, but stays far above ETLP, the sparse-backward baseline; ABP (b) performs below ABP (d) on this architecture. On the spiking CNN (Fig. 2c, top; Table 1), in contrast, ABP (b) reaches 90.23 ± 0.78, comparable to the E-prop family, whose best member (OSTL) attains 93.23 ± 0.81. The gap between ABP (d) (87.11 ± 1.41) and OSTL narrows to $\sim 5\%$, and ABP (d) is statistically indistinguishable from DECOLLE. On this architecture, both ABP (b) and ABP (d) significantly outperform ETLP ($^{***}p < 0.001$), the method with a sparse backward pass.

Forward energy All SNN methods are far more energy-efficient than the equivalent ANN models (Fig. 2b, c, middle; Table 1), with ETLP attaining the lowest forward energy (for forward energy counting details, please refer Appendix A.5).

Backward energy Both ternary-coded ABP variants reduce the backward energy by orders of magnitude relative to the dense analog error coding of SG-BPTT and the E-prop family, on both architectures (Fig. 2b, c, bottom; Table 1). On the spiking CNN, ABP (b) reduces the backward energy by $\sim 31\times$ while remaining comparable to OSTL in accuracy, and ABP (d) attains the largest reduction, $\sim 53\times$. ETLP's sparse backward pass also improves over the dense baselines, yet ABP improves on it by a further factor of $6\times$ to $11\times$, showing that ternary error coding is more energy-efficient than a sparse analog backward pass. Since forward energies differ only modestly across methods while backward energies span orders of magnitude, ABP achieves the lowest total energy per sample on both architectures. In short, on a suitable architecture, ABP matches SOTA online spiking learning methods in accuracy while delivering extreme backward energy efficiency, making it the most energy-efficient method in the comparison (for backward energy counting details, please check Appendix A.5).

4.2 Ternary burst coding is the optimal coding scheme

We next ask whether the ternary code itself is the optimal operating point. Scaling the burst alphabet keeps the backward pass multiply-free, since every payload remains a shift-add accumulate, but carries finer error fidelity (Fig. 3a), and it is not obvious a priori whether the added fidelity helps. For ABP (d), we sweep the alphabet half-width L from 1 (ternary $\{-1, 0, +1\}$) to 2, 4, and 8 on the spiking CNN with the DVS-Gesture dataset (Fig. 3b, c, d). Increasing the burst fidelity yields no accuracy gain (Fig. 3b), while the forward energy rises (Fig. 3c) and the backward energy multiplies with the alphabet size (Fig. 3d), because wider alphabets both emit events at smaller drives and cost more shift-add accumulates per event. Scaling the burst fidelity therefore buys no performance but costs significant energy, confirming the ternary code as the optimal coding scheme of the family.

4.3 Widening layers, rather than raising the burst rate, recovers the depth-induced accuracy drop

Error signals attenuate with depth in deep SNNs. To investigate this depth-scaling effect, we fix the first spiking convolutional block and progressively increase the number of spiking MLP layers, sweeping the depth over $\{2, 4, 6, 8\}$ for ABP and all comparison groups. As shown in Fig. 4a, SG-BPTT and DECOLLE maintain their accuracy across all depths: SG-BPTT because its dense backward transports the full gradient, and DECOLLE because its layer-local errors never propagate through the network at all. All methods that transport a global error, the E-prop family, ETLP, and ABP, degrade with depth, and the degradation is strongest for ABP, whose transport is the sparsest. We then test two compensation arms, both designed to double the nominal error-channel budget: doubling the layer width, and doubling the burst rate (for details of the burst rate control, please check Appendix A.4). Only width doubling restores accuracy across depths, while doubling the burst rate does not (Fig. 4b), indicating that the starved resource is the population of error-

carrying neurons rather than the per-neuron burst rate. Width doubling increases both the forward and backward energy (Fig. 4c, d), yet even the compensated ABP remains far more energy-efficient than the dense-backward comparison methods.

4.4 ABP REMAINS COMPETITIVE ON THE TEMPORALLY RICH DATASET

Finally, we ask whether ABP still works on a dataset whose information is carried predominantly in the temporal domain. We test ABP (b) and ABP (d) against the same comparison groups on the Spiking Heidelberg Digits (SHD) dataset, using two recurrent architectures, LSTM and Mamba (Supp. Fig. 5d, e; Appendix A.2). Please see Appendix Table 3 for all hyperparameter settings.

