VARIABLE-ORDER SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most existing spiking neural networks (SNNs) describe membrane dynamics with first-order ordinary differential equations, so earlier activity affects the membrane only through an exponential decay with a fixed time constant. Fractional-order SNNs (f-SNNs) address this limitation by replacing the first-order derivative with a fractional one, whose power-law kernel lets the membrane potential depend on long-range history. However, f-SNNs fix the derivative order for the entire input sequence, so the balance between recent and distant history cannot adapt as the input evolves. To address this problem, we propose Variable-Order Spiking Neural Networks (*VO*-SNNs). Specifically, we introduce variable-order leaky integrate-and-fire (*VO*-LIF) neurons, whose fractional order varies over time and depends on the current input and the accumulated membrane memory. Based on a variable-order Grünwald–Letnikov discretization, *VO*-LIF neurons admit a direct charge–spike–reset update that can be trained end to end with surrogate gradients and adds only a small number of order parameters. *VO*-SNNs recover first-order SNNs and fixed-order f-SNNs as special cases, and we prove that their discrete dynamics are causal, remain bounded under an explicit step-size condition, and that time-varying orders produce non-stationary history kernels that no fixed order can reproduce. Extensive experiments show that *VO*-SNNs perform well on event-based recognition with multiple architectures, event-based object detection, graph-spike node classification, and neural response prediction from mouse visual cortex recordings, improving accuracy over fixed-order f-SNNs by up to 7.44 percentage points while consuming energy comparable to f-SNNs on neuromorphic hardware.

1 INTRODUCTION

Spiking neural networks (SNNs) draw inspiration from biological neurons and communicate through discrete, event-driven spikes, which enables substantial energy savings on neuromorphic hardware (Maass, 1997; Roy et al., 2019; Pei et al., 2019; Davies et al., 2021). Because their neurons carry state across time, SNNs are also naturally suited to temporal processing (Wu et al., 2018; Yao et al., 2021; 2023). How well an SNN processes temporal information therefore depends not only on its synaptic architecture but also on how each neuron integrates its input history.

Most SNNs use integrate-and-fire (IF) or leaky integrate-and-fire (LIF) neurons, whose membrane dynamics follow first-order ordinary differential equations (Stein, 1967; Gerstner et al., 2014). These dynamics are Markovian: the next state depends only on the present state and input, and past activity is summarized by a single exponential decay. Real neurons, in contrast, exhibit long-range temporal correlations and history-dependent responses (Gilboa et al., 2005; La Camera et al., 2006). Fractional-order models capture such behavior by replacing the first-order derivative with a fractional derivative of order $\alpha \in (0, 1]$, whose power-law kernel weights the entire past trajectory (Diethelm, 2010). The fractional LIF (f-LIF) neuron reproduces spike-frequency adaptation in neocortical pyramidal neurons (Lundstrom et al., 2008; Teka et al., 2014), and fractional-order SNNs (f-SNNs) bring these dynamics into deep, trainable networks with consistent gains over integer-order models (Ge et al., 2026).

However, f-SNNs fix the order α for the entire input sequence, so a neuron applies the same history weighting whether it is at rest, strongly driven, or recovering from strong activity. Neocortical pyramidal neurons do not integrate input in this way. Under intense synaptic input they enter a high-conductance state in which the effective membrane time constant shortens and integration becomes

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.

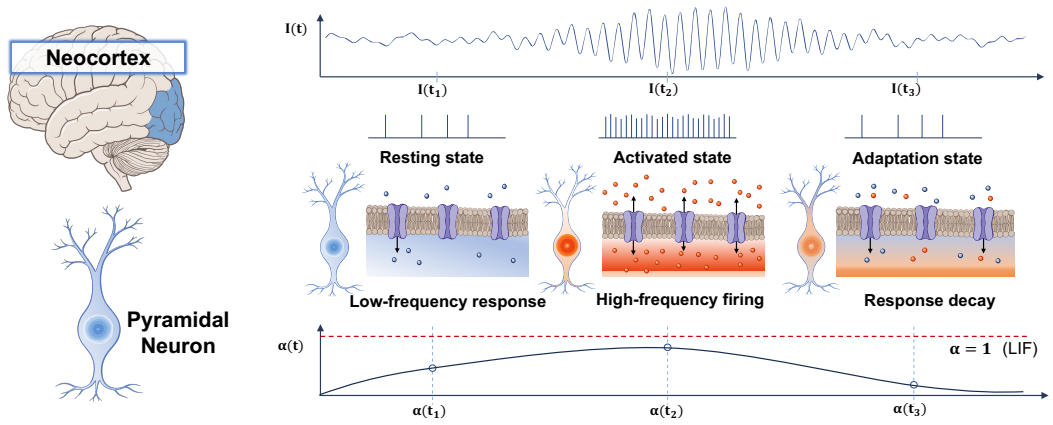

Figure 1: Biological motivation for variable-order neuronal memory. **Left:** A neocortical pyramidal neuron, whose responses have been linked to history-dependent dynamics. **Top:** Schematic stimulation, firing, and membrane-channel activity across resting, activated, and adaptation phases. **Bottom:** An illustrative order trajectory shows how a variable-order neuron could shift the relative weight of recent and distant history across these phases; the diagram is a conceptual illustration rather than a fitted trajectory.

more local (Destexhe et al., 2003); after sustained firing, slow adaptation currents retain a long-lasting trace of that activity and suppress subsequent responses (Ha & Cheong, 2017; La Camera et al., 2006). Moreover, the time scale of adaptation is not fixed but set by the recent stimulus history (Lundstrom et al., 2008; Ulanovsky et al., 2004). How far back a neuron effectively remembers thus changes with its state, which a single fixed order, and hence a stationary memory kernel, cannot represent. Variable-order fractional operators, whose order is itself a function of time, were introduced to model systems with such evolving memory (Samko & Ross, 1993; Lorenzo & Hartley, 2002). Fractional-order neural networks, however, have mainly been studied as dynamical systems with a fixed order (Anastasio, 1994; Kaslik & Sivasundaram, 2012; Yang et al., 2018), and to our knowledge variable-order dynamics have not yet been brought into deep, trainable SNNs.

To address this limitation, we propose Variable-Order Spiking Neural Networks (*VO*-SNN), built from variable-order leaky integrate-and-fire (*VO*-LIF) neurons whose fractional order changes over time. As illustrated in Figure 1, the order follows the neuron's state: it takes an intermediate value at rest, rises toward $\alpha = 1$ under strong drive so that integration becomes nearly local, and falls during adaptation so that the preceding activity weighs heavily on the membrane. Similar inputs at t_1 and t_3 thus receive different orders because the histories differ. Since the current order re-weights the entire membrane history, a change in order reshapes the memory kernel itself. The order is produced by a causal controller from the current input and the accumulated membrane memory, and we use it as a computational abstraction rather than a measured physiological quantity. A constant order recovers the f-LIF neuron, and $\alpha = 1$ recovers the LIF neuron. Discretizing these dynamics with the Grünwald–Letnikov (GL) scheme (Podlubny, 1999) yields an explicit charge–spike–reset update, like that of the LIF neuron, which can be trained end to end with surrogate gradients. *VO*-LIF neurons therefore serve as drop-in replacements for LIF neurons in existing residual, Transformer-based, detection, and graph SNNs, leaving the rest of the architecture unchanged.

Our contributions are as follows.

- We propose *VO*-SNN, to our knowledge the first SNN framework with variable-order fractional neuron dynamics, whose *VO*-LIF neurons adjust their memory profile during a sequence and recover f-LIF and LIF neurons as special cases.
- We derive a direct GL update with a compact order parameterization and prove that the dynamics are causal, bounded under an explicit step-size condition, and produce non-stationary history kernels that no fixed order can reproduce.
- On event-based recognition, event-based object detection, graph-spike node classification, and neural response prediction, *VO*-SNN improves over LIF and f-SNN baselines, by up to 7.44 accuracy points over f-SNNs, with energy comparable to f-SNNs on neuromorphic hardware.

2 Related Work

2.1 Spiking Neural Networks

SNNs are often regarded as the third generation of neural networks (Maass, 1997). They communicate through sparse, asynchronous spikes, which matches event-driven neuromorphic hardware and saves energy (Davies et al., 2021; Tavanaei et al., 2019). The LIF neuron is the most common spiking neuron because it balances biological plausibility and simplicity (Stein, 1967; Gerstner et al., 2014). Its first-order dynamics, $\tau\,\mathrm{d}U/\mathrm{d}t = -U + RI_{\mathrm{in}}$, make the membrane potential decay exponentially with a single fixed time constant, so earlier activity is summarized by a single exponential decay (Teeter et al., 2018; Trushechkin, 2024). Variants with adaptive thresholds, nonlinear charging, or parallelizable dynamics (Bellec et al., 2018; Fang et al., 2023b; Huang et al., 2024; Zhang et al., 2025) largely keep this first-order, Markovian form (Ge et al., 2026). *VO*-SNN instead replaces the local derivative with a non-local operator whose memory profile can change during a sequence.

2.2 Fractional and Variable-Order Neuron Models

Fractional calculus has been used in neuroscience to model anomalous diffusion and long memory in neuronal dynamics (Anastasio, 1994; Lundstrom et al., 2008). The f-LIF model, $\tau D^{\alpha}U(t) = -U(t) + RI_{\mathrm{in}}(t)$ with a Caputo derivative, reproduces spike-frequency adaptation in pyramidal neurons (Teka et al., 2014; Deng et al., 2022); its Mittag-Leffler solution (Podlubny, 1999; Diethelm, 2010) decays as a power law rather than exponentially, so past inputs persist (Chen et al., 2014; Ding et al., 2016), which improves robustness to noisy inputs (Ding et al., 2025; Rubin et al., 2017). Earlier network-level work analyzed shallow fractional Hopfield-type spiking networks without training deep architectures (Zhang et al., 2026), and f-SNN (Ge et al., 2026) integrated fractional dynamics into deep SNNs with end-to-end training. All these models fix the order α over the whole sequence. Variable-order operators, whose order is a function of time, describe systems with evolving memory (Samko & Ross, 1993; Lorenzo & Hartley, 2002), and single-neuron studies show that responses depend on stimulation history across several time scales (Gilboa et al., 2005; Buzsáki, 2019). *VO*-SNN brings this idea to deep SNNs through a direct Grünwald–Letnikov update in which the order varies during the sequence (Figure 1); we treat the learned order as a computational variable rather than a measured physiological quantity (see also Appendix A).

3 Variable-Order Spiking Neural Networks

We present *VO*-SNN, which endows spiking neurons with variable-order fractional dynamics. An LIF neuron updates its membrane potential from the previous state alone, and an f-LIF neuron weights its history with a kernel that is fixed for the entire sequence. A *VO*-LIF neuron instead recomputes its history kernel at every time step from the current order, so the balance between recent and distant history can change as the input evolves. Section 3.1 defines the neuron, Section 3.2 derives its discrete charge–spike–reset update, Section 3.3 describes how the order is computed, and Section 3.4 analyzes the resulting dynamics (background in Appendix B).

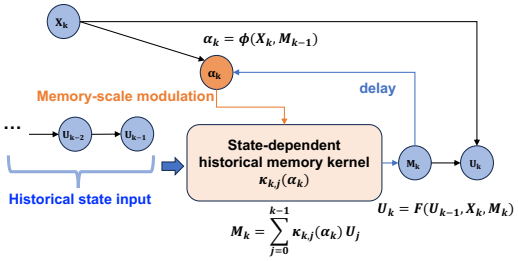

Figure 2: Computational graph of a *VO*-LIF neuron. The current input X_k and the previous memory M_{k-1} determine the order $\alpha_k = \phi(X_k, M_{k-1})$, which sets the history kernel $\kappa_{k,j}(\alpha_k)$; the resulting memory M_k enters the update of U_k.

3.1 Variable-Order LIF Neuron

Let $U(t)$ denote the membrane potential, $X(t)$ the presynaptic input, and $\tau > 0$ the membrane time constant. The f-LIF neuron (Teka et al., 2014; Ge et al., 2026) obeys

$$\tau D^{\alpha}U(t) = -U(t) + X(t), \qquad 0 < \alpha \le 1, \tag{1}$$

where D^α is a fractional derivative of order α, and $\alpha = 1$ recovers the LIF neuron. A *VO*-LIF neuron replaces the constant order with a time-varying order $\alpha(t)$:

$$\tau D^{\alpha(t)} U(t) = -U(t) + X(t), \qquad 0 < \alpha(t) \leq 1. \tag{2}$$

Because different definitions of the continuous variable-order derivative need not coincide (Samko & Ross, 1993; Lorenzo & Hartley, 2002), we take the discrete operator of Section 3.2 as the definition of how the current order acts on past states. Spikes are emitted when the membrane potential crosses a threshold, followed by a reset.

3.2 Discretization and *VO*-SNN

Consider the grid $t_k = kh$ with step size $h > 0$, and write $\alpha_k = \alpha(t_k)$. Let U_k denote the post-reset membrane potential and X_k the presynaptic input at step k, which may be a spike vector or a feature map produced by a convolutional, residual, or Transformer block. We use the Grünwald–Letnikov (GL) coefficients (Podlubny, 1999; Diethelm, 2010) and the associated history weights

$$c_0(\alpha) = 1, \qquad c_j(\alpha) = \Big(1 - \frac{1+\alpha}{j}\Big) c_{j-1}(\alpha), \qquad w_j(\alpha) = -c_j(\alpha), \quad j \geq 1. \tag{3}$$

For $0 < \alpha < 1$, all weights are positive, $w_1(\alpha) = \alpha$, and $\sum_{j\geq 1} w_j(\alpha) = 1$ (Lemma 1). At step k, we apply the initial-value-corrected GL operator with the *current* order to the post-reset history $U_0, \ldots, U_{k-1}$ and the current pre-reset charge Z_k,

$$\mathcal{D}_h^{\alpha_k} Z_k = h^{-\alpha_k} \Big[(Z_k - U_0) + \sum_{j=1}^{k} c_j(\alpha_k)\, (U_{k-j} - U_0) \Big], \tag{4}$$

and evaluate the leak and input at the previous post-reset state, $\mathcal{D}_h^{\alpha_k} Z_k = (X_k - U_{k-1})/\tau$. Solving for Z_k and then applying the spike and reset gives the charge–spike–reset update of a *VO*-LIF neuron:

$$\text{Memory:} \quad M_k = \sum_{j=1}^{k} w_j(\alpha_k)\, (U_{k-j} - U_0), \tag{5}$$

$$\text{Charge:} \quad Z_k = U_0 + M_k + \frac{h^{\alpha_k}}{\tau}\, (X_k - U_{k-1}), \tag{6}$$

$$\text{Spike:} \quad S_k = \Theta(Z_k - \vartheta), \tag{7}$$

$$\text{Reset:} \quad U_k = \begin{cases} Z_k - \vartheta S_k, & \text{(soft reset)}, \\ (1 - S_k) Z_k + S_k U_{\text{reset}}, & \text{(hard reset)}, \end{cases} \tag{8}$$

where $\vartheta > 0$ is the firing threshold and Θ is the Heaviside function. The memory M_k aggregates the entire post-reset history with weights generated by the current order, so every past state is reweighted whenever α_k changes; the history is not cleared after a spike. We set $U_0 = 0$ unless stated otherwise, in which case $M_k = \sum_{i=0}^{k-1} \kappa_{k,i}(\alpha_k) U_i$ with $\kappa_{k,i}(\alpha_k) = w_{k-i}(\alpha_k)$, the aggregation shown in Figure 2. For a constant order, Equations (5)–(8) give the GL-discretized f-LIF neuron, which we also use for all f-LIF baselines; for $\alpha_k = 1$ they reduce to the forward-Euler LIF update (Proposition 1).

Training uses surrogate gradients for Θ (Wu et al., 2018; Neftci et al., 2019). Because the weights w_j are differentiable in the order (Appendix D), the task loss updates both the synaptic weights and the order controller. Direct evaluation of M_k over a length-T sequence costs $O(T^2)$ operations per neuron; our GPU implementation computes the weights of each step once and shares them across all neurons that use the same order, without a general-purpose differential-equation solver (Appendix E).

3.3 Parameterizing the Order

A single order α_k per sample and time step is shared by all spiking neurons in the network and is produced by a lightweight causal controller (Figure 2):

$$\alpha_k = \alpha_- + (\alpha_+ - \alpha_-)\, \sigma\Big(b_k + s \tanh\big(\phi_\theta(t_k,\, s_X(X_k),\, s_M(M_{k-1}))\big)\Big), \qquad M_0 = 0, \tag{9}$$

where $0 < \alpha_- < \alpha_+ < 1$ bound the order, σ is the logistic function, b_k is a learnable base schedule, $s > 0$ scales the residual, and $t_k \in [-1, 1]$ is the normalized time step. The summary s_X describes

the current input event frame (the mean and standard deviation of each polarity channel, the fraction of active events, and the difference between the channel means), and s_M describes the previous memory of the last spiking layer (its mean, standard deviation, mean magnitude, and RMS). The input tells the controller what is happening now, whereas the memory tells it what the network has already accumulated; reading M_{k-1} rather than M_k keeps the update explicit. The controller ϕ_θ is a small MLP with Tanh activations whose last layer is zero-initialized, so training starts from the base schedule, and it adds fewer than 800 parameters (below 0.03% of the backbone). We stop gradients through the memory summary, while the controller and the order still receive gradients through the GL weights. A constant order recovers the f-LIF (GL) neuron.

