Filtered-Trace Online Variational Training for Probabilistic Spiking Neural Networks

Yaokun Wang[1] **Tiantian Xiao**[1] **Hongyan Ding**[2] **Zhi Yan**[3*]

[1]School of Artificial Intelligence, Beijing Normal University, China
[2]State Key Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, China
[3]School of Mathematics and Statistics, Shandong University of Technology, China

Abstract

Probabilistic spiking neural networks (SNNs) model spike trains as temporal point processes and provide a principled framework for learning with latent spikes. Recent differentiable point-process methods enable path-wise variational learning, but their training still relies on full-sequence backpropagation through time (BPTT), leading to memory costs that grow with the temporal horizon. In this paper, we develop an online variational training framework for probabilistic SNNs based on discrete-time spike response model dynamics. By representing synaptic history with finite-dimensional Markovian traces, our method updates both generative and variational parameters without storing the full temporal computation graph. To handle future-dependent credit assignment from recurrent spike histories, we introduce a horizon-R family of online estimators. Theoretically, we show that the truncation-induced bias decays exponentially with R under the SRM kernel-decay condition. Synthetic experiments validate the predicted horizon-dependent behavior and bias decay, while N-MNIST experiments show competitive classification accuracy and avoid the sequence-length-dependent memory growth of BPTT.

1 Introduction

Spiking neural networks (SNNs) provide an event-driven model for temporal neural computation [20, 39, 32]. Most trainable SNNs are based on deterministic hard-threshold dynamics and are optimized with surrogate gradients or differentiable relaxations [41, 37, 31, 26]. While this paradigm has been effective for supervised prediction [19, 35, 36], it is less natural for probabilistic modeling of spike trains, latent spike inference, and uncertainty-aware generative learning [12]. Probabilistic SNNs instead model spike trains as temporal point processes with history-dependent conditional intensities, providing a likelihood-based framework for learning with hidden spike variables [28, 33, 15].

A classical approach to probabilistic SNNs with hidden neurons is stochastic variational inference, where hidden spike trains are treated as latent variables and learning is formulated by maximizing an evidence lower bound (ELBO) [15]. However, its score-function gradient estimator can suffer from high variance [24, 29]. Kajino introduced a differentiable point-process formulation for SNNs, enabling path-wise gradient estimation through a differentiable thinning procedure and a differentiable SNN relaxation [16]. This provides a principled reparameterization-based approach to variational learning in probabilistic SNNs.

Despite this progress, existing differentiable point-process training still relies on full-sequence backpropagation through time (BPTT), which stores the temporal computation graph over the entire sequence [34, 40, 9]. Online training methods for deterministic SNNs reduce this cost through eligibility traces [2], local credit assignment [25], or forward-in-time approximations [38, 14], but

*Correspondence to Zhi Yan <yanzhi@sdut.edu.cn>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

they do not directly address variational learning in probabilistic SNNs. In this setting, hidden spikes influence both the generative and variational models through future synaptic traces, making online credit assignment substantially more involved than in deterministic surrogate-gradient training.

Contributions. In this paper, we develop an online variational training framework for probabilistic SNNs under the differentiable point-process formulation. The key idea is to represent discrete-time spike response model (SRM) dynamics using finite-dimensional Markovian synaptic traces, so that the temporal dependence of the ELBO gradient can be propagated through recursively updated local traces and filtered feedback signals. This leads to streaming updates for both the generative and variational parameters without unrolling the full temporal computation graph.

We then provide a theoretical characterization of the resulting online recursion. Specifically, we formulate the algorithm as a controlled Markov stochastic-approximation process and identify its limiting mean-field ordinary differential equation (ODE). This formalizes the behavior of the proposed online variational updates and clarifies the approximation introduced by truncating future-dependent credit assignment.

Finally, we introduce a horizon-R family of truncated online rules and prove that the additional truncation-induced mean-field bias relative to the full-horizon online rule decays exponentially with R under the SRM kernel-decay condition. Experiments on synthetic spike sequences validate the predicted dependence on R, while experiments on N-MNIST show that the proposed online method reduces memory usage compared with BPTT while maintaining competitive predictive performance.

2 Related work

Online training in deterministic SNNs. Most trainable spiking neural networks are based on deterministic neuron dynamics, such as leaky integrate-and-fire (LIF) models [8], and are commonly optimized with surrogate gradients or differentiable relaxations through backpropagation through time (BPTT) [41, 37, 31, 26]. However, BPTT stores intermediate neuronal states and computation graphs across the entire sequence, resulting in memory costs that grow with the number of timesteps. Moreover, it can also suffer from vanishing or exploding gradients over long temporal horizons. To alleviate these limitations, subsequent studies have proposed forward-in-time or local online credit-assignment schemes, such as e-prop [2], OTTT [38], SLTT [23], and NDOT [14], which approximate temporal gradients using eligibility traces, instantaneous local signals, or neuronal-dynamics-based updates. However, these methods are mainly designed for deterministic spiking dynamics and surrogate-gradient training, and therefore do not directly address variational learning in probabilistic SNNs with latent spike trains.

Point processes and probabilistic SNNs. A separate line of work models spike trains as realizations of point processes and casts learning in a probabilistic or variational framework [28, 33]. Neural temporal point-process models further use recurrent or attention-based representations to encode event histories and parameterize history-dependent conditional intensities [6, 22, 42]; see Shchur et al. [30] for a review. This literature provides a broader neural point-process perspective for modeling temporal event dependencies, while probabilistic SNNs additionally involve latent spike trajectories and inference over hidden neuronal events.