We first impose a strict online-learning setting in which only the events of the current timestep are fed to the models. Under this setting (Table 2), all methods perform poorly on both architectures, including SG-BPTT. We then relax the restriction by allowing a single online-legal spiking temporal-convolution layer in front of the LSTM and Mamba backbones, which integrates recent temporal history through a fixed causal FIR window. Under this relaxed setting, SG-BPTT achieves the best performance (34.90 ± 2.10 on TCNN-BurstLSTM, 29.62 ± 6.41 on TCNN-BurstMamba), and ABP (b) attains the smallest gap to SG-BPTT on TCNN-BurstLSTM among all online methods; in the remaining conditions, both ABP variants are comparable to the other online learning methods. ABP also preserves its backward-energy advantage in every condition (Table 2), for example $\sim 25\times$ below the dense-backward baselines on the temporal-convolution architectures.

We stress that these accuracies sit below the benchmark values reported by task-specific prior work. That is by design: published methods pair with their own optimally tuned architectures and neuron models, whereas our goal is not a new SOTA but a controlled comparison, in which every method shares the same simple architecture and the same LIF neuron model, so that differences reflect the learning rule alone. The comparison between the two settings also carries a methodological message: introducing an online-legal temporal kernel recovers a substantial part of the performance on this temporally rich task. For such tasks, faithfully approximating BPTT's credit signal is only the first step; an architecture that captures temporal structure without breaking the online-learning requirement is equally critical.

5 DISCUSSION

We developed Advanced Burstprop to make SNN training as event-driven as inference: ABP keeps the e-prop family's online $\mathcal{O}(1)$ skeleton but encodes the backward error as ternary bursts, processed and triggered by the apical dendrite. On DVS-Gesture, ABP matches the accuracy of existing online spiking learning methods at an extreme backward energy advantage. On SHD, we learned that on temporally rich workloads, temporal architecture matters as much as BPTT fidelity in the learning rule. ABP's operating regime is bounded: accuracy attenuates with depth beyond the feasible band, currently remedied by layer widening.

Deployment on neuromorphic hardware follows directly from this design. Online learning is silicon-proven: e-prop has been trained from scratch in real time on SpiNNaker 2 (Rostami et al., 2022) and Loihi (Zhao et al., 2025), and ABP's identical online $\mathcal{O}(1)$ skeleton inherits the memory and control flow these platforms support, while its backward replaces dense floating-point updates with ternary burst events and accumulations, strictly simpler operations natively supported on both chips. The one nontrivial hardware requirement is the exact transpose $\mathbf{W}^\top$ for error transport: on digital many-cores it needs a duplicated reverse-routing table, but one that resynchronizes only at the $\sim 5\%$ burst rate with a ternary payload, so the overhead scales with an already-sparse channel; and symmetric transport is a preference, not a prerequisite, since burst-coded learning retains performance under fixed random feedback (Payeur et al., 2021; Stuck et al., 2025), making FA/DFA variants of ABP drop-in replacements, with verification at depth ongoing. The silicon feasibility of the e-prop family thus carries over to ABP, whose backward promises a further, substantial reduction of on-chip learning energy.

These lessons and this deployment path point to our next step: harder tasks such as DVS-Lip will require architectures co-designed with ABP.

AI USE STATEMENT

In this work, we used generative AI tools to extend our experiment codes based on our core Advanced Burstprop codes. We have not used generative AI tools for core Advanced Burstprop codes. Additionally, we used generative AI tools for rephrasing our manuscript. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the first author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

A patent application covering the ABP-based spiking Transformer has been filed. The authors declare no other competing interests.

REPRODUCIBILITY STATEMENT

All details needed to reproduce experimental results are described in the main context and appendix of manuscript. All codes to reproduce experimental results are in supplementary files.

REFERENCES

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 787–797, 2018.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11:3625, 2020.

Thomas Bohnstingl, Stanisław Woźniak, Angeliki Pantazi, and Evangelos Eleftheriou. Online spatio-temporal learning in deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8894–8908, 2022.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

Charlotte Frenkel and Giacomo Indiveri. Reckon: A 28nm sub-mm^2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales. In *IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pp. 468–469, 2022.

Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. A 0.086-mm^2 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. *IEEE Transactions on Biomedical Circuits and Systems*, 13(1):145–156, 2019.