3.4 Theoretical Analysis

We analyze the update in Equations (5)–(8). Proofs are given in Appendix C.

Proposition 1 (Reduction and causality). *(i) If $\alpha_k \equiv \bar{\alpha}$, the update coincides step by step with the GL-discretized* f*-LIF neuron of order $\bar{\alpha}$. (ii) At $\alpha = 1$, $c_1(1) = -1$ and $c_j(1) = 0$ for $j \geq 2$, and the charge reduces to the forward-Euler LIF update $Z_k = U_{k-1} + \frac{h}{\tau}(X_k - U_{k-1})$. (iii) If two input sequences agree up to step K, the orders, charges, spikes, and post-reset states they produce agree up to step K.*

Proposition 2 (The order modulates the memory kernel). *For $0 < \alpha < 1$ and lags $1 \leq r < j$,*

$$\frac{\partial}{\partial \alpha} \log \frac{w_j(\alpha)}{w_r(\alpha)} = -\sum_{\ell=r+1}^{j} \frac{1}{\ell - 1 - \alpha} < 0, \qquad w_j(\alpha) = \frac{\alpha}{\Gamma(1-\alpha)}\, j^{-1-\alpha}\big(1 + O(j^{-1})\big). \quad (10)$$

A smaller order therefore strictly increases the weight of distant history relative to recent history, and the kernel at step k decays with an exponent set by α_k.

Proposition 3 (Uniform boundedness). *Let $\alpha_k \in [\alpha_-, \alpha_+] \subset (0,1)$, $\|X_k\|_\infty \leq X_*$, and $A = \max_\alpha h^\alpha/\tau$ (Assumption 1), and suppose $0 < h^\alpha/\tau < 2\alpha$ on $[\alpha_-, \alpha_+]$. Then $\rho = \max_{\alpha \in [\alpha_-, \alpha_+]}\big(|\alpha - h^\alpha/\tau| + 1 - \alpha\big) < 1$, and for either reset,*

$$\sup_{k \geq 0} \|U_k\|_\infty \leq \frac{AX_* + \|U_0\|_\infty}{1 - \rho} \quad (11)$$

for every admissible order sequence, including orders that depend on the input and memory.

Proposition 4 (Non-stationary memory). *Write $M_k = \sum_{i=0}^{k-1} \kappa_{k,i}(U_i - U_0)$ with $\kappa_{k,i} = w_{k-i}(\alpha_k)$. Every fixed-order kernel, and every fixed weighted combination of fixed-order kernels, is stationary: $\kappa_{k,i}$ depends only on the lag $k - i$. If $\alpha_k \neq \alpha_{k'}$ for two steps $k, k' \geq 1$, the* VO*-LIF kernel is not stationary and therefore cannot be reproduced by any such kernel.*

Remark. When the controller raises α_k, the memory concentrates on recent states; when it lowers α_k, distant states gain weight, and because the kernel changes with k, similar inputs can be integrated differently depending on the preceding history (Figure 1). The condition in Proposition 3 is sufficient but not necessary; we also observe stable training empirically for broader order ranges.

4 Experiments

We evaluate *VO*-SNN on four families of temporal tasks (Figures 3 and 5): event-based recognition, event-based object detection, graph-spike node classification, and neural response prediction. In every comparison, the backbone, input representation, data split, and training protocol are fixed, and only the spiking neuron is replaced: the LIF neuron, the GL-discretized f-LIF neuron (both run with the official open-source f-SNN code (Ge et al., 2026)), or the *VO*-LIF neuron with the order controller of Section 3.3. Dataset details and training settings are given in Appendix F.

4.1 Event-based recognition

We evaluate on N-Caltech101 (Orchard et al., 2015), DVS-Lip (Tan et al., 2022), and HAR-DVS (Wang et al., 2024) with SEW-ResNet-18 (Fang et al., 2021) and Spikformer-E256 (Zhou et al., 2023) backbones. Table 1 shows that *VO*-LIF achieves the highest accuracy in every setting.

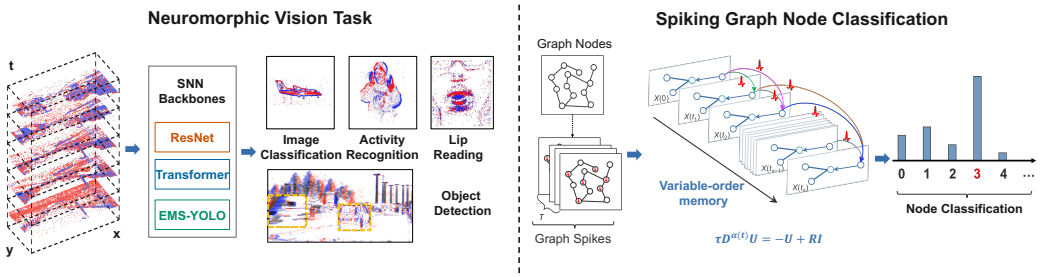

Figure 3: Task workflows for spiking temporal processing. **Left:** Neuromorphic event streams are processed by spiking ResNet, Transformer, or EMS-YOLO backbones for object recognition, activity recognition, lip reading, and object detection. **Right:** A temporal spiking graph sequence is integrated with variable-order memory to produce node representations and node-class predictions.

Table 1: Event-based recognition accuracy (%, mean ± standard deviation over five seeds). All models use the same backbone and training protocol and differ only in the spiking neuron. Best results are in bold.

Dataset	Backbone	Steps	LIF	f-LIF (GL)	*VO*-LIF
N-Caltech101	SEW-ResNet-18	16	69.79 ± 0.32	70.66 ± 0.51	$\mathbf{73.45 \pm 0.64}$
	Spikformer-E256	16	71.03 ± 0.78	73.53 ± 1.37	$\mathbf{77.42 \pm 0.85}$
DVS-Lip	SEW-ResNet-18	16	33.27 ± 0.47	42.05 ± 0.33	$\mathbf{45.32 \pm 0.61}$
	Spikformer-E256	16	20.42 ± 0.71	32.47 ± 0.82	$\mathbf{39.91 \pm 0.91}$
HAR-DVS	SEW-ResNet-18	8	45.93 ± 0.05	47.29 ± 0.12	$\mathbf{49.63 \pm 0.11}$
	Spikformer-E256	8	45.37 ± 0.14	46.47 ± 0.09	$\mathbf{48.44 \pm 0.16}$

Relative to f-LIF, the gain ranges from 1.97 to 7.44 points and is largest on DVS-Lip, where Spikformer improves from 32.47% to 39.91%; recognizing words in this dataset depends on brief lip movements embedded in longer sequences, which is consistent with the motivation for a memory that can change within a sequence. The gains hold for both convolutional and Transformer backbones and exceed the run-to-run standard deviation in all six settings. Because f-LIF also improves over LIF in every setting, long memory and its adaptation over time contribute separately.

4.2 Event-based object detection

We further evaluate *VO*-SNN on event-based object detection with the Gen1 (de Tournemire et al., 2020) and Gen4 (Perot et al., 2020) automotive datasets, using the EMS-YOLO detector (Su et al., 2023) with ResNet-10 and ResNet-34 backbones. As in the recognition experiments, only the spiking neurons are replaced. Table 2 shows that *VO*-LIF improves over both LIF and f-LIF neurons on both datasets and both backbones. Relative to f-LIF, the gain in mAP@0.5:0.95 ranges from 2.2 to 3.9 points, and the gains are consistent across the two backbone sizes, indicating that the benefit of a time-varying order carries over from recognition to dense prediction. Figure 4 shows a representative Gen1 scene: with the same detector, *VO*-LIF neurons produce boxes that align more closely with the ground truth and assign higher confidence to the vehicles than LIF and f-LIF neurons. Further Gen1 and Gen4 scenes are shown in Appendix J.

4.3 Graph-spike node classification

The right panel of Figure 3 illustrates the processing of graph-structured spike information: node spikes enter the network over time, and the current order reweights the contribution of earlier states to each neuronal update, yielding node representations for classification. Following this workflow, we evaluate LIF, f-LIF, and *VO*-LIF in the SGCN (Zhu et al., 2022) and DRSGNN (Zhao et al., 2024) backbones.

We evaluate eight benchmarks: the citation networks Cora (McCallum et al., 2000), Citeseer (Sen et al., 2008), PubMed (Namata et al., 2012), and ogbn-arxiv (Hu et al., 2020); the Amazon co-purchase networks Photo and Computers (Shchur et al., 2018); and the heterophilous graphs squirrel

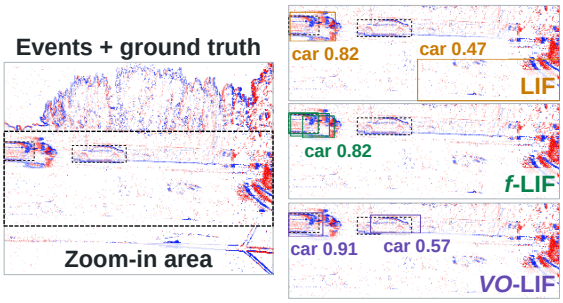

Figure 4: Qualitative detections on Gen1. Left: events with ground-truth boxes; the dashed box marks the zoom-in area. Right, top to bottom: EMS-YOLO predictions in this area with LIF, f-LIF, and *VO*-LIF neurons.

Table 2: Event-based object detection with EMS-YOLO (ResNet-10/34 backbones, %, single run). Best results are in bold.

	Gen1		Gen4	
Neuron	Res10	Res34	Res10	Res34
mAP@0.5:0.95				
LIF	26.7	31.0	21.5	26.4
f-LIF	27.6	32.8	23.2	30.4
VO-LIF	**31.4**	**36.7**	**25.6**	**32.6**
mAP@0.5				
LIF	54.7	59.0	50.4	58.0
f-LIF	56.5	62.4	52.6	60.0
VO-LIF	**60.9**	**66.1**	**55.7**	**62.7**

Table 3: Node classification accuracy (%) on eight graph benchmarks with SGCN and DRSGNN backbones (mean ± standard deviation over five seeds). Best results are in bold.

	SGCN			DRSGNN		
Dataset	LIF	f-LIF	*VO*-LIF	LIF	f-LIF	*VO*-LIF
Cora	79.78±0.50	80.83±0.53	**82.43**±0.54	80.35±0.23	80.70±0.20	**82.05**±0.33
Citeseer	67.24±0.69	68.01±0.76	**71.04**±0.61	67.22±1.18	68.88±0.69	**70.63**±0.94
PubMed	78.14±0.54	78.53±0.54	**79.54**±0.41	78.80±0.23	78.88±0.19	**79.93**±0.36
Photo	93.80±0.12	93.95±0.11	**94.18**±0.30	93.99±0.09	94.01±0.09	**94.22**±0.21
Computers	88.52±0.32	88.72±0.12	**90.02**±0.09	89.37±0.10	89.24±0.17	**90.79**±0.15
ogbn-arxiv	66.16±0.32	66.51±0.30	**68.99**±0.25	66.42±0.21	66.67±0.23	**68.73**±0.45
squirrel	48.86±0.57	49.43±0.39	**52.59**±0.52	49.12±0.38	49.56±0.44	**53.10**±0.49
chameleon	64.61±0.39	64.86±0.45	**66.48**±0.61	64.75±0.46	65.58±0.37	**67.43**±0.83

and chameleon (Rozemberczki et al., 2021; Pei et al., 2020). We report test accuracy over five training seeds for each dataset–backbone combination, and the accuracy of each seed is averaged over eight random spike encodings; splits and dataset statistics are given in Appendix F.

Table 3 compares LIF, f-LIF, and *VO*-LIF across the two graph backbones. *VO*-LIF achieves the highest reported accuracy in all 16 dataset–backbone settings. The reported gaps over f-LIF are particularly large on squirrel (3.16 and 3.54 percentage points for SGCN and DRSGNN) and ogbn-arxiv (2.48 and 2.06 points, respectively). The fixed order adds at most 1.66 points over LIF across the eight graphs, whereas the variable order adds 0.21 to 3.54 points over f-LIF. The results span citation, co-purchase, and heterophilous graphs, indicating that *VO*-LIF can be applied across different graph structures and spiking graph architectures. The learned order schedules for both graph backbones are examined in Appendix G.

4.4 Neural Response Prediction

To test whether the neuron dynamics capture the temporal structure of real neural activity, we predict responses recorded from mouse visual cortex in the Allen Brain Observatory Visual Coding Neuropixels dataset (Siegle et al., 2021) under drifting gratings (DG), natural movies, and natural scenes (Figure 5). Models predict either single-trial activity (Single) or the trial-averaged peri-stimulus time histogram (PSTH). We compare *VO*-LIF neurons and their leak-free counterparts (VO-IF) with IF and LIF neurons, the GL-discretized f-LIF neuron, and the TC-LIF (Zhang et al., 2024) and TS-LIF (Feng et al., 2025) neurons. Table 4 shows that variable-order neurons achieve the lowest error in every setting. The gains are largest for trial-averaged responses: relative to f-LIF, the PSTH prediction MSE decreases from 0.0425 to 0.0163 on drifting gratings, from 0.0501 to 0.0217 on natural movies, and from 0.0467 to 0.0309 on natural scenes. Averaged over all 72 settings, the MSE decreases from 0.0479 for f-LIF to 0.0341 for VO-IF. Single-trial margins are smaller; the standard deviations are taken across settings, not seeds, and a variable-order neuron is best in every one of the

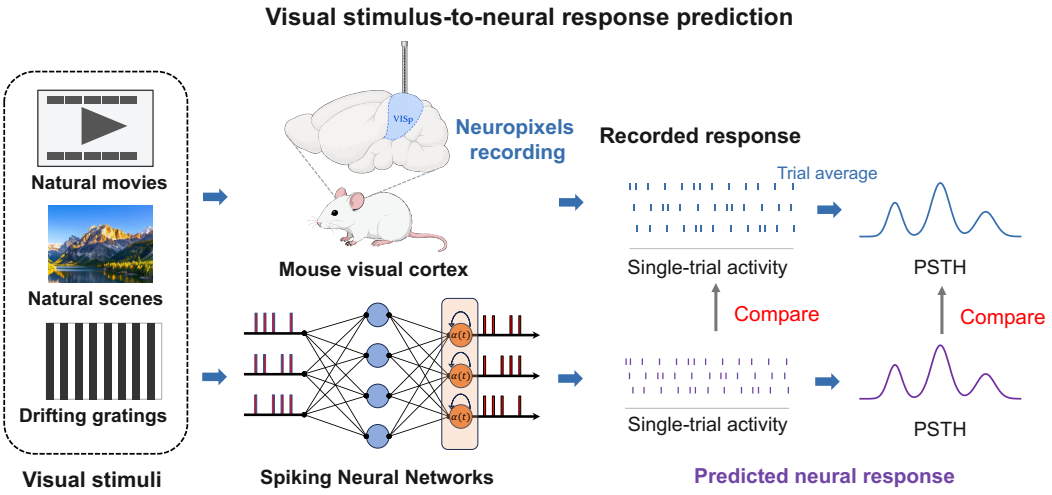

Figure 5: Allen visual-response prediction workflow. Drifting gratings, natural movies, and natural scenes evoke single-trial responses in mouse visual cortex that are recorded with Neuropixels and summarized by peri-stimulus time histograms (PSTHs). A spiking neural network predicts the corresponding single-trial activity or PSTH for direct comparison with the recorded response.

Table 4: Neural response prediction on the Allen Neuropixels dataset. Results are test mean squared error (MSE; lower is better), reported as mean ± standard deviation. "All 72" comprises 72 evaluation settings across three stimuli (DG, Movie, and Scenes), three recording sessions, four stimulus–response lags (0, 40, 80, and 100 ms), and two response modes (single-trial and PSTH). Results are first averaged over five random seeds for each setting and then aggregated across the 72 settings. Best results are in bold.

Stimulus	Mode	VO-IF	*VO*-LIF	TC-LIF	TS-LIF	IF	LIF	f-LIF
DG	Single	**0.0400**±0.0199	0.0404±0.0200	0.0471±0.0212	0.0455±0.0210	0.0465±0.0217	0.0498±0.0234	0.0419±0.0205
	PSTH	**0.0163**±0.0026	0.0178±0.0026	0.0398±0.0022	0.0513±0.0037	0.0453±0.0030	0.0542±0.0039	0.0425±0.0033
Movie	Single	**0.0550**±0.0285	0.0555±0.0287	0.0746±0.0298	0.0704±0.0309	0.0707±0.0303	0.0725±0.0313	0.0644±0.0301
	PSTH	0.0219±0.0059	**0.0217**±0.0057	0.0522±0.0058	0.0611±0.0114	0.0564±0.0102	0.0665±0.0121	0.0501±0.0103
Scenes	Single	**0.0404**±0.0071	0.0406±0.0072	0.0457±0.0079	0.0459±0.0091	0.0463±0.0089	0.0521±0.0121	0.0417±0.0079
	PSTH	**0.0309**±0.0060	0.0311±0.0058	0.0497±0.0051	0.0586±0.0030	0.0514±0.0041	0.0609±0.0030	0.0467±0.0032
All 72	Mean	**0.0341**±0.0193	0.0345±0.0193	0.0515±0.0187	0.0555±0.0183	0.0528±0.0181	0.0593±0.0188	0.0479±0.0173

72 settings (Appendix H). The learned orders on this task are shown in Appendix I, and preprocessing details are given in Appendix F.