Within probabilistic SNNs, Rezende and Gerstner introduced stochastic variational learning for recurrent spiking networks with hidden neurons, where learning is formulated by maximizing an ELBO [15]. Their optimization of the variational distribution relies on a score-function estimator, which can suffer from high variance. Building on this variational formulation, Kajino introduced a differentiable point process together with a differentiable SNN model, enabling reparameterized path-wise gradient estimation through relaxed hidden spike trajectories and substantially improving gradient estimation quality [16]. However, temporal credit assignment in this differentiable formulation still relies on full-sequence BPTT [37, 26], requiring gradients to propagate through the recurrent influence of sampled hidden spikes over the complete temporal computation graph.

A complementary line of work studies online or local learning directly in probabilistic SNNs. VOWEL derives a variational online local learning rule for recurrent probabilistic SNNs using local likelihood-gradient terms modulated by a shared learning signal [13], while GEM-SNN develops multi-sample online learning based on generalized expectation maximization and three-factor updates [11]. These methods directly sample discrete hidden spikes and construct their learning rules from

score-function or generalized-EM estimators, without path-wise differentiation through the sampled spike trajectories. They therefore address a different gradient-estimation setting from Kajino's Concrete-relaxed differentiable point-process formulation.

Our work builds on the latter path-wise formulation. Because a reparameterized hidden spike affects both future generative and variational intensities through recurrent synaptic dynamics, its ELBO gradient contains future-dependent credit that is non-causal under direct differentiation. We develop a filtered-trace formulation that causally propagates this future-dependent variational credit without full temporal unrolling, together with a horizon-R family for controlling and analyzing the resulting future-credit approximation.

3 Preliminaries

This section introduces the probabilistic spiking neural network model, its discrete-time approximation, and the variational and differentiable-sampling ingredients that underlie the subsequent method and analysis.

3.1 Probabilistic model of spiking neural networks

Point-process model. We model a continuous-time spiking neural network (SNN) as an N-marked temporal point process [28, 33], with spike train $\mathcal{T} = \{(t_n, m_n)\}_{n=1}^{|\mathcal{T}|}$ of all neurons on $[0, T_{\mathrm{end}}]$. Each event consists of a timestamp t and a one-hot mark $m \in \mathcal{P} := \{\mathbf{1}_1, \dots, \mathbf{1}_N\} \subset \{0,1\}^N$, indicating which neuron fired. Let $\mathcal{T}^{<t}$ denote the event history strictly before time t, and let $d\mathcal{N}(s)$ be the associated $\mathbb{R}^N$-valued counting measure, $d\mathcal{N}(s) := \sum_{n=1}^{|\mathcal{T}|} m_n \, \delta_{t_n}(ds)$. Here s denotes the continuous-time integration variable over past time; when it appears as the argument of a kernel such as $\mathbf{K}(s)$ or $\kappa(s)$, it denotes the corresponding nonnegative time lag. For the discrete-time formulation used later, we write z_j^t for the spike variable of neuron j in time bin t; in the original spiking model, $z_j^t \in \{0,1\}$ indicates whether neuron j fires in bin t. Thus, the membrane potential $u(t) \in \mathbb{R}^N$ is defined as

$$u(t \mid \mathcal{T}^{<t}) = \bar{u} + \int_{[0,t)} \mathbf{K}(t-s)\, d\mathcal{N}(s), \tag{1}$$

where $\bar{u} \in \mathbb{R}^N$ is a constant baseline and $\mathbf{K}(\cdot) \in \mathbb{R}^{N \times N}$ is a causal synaptic filter, i.e., $\mathbf{K}(s) = 0$ for $s < 0$. Equivalently,

$$u(t \mid \mathcal{T}^{<t}) = \bar{u} + \sum_{n : t_n < t} \mathbf{K}(t - t_n)\, m_n. \tag{2}$$

The conditional intensity of emitting a spike with mark m at time t is

$$\lambda_{\mathrm{SNN}}(t, m \mid \mathcal{T}^{<t}) = m^\top \sigma\big(u(t \mid \mathcal{T}^{<t})\big), \tag{3}$$

where $\sigma : \mathbb{R} \to \mathbb{R}_{>0}$ is a positive differentiable nonlinearity, and for vector inputs it is applied element-wise. For one-hot marks, the total intensity is $\sum_{m \in \mathcal{P}} \lambda_{\mathrm{SNN}}(t, m \mid \mathcal{T}^{<t}) = \sum_{i=1}^N \sigma\big(u_i(t \mid \mathcal{T}^{<t})\big)$. The log-likelihood admits the standard form

$$\log p(\mathcal{T}) = \sum_{(t,m) \in \mathcal{T}} \log \lambda_{\mathrm{SNN}}\big(t, m \mid \mathcal{T}^{<t}\big) - \int_0^{T_{\mathrm{end}}} \sum_{m \in \mathcal{P}} \lambda_{\mathrm{SNN}}\big(t, m \mid \mathcal{T}^{<t}\big)\, dt. \tag{4}$$