Steve B. Furber, Francesco Galluppi, Steve Temple, and Luis A. Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

Will Greedy, Heng Wei Zhu, Joseph Pemberton, Jack Mellor, and Rui Ponte Costa. Single-phase deep learning in cortico-cortical networks. *Advances in Neural Information Processing Systems*, 35:24213–24225, 2022.

Jordan Guerguiev, Timothy P. Lillicrap, and Blake A. Richards. Towards deep learning with segregated dendrites. *Nature Communications*, 8:1676, 2017.

Sebastian Höppner, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Love Cederstroem, Matthias Eberlein, Christian Mayr, et al. The spinnaker 2 processing element architecture for hybrid digital neuromorphic computing. *arXiv preprint arXiv:2103.08392*, 2021.

Mark Horowitz. Computing's energy problem (and what we can do about it). In *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 10–14, 2014.

Intel Corporation. Loihi 2. Technology brief, 2021.

Jacques Kaiser, Hesham Mostafa, and Emre Neftci. Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14:424, 2020.

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7: 13276, 2016.

James M. Murray. Local online learning in recurrent networks with random feedback. *eLife*, 8: e43299, 2019.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):61–73, 2019.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pp. 1037–1045, 2016.

Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A. Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature Neuroscience*, 24(7):1010–1019, 2021.

Fernando M. Quintana, Fernando Perez-Peña, Pedro L. Galindo, Emre O. Neftci, Elisabetta Chicca, and Lyes Khacef. Etlp: event-based three-factor local plasticity for online learning with neuromorphic hardware. *Neuromorphic Computing and Engineering*, 4(3):034006, 2024.

Amirhossein Rostami, Bernhard Vogginger, Yexin Yan, and Christian G. Mayr. E-prop on spinnaker 2: Exploring online learning in spiking rnns on neuromorphic hardware. *Frontiers in Neuroscience*, 16:1018006, 2022.

Sumit B. Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Michael Stuck, Xingyun Wang, and Richard Naud. A burst-dependent algorithm for neuromorphic on-chip learning of spiking neural networks. *Neuromorphic Computing and Engineering*, 2025.

Paul J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Online training through time for spiking neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018.

Kejie Zhao, Wenjia Hua, Aiersi Tuerhong, Luziwei Leng, Yuxin Ma, and Qinghai Guo. Threshold modulation for online test-time adaptation of spiking neural networks. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2025.

A Appendix

A.1 From BPTT to Advanced Burstprop: the derivation chain

Here, we build the whole approximation ladder slowly, starting from the exact BPTT formula and deriving, how each family arises from it and what each step costs. The ladder is BPTT $\rightarrow$ E-prop family $\rightarrow$ ABP.

A.1.1 THE STANDARD BPTT FORMULA, DERIVED FROM THE CHAIN RULE

Dynamics and objective: A spiking layer l of neurons j evolves by $h^l(t) = F(h^l(t-1), x(t); W^l)$; for the LIF case $h = V$ (membrane potential), $V_j^l(t) = \alpha V_j^l(t-1) + \sum_i W_{ij}^l z_i^{l-1}(t)$, spikes $s_j^l(t) = \Theta(V_j^l(t) - V_{th})$, $\alpha = 1 - dt/\tau_m$, and the non-differentiable spike is handled by the surrogate $\partial s_j/\partial V_j \approx \psi_j$ (Neftci et al., 2019). And, the objective loss is $\mathcal{L} = \sum_t \mathcal{L}_t$.

The adjoint trick (where the backward recursion comes from): Define the *total* derivative of the loss w.r.t. the state, $\delta(t) \equiv d\mathcal{L}/dh(t)$, and note the Markov property of the dynamics: $h(t)$ influences the loss in exactly two ways — the current loss $\mathcal{L}_t$ directly, and all future losses $\mathcal{L}_{>t}$ only through $h(t+1)$. Splitting the loss and applying the chain rule exactly once,

$$\frac{d\mathcal{L}}{dh(t)} = \underbrace{\frac{\partial \mathcal{L}_t}{\partial h(t)}}_{\text{direct}} + \underbrace{\Big(\frac{\partial h(t+1)}{\partial h(t)}\Big)^\top}_{J(t+1)^\top} \cdot \frac{d\mathcal{L}_{>t}}{dh(t+1)}. \tag{14}$$