4.5 What does the learned order do?

Figure 6 shows the order schedules learned on three event-based datasets. In every case the learned order changes over time instead of staying at a constant value, and the shape of the schedule depends on the dataset. On N-Caltech101, both backbones follow a U-shaped schedule: the order drops during the first steps, stays nearly flat in the middle of the sequence, and rises again toward the end. On HAR-DVS, the two backbones move in opposite directions: the order of Spikformer decreases and then recovers slightly, whereas the order of SEW-ResNet-18 rises, peaks at the fifth step, and then declines. On DVS-Lip, the order of both backbones rises until about the eleventh step; afterwards it stays almost constant for Spikformer and decreases for SEW-ResNet-18. The range of variation also differs, from about 0.01 for Spikformer on DVS-Lip to about 0.24 for SEW-ResNet-18 on N-Caltech101. The network therefore uses the additional freedom in a data-dependent way rather than converging to a common schedule.

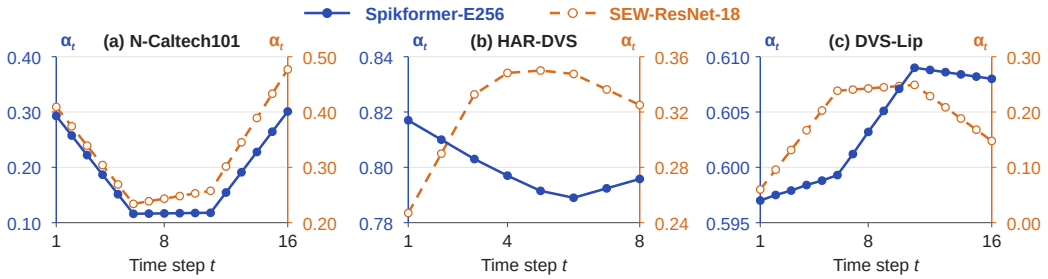

Figure 6: Learned order schedules α_t on three event-based datasets. Solid blue lines show Spikformer-E256 (left axis) and dashed orange lines show SEW-ResNet-18 (right axis). The two backbones use separate vertical scales, so the size of the variation should be read from the axis values rather than from the slope.

4.6 Efficiency and Energy

We measure inference cost on an NVIDIA RTX 3090 GPU and a Lynxi HP100 neuromorphic accelerator with the N-Caltech101 models (Table 5). We use a batch size of 32 on the GPU and 1 on HP100 and report the mean and standard deviation over five runs. Dynamic energy per sample is $E_{\text{dyn}} = (P_{\text{active}} - P_{\text{idle}})\, t/B$, where t is the batch latency and B the batch size; because the batch sizes differ, energies should not be compared across devices. On HP100, *VO*-LIF runs at the same latency as f-LIF, and its dynamic energy is 1.18% and 3.56% higher for SEW-ResNet-18 and Spikformer-E256, which is within the run-to-run standard deviation, while accuracy improves by 2.41 and 5.55 points on HP100 (2.93 and 5.67 points on the GPU). On the GPU, *VO*-LIF uses 1.63% and 2.97% less dynamic energy per sample than f-LIF, with 3.96% and 0.07% lower throughput. Both fractional neurons cost more than LIF, mainly because they keep the full membrane history. On DVS-Lip with SEW-ResNet-18, the overall firing rate over all spiking neurons falls from 8.60% for LIF to 5.05% for f-LIF and 5.14% for *VO*-LIF, so the time-varying order adds almost no spiking activity (Appendix K).

Table 5: Inference cost of N-Caltech101 models on an RTX 3090 GPU (batch 32) and the HP100 neuromorphic accelerator (batch 1). Accuracies come from a single training run of each model and therefore differ from the five-seed means in Table 1. Energy is dynamic energy per sample. Cost values are mean $\pm$ standard deviation over five runs.

		RTX 3090			HP100		
Backbone	Neuron	Acc. (%)	Samples/s	mJ/sample	Acc. (%)	Latency (ms)	mJ/sample
SEW-ResNet-18	LIF	68.89	733.58±2.84	354.88±6.06	68.60	113.58±0.01	148.00±7.77
	f-LIF	71.35	710.46±1.13	398.28±4.63	71.99	116.32±0.01	150.32±3.95
	VO-LIF	**74.28**	682.36±0.93	391.79±2.59	**74.40**	116.33±0.01	152.09±1.99
Spikformer-E256	LIF	71.24	602.02±1.33	434.00±1.57	71.53	104.40±0.02	78.45±1.95
	f-LIF	73.26	578.33±0.33	489.02±3.10	73.21	119.81±0.01	92.88±4.86
	VO-LIF	**78.93**	577.91±1.56	474.49±3.10	**78.76**	119.82±0.01	96.19±5.88

5 Conclusion

We introduced *VO*-SNN, whose *VO*-LIF neurons recompute their history kernel at every step from an order produced by a lightweight causal controller that reads the current input and the accumulated membrane memory. Discretized with the Grünwald–Letnikov scheme, the resulting charge–spike–reset update recovers LIF and f-LIF neurons as special cases, is provably causal and bounded under an explicit step-size condition, and yields non-stationary memory kernels that no fixed order can reproduce. *VO*-LIF neurons serve as drop-in replacements for LIF neurons in existing SNNs. Across event-based recognition, event-based object detection, graph-spike node classification, and neural response prediction from mouse visual cortex, *VO*-SNN consistently improves over LIF and f-SNN baselines, with energy comparable to f-SNNs on neuromorphic hardware and almost no additional spiking activity. These results suggest that adaptive neuronal memory is a simple and effective ingredient for temporal processing in SNNs. Limitations are discussed in Appendix L.

LLM USAGE STATEMENT

Large language models were used only to polish the writing of this paper, for example to improve grammar and wording. They were not used for research ideation, method design, implementation, experiments, data analysis, or generating results. All content was written, checked, and approved by the authors, who take full responsibility for it.

REFERENCES

Thomas J Anastasio. The fractional-order dynamics of brainstem vestibulo-oculomotor neurons. *Biological cybernetics*, 72(1):69–79, 1994.

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*, volume 31, 2018.

Jan Benda. Neural adaptation. *Current Biology*, 31(3):R110–R116, 2021.

György Buzsáki. *The brain from inside out*. Oxford University Press, 2019.

Michele Caputo. Linear models of dissipation whose q is almost frequency independent—ii. *Geophysical journal international*, 13(5):529–539, 1967.

Jiejie Chen, Zhigang Zeng, and Ping Jiang. Global mittag-leffler stability and synchronization of memristor-based fractional-order neural networks. *Neural Networks*, 51:1–8, 2014.

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.

Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020.

Yabin Deng, Bijing Liu, Zenan Huang, Xiaojie Liu, Shan He, Qiuhong Li, and Donghui Guo. Fractional spiking neuron: Fractional leaky integrate-and-fire circuit described with dendritic fractal model. *IEEE Transactions on Biomedical Circuits and Systems*, 16(6):1375–1386, 2022.

Alain Destexhe, Michael Rudolph, and Denis Paré. The high-conductance state of neocortical neurons in vivo. *Nature Reviews Neuroscience*, 4(9):739–751, 2003. doi: 10.1038/nrn1198.

Kai Diethelm. *The analysis of fractional differential equations: an application-oriented exposition using differential operators of Caputo type*, volume 2004 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-14574-2.

Kai Diethelm, Neville J Ford, and Alan D Freed. Detailed error analysis for a fractional adams method. *Numer. Algorithms*, 36:31–52, 2004.

Jianhao Ding, Zhaofei Yu, Jian K. Liu, and Tiejun Huang. Neuromorphic computing paradigms enhance robustness through spiking neural networks. *Nature Communications*, 16(1):10175, 2025. doi: 10.1038/s41467-025-65197-x.

Zhixia Ding, Yi Shen, and Leimin Wang. Global mittag-leffler synchronization of fractional-order neural networks with discontinuous activations. *Neural Networks*, 73:77–85, 2016.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21056–21069, 2021.

Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023a.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53674–53687, 2023b.

Shibo Feng, Wanjin Feng, Xingyu Gao, Peilin Zhao, and Zhiqi Shen. TS-LIF: A temporal segment spiking neuron network for time series forecasting. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rDe9yQQYKt.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

Chengjie Ge, Yufeng Peng, Zihao Li, Qiyu Kang, Xueyang Fu, Xuhao Li, Qixin Zhang, Junhao Ren, and Zheng-Jun Zha. Fractional-order spiking neural network. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=NJhBSLJ0nL.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Gail Gilboa, Ronen Chen, and Naama Brenner. History-dependent multiple-time-scale dynamics in a single-neuron model. *Journal of Neuroscience*, 25(28):6479–6489, 2005.

Go Eun Ha and Eunji Cheong. Spike frequency adaptation in neurons of the central nervous system. *Experimental neurobiology*, 26(4):179–185, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

Yulong Huang, Xiaopeng Lin, Hongwei Ren, Haotian Fu, Yue Zhou, Zunchang Liu, Biao Pan, and Bojun Cheng. CLIF: Complementary leaky integrate-and-fire neuron for spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19949–19972. PMLR, 2024. URL https://proceedings.mlr.press/v235/huang24n.html.

Murat Isik, Karn Tiwari, Muhammed Burak Eryilmaz, and I Can Dikmen. Accelerating sensor fusion in neuromorphic computing: A case study on loihi-2. In *2024 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7. IEEE, 2024.

Qiyu Kang, Kai Zhao, Qinxu Ding, Feng Ji, Xuhao Li, Wenfei Liang, Yang Song, and Wee Peng Tay. Unleashing the potential of fractional calculus in graph neural networks with FROND. In *Proc. International Conference on Learning Representations*, Vienna, Austria, 2024.

Eva Kaslik and Seenith Sivasundaram. Nonlinear dynamics and chaos in fractional-order neural networks. *Neural networks*, 32:245–256, 2012.

Shujaat Khan, Jawwad Ahmad, Imran Naseem, and Muhammad Moinuddin. A novel fractional gradient-based learning algorithm for recurrent neural networks. *Circuits, Systems, and Signal Processing*, 37(2):593–612, 2018.

Christof Koch, Moshe Rapp, and Idan Segev. A brief history of time (constants). *Cerebral cortex*, 6 (2):93–101, 1996.

Giancarlo La Camera, Alexander Rauch, David Thurbon, Hans-R Lüscher, Walter Senn, and Stefano Fusi. Multiple time scales of temporal response in pyramidal and fast spiking cortical neurons. *Journal of neurophysiology*, 96(6):3448–3464, 2006.

Carl F Lorenzo and Tom T Hartley. Variable order and distributed order fractional operators. *Nonlinear dynamics*, 29(1–4):57–98, 2002.

Brian N Lundstrom, Matthew H Higgs, William J Spain, and Adrienne L Fairhall. Fractional differentiation by neocortical pyramidal neurons. *Nature neuroscience*, 11(11):1335–1342, 2008.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.

Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *Mining and Learning with Graphs Workshop*, 2012.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Gabriel Nobis, Maximilian Springenberg, Marco Aversa, Michael Detzel, Rembert Daems, Roderick Murray-Smith, Shinichi Nakajima, Sebastian Lapuschkin, Stefano Ermon, Tolga Birdal, et al. Generative fractional diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 25469–25509, 2024.

Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

Guofei Pang, Lu Lu, and George Em Karniadakis. fpinns: Fractional physics-informed neural networks. *SIAM J. Sci. Comput.*, 41(4):A2603–A2626, 2019.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16639–16652, 2020.

Igor Podlubny. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, volume 198 of *Mathematics in Science and Engineering*. Academic Press, 1999. ISBN 9780125588409.

Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, pp. 3857–3866, 2019.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

Ran Rubin, L. F. Abbott, and Haim Sompolinsky. Balanced excitation and inhibition are required for high-capacity, noise-robust neuronal selectivity. *Proceedings of the National Academy of Sciences*, 114(44):E9366–E9375, 2017. doi: 10.1073/pnas.1705841114.

Stefan G. Samko and Bertram Ross. Integration and differentiation to a variable fractional order. *Integral Transforms and Special Functions*, 1(4):277–300, 1993. doi: 10.1080/10652469308819027.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS 2018*, 2018. URL https://arxiv.org/abs/1811.05868.

Yeonjong Shin, Jérôme Darbon, and George Em Karniadakis. Accelerating gradient descent and adam via fractional gradients. *Neural Networks*, 161:185–201, 2023.

Joshua H. Siegle, Xiaoxuan Jia, Séverine Durand, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, 2021. doi: 10.1038/s41586-020-03171-x.

Richard B Stein. Some models of neuronal variability. *Biophysical journal*, 7(1):37–68, 1967.

Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6555–6565, 2023.

OI Tacha, JM Munoz-Pacheco, E Zambrano-Serrano, IN Stouboulos, and V-T Pham. Determining the chaotic behavior in a fractional-order finance system with negative parameters. *Nonlinear Dynamics*, 94(2):1303–1317, 2018.

Ganchao Tan, Yang Wang, Han Han, Yang Cao, Feng Wu, and Zheng-Jun Zha. Multi-grained spatio-temporal features perceived network for event-based lip-reading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20094–20103, 2022.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural networks*, 111:47–63, 2019.

Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Aaron Szafer, Nicholas Cain, Hongkui Zeng, Michael Hawrylycz, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature communications*, 9(1):709, 2018.

Wondimu Teka, Toma M Marinov, and Fidel Santamaria. Neuronal spike timing adaptation described with a fractional leaky integrate-and-fire model. *PLoS computational biology*, 10(3):e1003526, 2014.

Anton Trushechkin. Long-term behaviour in an exactly solvable model of pure decoherence and the problem of markovian embedding. *Mathematics*, 12(1):1, 2024. doi: 10.3390/math12010001.

Nachum Ulanovsky, Liora Las, Dina Farkas, and Israel Nelken. Multiple time scales of adaptation in auditory cortex neurons. *Journal of Neuroscience*, 24(46):10440–10453, 2004.

Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. Hardvs: Revisiting human activity recognition with dynamic vision sensors. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 5615–5623, 2024.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Changjin Xu, Maoxin Liao, Peiluan Li, Ying Guo, Qimei Xiao, and Shuai Yuan. Influence of multiple time delays on bifurcation of fractional-order neural networks. *Applied Mathematics and Computation*, 361:565–582, 2019.

Ying Yang, Yong He, Yong Wang, and Min Wu. Stability analysis of fractional-order neural networks: an lmi approach. *Neurocomputing*, 285:82–93, 2018.

Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10221–10230, 2021.

Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9393–9410, 2023.

Chengting Yu, Zheming Gu, Da Li, Gaoang Wang, Aili Wang, and Erping Li. Stsc-snn: Spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks. *Frontiers in Neuroscience*, 16:1079357, 2022.

Jiqing Zhang, Malu Zhang, Yuanchen Wang, Qianhui Liu, Baocai Yin, Haizhou Li, and Xin Yang. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE Transactions on Image Processing*, 34:1009–1021, 2025. doi: 10.1109/TIP.2025.3533213.

Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. TC-LIF: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16838–16847, 2024.

Shuo Zhang, Lu Liu, Chunhua Wang, Xiaomeng Zhang, and Rong Ma. Multistability and global attractivity for fractional-order spiking neural networks. *Applied Mathematics and Computation*, 508:129617, 2026.

Han Zhao, Xu Yang, Cheng Deng, and Junchi Yan. Dynamic reactive spiking graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16970–16978, 2024. doi: 10.1609/aaai.v38i15.29640.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

Zulun Zhu, Jiaying Peng, Jintang Li, Liang Chen, Qi Yu, and Siqiang Luo. Spiking graph convolutional networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2434–2440, 2022. doi: 10.24963/ijcai.2022/338.

A More Related Work

A.1 Fractional Calculus in Deep Learning

Beyond neuron modeling, fractional calculus has found other applications in deep learning. Fractional operators have been used to enhance continuous neural networks, for example fractional graph neural networks that improve performance and robustness on graph-structured data (Kang et al., 2024), and fractional diffusion processes in generative models that promote sample diversity (Nobis et al., 2024). Physics-informed neural networks have been extended to fractional partial differential equations, treating the network as a function approximator for the solution of a given physical law (Pang et al., 2019). A distinct line of work uses fractional derivatives to define new gradient operators for training, replacing the integer-order update rule of gradient descent with a fractional counterpart (Khan et al., 2018; Shin et al., 2023). Our work differs from these approaches: *VO*-SNN neither solves an external equation nor changes the optimizer, but embeds variable-order fractional dynamics into the spiking neuron itself.

A.2 Variable-Order Fractional Operators

Variable-order operators generalize fractional calculus by letting the order depend on time or on other variables (Samko & Ross, 1993; Lorenzo & Hartley, 2002). They have been used to model physical systems whose memory characteristics evolve, such as materials whose relaxation behavior changes with their state. Earlier studies that combine fractional dynamics with neural networks mainly analyze the stability and bifurcation of fixed-order network models (Anastasio, 1994; Kaslik & Sivasundaram, 2012; Yang et al., 2018; Xu et al., 2019). *VO*-SNN instead uses a learned, time-varying order inside deep, trainable SNNs.