Let O and H denote the sets of observed and hidden neurons, respectively, with $O \cup H = [N]$ and $O \cap H = \emptyset$, and $[N] := \{1, \dots, N\}$. The spike train $\mathcal{T}$ is partitioned as $\mathcal{T} = \mathcal{T}_O \cup \mathcal{T}_H$, where $\mathcal{T}_O$ contains spikes from observed neurons and $\mathcal{T}_H$ those from hidden neurons. Under parameters θ, the intensity (3) induces a joint point-process likelihood $p_\theta(\mathcal{T}_O, \mathcal{T}_H)$ via (4). We learn θ by maximizing the marginal log-likelihood $\log p_\theta(\mathcal{T}_O) = \log \int p_\theta(\mathcal{T}_O, \mathcal{T}_H)\, d\mathcal{T}_H$, which is generally intractable. Therefore, we introduce a variational distribution $q_\phi(\mathcal{T}_H \mid \mathcal{T}_O)$ and optimize the evidence lower bound (ELBO) [17, 15, 16]. Following Kajino's framework, one convenient choice is to parameterize $q_\phi(\mathcal{T}_H \mid \mathcal{T}_O)$ as an SNN through its own conditional intensity. We keep θ and ϕ separate in the derivation, while allowing parameter tying in implementation [16]. The ELBO $\mathcal{L}(\theta, \phi; \mathcal{T}_O)$ is defined as:

$$\mathcal{L}(\theta, \phi; \mathcal{T}_O) = \mathbb{E}_{\mathcal{T}_H \sim q_\phi(\cdot \mid \mathcal{T}_O)} \big[\log p_\theta(\mathcal{T}_O, \mathcal{T}_H) - \log q_\phi(\mathcal{T}_H \mid \mathcal{T}_O) \big]. \tag{5}$$

In the following sections, we derive an online training rule for optimizing (5) and establish its theoretical properties.

Differentiable variational distribution. The main obstacle to using the path-wise gradient estimator in SNNs is that point-process samples are not differentiable with respect to the variational parameters. Kajino overcomes this by introducing a differentiable point process $\partial\mathrm{PP}$, which replaces the categorical thinning step with a reparameterizable Concrete relaxation [16, 21, 10]. Because the resulting relaxed hidden marks are not compatible with the standard SNN intensity, Kajino further proposes a differentiable SNN (∂SNN) to evaluate the relaxed ELBO consistently. The original SNN is recovered as the relaxation temperature $\tau_{\mathrm{SNN}} \to 0$. We accordingly parameterize the variational distribution as

$$q_\phi(\mathcal{T}_H \mid \mathcal{T}_O) = \partial\mathrm{PP}\big(\lambda_q(\cdot \mid \cdot\,; \phi), \bar{\lambda}, \tau_{\mathrm{SNN}}\big). \tag{6}$$

Here, λ_q denotes the variational conditional intensity for hidden spikes, $\bar{\lambda}$ is a constant upper bound used by the base Poisson process in thinning, and $\tau_{\mathrm{SNN}} > 0$ is the Concrete relaxation temperature. We approximate the ELBO gradient by differentiating the intensity-based relaxed objective under $\partial\mathrm{PP}$, evaluated using the differentiable SNN. Building on this differentiable sampling framework, we focus on deriving an online training algorithm that differs from BPTT-style updates within Kajino's variational framework, and on analyzing its mean-field dynamics and truncation error.

4 Filtered-trace online variational training

This section derives the proposed online training algorithm for variational probabilistic SNNs. Starting from the discrete-time ELBO objective, we develop per-time-step update rules for both the generative parameters and the inference parameters, so that learning can be carried out online without backpropagating through the entire sequence.

4.1 Markovian SRM dynamics and local traces

We consider a discrete-time approximation with bin width Δt and time index $t = 1, \ldots, T$. We denote by B the number of exponential basis functions used to approximate the continuous-time synaptic filter. Following the standard SRM representation with exponentially decaying synaptic filters [8], we introduce a multi-exponential template for scalar synaptic kernels:

$$\kappa(s) \approx \sum_{k=1}^{B} c_k e^{-s/\rho_k}, \tag{7}$$

where $c_k \in \mathbb{R}$ are basis coefficients and $\rho_k > 0$ are time constants. For each synaptic connection (j, i), we use the same exponential basis with connection-specific coefficients $c_k w_{ji,k}$. This yields $\alpha_k := e^{-\Delta t/\rho_k} \in (0, 1)$. For each presynaptic neuron i and basis index $k \in [B] := \{1, \ldots, B\}$, let z_i^t denote the spike variable in bin t, and maintain the exponentially filtered synaptic state $x_{i,k}^t = \alpha_k x_{i,k}^{t-1} + z_i^{t-1}$, with initialization $x_{i,k}^0 = 0$ and the convention $z_i^0 = 0$, so that $x_{i,k}^1 = 0$. The membrane potential of neuron j is then

$$u_j^t = b_j + \sum_i \sum_{k=1}^{B} (c_k w_{ji,k})\, x_{i,k}^t, \tag{8}$$

and the discretized conditional intensity in bin t is parameterized as $\lambda_j^t = \sigma(u_j^t)$, where b_j is the discrete-time baseline potential and $\sigma(\cdot)$ is the same positive nonlinearity as defined earlier.

The synaptic states $\{x_{i,k}^t\}_{i,k}$ define a finite-dimensional Markov state-space model: the collection of states $\{x_{i,k}^t\}_{i,k}$ is sufficient to advance the dynamics without storing the full spike history. This Markovian structure is the key that enables online training. From (8), the local derivative of the membrane potential with respect to a synaptic parameter is

$$\frac{\partial u_j^t}{\partial w_{ji,k}} = c_k x_{i,k}^t, \tag{9}$$

which is available online because $x_{i,k}^t$ is updated forward in time via the synaptic states $\{x_{i,k}^t\}_{i,k}$. We will use (9) as the synapse-level trace factor in the online gradient rules.