Since $\mathcal{L}_t$ does not depend on $h(t+1)$, the future factor is itself the total derivative at $t+1$, which yields the backward recursion with terminal condition $\delta(T) = \partial \mathcal{L}_T/\partial h(T)$:

$$\boxed{\delta^l(t) = \frac{\partial \mathcal{L}_t}{\partial h^l(t)} + J^l(t+1)^\top \, \delta^l(t+1)\,, \qquad J(s)_{jk} = \frac{\partial h_j(s)}{\partial h_k(s-1)}} \tag{15}$$

A.1.2 WHAT THE JACOBIAN PRODUCT ACTUALLY CONTAINS

For the LIF layer the Jacobian has four kinds of terms,

$$J_{jk}(t) = \underbrace{\alpha\, \delta_{jk}}_{\text{own membrane decay}} + \underbrace{(-\alpha V_{th}\, \psi_k\, \delta_{jk})}_{\text{reset path}} + \underbrace{W_{jk}^{rec}\, \psi_k(t-1)}_{\text{cross-neuron recurrent}} + \underbrace{(\beta_a \psi\text{-coupling})}_{\text{ALIF adaptation}}, \tag{16}$$

and $\prod J$ measures how much a synaptic contribution made at t' still rings in the state at t — including paths that leave the neuron and return through the recurrence. Only the first term is diagonal.

A.1.3 MOVE 1 (E-PROP): WITHIN-NEURON TRUNCATION TURNS $\prod J$ INTO THE TRACE

Keep only the diagonal: $J(t) \to \mathrm{diag}(\alpha)$. The propagator then becomes a scalar exponential per neuron,

$$\prod_{s=t'+1}^{t} J(s) \;\longrightarrow\; \alpha^{\,t-t'}, \tag{17}$$

and, per synapse,

$$\varepsilon_{ij}(t) = \alpha\, \varepsilon_{ij}(t-1) + z_i(t). \tag{18}$$

That scalar filter is the presynaptic eligibility trace: the matrix product (the network-wide echo) became a per-synapse exponential echo of the spike train, and the update becomes the online, local, $O(1)$ three-factor rule. Then, we can get e-prop family as

$$\frac{d\mathcal{L}}{dW_{ij}^l} = \sum_t \delta_j^l(t)\, \psi_j^l(t)\, \varepsilon_{ij}^l(t) \tag{19}$$

$$\varepsilon_{ij}(t) \equiv \frac{\partial V_j(t)}{\partial W_{ij}} = \alpha\, \varepsilon_{ij}(t-1) + z_i(t) = \sum_{\Delta t \geq 0} \alpha^{\Delta t} z_i(t - \Delta t). \tag{20}$$

The E-prop weight update is

$$w_{ij}^{EP} = \sum_t \delta_j(t)\, \psi_j(t)\, \varepsilon_{ij}(t). \tag{21}$$

A.1.4 FROM E-PROP TO ADVANCED BURSTPROP

Advanced Burstprop keeps the three-factor object and replaces its components as

component	E-prop	Advanced Burstprop
learning signal	dense analog $\delta_j(t)$	ternary burst $b_j(t) \in \{-1, 0, +1\}$, Bernoulli
surrogate factor	per-neuron $\psi_j(t)$	per-layer scalar $\bar{\psi}_l$ (direction-safe)
trace kernel	membrane $\alpha^{\Delta t}$	log-spread bank $\beta_i^{\Delta t}$, $[2\,\mathrm{ms}, 1\,\mathrm{s}]$
transport	broadcast B	symmetric $W^\top$ — unchanged from BPTT

The Advanced Burstprop estimator is

$$w_{ij}^{ABP} = \sum_t b_j(t)\, \bar{\psi}_l\, et_i(t), \qquad et_i(t) = \sum_{\Delta t \geq 0} \beta_i^{\Delta t} z_i(t - \Delta t). \tag{22}$$