A.3 Event Cameras

Event cameras are bio-inspired vision sensors that operate asynchronously: each pixel independently monitors its brightness and emits an event (x, y, t, p) when the change in log intensity exceeds a threshold, where (x, y) is the pixel location, t is a timestamp with microsecond resolution, and $p \in \{+1, -1\}$ is the polarity of the change (Gallego et al., 2022). This sensing mechanism provides high temporal resolution, high dynamic range, negligible motion blur, and sparse output (Gallego et al., 2022), making event cameras suitable for high-speed and low-latency perception. Their output has been used for optical flow and visual odometry (Gallego et al., 2022), intensity reconstruction (Rebecq et al., 2019), and recognition with specialized SNNs (Yu et al., 2022), and it can be processed efficiently on neuromorphic hardware (Isik et al., 2024). In our experiments, event streams are converted into frames and fed to *VO*-SNN directly, where the time-varying order lets the network adapt its memory to the changing temporal structure of the events.

B Preliminaries

B.1 Fractional and Variable-Order Derivatives

Fractional calculus generalizes differentiation to non-integer orders and provides a framework for modeling temporal memory and non-local dependence (Podlubny, 1999). For a function $y(t)$ on $[0, T]$, the Caputo fractional derivative of order $\alpha \in (0, 1)$ is (Caputo, 1967; Diethelm, 2010)

$$D^{\alpha} y(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha}\, y'(s)\, \mathrm{d}s, \tag{12}$$

which weights the entire history of y' with a power-law kernel and recovers the first-order derivative as $\alpha \to 1$. Letting the order vary with time gives a variable-order Caputo derivative,

$$D^{\alpha(t)} y(t) = \frac{1}{\Gamma(1-\alpha(t))} \int_0^t (t-s)^{-\alpha(t)}\, y'(s)\, \mathrm{d}s, \tag{13}$$

whose kernel changes with t (Samko & Ross, 1993; Lorenzo & Hartley, 2002). Different variable-order definitions need not coincide, which is why we define *VO*-LIF neurons directly through the discrete operator in Equation (4).

B.2 Grünwald–Letnikov Discretization

The GL formula approximates a fractional derivative by a weighted sum over past values (Tacha et al., 2018; Podlubny, 1999):

$$D^{\alpha}y(t) \approx \frac{1}{h^{\alpha}} \sum_{j=0}^{\lfloor t/h \rfloor} (-1)^j \binom{\alpha}{j} y(t-jh), \tag{14}$$

where h is the step size. The coefficients $c_j(\alpha) = (-1)^j\binom{\alpha}{j}$ satisfy $c_j(\alpha)/c_{j-1}(\alpha) = -(\alpha - j + 1)/j = 1-(1+\alpha)/j$, which gives the recursion in Equation (3) and avoids evaluating large binomial coefficients directly.

B.3 Fractional Spiking Neurons and f-SNNs

Existing SNNs mostly use discretized first-order dynamics, such as IF and LIF neurons and their variants (Stein, 1967; Gerstner et al., 2014; Koch et al., 1996; Bellec et al., 2018; Benda, 2021; Zhang et al., 2025). The f-SNN framework (Ge et al., 2026) replaces the first-order derivative with a fractional derivative, giving the f-LIF neuron in Equation (1), and discretizes it with an Adams–Bashforth–Moulton predictor (Diethelm et al., 2004). In this work, all LIF and f-LIF baseline results are obtained with the official open-source f-SNN implementation, using its built-in Grünwald–Letnikov solver for f-LIF; the f-LIF baselines thus use the same GL discretization as *VO*-SNN, so that differences between the two come from the time-varying order rather than from the numerical scheme. Training in all cases uses surrogate gradients (Neftci et al., 2019; Wu et al., 2018).

B.4 Discretized IF, LIF, and GL-Based f-LIF Neurons

For reference, we list the discrete updates of the baseline neurons on the grid $t_k = kh$ used in Section 3.2. All neurons share the spike and reset steps in Equations (7) and (8), and differ only in how the pre-reset charge Z_k is computed from the post-reset history $U_0, \dots, U_{k-1}$ and the input X_k.

IF neuron. The integrate-and-fire neuron has no leak, $\tau\,\mathrm{d}U/\mathrm{d}t = X(t)$, and its forward-Euler update is

$$Z_k = U_{k-1} + \frac{h}{\tau} X_k. \tag{15}$$

LIF neuron. The LIF neuron, $\tau\,\mathrm{d}U/\mathrm{d}t = -U + X(t)$, is discretized in the same way:

$$Z_k = U_{k-1} + \frac{h}{\tau}\,(X_k - U_{k-1}). \tag{16}$$

GL-based f-LIF neuron. For the f-LIF neuron in Equation (1) with a constant order $\bar{\alpha} \in (0,1)$, applying the GL operator in Equation (4) with $\alpha_k \equiv \bar{\alpha}$ and evaluating the leak and input at the previous post-reset state gives

$$Z_k = U_0 + \sum_{j=1}^{k} w_j(\bar{\alpha})\,(U_{k-j} - U_0) + \frac{h^{\bar{\alpha}}}{\tau}\,(X_k - U_{k-1}), \tag{17}$$

where $w_j(\bar{\alpha}) = -c_j(\bar{\alpha})$ are the GL weights in Equation (3). The history weights are the same at every step, so the memory kernel is stationary. At $\bar{\alpha} = 1$, only $w_1(1) = 1$ is nonzero and Equation (17) reduces to the LIF update in Equation (16). Analogously, removing the leak term gives the fractional IF update $Z_k = U_0 + \sum_{j=1}^{k} w_j(\bar{\alpha})(U_{k-j} - U_0) + \frac{h^{\bar{\alpha}}}{\tau} X_k$, and replacing $\bar{\alpha}$ by the time-varying order α_k gives the VO-IF neuron used in Section 4.4. The *VO*-LIF update in Equations (5)–(6) is obtained from Equation (17) in the same way.

C Proofs for Section 3.4

Throughout, $w_j(\alpha) = -c_j(\alpha)$ with c_j from Equation (3), and $q_m(\alpha) = \prod_{r=1}^{m}(1-\alpha/r)$ with $q_0 = 1$.

Assumption 1. *The step size h and time constant τ are positive; $\alpha_k \in [\alpha_-, \alpha_+] \subset (0,1)$ for all k; $\|X_k\|_\infty \le X_*$; $\vartheta > 0$; and for the hard reset, $|U_{\mathrm{reset}}| \le \vartheta$. Let $A = \max_{\alpha\in[\alpha_-,\alpha_+]} h^{\alpha}/\tau$.*

Discussion. Propositions 2 and 4 make the picture in Figure 1 precise. When the controller raises α_k under strong drive, the memory concentrates on recent states; when it lowers α_k during adaptation, distant states gain relative weight; and because the kernel changes with k, similar inputs can be integrated differently depending on the preceding history. The condition in Proposition 3 is sufficient rather than necessary. At $\alpha = 1$ it reduces to the forward-Euler stability condition $h/\tau < 2$, and with $h = 1$ it requires $\alpha_- > 1/(2\tau)$; without such a condition, bounded orders and inputs do not guarantee bounded states (see the counterexample below). Our experiments also use broader order ranges, for which we observe stable training empirically. None of these results by itself implies higher accuracy.

Lemma 1. *For $0 < \alpha < 1$ and $j \geq 1$,*

$$w_j(\alpha) = \frac{\alpha}{j} \prod_{r=1}^{j-1} \Big(1 - \frac{\alpha}{r}\Big) = \frac{\alpha\,\Gamma(j-\alpha)}{\Gamma(1-\alpha)\Gamma(j+1)} > 0, \tag{18}$$

$$\sum_{j=1}^{m} w_j(\alpha) = 1 - q_m(\alpha), \qquad \sum_{j=1}^{\infty} w_j(\alpha) = 1.$$

Proof. From Equation (3), $c_1 = -\alpha$, and for $j \geq 2$ the factor $(j - 1 - \alpha)/j$ is positive, so all c_j with $j \geq 1$ are negative and the product form follows. The Gamma form follows from $\Gamma(z+1) = z\Gamma(z)$. Since $w_j = (\alpha/j)q_{j-1} = q_{j-1} - q_j$, the partial sums telescope to $1 - q_m$. Finally, $\log q_m \leq -\alpha \sum_{r=1}^{m} 1/r \to -\infty$, so $q_m \to 0$. □

By Lemma 1, the charge in Equation (6) can be written as

$$Z_k = q_k(\alpha_k)U_0 + \sum_{j=1}^{k} w_j(\alpha_k)U_{k-j} + a_k(X_k - U_{k-1}), \qquad a_k = h^{\alpha_k}/\tau, \tag{19}$$

where the coefficients $q_k(\alpha_k)$ and $w_j(\alpha_k)$ are nonnegative and sum to one.

Proof of Proposition 1. (i) If $\alpha_k \equiv \bar{\alpha}$, every step uses the weights $w_j(\bar{\alpha})$ and the input scale $h^{\bar{\alpha}}/\tau$. The initial states coincide; if the states coincide up to step $k-1$, the memory, charge, spike, and reset at step k coincide, and induction completes the proof. (ii) At $\alpha = 1$, the recursion gives $c_1 = -1$ and $c_2 = (1 - 2/2)c_1 = 0$, hence $c_j = 0$ for $j \geq 2$. Equation (4) becomes $(Z_k - U_{k-1})/h$, the U_0 terms cancel, and the charge is $U_{k-1} + \frac{h}{\tau}(X_k - U_{k-1})$ for any U_0. (iii) At step 1, the controller reads t_1, X_1, and M_0, so α_1 and the step-1 quantities agree. If all quantities agree up to step $k-1$, the controller at step k reads the same t_k, X_k, and M_{k-1}, and the memory uses only $U_0, \ldots, U_{k-1}$, so all step-k quantities agree. Induction up to K completes the proof.

Proof of Proposition 2. For $\ell \geq 2$, $w_\ell/w_{\ell-1} = c_\ell/c_{\ell-1} = (\ell-1-\alpha)/\ell$, so $w_j/w_r = \prod_{\ell=r+1}^{j}(\ell - 1 - \alpha)/\ell$ for $1 \leq r < j$. Taking the logarithm and differentiating gives the first identity; every denominator $\ell - 1 - \alpha \geq 1 - \alpha$ is positive. The asymptotic form follows from Lemma 1 and the ratio asymptotics $\Gamma(j-\alpha)/\Gamma(j+1) = j^{-1-\alpha}(1 + O(j^{-1}))$. The same argument shows that, for a finite window L and $1 \leq m < L$, the normalized distant mass $\sum_{j=m+1}^{L} w_j / \sum_{j=1}^{L} w_j$ strictly increases as α decreases: its derivative has the sign of $\sum_{j>m} \sum_{r\leq m} w_j w_r\, \partial_\alpha \log(w_j/w_r) < 0$. Note that this concerns relative weights; individual weights need not increase, e.g., $w_1(\alpha) = \alpha$.

Proof of Proposition 3. Let $a(\alpha) = h^\alpha/\tau$. Then $|\alpha - a| + 1 - \alpha = 1 - \min\{a,\, 2\alpha - a\}$, which is smaller than one if and only if $0 < a < 2\alpha$; by continuity on the compact interval, $\rho < 1$. Both resets satisfy $|U_k| \leq |Z_k|$ coordinatewise: without a spike, $U_k = Z_k$; with a spike, $Z_k \geq \vartheta > 0$, so $0 \leq Z_k - \vartheta < Z_k$ for the soft reset and $|U_{\text{reset}}| \leq \vartheta \leq Z_k$ for the hard reset. Using $w_1(\alpha_k) = \alpha_k$, rewrite Equation (19) as

$$Z_k = q_k U_0 + a_k X_k + (\alpha_k - a_k)U_{k-1} + \sum_{j=2}^{k} w_j(\alpha_k)U_{k-j}.$$

Let $H_{k-1} = \max_{0\le r\le k-1} \|U_r\|_\infty$. Since $q_k \le 1$ and $\sum_{j=2}^{k} w_j \le 1-\alpha_k$,

$$\|Z_k\|_\infty \le \|U_0\|_\infty + AX_* + \rho H_{k-1}.$$

Let $B_* = (AX_* + \|U_0\|_\infty)/(1-\rho) \ge \|U_0\|_\infty$. If $H_{k-1} \le B_*$, then $\|U_k\|_\infty \le \|Z_k\|_\infty \le (1-\rho)B_* + \rho B_* = B_*$. Since $H_0 = \|U_0\|_\infty \le B_*$, induction completes the proof. The argument does not use how α_k is chosen within $[\alpha_-, \alpha_+]$.

The step-size condition cannot simply be dropped. Take $h = 1$, $\tau = 1/4$, $\alpha_k \equiv 1/2$, $\vartheta = 1$, a soft reset, $U_0 = 0$, $X_1 = 1$, and $X_k = 0$ for $k \ge 2$. Then $Z_1 = 4$ and $U_1 = 3$. For $k \ge 2$, $Z_k = -3.5\, U_{k-1} + \sum_{j=2}^{k} w_j(1/2)U_{k-j}$. If $|U_r| \le |U_{k-1}|$ for all $r < k$, then $\sum_{j\ge 2} w_j \le 1/2$ gives $|Z_k| \ge 3|U_{k-1}|$ and hence $|U_k| \ge 3|U_{k-1}| - 1 > |U_{k-1}|$. Starting from $|U_1| = 3$, the magnitude grows without bound although the order and input are bounded. Here $h^\alpha/\tau = 4 > 2\alpha$, so the condition of Proposition 3 fails; as with forward-Euler LIF at $h/\tau > 2$, the instability comes from the explicit treatment of the leak.

Proof of Proposition 4. A fixed-order kernel has $\kappa_{k,i} = w_{k-i}(\bar\alpha)$, and a fixed combination has $\kappa_{k,i} = \sum_m \omega_m w_{k-i}(\bar\alpha_m)$; both depend only on $k-i$. For the *VO*-LIF kernel, the lag-one coefficient at step k is $\kappa_{k,k-1} = w_1(\alpha_k) = \alpha_k$. A stationary kernel would require $\alpha_k = \alpha_{k'}$ for all $k, k' \ge 1$, a contradiction.

Sensitivity to the order. We finally show that the update depends smoothly on the order and that this dependence grows only logarithmically with the sequence length, which complements the gradient expressions in Appendix D.

Lemma 2. *For $0 < \alpha < 1$ and $k \ge 1$,*

$$\sum_{j=1}^{k} \big|w_j'(\alpha)\big| \le \frac{1}{\alpha} + \frac{1}{1-\alpha} + 1 + \log k. \tag{20}$$

Proof. Taking the logarithm of the product form in Lemma 1 and differentiating gives $w_j'(\alpha) = w_j(\alpha)\big[\frac{1}{\alpha} - \sum_{r=1}^{j-1} \frac{1}{r-\alpha}\big]$. For $j \le k$, the first term of the sum is at most $1/(1-\alpha)$, and for $r \ge 2$, $1/(r-\alpha) \le 1/(r-1)$, so $\sum_{r=1}^{j-1} 1/(r-\alpha) \le 1/(1-\alpha) + 1 + \log k$. Hence $|w_j'(\alpha)| \le w_j(\alpha)\big(1/\alpha + 1/(1-\alpha) + 1 + \log k\big)$, and summing over j with $\sum_{j=1}^{k} w_j(\alpha) \le 1$ (Lemma 1) completes the proof. □

Proposition 5 (Lipschitz dependence on the order)**.** *Under the conditions of Proposition 3, let B_* be the bound in Equation* (11)*, and fix the history $U_0, \dots, U_{k-1}$ and the input X_k. Write $Z_k(\alpha)$ for the charge in Equation* (6) *computed with order α. Then for all $\alpha, \alpha' \in [\alpha_-, \alpha_+]$,*

$$\|Z_k(\alpha) - Z_k(\alpha')\|_\infty \le L_k\, |\alpha - \alpha'|, \qquad L_k = 2B_*\Big(\frac{1}{\alpha_-} + \frac{1}{1-\alpha_+} + 1 + \log k\Big) + A\,|\log h|\,(X_* + B_*). \tag{21}$$

Proof. By Equation (6), $\partial Z_k/\partial\alpha = \sum_{j=1}^{k} w_j'(\alpha)(U_{k-j} - U_0) + \frac{h^\alpha \log h}{\tau}(X_k - U_{k-1})$. By Proposition 3, $\|U_{k-j} - U_0\|_\infty \le 2B_*$ and $\|X_k - U_{k-1}\|_\infty \le X_* + B_*$, and $h^\alpha/\tau \le A$. Lemma 2 with $1/\alpha \le 1/\alpha_-$ and $1/(1-\alpha) \le 1/(1-\alpha_+)$ bounds the first term, and the mean value theorem on $[\alpha, \alpha']$ completes the proof. □

Proposition 5 has two consequences. First, a small change of the order produces a proportionally small change of the charge, so the order controller perturbs the dynamics smoothly rather than abruptly. Second, the sensitivity $\partial Z_k/\partial\alpha_k$ that drives the gradient of the order is bounded uniformly over admissible order sequences and grows only as $\log k$, so the order receives well-scaled gradients even for long sequences. For $h = 1$, the second term of L_k vanishes.