4.1.1 Online updates for generative parameters θ

Our objective is to minimize the negative ELBO, $E(\theta, \phi) := -\mathrm{ELBO}(\theta, \phi)$. For a fixed observed realization $\mathcal{T}_O$ and a sampled hidden completion $\mathcal{T}_H \sim q_\phi(\cdot \mid \mathcal{T}_O)$, when deriving the update for the generative parameters θ, we only need the sample-wise negative joint log-likelihood $E_\theta(\mathcal{T}_O, \mathcal{T}_H) := -\log p_\theta(\mathcal{T}_O, \mathcal{T}_H)$. Using the discrete-time approximation with bin width Δt (and dropping additive constants independent of θ), E_θ decomposes into a sum of per-neuron, per-bin contributions, $E_\theta \approx \sum_{t=1}^{T} \sum_j E_{\theta,j}^t$. Let $\lambda_{p,j}^t = \sigma(u_j^t)$ denote the conditional intensity of the generating process, where

$$E_{\theta,j}^t = \lambda_{p,j}^t \Delta t - z_j^t \log \lambda_{p,j}^t. \tag{10}$$

We obtain the neuron-level learning signal for neuron j at time t

$$L_{\theta,j}^t \;:=\; \frac{\partial E_{\theta,j}^t}{\partial u_j^t} = (\lambda_{p,j}^t \Delta t - z_j^t) \frac{\sigma'(u_j^t)}{\lambda_{p,j}^t}, \tag{11}$$

and combining (11) with the local trace (9) yields, for the weight $w_{ji,k}^{(\theta)}$ in the generative model,

$$\frac{\partial E_\theta}{\partial w_{ji,k}^{(\theta)}} = \sum_{t=1}^{T} L_{\theta,j}^t \, (c_k x_{i,k}^t). \tag{12}$$

Hence the fully online update is

$$w_{ji,k}^{(\theta)} \leftarrow w_{ji,k}^{(\theta)} - \eta_t \, L_{\theta,j}^t \, (c_k x_{i,k}^t), \tag{13}$$

where η_t is a learning rate.

4.1.2 Online updates for variational parameters ϕ

The ϕ-gradient is more involved because the hidden trajectory is sampled from q_ϕ. We use the reparameterization form and draw a single hidden trajectory via $\mathcal{T}_H = G_\phi(\xi; \mathcal{T}_O)$, where $G_\phi(\cdot)$ is a differentiable sampling function and $\xi \sim p(\xi)$ is independent of ϕ. The corresponding sample-wise negative-ELBO objective for a single noise draw is

$$\tilde{E}(\theta, \phi; \xi) := -\log p_\theta\big(\mathcal{T}_O, G_\phi(\xi; \mathcal{T}_O)\big) + \log q_\phi\big(G_\phi(\xi; \mathcal{T}_O) \mid \mathcal{T}_O\big). \tag{14}$$

For the sampled hidden trajectory $\mathcal{T}_H = G_\phi(\xi; \mathcal{T}_O)$, the same quantity may be written as

$$\tilde{E}(\theta, \phi; \mathcal{T}_H) = -\log p_\theta(\mathcal{T}_O, \mathcal{T}_H) + \log q_\phi(\mathcal{T}_H \mid \mathcal{T}_O). \tag{15}$$

For the inference SNN, we denote the corresponding membrane potential and intensity by $u_{q,j}^t$ and $\lambda_{q,j}^t = \sigma(u_{q,j}^t)$, respectively. For the inference parameters $w_{ji,k}^{(\phi)}$, we have $\frac{\partial \tilde{E}}{\partial w_{ji,k}^{(\phi)}} = \sum_{t=1}^{T} \frac{d\tilde{E}}{du_{q,j}^t} \cdot \frac{\partial u_{q,j}^t}{\partial w_{ji,k}^{(\phi)}}$. For the online update, we retain the diagonal component of the instantaneous path-wise sampling Jacobian, while temporal credit is handled through the filtered-trace construction below. This gives

$$\frac{d\tilde{E}}{du_{q,j}^t} \approx \frac{d\tilde{E}}{dz_j^t} \frac{\partial z_j^t}{\partial u_{q,j}^t} + \frac{\partial \tilde{E}}{\partial u_{q,j}^t}. \tag{16}$$

Combining (16) with (9) applied to the inference membrane potential, we obtain:

$$\frac{\partial \tilde{E}}{\partial w_{ji,k}^{(\phi)}} \approx \sum_{t=1}^{T} \left[L_{\phi,j}^{t,\mathrm{ideal}} \, e_{\phi,ji,k}^t + \frac{\partial \tilde{E}}{\partial u_{q,j}^t} \, (c_k x_{i,k}^t) \right], \tag{17}$$

where $L_{\phi,j}^{t,\mathrm{ideal}} := \frac{d\tilde{E}}{dz_j^t}$, $e_{\phi,ji,k}^t := \frac{\partial z_j^t}{\partial u_{q,j}^t}(c_k x_{i,k}^t)$. To obtain a strictly online training rule, we replace the non-causal total derivative with a local partial derivative, an approximation similar in spirit to e-prop-style online credit assignment [2] but applied here to the variational inference update in probabilistic SNNs: $L_{\phi,j}^{t,\mathrm{ideal}} = \frac{d\tilde{E}}{dz_j^t} \approx \hat{L}_{\phi,j}^t := \frac{\partial \tilde{E}}{\partial z_j^t}$, thereby obtaining an online estimator