A.1.5 THE LAYER-SPECIFIC SCALAR $\bar{\psi}_l$

The scalar is not a free constant; it is measured online from the layer's own membrane statistics. Each neuron evaluates the arctan-family surrogate derivative at its pre-reset membrane potential,

$$\psi_j[t] = \frac{1}{1 + \big(\alpha_\psi\,(V_{pre,j}[t] - V_{th,\mathrm{eff},j}[t])\,\big)^2}, \qquad \alpha_\psi = 2, \tag{23}$$

and the layer scalar is the leaky running mean of the population average, folded once per batch,

$$\bar{\psi}_l \;\leftarrow\; \rho\,\bar{\psi}_l + (1-\rho)\sum_j \psi_j[t]/N, \qquad \rho = 0.99, \tag{24}$$

initialized to the first measured batch mean. $\bar{\psi}_l$ therefore tracks the layer's actual sub-threshold membrane distribution rather than an assumed slope; measured values across our benchmarked architectures fall in a narrow band of $\approx$ 0.18–0.24 (convolutional and recurrent layers, on both datasets) and drift only slowly during training. Because $\bar{\psi}_l$ is constant within a batch and uniform across the layer's coordinates, it carries no direction information, and it is hardware-absorbable in the strict sense: it can be folded either into the per-layer learning rate, $\eta_{\mathrm{eff},l} = \eta \cdot \bar{\psi}_l$, or into the backward routing table as a fixed per-layer gain on the ternary events, leaving the event stream itself integer and multiply-free.

A.2 DEEP SPIKING NEURAL NETWORK ARCHITECTURES

The BurstLIF Module (Unifying Compartmental Dynamics) To operationalize the advanced Burstprop paradigm within modern deep learning architectures, we encapsulate somatic forward dynamics, and apical backward dynamics into a single, highly optimized structural primitive: the **BurstLIF** module.

BurstCNN (Spiking Convolutional Networks) Designed for localized saccadic vision tasks (e.g., Dynamic Vision Sensors, DVS), BurstCNN utilizes overlapping spiking convolutions to preserve rich 2D edge geometries over time, as shown in Fig. 5 (b). Let a spatiotemporal event stream be denoted as $X[t] \in \{0,1\}^{B\times C\times H\times W}$. A BurstConv block applies a 2D spatial convolution mapping via basal dendrites:$I_{conv}^{(l)}[t] = W^{(l)} * S^{(l-1)}[t] + b^{(l)}$. Where $*$ denotes the spatial convolution operator. This continuous current map is normalized and thresholds into a new spike map via the 'BurstLIF' soma:$S_{conv}^{(l)}[t] = \mathrm{BurstLIF}\Big(\mathrm{BatchNorm}(I_{conv}^{(l)}[t])\Big)$. To achieve spatiotemporal noise filtering, BurstCNN utilizes Spiking Max Pooling. Because it operates on discrete spikes, Max Pooling actively discards trailing sensor noise by exclusively propagating the earliest and strongest action potentials within a localized spatial window:$S_{pool}^{(l)}[t] = \mathrm{MaxPool2D}\Big(S_{conv}^{(l)}[t]\Big)$.

BurstFormer (Spiking Transformers) Vision neuromorphic datasets inherently contain both spatial and temporal dimensions. Applying standard Transformer attention across the time dimension (T) incurs a massive $\mathcal{O}(T^2)$ computational bottleneck. BurstFormer calculates attention strictly across N spatial patches intra-frame, relegating temporal integration to the individual