D GRADIENTS WITH RESPECT TO THE ORDER

Differentiating Equation (3) gives $c'_0 = 0$ and $c'_j = (1 - \frac{1+\alpha}{j})c'_{j-1} - c_{j-1}/j$, or equivalently, from Lemma 1,

$$w'_j(\alpha) = w_j(\alpha)\Big[\frac{1}{\alpha} - \sum_{r=1}^{j-1}\frac{1}{r-\alpha}\Big], \tag{22}$$

which is bounded on $[\alpha_-, \alpha_+]$. With the history fixed, the sensitivity of the charge to the current order is

$$\frac{\partial Z_k}{\partial \alpha_k} = \sum_{j=1}^{k} w'_j(\alpha_k)(U_{k-j} - U_0) + \frac{h^{\alpha_k}\log h}{\tau}(X_k - U_{k-1}). \tag{23}$$

For $h = 1$, the second term vanishes, and at $k = 1$ the only history term is $U_0 - U_0 = 0$, so the first-step order receives no local gradient; this is a property of the scheme rather than a failure of learning. The full gradient combines these terms with the chain rule through ϕ_θ and M_{k-1}. Because the forward pass uses the hard threshold Θ, training with a smooth surrogate derivative yields a surrogate-gradient learning rule rather than the exact gradient of the hard-threshold loss.

E ALGORITHM

Algorithm 1 summarizes the forward pass of a *VO*-LIF layer. In the full network, the order α_k is computed once per sample and time step and shared by all layers; the memory summary is taken from the last spiking layer.

Algorithm 1 Forward pass of a *VO*-LIF layer with the order controller

Require: inputs $X_1, \dots, X_T$; initial state U_0; controller ϕ_θ; bounds α_-, α_+; threshold ϑ

$M_0 \leftarrow 0$
for $k = 1, \dots, T$ **do**
  $\alpha_k \leftarrow \alpha_- + (\alpha_+ - \alpha_-)\,\sigma\big(b_k + s\tanh(\phi_\theta(t_k, s_X(X_k), s_M(M_{k-1})))\big)$
  compute $w_1(\alpha_k), \dots, w_k(\alpha_k)$ by the recursion in Equation (3)
  $M_k \leftarrow \sum_{j=1}^{k} w_j(\alpha_k)\,(U_{k-j} - U_0)$
  $Z_k \leftarrow U_0 + M_k + \frac{h^{\alpha_k}}{\tau}(X_k - U_{k-1})$
  $S_k \leftarrow \Theta(Z_k - \vartheta)$
  $U_k \leftarrow Z_k - \vartheta S_k$ **(soft)** or $U_k \leftarrow (1 - S_k)Z_k + S_k U_{\text{reset}}$ **(hard)**
end for
return $S_1, \dots, S_T$

F DATASETS AND EXPERIMENTAL SETTINGS

Table 6 summarizes the event-based datasets; the graph datasets are summarized in Table 7.

Table 6: Summary of the event-based datasets.

Dataset	Task	Sensor (resolution)	Classes	Scale
N-Caltech101	object recognition	ATIS (240×180)	101	8,709 samples
DVS-Lip	lip reading	iniVation DAVIS346 (128×128 crop)	100	19,871 samples
HAR-DVS	action recognition	iniVation DAVIS346 (346×260)	300	107,646 samples
Gen1	object detection	Prophesee Gen1.0 (304×240)	2	39 h, 255,781 boxes
Gen4 (1Mpx)	object detection	Prophesee Gen4.0 (1280×720)	3	15 h, 25M boxes

F.1 EVENT-BASED RECOGNITION

N-Caltech101. N-Caltech101 (Orchard et al., 2015) converts the static Caltech101 images into event streams by displaying each image on a monitor and moving an ATIS sensor mounted on a pan-tilt unit, which mimics saccadic eye movements. It contains 100 object categories and a background class. Since no official split exists, we use an 80%/20% random split into training and test sets.

DVS-Lip. DVS-Lip (Tan et al., 2022) is an event-based lip-reading dataset recorded with a DAVIS346 camera, with the lip region cropped to 128×128. It contains 100 words selected from the LRW benchmark, including many words that are easily confused, so recognition depends on subtle and fast lip movements. We use the official training and test partitions.

HAR-DVS. HAR-DVS (Wang et al., 2024) is a large-scale human activity recognition dataset recorded with a DAVIS346 camera at 346×260 resolution. It contains 107,646 event sequences of 300 daily activities, each lasting about 5 to 10 seconds, recorded under varying viewpoints, illumination, motion speeds, backgrounds, occlusions, and distances. We use the official training and test partitions.

Settings. We use SEW-ResNet-18 (Fang et al., 2021) and Spikformer (Zhou et al., 2023) backbones and vary only the spiking neurons: LIF neurons from SpikingJelly (Fang et al., 2023a), GL-discretized f-LIF neurons, and *VO*-LIF neurons. Raw event streams are converted into frames with the SpikingJelly preprocessing pipeline and resized to 128×128. We use $T = 16$ time steps for N-Caltech101 and DVS-Lip, and $T = 8$ for HAR-DVS. SEW-ResNet-18 models are trained for 100 epochs and Spikformer-E256 models for 500 epochs. All models use a batch size of 16 and the Adam optimizer, and results are averaged over five random seeds.

F.2 Event-Based Object Detection

Gen1. The Gen1 Automotive Detection dataset (de Tournemire et al., 2020) contains 39 hours of driving recordings captured with a Prophesee Gen1.0 (ATIS) camera at a resolution of 304×240, covering urban, highway, suburban, and countryside scenes. It provides 228,123 manually annotated bounding boxes of cars and 27,658 of pedestrians.

Gen4 (1Mpx). The 1Mpx dataset (Perot et al., 2020), also called Gen4, contains about 15 hours of driving recordings captured with a Prophesee Gen4.0 camera at a resolution of 1280×720. Its 25 million bounding boxes are annotated semi-automatically at 60 Hz; following the official evaluation, we use the car, pedestrian, and two-wheeler classes. Its higher resolution and denser annotations make it considerably more challenging than Gen1.

Settings. We use the EMS-YOLO detector (Su et al., 2023), a directly trained spiking detector, and replace its spiking neurons with each neuron type while keeping the architecture and training pipeline unchanged. The event representation, number of time steps, input resolution, training schedule, and COCO-style evaluation (mAP@0.5:0.95 and mAP@0.5) follow the EMS-YOLO paper (Su et al., 2023).

F.3 Graph-Spike Node Classification

We evaluate node classification with the spiking graph neural networks SGCN (Zhu et al., 2022) and DRSGNN (Zhao et al., 2024) on eight datasets (Table 7): the citation graphs Cora (McCallum et al., 2000), Citeseer (Sen et al., 2008), and PubMed (Namata et al., 2012); the Amazon co-purchase graphs Photo and Computers (Shchur et al., 2018); the large-scale citation graph ogbn-arxiv (Hu et al., 2020); and the heterophilous Wikipedia graphs squirrel and chameleon (Rozemberczki et al., 2021; Pei et al., 2020), in which connected nodes often belong to different classes. Cora, Citeseer, and PubMed use their public splits and ogbn-arxiv uses its official split; Photo and Computers use a fixed stratified 60/20/20 split; and squirrel and chameleon use split 0 of Geom-GCN (Pei et al., 2020). Node features are encoded as spike trains, and the test accuracy of each seed is averaged over eight random encodings; we report results over five training seeds. Only the spiking neurons are replaced.

F.4 Neural Response Prediction

Data source and recording sessions. We use extracellular spike recordings from the Allen Brain Observatory Visual Coding Neuropixels dataset (Siegle et al., 2021). The public Neurodata Without Borders (NWB) files, available through the AllenSDK, contain spike-sorted unit spike times, unit-quality metrics, recording-location metadata, and stimulus-presentation timestamps. These are electrophysiological recordings rather than two-photon calcium signals. We use responses from the primary visual cortex (VISp) in three sessions (732592105, 754312389, and 791319847), under

Table 7: Statistics of the graph datasets.

Dataset	Nodes	Classes	Features	Edges
Cora	2,708	7	1,433	10,556
Citeseer	3,327	6	3,703	9,104
PubMed	19,717	3	500	88,648
Photo	7,650	8	745	238,162
Computers	13,752	10	767	491,722
ogbn-arxiv	169,343	40	128	1,166,243
squirrel	5,201	5	2,089	217,073
chameleon	2,277	5	2,325	36,101

drifting gratings (DG), natural movies, and natural scenes. The task is stimulus-to-response regression: the network receives a representation of the visual stimulus and predicts the recorded neural response over time, rather than a stimulus class label.

Unit selection and temporal alignment. We identify VISp units using the location of their peak recording channel. We retain units with firing rate above 0.1 Hz, ISI violations below 0.5, amplitude cutoff below 0.1, and presence ratio above 0.9, rank the retained units by firing rate, and keep at most 128 per session. This yields 113, 112, and 106 units in the three sessions, respectively. For a stimulus segment starting at time t_0, we count each unit's spikes in 100 consecutive 20-ms bins, producing a 2-s response sequence. A lag $\delta \in \{0, 40, 80, 100\}$ ms shifts the response window to start at $t_0 + \delta$, while the stimulus input remains aligned to t_0. The target $Y \in \mathbb{R}^{100\times N}$ therefore contains spike counts per bin for the N selected units, not raw voltage traces or firing rates in Hz. The single-trial datasets contain 512 segments per session for DG and scenes, and 180 for movies.

Stimulus representation. For DG, the input is a seven-dimensional descriptor containing $\sin(2\theta)$, $\cos(2\theta)$, $\log(1 + f_{\text{temporal}})$, $\log(1 + f_{\text{spatial}})$, contrast, $\sin(\varphi)$, and $\cos(\varphi)$, where θ and φ are orientation and phase in radians. The descriptor is repeated over the 100 time steps. Natural scenes are represented by the scene ID at the segment onset, repeated across the sequence. For movies, the frame ID active at each 20-ms input-bin start is read from the presentation timestamps, forming a time-varying ID sequence. Scene and movie IDs pass through a jointly learned 64-dimensional embedding. Thus, the models use stimulus metadata and identity embeddings, not raw image pixels or a pretrained visual encoder.

Single-trial and PSTH targets. In single-trial mode, each sample retains one recorded response segment. In PSTH mode, we group repetitions with identical grating parameters, scene IDs, or movie frame-ID sequences, and average their spike counts separately for every unit and time bin. Groups with fewer than two repetitions are excluded; each retained group yields one input sequence and one trial-averaged peri-stimulus time histogram (PSTH). No additional temporal smoothing is applied. PSTH targets reduce trial-to-trial variability, whereas single-trial targets retain it. The three stimulus classes, three sessions, four lags, and two response modes give $3 \times 3 \times 4 \times 2 = 72$ settings, each trained and evaluated separately.

Splitting and target scaling. Approximately 80% of the preprocessed samples are randomly assigned to training and the remaining 20% to testing. In PSTH mode, averaging precedes splitting, so the partition is over stimulus groups rather than individual repetitions. In single-trial mode, the partition is over response segments; repetitions of the same stimulus identity may occur in both partitions; since all neurons share the same split, this affects all models equally. The saved split is reused across models and training seeds. For each setting, targets are divided by one scalar, the 99th percentile of all target values across samples, bins, and units. A nonpositive or nonfinite scale is replaced by one, and values are not clipped. This target scaling is not applied to stimulus inputs.

Network input, readout, and evaluation. After stimulus encoding, a minibatch has shape $n_{\text{b}} \times 100 \times d$, where n_{b} is the batch size, with $d = 7$ for DG and $d = 64$ for scenes and movies. The direct SNN regressor applies a linear projection to 128 hidden units, a spiking-neuron layer, a second 128-unit linear/spiking layer, and a linear readout with N outputs at each time step. The readout is continuous, allowing comparison with normalized spike counts or PSTH values rather than requiring

binary output spikes. Sequences are processed in time-major form, and neuronal states are reset between minibatches so that separate samples do not share membrane histories. Training minimizes mean squared error over samples, time bins, and output units; test MSE is computed on the same normalized target scale. Results are averaged over five training seeds for each setting, followed by aggregation over settings as described in Table 4. Per-setting results are given in Appendix H.

G Learned Orders on Graph-Spike Node Classification

Figure 7 shows the learned order schedules on Computers, Photo, and PubMed with SGCN and DRSGNN, complementing the classification results in Section 4.3. Both backbones learn non-constant schedules, but their temporal profiles differ. On Computers, SGCN follows a U-shaped trajectory, decreasing from about 0.579 to 0.556 before rising to 0.585; DRSGNN decreases from about 0.428 to 0.385, with a slight recovery near the end. On Photo, the SGCN order first rises and then falls, whereas the DRSGNN order decreases throughout the sequence. On PubMed, SGCN decreases rapidly before stabilizing near 0.75, while DRSGNN first increases and then decreases.

These contrasting profiles extend the observations on event-based recognition in Figure 6 to graph-spike processing. The learned schedules depend on both the dataset and the backbone, rather than converging to a common constant order. Through the history weights in Equation (5), these temporal changes vary the relative contribution of recent and distant membrane states during node representation learning.

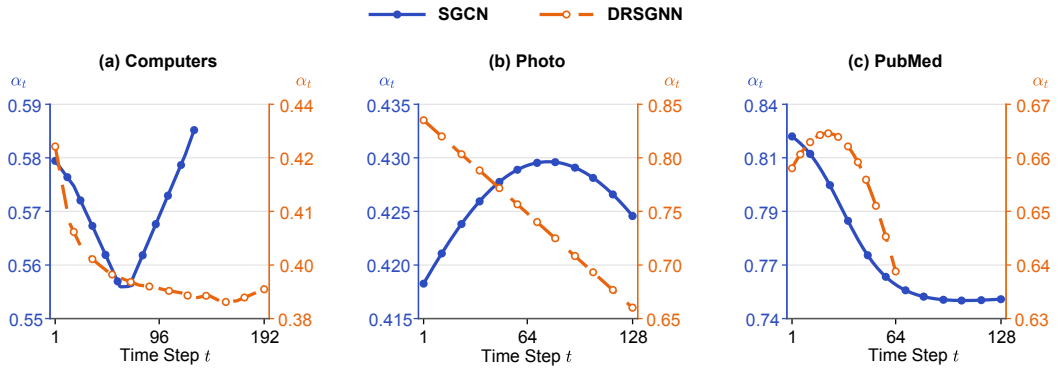

Figure 7: Learned order schedules α_t on Computers, Photo, and PubMed. Solid blue lines show SGCN (left axis), and dashed orange lines show DRSGNN (right axis). The two backbones use separate vertical scales, and each curve ends at the simulation horizon of its configuration.

H PER-SETTING RESULTS ON NEURAL RESPONSE PREDICTION

Tables 8–10 list the test MSE of all 72 settings summarized in Table 4. Each stimulus class uses three recording sessions, four stimulus–response lags of 0, 40, 80, and 100 ms (0, 2, 4, and 5 bins of 20 ms), and two response modes. Each entry is the mean over five seeds; Table 4 reports the mean and sample standard deviation of these entries within each group. A variable-order neuron attains the lowest (or tied-lowest) MSE in every setting: VO-IF in most settings and *VO*-LIF in the others, mainly PSTH settings on natural movies. The margin over the baselines is large for PSTH targets and small for single-trial targets.

Table 8: Per-setting test MSE on the Allen Neuropixels task for drifting gratings (mean over five seeds; lower is better). Best results are in bold.