$$\frac{\partial \tilde{E}}{\partial w_{ji,k}^{(\phi)}} \approx \sum_{t=1}^{T} \Big[\hat{L}_{\phi,j}^t \, e_{\phi,ji,k}^t + D_{q,j}^t \, (c_k x_{i,k}^t) \Big], \tag{18}$$

where the explicit term is

$$D^t_{q,j} := \frac{\partial \tilde{E}}{\partial u^t_{q,j}} = \frac{\partial \log q_\phi(\mathcal{T}_H \mid \mathcal{T}_O)}{\partial u^t_{q,j}} = (z^t_j - \lambda^t_{q,j}\Delta t)\frac{\sigma'(u^t_{q,j})}{\lambda^t_{q,j}}. \tag{19}$$

Here the derivative is the explicit local derivative with respect to the current inference membrane potential $u^t_{q,j}$, i.e., future-state dependencies are not included in this term. Throughout the variational update derivation, $j \in H$. The generative-feedback sums run over $\ell \in O \cup H$, whereas the variational-feedback sums run over $\ell \in H$. When both contributions appear under a common summation, the variational-feedback term is understood to vanish for $\ell \in O$.

Direct learning signal. A minimal fully online choice keeps only the instantaneous dependence of $\tilde{E}$ on z^t_j through the per-bin likelihood terms, resulting in the log-ratio learning signal

$$\hat{L}^{t,\mathrm{dir}}_{\phi,j} \approx -\log \lambda^t_{p,j} + \log \lambda^t_{q,j} = \log \frac{\lambda^t_{q,j}}{\lambda^t_{p,j}}. \tag{20}$$

Filtered-trace correction. The partial credit signal $\partial\tilde{E}/\partial z^t_j$ generally contains future contributions, because a hidden spike at time t perturbs later membrane potentials through the synaptic filters. Under the multi-exponential SRM state representation, these future-dependent terms decompose into a generative-feedback part and a variational-feedback part:

$$\frac{\partial \tilde{E}}{\partial z^t_j} = \log \frac{\lambda^t_{q,j}}{\lambda^t_{p,j}} + \sum_\ell \sum_{t'>t} L^{t'}_{\theta,\ell} \sum_{k'} (c_{k'} w^{(\theta)}_{\ell j,k'}) \alpha_{k'}^{t'-t-1} + \sum_\ell \sum_{t'>t} D^{t'}_{q,\ell} \sum_{k'} (c_{k'} w^{(\phi)}_{\ell j,k'}) \alpha_{k'}^{t'-t-1}. \tag{21}$$

Reordering the future-dependent double sums in (21) shows that these terms can be expressed in terms of exponentially filtered eligibility traces. This observation enables an exact causal re-indexing of the cumulative future correction. For each pair of decay indices (k, k'), we define the filtered eligibility trace

$$\bar{e}^{t,k'}_{\phi,ji,k} := \sum_{\tau<t} \alpha_{k'}^{t-\tau-1}\, e^\tau_{\phi,ji,k}, \qquad \bar{e}^{1,k'}_{\phi,ji,k} = 0, \tag{22}$$

which is updated online by $\bar{e}^{t+1,k'}_{\phi,ji,k} = \alpha_{k'}\, \bar{e}^{t,k'}_{\phi,ji,k} + e^t_{\phi,ji,k}$. We also define a per-time feedback coefficient for the filtered-trace correction,

$$F^{t,k'}_{j,\mathrm{fb}} := \sum_\ell c_{k'} \Big(w^{(\theta)}_{\ell j,k'} L^t_{\theta,\ell} + w^{(\phi)}_{\ell j,k'} D^t_{q,\ell} \Big), \tag{23}$$

where $L^t_{\theta,\ell}$ is the generative learning signal in (11), and $D^t_{q,\ell}$ is the explicit inference term defined in (19).

Final online update for ϕ. The resulting online update consists of three parts: an instantaneous log-ratio term, a filtered future-feedback correction, and the explicit inference term. Our final streaming update for inference parameters is

$$w^{(\phi)}_{ji,k} \leftarrow w^{(\phi)}_{ji,k} - \eta_t \Big[\hat{L}^{t,\mathrm{dir}}_{\phi,j}\, e^t_{\phi,ji,k} + \sum_{k'} F^{t,k'}_{j,\mathrm{fb}}\, \bar{e}^{t,k'}_{\phi,ji,k} + D^t_{q,j}\,(c_k x^t_{i,k}) \Big]. \tag{24}$$

For supervised classification, the filtered-trace construction uses local gradients of the supervised variational objective as feedback signals.

Memory complexity. For a fixed batch size, the synaptic and local eligibility states scale as $O(EB)$, while the filtered eligibility traces scale as $O(EB^2)$, where E is the number of trainable synapses and B is the number of exponential basis functions. Importantly, these online states have fixed size with respect to the sequence length T, avoiding the temporal activation storage required by full-sequence BPTT.

5 Theory of filtered-trace online variational training

Conditioned on the local approximations adopted in Section 4, this section analyzes the long-run behavior of the resulting filtered-trace online training rules. For the full-horizon case $R = \infty$, the cumulative future-credit correction admits an exact causal re-indexing through the filtered-trace

recursion, yielding a Markovian stochastic-approximation (SA) recursion that tracks a limiting ordinary differential equation (ODE) under standard conditions [18, 5]. We then introduce a horizon-R family of truncated online rules and compare their mean-field dynamics with the full-horizon rule. This separation isolates the approximation specifically attributable to truncating future-dependent credit assignment and allows us to characterize how the finite-horizon dynamics approach the full-horizon online limit. These local approximations are shared by all members of this estimator family and are therefore separated from the horizon-dependent effect analyzed here. All technical derivations are deferred to Appendix A.