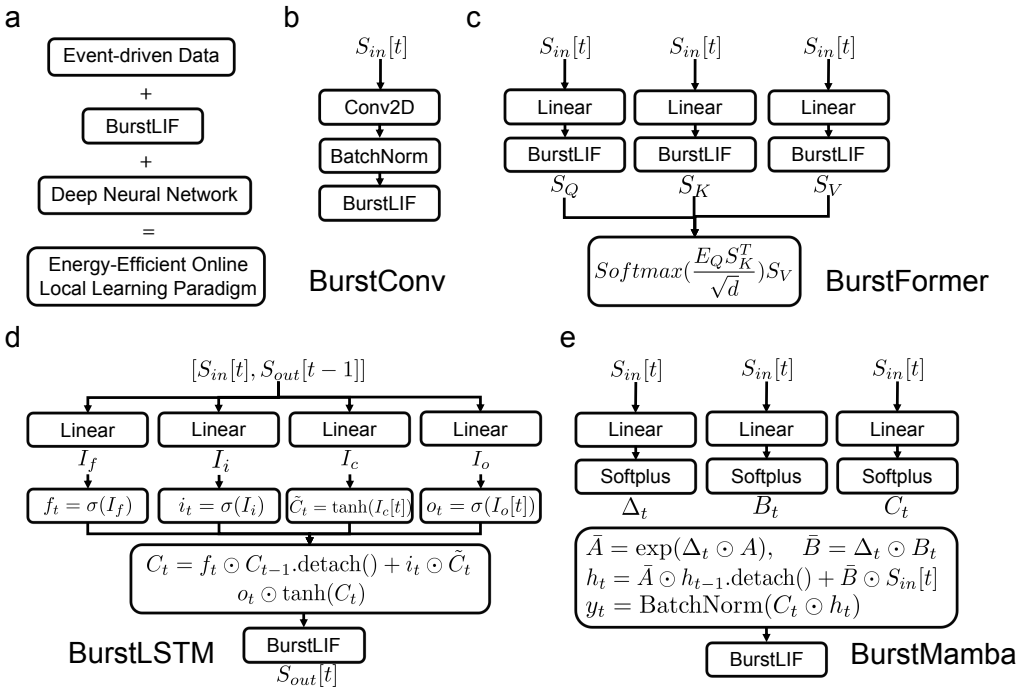


Figure 5: **Advanced Burstprop supports development of deep SNNs for hard Neuromorphic task in online learning setting. (a)** By combination of Advanced Burstprop with deep SNNs, we can obtain an energy-efficient online local learning paradigm for event-driven environment. **(b)-(e)** Schematic of BurstConv, BurstFormer, BurstLSTM and BurstMamba; Advanced Burstprop trained spiking network modules of convolutional neural network, Transformer encoder block, LSTM and Mamba.

LIF neurons, as shown in Fig. 5(c). Given a sequence of spatially flattened tokens $S_{in}[t] \in \{0,1\}^{B\times N\times D}$, the Query ($Q_t$), Key ($K_t$), and Value ($V_t$) tensors are projected using basal dendrites and converted into spikes: $Q_t = \text{BurstLIF}(W_Q S_{in}[t]), K_t = \text{BurstLIF}(W_K S_{in}[t]), V_t = \text{BurstLIF}(W_V S_{in}[t])$. To achieve dynamic spatial routing without breaking the spiking paradigm, the Attention matrix evaluates the overlap of the discrete spike trains. The output routes the spiking Values (V_t):$\text{Attn}(Q_t, K_t, V_t) = \text{softmax}\left(\frac{E_{Q_t} K_t^T}{\sqrt{d_{head}}}\right) V_t$. In more complex hybrid variations (BioBurstFormer), the $Q \cdot K^T$ interaction can be computed in the continuous analog domain as dendritic voltage integration, before being thresholded back into sparse attention-routing spikes.

BurstLSTM (Spiking Long Short-Term Memory) Traditional LSTMs are trained using BPTT, unrolling vast computational graphs. In Fig. 5(d), BurstLSTM implements recurrent temporal sequences using strictly $\mathcal{O}(1)$ online memory by physically decoupling the continuous Cell State from the backward gradient flow. At each timestep t, the basal dendrites receive a concatenation of the current input spikes $S_{in}[t]$ and the previous output spikes $S_{out}[t-1]$. These sparse binary vectors drive four independent analog currents:$I_f[t] = W_f \cdot [S_{in}[t] \parallel S_{out}[t-1]], I_i[t] = W_i \cdot [S_{in}[t] \parallel S_{out}[t-1]], I_c[t] = W_c \cdot [S_{in}[t] \parallel S_{out}[t-1]], I_o[t] = W_o \cdot [S_{in}[t] \parallel S_{out}[t-1]]$. Because inputs are binary spikes, these operations are computed using highly efficient Accumulate (AC) operations. The currents are processed via standard non-linearities to form the dynamic gates: $f_t = \sigma(I_f[t]), i_t = \sigma(I_i[t]), \tilde{C}_t = \tanh(I_c[t])$, and $o_t = \sigma(I_o[t])$. The continuous Cell State C_t serves as the analog memory of the neuron. To enforce $\mathcal{O}(1)$ Burstprop, the recurrent connection is explicitly detached from the autograd graph: $C_t = f_t \odot C_{t-1}.\text{detach}() + i_t \odot \tilde{C}_t$. Finally, the updated cell state is gated and converted back into a sparse binary spike train $S_{out}[t]$ via the somatic threshold:$S_{out}[t] = \text{BurstLIF}\Big(o_t \odot \tanh(C_t)\Big)$.