Mode	Session	Lag (ms)	VO-IF	*VO*-LIF	TC-LIF	TS-LIF	IF	LIF	f-LIF
PSTH	732592105	0	**0.0147**	0.0161	0.0381	0.0481	0.0421	0.0513	0.0393
PSTH	732592105	40	**0.0123**	0.0144	0.0370	0.0447	0.0407	0.0475	0.0372
PSTH	732592105	80	**0.0132**	0.0142	0.0361	0.0462	0.0403	0.0493	0.0381
PSTH	732592105	100	**0.0150**	0.0159	0.0381	0.0526	0.0447	0.0561	0.0440
PSTH	754312389	0	**0.0186**	0.0199	0.0401	0.0505	0.0450	0.0531	0.0409
PSTH	754312389	40	**0.0203**	0.0221	0.0438	0.0540	0.0481	0.0570	0.0446
PSTH	754312389	80	**0.0188**	0.0203	0.0411	0.0521	0.0459	0.0548	0.0431
PSTH	754312389	100	**0.0194**	0.0209	0.0412	0.0551	0.0481	0.0580	0.0450
PSTH	791319847	0	**0.0171**	0.0185	0.0414	0.0582	0.0498	0.0620	0.0487
PSTH	791319847	40	**0.0153**	0.0177	0.0419	0.0524	0.0474	0.0545	0.0445
PSTH	791319847	80	**0.0147**	0.0164	0.0395	0.0518	0.0457	0.0543	0.0435
PSTH	791319847	100	**0.0159**	0.0174	0.0391	0.0503	0.0451	0.0524	0.0414
Single	732592105	0	**0.0292**	0.0295	0.0358	0.0346	0.0347	0.0374	0.0306
Single	732592105	40	**0.0285**	0.0289	0.0353	0.0340	0.0341	0.0360	0.0301
Single	732592105	80	**0.0277**	0.0278	0.0338	0.0325	0.0326	0.0346	0.0287
Single	732592105	100	**0.0288**	0.0289	0.0350	0.0338	0.0339	0.0364	0.0300
Single	754312389	0	**0.0676**	0.0682	0.0765	0.0745	0.0767	0.0823	0.0704
Single	754312389	40	**0.0662**	0.0669	0.0756	0.0737	0.0756	0.0811	0.0693
Single	754312389	80	**0.0668**	0.0673	0.0755	0.0736	0.0755	0.0809	0.0695
Single	754312389	100	**0.0666**	0.0671	0.0750	0.0733	0.0752	0.0817	0.0691
Single	791319847	0	**0.0262**	0.0266	0.0322	0.0306	0.0316	0.0341	0.0279
Single	791319847	40	**0.0251**	0.0256	0.0317	0.0299	0.0308	0.0330	0.0271
Single	791319847	80	**0.0239**	0.0242	0.0299	0.0284	0.0292	0.0312	0.0257
Single	791319847	100	**0.0231**	0.0233	0.0287	0.0272	0.0278	0.0295	0.0245

Table 9: Per-setting test MSE on the Allen Neuropixels task for natural movies (mean over five seeds; lower is better). Best results are in bold.

Mode	Session	Lag (ms)	VO-IF	*VO*-LIF	TC-LIF	TS-LIF	IF	LIF	f-LIF
PSTH	732592105	0	**0.0137**	0.0138	0.0437	0.0487	0.0451	0.0537	0.0389
PSTH	732592105	40	0.0182	**0.0180**	0.0482	0.0706	0.0650	0.0770	0.0590
PSTH	732592105	80	**0.0144**	0.0146	0.0460	0.0511	0.0469	0.0562	0.0404
PSTH	732592105	100	**0.0173**	0.0179	0.0492	0.0478	0.0445	0.0521	0.0383
PSTH	754312389	0	0.0290	**0.0286**	0.0586	0.0691	0.0634	0.0747	0.0570
PSTH	754312389	40	**0.0287**	0.0291	0.0609	0.0736	0.0678	0.0795	0.0612
PSTH	754312389	80	0.0215	**0.0213**	0.0521	0.0520	0.0488	0.0567	0.0419
PSTH	754312389	100	0.0328	**0.0317**	0.0626	0.0835	0.0765	0.0901	0.0701
PSTH	791319847	0	**0.0212**	0.0213	0.0503	0.0641	0.0586	0.0696	0.0527
PSTH	791319847	40	0.0227	**0.0226**	0.0516	0.0599	0.0557	0.0651	0.0493
PSTH	791319847	80	0.0236	**0.0226**	0.0523	0.0621	0.0573	0.0674	0.0509
PSTH	791319847	100	0.0195	**0.0191**	0.0505	0.0512	0.0476	0.0558	0.0410
Single	732592105	0	**0.0365**	0.0367	0.0549	0.0483	0.0492	0.0498	0.0429
Single	732592105	40	**0.0379**	0.0382	0.0561	0.0522	0.0525	0.0539	0.0462
Single	732592105	80	**0.0374**	0.0375	0.0559	0.0516	0.0519	0.0533	0.0457
Single	732592105	100	**0.0390**	0.0392	0.0578	0.0559	0.0554	0.0578	0.0494
Single	754312389	0	**0.0906**	0.0914	0.1108	0.1072	0.1071	0.1096	0.1002
Single	754312389	40	**0.0958**	0.0970	0.1180	0.1163	0.1153	0.1189	0.1089
Single	754312389	80	**0.0907**	0.0913	0.1112	0.1069	0.1067	0.1093	0.1000
Single	754312389	100	**0.0965**	0.0972	0.1191	0.1176	0.1168	0.1203	0.1102
Single	791319847	0	**0.0341**	0.0346	0.0531	0.0477	0.0486	0.0497	0.0425
Single	791319847	40	**0.0338**	0.0343	0.0528	0.0475	0.0485	0.0495	0.0423
Single	791319847	80	**0.0340**	0.0343	0.0531	0.0477	0.0487	0.0497	0.0425
Single	791319847	100	**0.0333**	0.0339	0.0525	0.0465	0.0476	0.0484	0.0414

Table 10: Per-setting test MSE on the Allen Neuropixels task for natural scenes (mean over five seeds; lower is better). Best results are in bold.

Mode	Session	Lag (ms)	VO-IF	*VO*-LIF	TC-LIF	TS-LIF	IF	LIF	f-LIF
PSTH	732592105	0	**0.0245**	0.0252	0.0436	0.0563	0.0468	0.0589	0.0444
PSTH	732592105	40	**0.0246**	0.0253	0.0443	0.0567	0.0473	0.0591	0.0437
PSTH	732592105	80	**0.0254**	0.0259	0.0459	0.0551	0.0474	0.0572	0.0427
PSTH	732592105	100	**0.0253**	0.0257	0.0455	0.0537	0.0465	0.0558	0.0422
PSTH	754312389	0	0.0400	**0.0398**	0.0583	0.0600	0.0570	0.0620	0.0498
PSTH	754312389	40	**0.0395**	0.0398	0.0566	0.0630	0.0575	0.0652	0.0519
PSTH	754312389	80	**0.0364**	0.0365	0.0534	0.0587	0.0541	0.0609	0.0483
PSTH	754312389	100	0.0378	**0.0377**	0.0548	0.0595	0.0555	0.0616	0.0490
PSTH	791319847	0	**0.0276**	0.0279	0.0465	0.0582	0.0493	0.0607	0.0455
PSTH	791319847	40	**0.0320**	0.0324	0.0517	0.0642	0.0546	0.0666	0.0506
PSTH	791319847	80	**0.0292**	0.0294	0.0488	0.0591	0.0510	0.0616	0.0465
PSTH	791319847	100	**0.0277**	0.0279	0.0469	0.0589	0.0496	0.0614	0.0456
Single	732592105	0	**0.0463**	0.0467	0.0522	0.0537	0.0538	0.0621	0.0469
Single	732592105	40	**0.0477**	0.0480	0.0537	0.0552	0.0554	0.0650	0.0482
Single	732592105	80	**0.0469**	0.0472	0.0531	0.0543	0.0546	0.0635	0.0520
Single	732592105	100	**0.0483**	0.0485	0.0545	0.0555	0.0561	0.0651	0.0487
Single	754312389	0	**0.0314**	0.0315	0.0356	0.0343	0.0351	0.0368	0.0317
Single	754312389	40	**0.0303**	0.0305	0.0347	0.0333	0.0340	0.0353	0.0310
Single	754312389	80	**0.0307**	**0.0307**	0.0350	0.0336	0.0342	0.0357	0.0309
Single	754312389	100	**0.0323**	**0.0323**	0.0366	0.0351	0.0359	0.0378	0.0326
Single	791319847	0	**0.0426**	0.0427	0.0478	0.0487	0.0488	0.0553	0.0429
Single	791319847	40	**0.0431**	0.0432	0.0487	0.0493	0.0496	0.0567	0.0435
Single	791319847	80	**0.0416**	0.0420	0.0475	0.0479	0.0482	0.0541	0.0447
Single	791319847	100	**0.0434**	0.0439	0.0494	0.0499	0.0502	0.0575	0.0472

I LEARNED ORDERS ON NEURAL RESPONSE PREDICTION

Figures 8 and 9 show the orders α_t learned by *VO*-LIF and VO-IF networks on the Allen Neuropixels task (Section 4.4) over 100 time steps, for each stimulus class and for single-trial and PSTH targets. Each curve is the mean order at each time step. Two patterns hold for both neuron types. First, the target sets the level of the order. For natural movies and natural scenes, PSTH targets lead to much larger orders than single-trial targets (for *VO*-LIF, about 0.81 vs. 0.22 on movies and 0.79 vs. 0.46 on scenes), whereas for drifting gratings the two targets give similar orders of about 0.3 to 0.4. Second, the schedule depends on the stimulus. On drifting gratings, the order is highest in the first few steps, drops within about ten steps, and then changes slowly. On natural movies it stays almost constant, varying by less than 0.03. On natural scenes it oscillates, most clearly with single-trial targets.

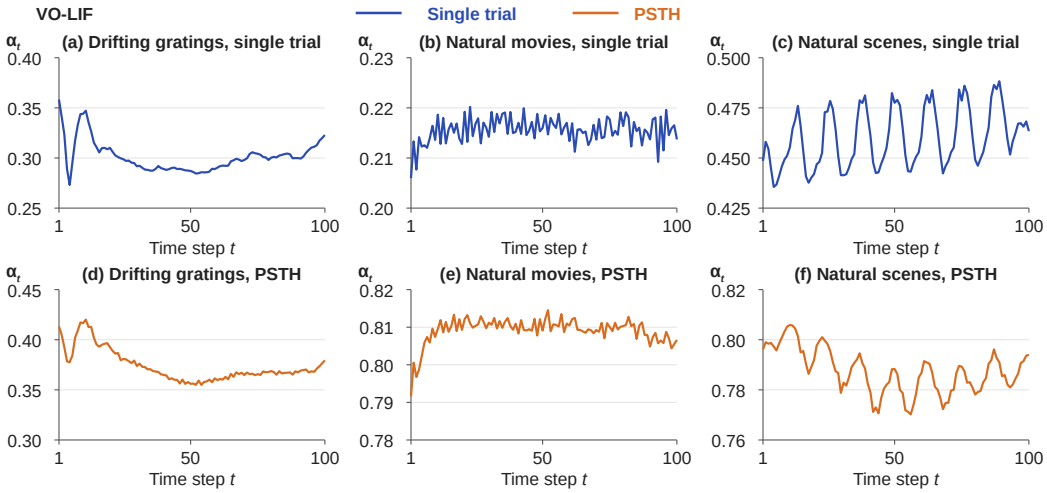

Figure 8: Learned orders α_t of *VO*-LIF on the Allen Neuropixels task for drifting gratings, natural movies, and natural scenes, with single-trial (top, blue) and PSTH (bottom, orange) targets. Curves show the mean order at each time step. Vertical scales differ across panels.

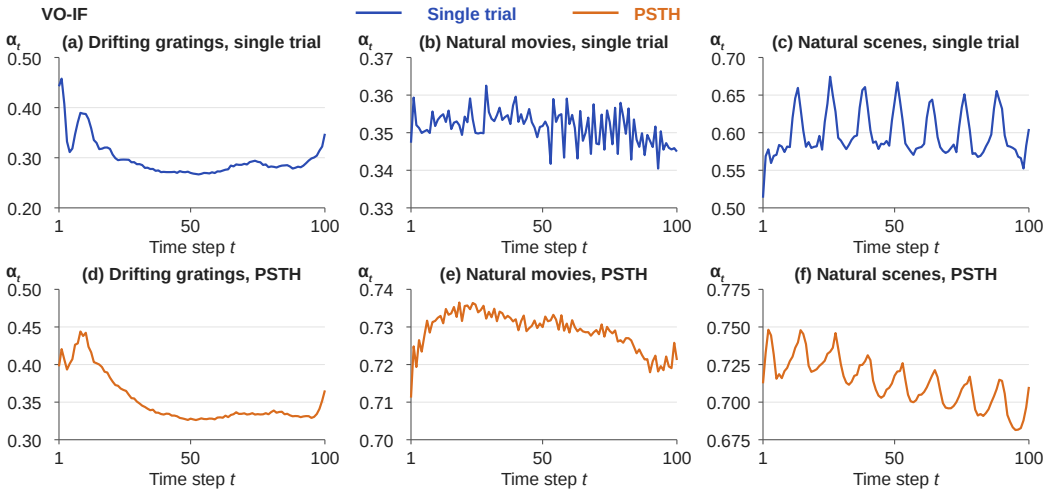

Figure 9: Learned orders α_t of VO-IF on the Allen Neuropixels task. Layout as in Figure 8.

J Additional Qualitative Detection Results

Figures 10–18 show nine further Gen1 test scenes and Figures 19–23 five Gen4 test scenes, in the format of Figure 4. For each scene, the left panel shows the event frame with ground-truth boxes (dashed black), and the right panels show EMS-YOLO (ResNet-10) predictions with LIF, *f*-LIF, and *VO*-LIF neurons on the same test sample and at the same confidence threshold. A prediction counts as a true positive (TP) if its IoU with a ground-truth box is at least 0.5, and only predictions with confidence at least 0.25 are shown; each panel reports TP, false positives (FP), false negatives (FN), and F1. In these scenes *f*-LIF mainly recovers objects that LIF misses, whereas *VO*-LIF mainly removes false positives: *VO*-LIF has no false positive in twelve of the fourteen scenes and the highest F1 in all of them. These scenes are illustrative examples; the quantitative comparison is given in Table 2.

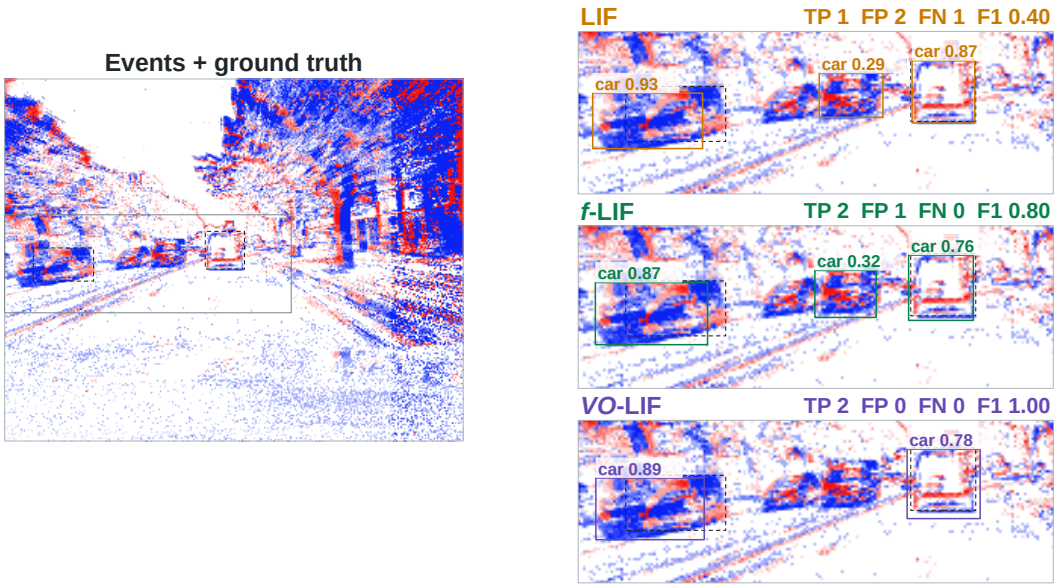

Figure 10: Gen1 detection, distant vehicles. *f*-LIF recovers the missed car and cuts false positives from 2 to 1; *VO*-LIF removes the remaining false positive. (LIF: TP 1 / FP 2 / FN 1 / F1 0.40; *f*-LIF: TP 2 / FP 1 / FN 0 / F1 0.80; *VO*-LIF: TP 2 / FP 0 / FN 0 / F1 1.00.)

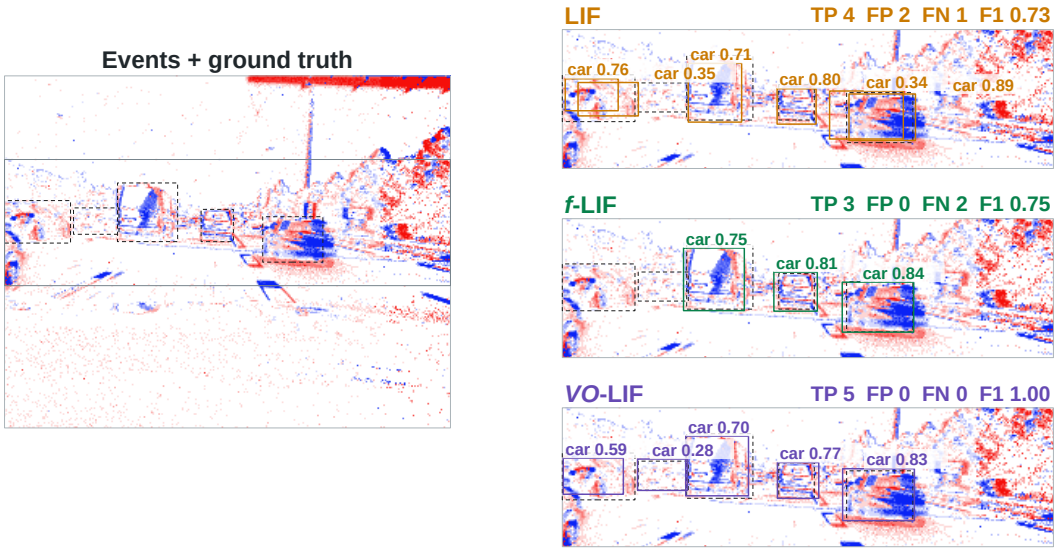

Figure 11: Gen1 detection, example 1. f-LIF removes both LIF false positives but misses one more target; *VO*-LIF detects all 5 targets without false positives. (LIF: TP 4 / FP 2 / FN 1 / F1 0.73; f-LIF: TP 3 / FP 0 / FN 2 / F1 0.75; *VO*-LIF: TP 5 / FP 0 / FN 0 / F1 1.00.)