5.1 Markovian stochastic-approximation formulation

Stochastic-approximation form. Let $\psi := (\theta, \phi) \in \mathbb{R}^d$ collect all learnable parameters, and let ψ_t denote the parameter iterate at time t. Let $\mathcal{S}$ be the state space of the extended Markovian online dynamics, and let $S_t \in \mathcal{S}$ denote the state at time t. The state S_t is generated by the finite-dimensional SRM trace dynamics in Section 4.1, augmented with the auxiliary traces used by the online update rule. For each fixed ψ and horizon R, these trace recursions together with the differentiable sampling mechanism define a transition kernel P_ψ on $\mathcal{S}$, with the dependence on R suppressed in the notation. We cast the recursion in the standard controlled Markov stochastic-approximation form [5]. Then the online algorithm can be written as

$$\psi_{t+1} \;=\; \psi_t - \eta_t \, g^{(R)}(S_t, \psi_t), \tag{25}$$

where $\eta_t > 0$ is the step-size sequence. For the theoretical analysis, we assume that the iterates remain in a compact set $\mathcal{K} \subset \mathbb{R}^d$, which may be enforced by projection or treated as a standard bounded-iterate condition. We use $g^{(\infty)}$ to denote the untruncated full-horizon per-step update direction induced by the online rule in (13) and (24). The future-dependent part of this direction is realized online by the filtered-trace construction in (22) and (23). For the auxiliary truncated analysis, $g^{(R)}$ denotes the corresponding direction in which future-dependent contributions beyond horizon R are removed from the ϕ-learning signal. For finite R and for $R = \infty$, the update direction preserves the same local eligibility structure and contains both generative and variational feedback components.

Horizon-truncated variants. The online rule in (13) and (24) corresponds to the full-horizon case $R = \infty$, for which the filtered-trace construction gives an exact causal re-indexing of the cumulative future-dependent correction in (21). To study the effect of truncating this correction, we introduce an auxiliary family of horizon-limited variants indexed by $R \in \{-1, 0, 1, 2, \dots\} \cup \{\infty\}$, where R denotes the credit-assignment horizon. Specifically,

- $R = -1$: the future-feedback correction is omitted from the ϕ-update;
- finite R: future-dependent terms with $t' \le t + R + 1$ are retained;
- $R = \infty$: the full future-dependent correction is retained.

Thus, finite R defines an auxiliary truncation family for comparing approximate and full-horizon dynamics. Let $\hat{L}_{\phi,j}^{t,(\infty)}$ denote the full learning signal realized by the filtered-trace construction. For a finite horizon $R \in \{-1, 0, 1, \dots\}$, let $\hat{L}_{\phi,j}^{t,(R)}$ denote the corresponding truncated signal obtained by removing future-dependent contributions beyond time $t + R + 1$. The associated per-step update direction is denoted by $g^{(R)}$, with $g^{(\infty)}$ corresponding to the untruncated online rule.

Mean-field drift and limiting ODE. With the update direction $g^{(R)}$ in place, we can define the deterministic mean-field dynamics associated with the online recursion. For each fixed ψ, let P_ψ denote the Markov transition kernel on $\mathcal{S}$, and let π_ψ be an invariant distribution of this kernel. We then define the mean-field drift

$$h^{(R)}(\psi) := \mathbb{E}_{S \sim \pi_\psi}[g^{(R)}(S, \psi)], \tag{26}$$

and the associated limiting ODE

$$\dot{\psi} = -\, h^{(R)}(\psi). \tag{27}$$

The next subsection shows that, for any fixed R, the online iterates track this ODE asymptotically.

5.2 Mean-field dynamics of the online algorithm

This subsection establishes three connected results on the dynamics of the online algorithm. We first show that, for a fixed horizon R, the stochastic recursion tracks its associated mean-field ODE; we then show that the gap between the truncated and full-horizon mean-field drifts is controlled by a geometrically decaying tail bound; finally, we combine the two results to interpret the truncated recursion as a bounded perturbation of the full-horizon dynamics.

We first consider the dynamics associated with a fixed horizon R. Once the online recursion has been written in SA form and its mean-field drift $h^{(R)}$ has been defined, the natural question is whether the stochastic iterates $\{\psi_t\}$ follow the deterministic flow generated by (27). The following proposition establishes this ODE-tracking property.

Proposition 5.1 (ODE tracking under Markovian noise)**.** *Fix $R \in \{-1, 0, 1, \dots\} \cup \{\infty\}$ and consider the recursion in* (25)*. Under Assumptions A.1 and A.2, stated in Appendix A.1, the piecewise-linear interpolation of the iterates $\{\psi_t\}$ is an asymptotic pseudo-trajectory of the ODE* (27)*. Consequently, by the standard asymptotic-pseudo-trajectory theory for stochastic approximation [3, 4], its limit set is internally chain transitive for* (27)*.*

The proposition above characterizes the dynamics associated with a fixed horizon R. We next compare the truncated mean-field drift $h^{(R)}$ with the full-horizon drift $h^{(\infty)}$, which corresponds to the untruncated online rule. The following theorem shows that this discrepancy decays exponentially with the truncation horizon.