BurstMamba (Spiking State Space Models) For extremely long temporal sequences (e.g., $T \geq 200$ in audio datasets), BurstMamba merges Burstprop with the Selective State Space Model framework. It processes 1D acoustic trajectories locally in $\mathcal{O}(N)$ time, as shown in Fig. 5(e). BurstMamba translates incoming spikes $S_{in}[t]$ into data-dependent gating parameters using basal dendritic projections: $\Delta_t = \text{Softplus}(W_\Delta \cdot S_{in}[t]), B_t = W_B \cdot S_{in}[t], C_t = W_C \cdot S_{in}[t]$. Using a Zero-Order Hold (ZOH) discretization, the base transition matrix A is mapped dynamically:$\bar{A} = \exp(\Delta_t \odot A), \quad \bar{B} = \Delta_t \odot B_t$. The dense hidden state $h_t \in \mathbb{R}^{B\times D\times N}$ integrates the temporal sequence identically to an analog dendritic tree. Similar to BurstLSTM, h_{t-1} is explicitly detached from the PyTorch graph to restrict the algorithm to pure $\mathcal{O}(1)$ local updates:$h_t = \bar{A} \odot h_{t-1}.\text{detach}() + \bar{B} \odot S_{in}[t]$. The continuous memory state is projected via C_t into an analog current y_t:$y_t = \text{BatchNorm}(C_t \odot h_t)$. This current forces the `BurstLIF` somatic compartment to emit sparse temporal feature spikes $S_{out}[t]$ for downstream processing, incorporating a discrete residual connection:$S_{out}[t] = \text{BurstLIF}(y_t) + S_{in}[t]$.

A.3 The payload alphabet: ternary by default, wider by quantization.

All alphabets share one pipeline — RMS-normalized feedback integrated into the apical trace V_d, the continuous drive $d_j = |\tanh(V_{d,j} \cdot \gamma_l)| \in [0, 1)$ held in the linear region by Loop B, and the bAP permission window — and differ only in the final encoding. The ternary default ($L = 1$) is stochastic. The rate budget is enforced per regime: in the stochastic (Bernoulli) regime the burst probability is the bAP-gated drive itself, so the event rate is an emergent property of the drive distribution, pinned by Loop B's linearity setpoint on γ together with the bAP window (Loop A's θ_l is inert under Bernoulli sampling); in the deterministic regime the same budget is enforced by Loop A through the adaptive threshold θ_l. Wider alphabets ($L > 1$) instead quantize the drive magnitude deterministically into L uniform bins and carry magnitude in the payload integer:

$$m_j = \text{clip}\big(\text{round}(d_j \cdot L),\, 0,\, L\big), \qquad b_j = m_j \cdot \text{sign}(V_{d,j}) \in \{-L, \ldots, 0, \ldots, +L\}, \tag{25}$$

so L=2 gives $\{-2..2\}$, L=4 gives $\{-4..4\}$, L=8 gives $\{-8..8\}$.

A.4 The unified control architecture

The channel has two constraints to hold simultaneously: the *burst rate* (the energy budget ν^*) and the *operating region* (tanh linearity, without which the drive is magnitude-blind).

Loop A—per-layer rate control (owns θ_l) With sens_l owned by Loop B, the rate target relocates to the threshold. The burst rate $\nu_l(\theta_l)$ is monotone decreasing in θ_l, and

$$\theta_l \leftarrow \theta_l \exp\big(-\eta\,(\nu^* - \hat{\nu}_l)\big), \qquad \theta_l \in [0.05, 0.95], \tag{26}$$

log-space integral control (monotone plant, timescale separation). $\hat{\nu}_l$ is a leaky estimate of the measured burst rate, folded once per batch. Loop A pins the energy budget ν^* while Loop B keeps the rate-coding channel faithful.

Loop B—tanh linearity control (owns γ_l) The control signal is measured, not assumed: $r = \mathbb{E}|\tanh x|/\mathbb{E}|x|$ with $x = V_d \cdot \gamma_l$ ($r \approx 1$ when linear, $\to 0$ when saturated). The per-layer gain $g_l^{lin} \leftarrow g_l^{lin} \exp(\eta_B(x^* - \bar{x}_l))$ holds $\bar{x}_l = \mathrm{RMS}(V_d \cdot \gamma_l)$ at the setpoint

$$x^* = \operatorname{artanh}(1/\sqrt{2}) \approx 0.88, \tag{27}$$

the half-slope point ($\mathrm{sech}^2 x^* = 0.5$), which coincides with the maximum of the dynamic-range throughput $x\,\mathrm{sech}^2(x)$ (peak at ≈ 0.75–0.88). Effect: the drive stays magnitude-graded, so event selection picks informative coordinates rather than occupancy-ranked ones.