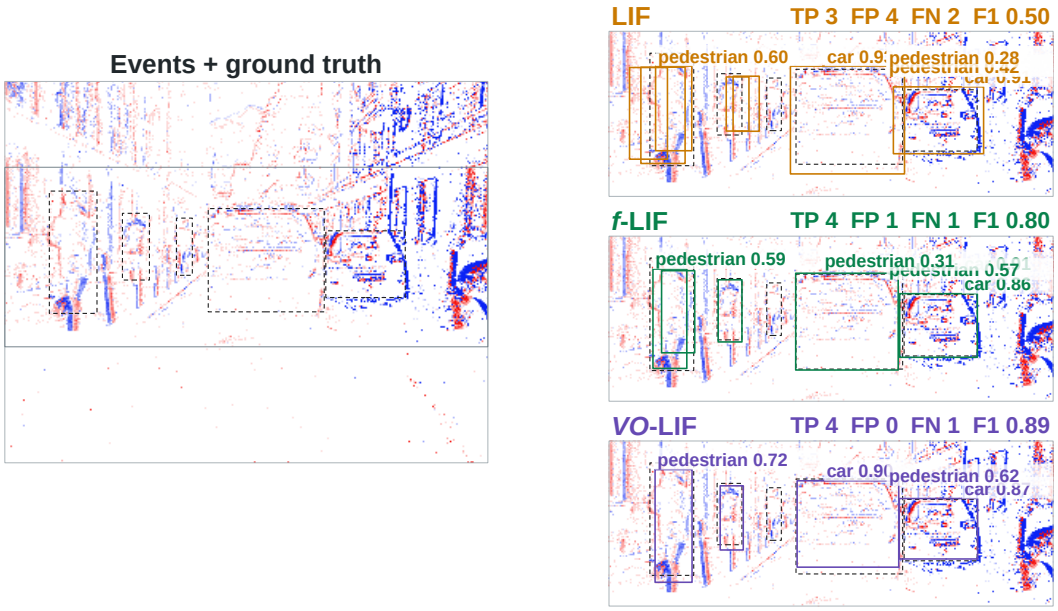

Figure 12: Gen1 detection, example 2. f-LIF recovers one target and cuts false positives from 4 to 1; *VO*-LIF keeps 4 true detections and removes the final false positive. (LIF: TP 3 / FP 4 / FN 2 / F1 0.50; f-LIF: TP 4 / FP 1 / FN 1 / F1 0.80; *VO*-LIF: TP 4 / FP 0 / FN 1 / F1 0.89.)

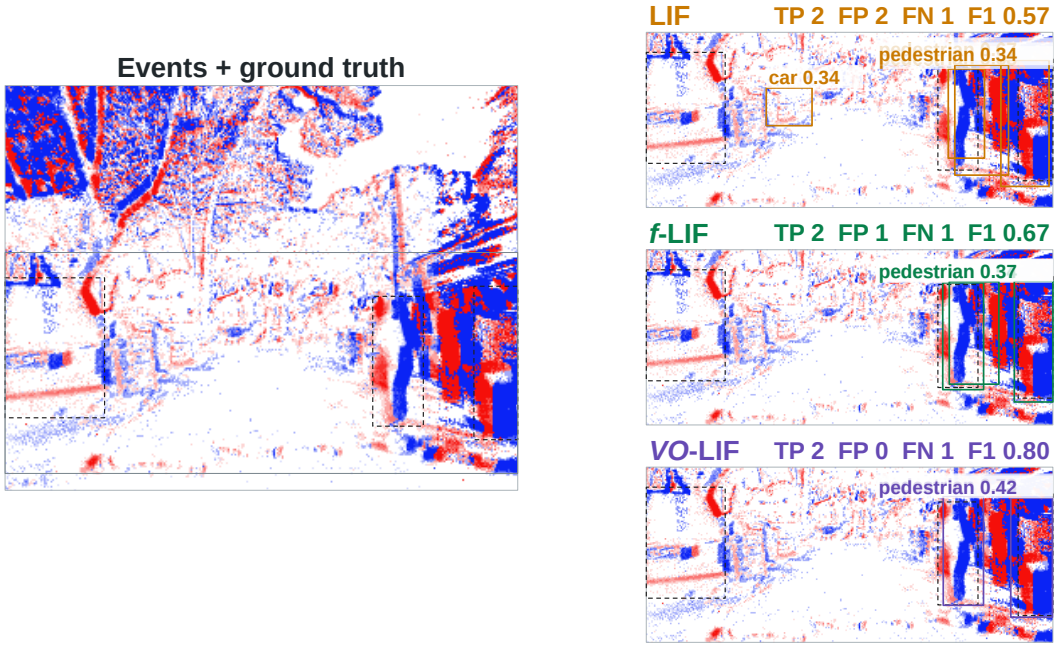

Figure 13: Gen1 detection, example 3. At the same 2 true detections, false positives fall from 2 (LIF) to 1 (*f*-LIF) to 0 (*VO*-LIF). (LIF: TP 2 / FP 2 / FN 1 / F1 0.57; *f*-LIF: TP 2 / FP 1 / FN 1 / F1 0.67; *VO*-LIF: TP 2 / FP 0 / FN 1 / F1 0.80.)

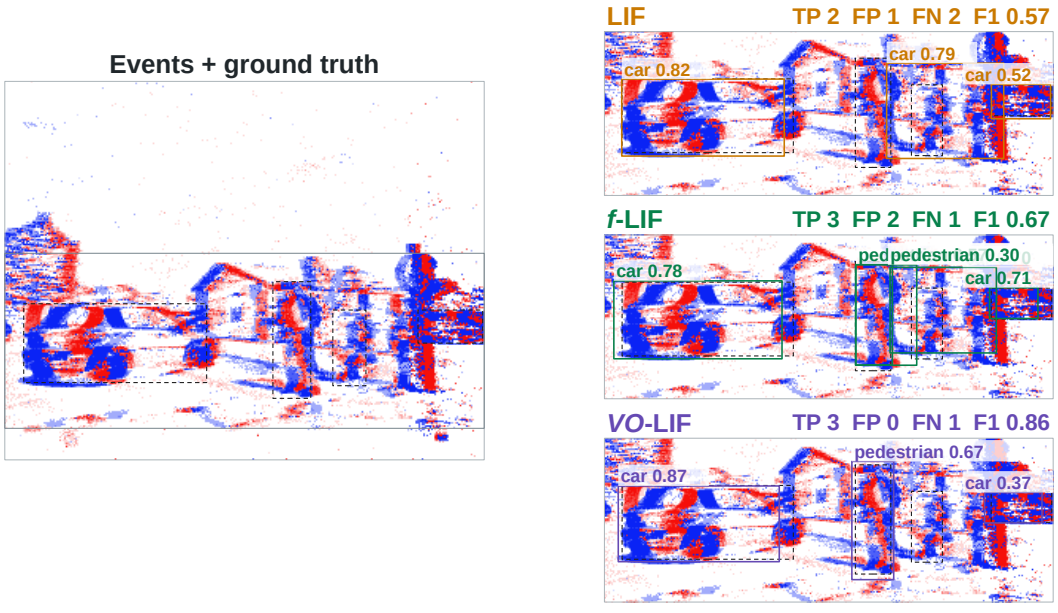

Figure 14: Gen1 detection, example 4. *f*-LIF detects one more target but adds a false positive; *VO*-LIF retains 3 true detections and removes both false positives. (LIF: TP 2 / FP 1 / FN 2 / F1 0.57; *f*-LIF: TP 3 / FP 2 / FN 1 / F1 0.67; *VO*-LIF: TP 3 / FP 0 / FN 1 / F1 0.86.)

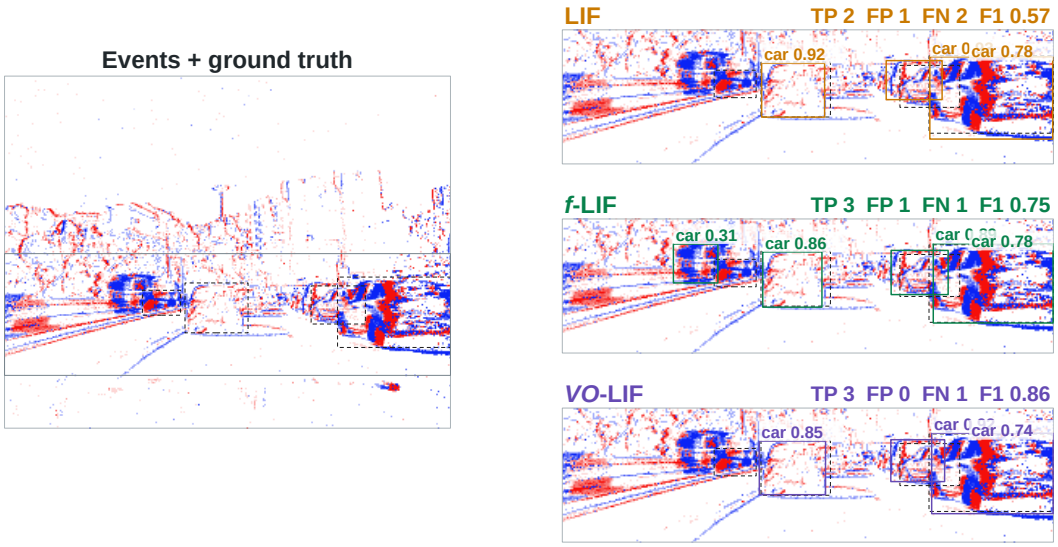

Figure 15: Gen1 detection, example 5. f-LIF recovers one missed target; *VO*-LIF keeps 3 true detections and removes the remaining false positive. (LIF: TP 2 / FP 1 / FN 2 / F1 0.57; f-LIF: TP 3 / FP 1 / FN 1 / F1 0.75; *VO*-LIF: TP 3 / FP 0 / FN 1 / F1 0.86.)

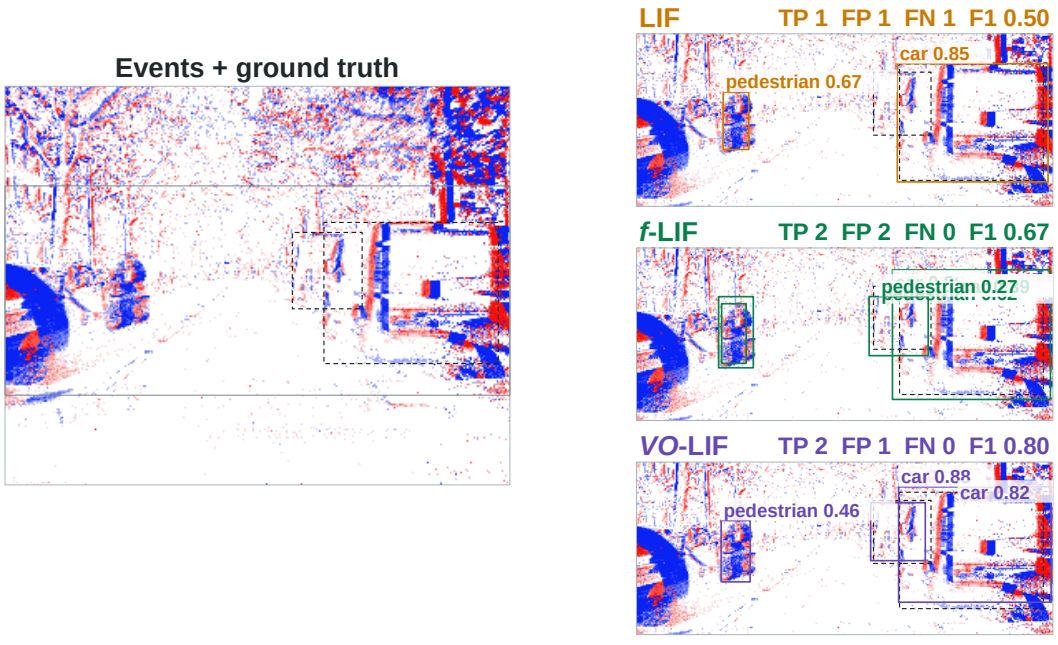

Figure 16: Gen1 detection, example 6. f-LIF recovers the missed target but adds a false positive; *VO*-LIF keeps both true detections and cuts false positives from 2 to 1. (LIF: TP 1 / FP 1 / FN 1 / F1 0.50; f-LIF: TP 2 / FP 2 / FN 0 / F1 0.67; *VO*-LIF: TP 2 / FP 1 / FN 0 / F1 0.80.)

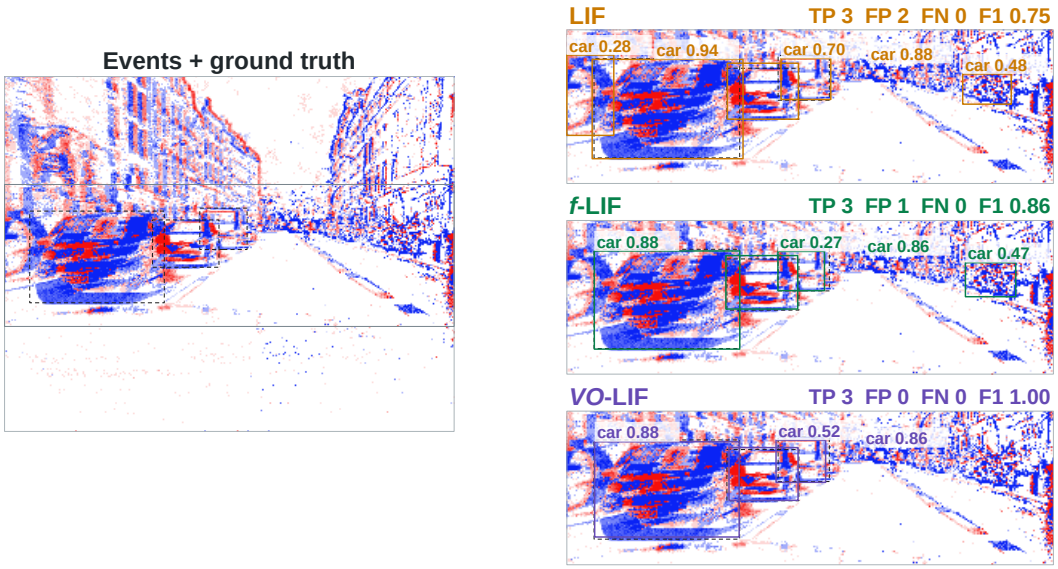

Figure 17: Gen1 detection, example 7. All three targets are detected; false positives fall from 2 (LIF) to 1 (*f*-LIF) to 0 (*VO*-LIF). (LIF: TP 3 / FP 2 / FN 0 / F1 0.75; *f*-LIF: TP 3 / FP 1 / FN 0 / F1 0.86; *VO*-LIF: TP 3 / FP 0 / FN 0 / F1 1.00.)

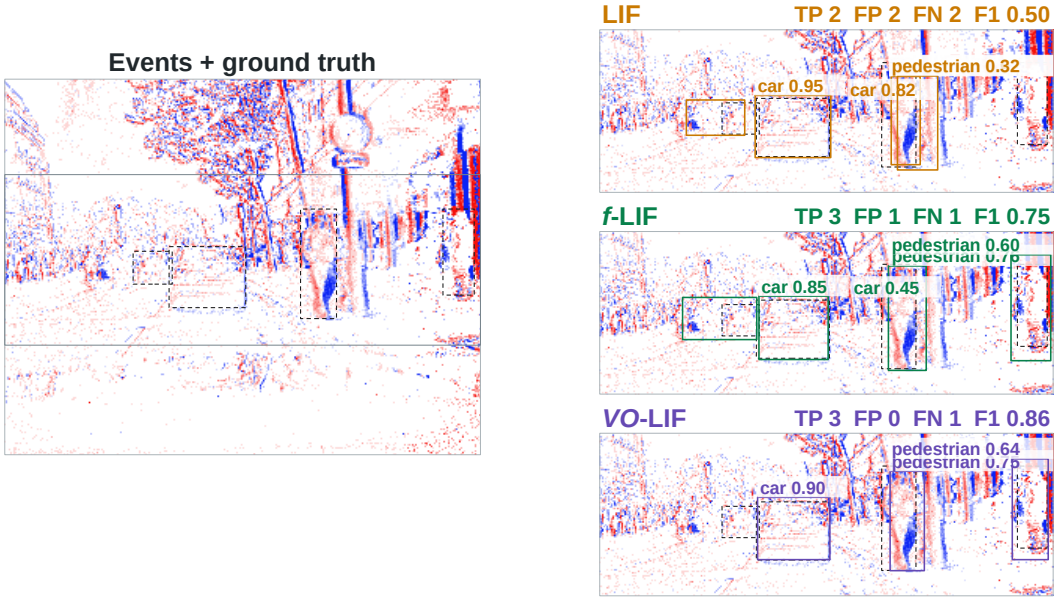

Figure 18: Gen1 detection, example 8. *f*-LIF recovers one target and cuts false positives from 2 to 1; *VO*-LIF removes the remaining false positive. (LIF: TP 2 / FP 2 / FN 2 / F1 0.50; *f*-LIF: TP 3 / FP 1 / FN 1 / F1 0.75; *VO*-LIF: TP 3 / FP 0 / FN 1 / F1 0.86.)