Theorem 5.2 (Exponential decay of truncation bias)**.** *Under Assumptions A.2 and A.3, define $\bar{\alpha} := \max_k \alpha_k < 1$. Let $h^{(\infty)}$ and $h^{(R)}$ be the mean-field drifts defined in* (26)*. Then there exists a constant $C < \infty$ such that, for every $R \in \{-1, 0, 1, 2, \dots\}$,*

$$\sup_{\psi \in \mathcal{K}} \left\| h^{(R)}(\psi) - h^{(\infty)}(\psi) \right\| \;\le\; C \, \frac{\bar{\alpha}^{R+1}}{1 - \bar{\alpha}}. \tag{28}$$

We now combine Proposition 5.1 and Theorem 5.2 to obtain the main dynamical consequence of truncation. Since the truncated drift differs from the full-horizon drift by at most ε_R, each finite-horizon recursion can be viewed as a bounded perturbation of the full-horizon mean-field dynamics. The perturbation size decreases geometrically as R increases, while the case $R = -1$ corresponds to the direct-only boundary case. Equation 28 therefore characterizes the horizon-dependent component of the approximation: as R increases, the truncated mean-field drift approaches the full-horizon online drift at a geometric rate governed by the synaptic decay factor. Since the local approximations are common to all values of R, the bound specifically quantifies the additional effect introduced by finite-horizon truncation.

Corollary 5.3 (Perturbation of full-horizon dynamics)**.** *Under Assumptions A.1–A.3, for any finite horizon $R \in \{-1, 0, 1, 2, \dots\}$, define $\varepsilon_R := \sup_{\psi \in \mathcal{K}} \|h^{(R)}(\psi) - h^{(\infty)}(\psi)\|$. Then* (28) *implies that the interpolated iterates form an asymptotic pseudo-trajectory of the differential inclusion*

$$\dot{\psi} \in -h^{(\infty)}(\psi) + \varepsilon_R \, \mathbb{B}, \tag{29}$$

in the sense of standard stochastic-approximation and differential-inclusion theory [4], where $\mathbb{B} := \{v \in \mathbb{R}^d : \|v\| \le 1\}$. In particular, the truncated dynamics is an $O(\varepsilon_R)$ perturbation of the full-horizon mean-field dynamics, where ε_R is controlled by

$$\varepsilon_R \le C \, \frac{\bar{\alpha}^{R+1}}{1 - \bar{\alpha}}. \tag{30}$$

6 Experiments

We conduct experiments on synthetic spike sequences and N-MNIST [27]. The synthetic study evaluates how the truncation horizon R affects optimization and truncation bias, while the N-MNIST study examines the practical accuracy–memory–runtime trade-off of the proposed online method, including different credit-assignment horizons and comparisons with BPTT and a score-function variational baseline. These experiments are designed to validate the theoretical predictions and assess the practical accuracy–memory–runtime trade-offs of the proposed online training framework.

Additional experiments further diagnose the gradient-level agreement between the online update and BPTT under different truncation horizons and synaptic decay settings (Appendix A.6, A.7). We additionally report results on the more temporally demanding DVS Gesture [1] dataset in Appendix A.9, providing a complementary evaluation of the filtered-trace correction when temporal dependencies are stronger.

6.1 Experimental setup

Datasets. For the synthetic study, we generate spike sequences from the discrete-time SRM approximation of the probabilistic SNN model described in Section 4.1. The network size is set to $N = 6$, including $|O| = 2$ observed neurons and $|H| = 4$ hidden neurons. In the discrete-time SRM implementation, we use $B = 2$ exponential basis functions with time constants $\rho_1 = 2.0$ and $\rho_2 = 5.0$, and the sequence length is set to $T = 50$. The Concrete relaxation temperature is set to $\tau_{\text{SNN}} = 0.3$, and we use a learning rate of 10^{-3}. For the real-data study, we use the official training and test splits of the N-MNIST dataset [27]. The N-MNIST dataset contains neuromorphic recordings of 10 digit classes, comprising 60,000 training and 10,000 test samples. Each frame has spatial size $2 \times 34 \times 34$ and is flattened into a 2312-dimensional input vector.

Experimental details. For the synthetic experiments, we use an SRM network with observed and hidden neurons and exponential synaptic traces. For N-MNIST, we use an input projection, an SRM hidden layer, and an output classifier with $T = 20$, 256 hidden units, batch size 128, and $B = 2$ exponential bases with $\{\rho_1, \rho_2\} = \{1.0, 1.25\}$. Training uses per-timestep cross-entropy with KL regularization, while predictions are obtained by averaging output logits over time. We use Adam with a learning rate of 10^{-3} and weight decay of 10^{-5}. All N-MNIST experiments are trained for 50 epochs. The main N-MNIST results are reported as mean $\pm$ standard deviation over five matched random seeds.

The BPTT baseline is a full-sequence backpropagation implementation adapted from Kajino's differentiable point-process framework [16], using the same SRM architecture and relaxation setting. We additionally include a score-function (REINFORCE) variational baseline following Jimenez Rezende and Gerstner [15], which uses the same probabilistic SRM, architecture, dataset, and training objective but replaces the path-wise relaxed gradient with discrete spike sampling and a score-function estimator. For the online method, we evaluate $R \in \{-1, 0, 1, 2\}$ under the same training configuration. For the memory-scaling experiments in Fig. 1 (c,d), we retain the direct online variant $R = -1$ to isolate the sequence-length scaling of the online recursion with minimal trace overhead.