A.5 Energy-Accounting Convention and Experimental Protocol

Energy model All energies are computed under the 45 nm CMOS model of Horowitz (2014), with $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ per multiply-accumulate and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ per accumulate. We count *synaptic operations only*, at every timestep, in both directions, using the identical convention for all methods. Per-neuron state updates are excluded uniformly: membrane leak/integration/threshold/reset, eligibility-trace decay, LSTM cell-state updates, Mamba SSM decay, gate nonlinearities, BatchNorm, and readout accumulation carry no charge for any method. A monitor hook is registered on every weight-bearing layer (linear, convolutional); since the model is invoked once per timestep, each hook fires once per timestep and measures that timestep's operations, so both directions accumulate over the full sequence.

Forward pass The forward pass is event-driven: a synapse operates only when its presynaptic neuron emits an event at that timestep, and each operation is an accumulate. With $n_{\mathrm{ev}}(\ell, t)$ presynaptic events in layer ℓ at timestep t and fan-out f_ℓ, $E_{\mathrm{fwd}} = \sum_{\ell,t} n_{\mathrm{ev}}(\ell, t) \cdot f_\ell \cdot E_{\mathrm{AC}}$. Layers receiving continuous (analog) currents are charged at E_{MAC} per operation.

Backward pass, per method family Let N_{ops} be the dense synaptic-op count accumulated over all timesteps of one sample. The factor 2 below accounts for the two operands of the backward, the activation-side and weight-side gradients.

- **Dense-analog rules** (SG-BPTT, SG-detached, OTTT, DECOLLE, E-prop): $E_{\mathrm{bwd}} = 2 \cdot N_{\mathrm{ops}} \cdot E_{\mathrm{MAC}}$; OSTL adds its per-synapse trace-state operations, $+N_{\mathrm{trace}} \cdot E_{\mathrm{MAC}}$.
- **ETLP** (event-triggered, analog): $E_{\mathrm{bwd}} = 2 \cdot N_{\mathrm{ops}} \cdot \theta_{\mathrm{trig}} \cdot E_{\mathrm{MAC}}$, where $\theta_{\mathrm{trig}} \approx 0.2$ is the realized fraction of triggered timesteps.
- **ABP (this work)**: $E_{\mathrm{bwd}} = 2 \cdot N_{\mathrm{ops}} \cdot \nu_b \cdot E_{\mathrm{AC}} \cdot \lceil \log_2(L{+}1) \rceil$, where ν_b is the *measured* burst-event rate and L the alphabet half-width. Ternary ($L{=}1$) has word-width factor 1; a payload from $\{-L, \ldots, +L\}$ is delivered as $\lceil \log_2(L{+}1) \rceil$ shift-add accumulates (powers-of-two shifts), so the backward remains multiply-free at any alphabet size.

ANN reference The ANN equivalent of each architecture is charged $N_{\mathrm{ops}} \cdot E_{\mathrm{MAC}}$ for the forward pass and $3\times$ that for the full training step (forward $+2\times$ backward), executed over the same number of timesteps as the SNN protocol.

Averaging convention Energies are per sample (batch totals divided by batch size) and are averaged over *all* epochs of training, then over the 3 seeds. Accuracies are best-epoch test accuracies; significance uses two-sided Welch's t-tests over seeds. Every method shares the same architecture and LIF neuron model within each comparison, baselines share the burst group's learning rate, and checkpoint selection is uniform across methods.

Table 3: Protocol of the main experiments

	DVS-Gesture	**SHD**
Classes	11	20
Input	$32\times32\times2$ events	700 cochlear channels
Time steps T	100	100
Batch size	256	256
Burn-in fraction	0.2	0.2
Epochs	150	50
Seeds	3	3
lr, Bernoulli group	10^{-4}	10^{-4}
lr, deterministic group	2×10^{-3}	10^{-4}
Burst budget ν^*	0.05	0.05
Architectures	BurstCNN, BurstFormer	BurstLSTM, BurstMamba, TCNN-LSTM, TCNN-Mamba