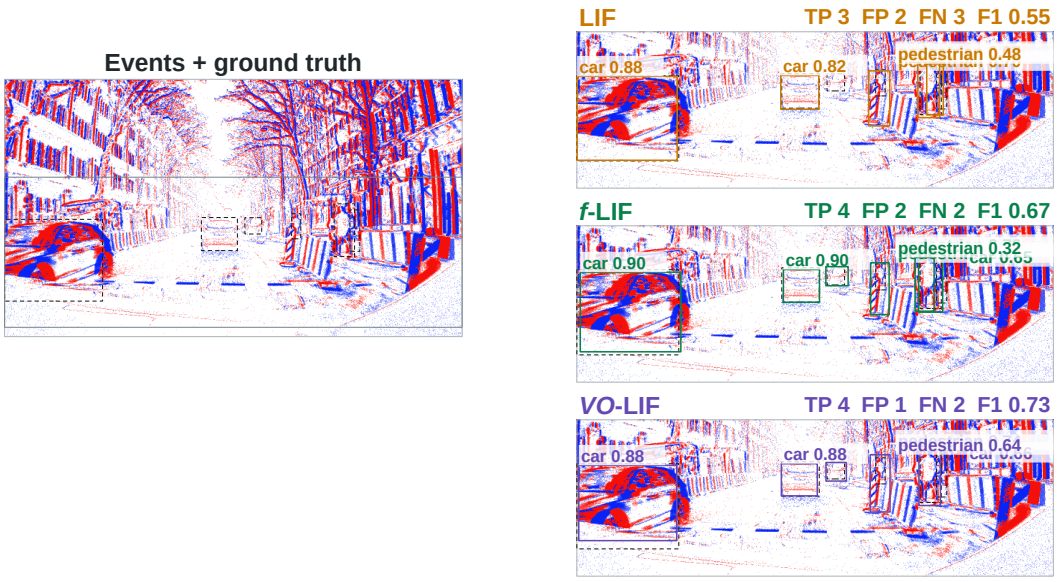

Figure 19: Gen4 detection, example 1. f-LIF recovers one missed target; *VO*-LIF keeps its 4 true detections and cuts false positives from 2 to 1. (LIF: TP 3 / FP 2 / FN 3 / F1 0.55; f-LIF: TP 4 / FP 2 / FN 2 / F1 0.67; *VO*-LIF: TP 4 / FP 1 / FN 2 / F1 0.73.)

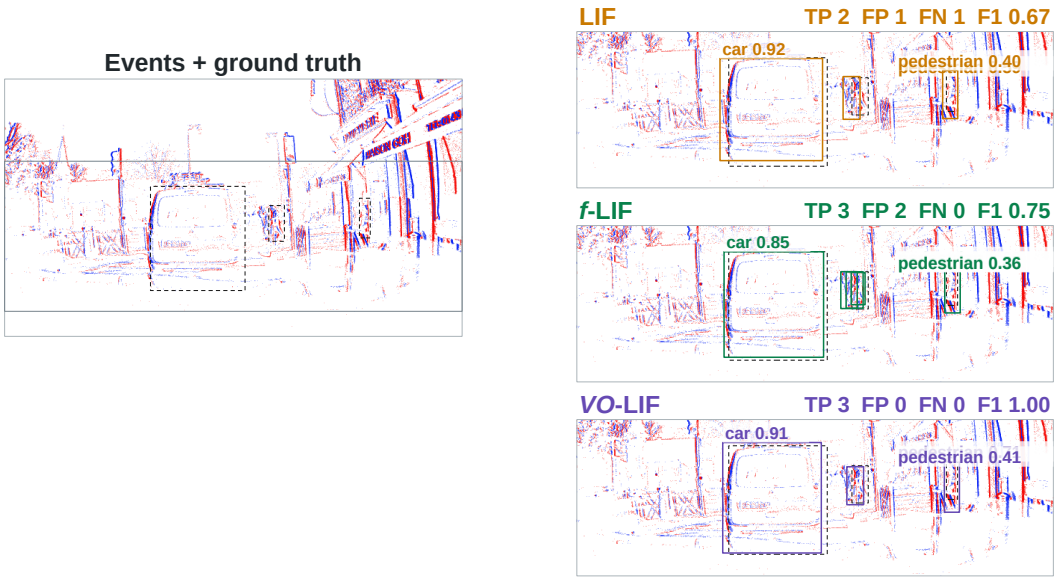

Figure 20: Gen4 detection, example 2. f-LIF recovers the missed target but adds a false positive; *VO*-LIF keeps all 3 detections and removes both false positives. (LIF: TP 2 / FP 1 / FN 1 / F1 0.67; f-LIF: TP 3 / FP 2 / FN 0 / F1 0.75; *VO*-LIF: TP 3 / FP 0 / FN 0 / F1 1.00.)

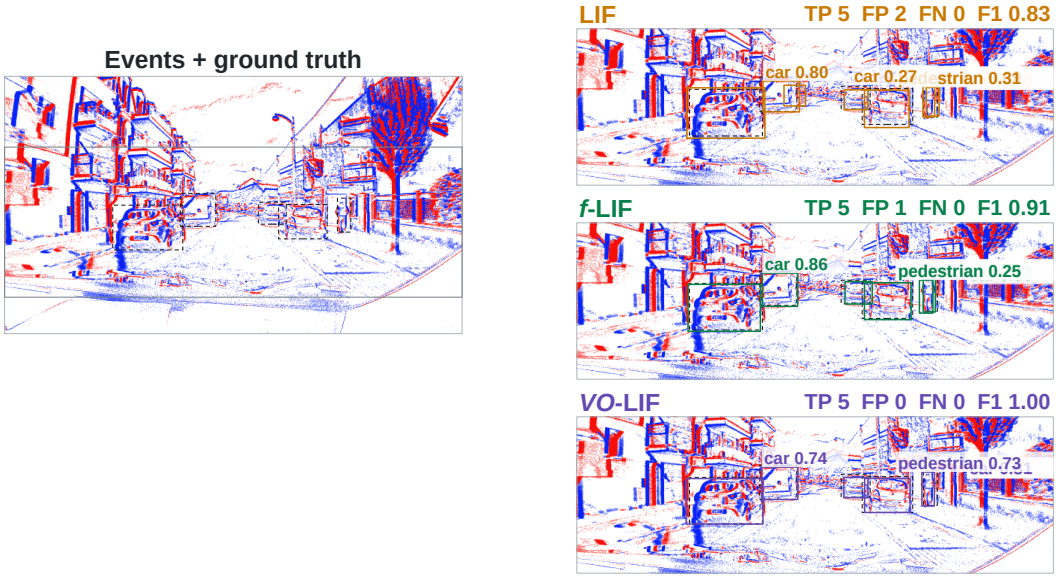

Figure 21: Gen4 detection, example 3. All 5 targets are detected; false positives fall from 2 (LIF) to 1 (f-LIF) to 0 (*VO*-LIF). (LIF: TP 5 / FP 2 / FN 0 / F1 0.83; f-LIF: TP 5 / FP 1 / FN 0 / F1 0.91; *VO*-LIF: TP 5 / FP 0 / FN 0 / F1 1.00.)

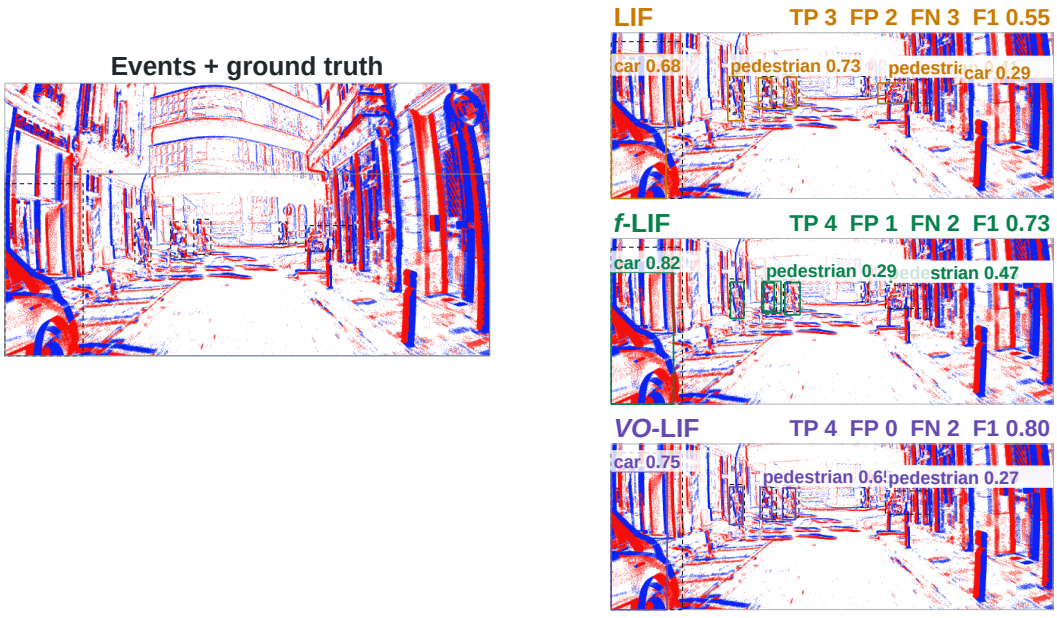

Figure 22: Gen4 detection, example 4. f-LIF recovers one target and cuts false positives from 2 to 1; *VO*-LIF keeps 4 true detections and removes the remaining false positive. (LIF: TP 3 / FP 2 / FN 3 / F1 0.55; f-LIF: TP 4 / FP 1 / FN 2 / F1 0.73; *VO*-LIF: TP 4 / FP 0 / FN 2 / F1 0.80.)

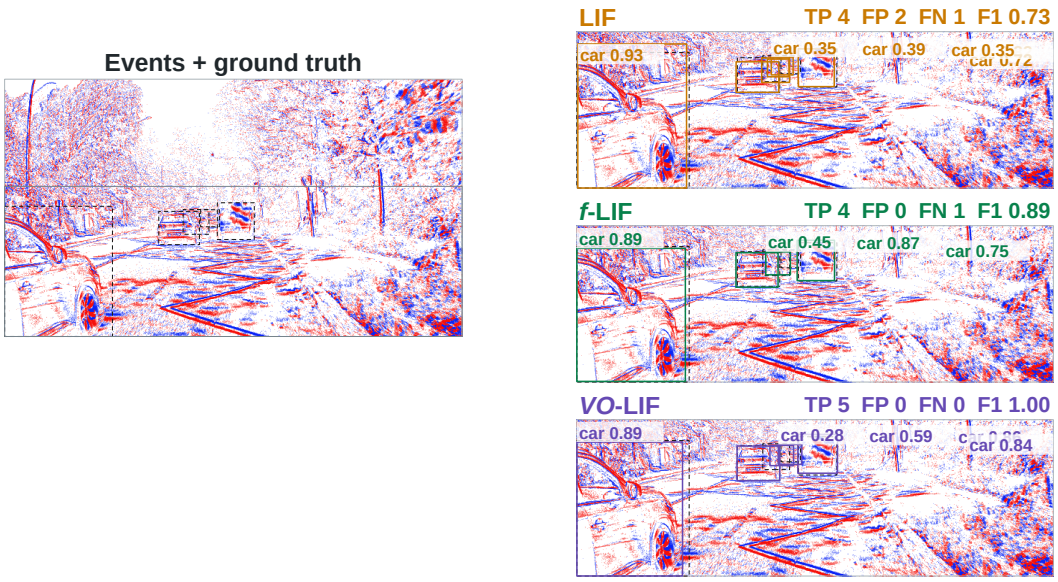

Figure 23: Gen4 detection, example 5. *f*-LIF removes both false positives; *VO*-LIF also recovers the missed target, reaching 5 true detections without false positives. (LIF: TP 4 / FP 2 / FN 1 / F1 0.73; *f*-LIF: TP 4 / FP 0 / FN 1 / F1 0.89; *VO*-LIF: TP 5 / FP 0 / FN 0 / F1 1.00.)

K Layer-wise Firing Rates

Figure 24 reports the average firing rate of each spiking layer of SEW-ResNet-18 on DVS-Lip for LIF, f-LIF, and *VO*-LIF neurons (mean over five seeds; exact values in Table 11). Both fractional neurons fire much less than LIF: the overall firing rate over all spiking neurons (i.e., weighting each layer by its number of neurons) drops from 8.60% for LIF to 5.05% for f-LIF and 5.14% for *VO*-LIF, a reduction of about 40%. The reduction is largest in the first two stages, for example from 23.38% to 9.30% and 10.32% in the second LIF layer of the first block of stage 2. In the last stage the three neurons are close, and the fractional neurons fire slightly more in the final layer (21.74% for *VO*-LIF vs. 19.49% for LIF). *VO*-LIF and f-LIF follow nearly the same profile, with an overall difference of 0.09 percentage points, so the time-varying order adds almost no spiking activity.

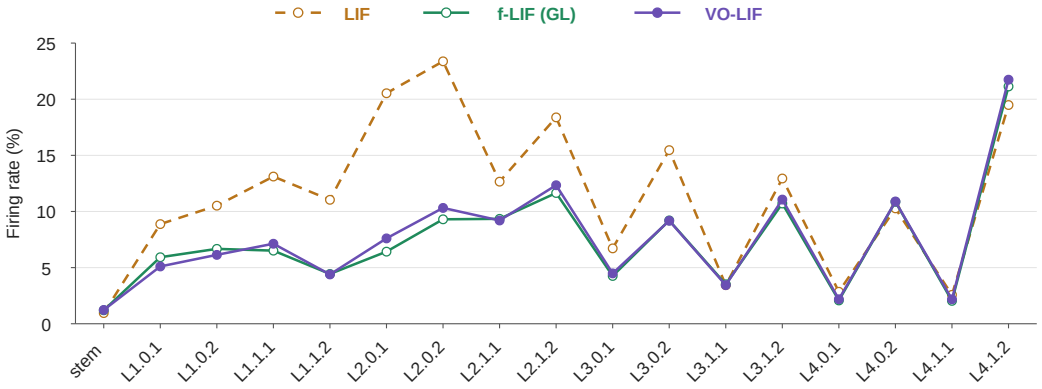

Figure 24: Layer-wise firing rates of SEW-ResNet-18 on DVS-Lip with LIF, f-LIF (GL), and *VO*-LIF neurons. "stem" denotes the spiking layer after the first convolution, and L$s.b.k$ denotes the k-th spiking layer of block b in stage s. Values are means over five seeds.

Table 11: Layer-wise firing rates (%) of SEW-ResNet-18 on DVS-Lip, reported as mean ± standard deviation over five seeds. These are the values plotted in Figure 24.

Layer	Module	LIF	f-LIF (GL)	*VO*-LIF
stem	`lif1`	0.95±0.36	1.20±0.11	1.23±0.12
L1.0.1	`layer1.0.lif1`	8.88±0.67	5.92±0.61	5.10±0.49
L1.0.2	`layer1.0.lif2`	10.52±0.53	6.67±1.11	6.14±0.80
L1.1.1	`layer1.1.lif1`	13.11±0.91	6.51±1.04	7.13±0.41
L1.1.2	`layer1.1.lif2`	11.04±0.85	4.43±0.54	4.39±0.19
L2.0.1	`layer2.0.lif1`	20.53±2.12	6.42±0.34	7.60±0.69
L2.0.2	`layer2.0.lif2`	23.38±1.61	9.30±0.56	10.32±0.38
L2.1.1	`layer2.1.lif1`	12.66±1.23	9.34±0.53	9.20±0.35
L2.1.2	`layer2.1.lif2`	18.38±1.33	11.65±0.37	12.34±0.26
L3.0.1	`layer3.0.lif1`	6.71±0.42	4.26±0.09	4.48±0.29
L3.0.2	`layer3.0.lif2`	15.47±0.44	9.20±0.16	9.18±0.32
L3.1.1	`layer3.1.lif1`	3.44±0.16	3.52±0.18	3.44±0.18
L3.1.2	`layer3.1.lif2`	12.94±0.48	10.71±0.26	11.06±0.20
L4.0.1	`layer4.0.lif1`	2.85±0.34	2.07±0.06	2.18±0.10
L4.0.2	`layer4.0.lif2`	10.28±0.18	10.86±0.12	10.89±0.39
L4.1.1	`layer4.1.lif1`	2.57±0.13	2.04±0.07	2.18±0.11
L4.1.2	`layer4.1.lif2`	19.49±5.42	21.14±0.15	21.74±0.48
Overall (all neurons)		8.60±0.47	5.05±0.18	5.14±0.13

L Limitations and Future Work

A single order per sample and time step is shared by all neurons in the network, which keeps the controller compact but does not tailor the memory profile to individual layers or neurons. Learning layer-wise or neuron-wise orders is a natural extension that we leave for future work.