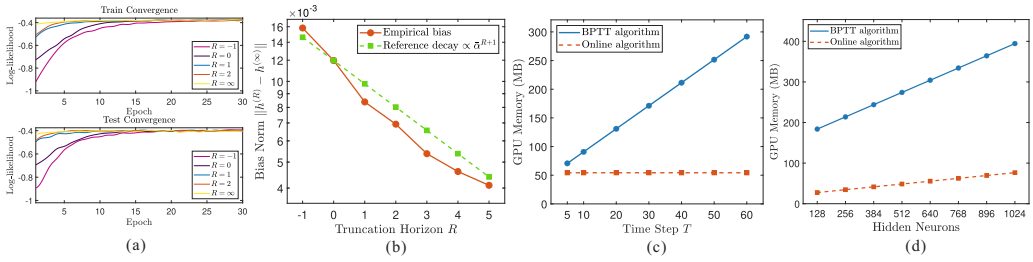

Figure 1: Experimental evaluation of filtered-trace online variational training. (a) Training and test average predictive log-likelihood under different truncation horizons R. (b) Empirical bias norm $\|h^{(R)} - h^{(\infty)}\|$ and the exponential reference decay. (c) GPU memory as the sequence length T increases under a fixed network architecture. (d) GPU memory as the number of hidden neurons increases under fixed T.

6.2 Synthetic experiments for theoretical validation

We first use synthetic data to examine the horizon-dependent learning behavior and the truncation-bias decay predicted by our theoretical analysis in Section 5.

Effect of truncation horizon R. Figure 1 (a) shows the training and test average predictive log-likelihood of the observed spike trains under different truncation horizons $R \in \{-1, 0, 1, 2, \infty\}$. Here, $R = -1$ denotes the direct-only boundary case, which retains only the instantaneous learning

signal and discards all future-dependent correction terms. All choices of R lead to stable learning, indicating that even the direct-only signal provides a usable online update. Increasing R mainly improves the early-stage optimization behavior: the curves with larger finite horizons approach the full-horizon case $R = \infty$ faster than the direct-only and shorter-horizon variants, and this trend is consistent on both the training and test sets. These results suggest that future-dependent credit assignment provides more informative learning signals and improves empirical optimization efficiency.

Exponential decay of truncation bias. Figure 1 (b) plots the empirical bias norm $\|h^{(R)} - h^{(\infty)}\|$ as a function of R in log scale, together with a normalized theoretical reference proportional to $\bar{\alpha}^{R+1}$. We estimate $h^{(R)}$ by averaging the per-step update direction $g^{(R)}$ over the same sampled states and noise realizations, and report the norm of the difference from the corresponding full-horizon estimate. The direct-only case $R = -1$ gives the largest bias, while the empirical bias decreases monotonically as R increases and closely follows the predicted exponential trend. This observation is consistent with our theoretical result that the truncation bias is controlled by a geometrically decaying tail bound.

6.3 Real-data evaluation on N-MNIST

We evaluate the practical memory behavior and predictive performance of the proposed online algorithm on N-MNIST. Figure 1 (c,d) first examines memory scaling using the lightweight $R = -1$ variant. As the sequence length T increases, the peak memory of BPTT grows steadily because intermediate activations must be retained throughout the temporal unrolling, whereas the online method maintains only the current neuronal states and fixed-size traces, resulting in nearly constant memory with respect to T. When varying the hidden dimension at fixed $T = 60$, both methods require more memory for wider networks, but BPTT grows more rapidly because its activation storage scales jointly with network width and sequence length.

Table 1: Comparison of different online credit-assignment horizons, REINFORCE, and BPTT on N-MNIST under the same architecture and training configuration.

	REINFORCE	$R = -1$	$R = 0$	$R = 1$	$R = 2$	BPTT
Accuracy (%)	80.10 ± 2.06	98.12 ± 0.03	98.30 ± 0.04	98.42 ± 0.04	98.44 ± 0.06	98.48 ± 0.04
Time/epoch (s)	11.69	14.47	17.09	18.01	18.82	12.82
Peak memory (MB)	81.7	48.8	50.3	51.8	53.3	78.9

Table 1 compares the score-function baseline, different online credit-assignment horizons, and BPTT under the same N-MNIST configuration. REINFORCE exhibits substantially lower accuracy (80.10 $\pm$ 2.06%) and considerably larger variability across runs, consistent with the known challenges of score-function gradient estimation. In contrast, the path-wise online variants approach BPTT accuracy while using substantially less peak GPU memory. Increasing the future-credit horizon improves the mean accuracy from 98.12 $\pm$ 0.03% at $R = -1$ to 98.44 $\pm$ 0.06% at $R = 2$, compared with 98.48 $\pm$ 0.04% for BPTT. Most of this improvement is already achieved at $R = 1$ (98.42 $\pm$ 0.04%), suggesting that relatively short horizons can capture useful future-dependent credit. This improvement comes with additional trace overhead: peak memory increases from 48.8 MB to 53.3 MB, while runtime per epoch increases from 14.47 s to 18.82 s. Overall, R provides a controllable accuracy–runtime–memory trade-off, while the online formulation avoids sequence-length-dependent activation storage.

7 Conclusion

We presented an online variational training framework for probabilistic SNNs under the differentiable point-process formulation. Finite-dimensional Markovian SRM traces enable streaming updates of generative and variational parameters without full temporal unrolling. A horizon-R filtered-trace rule handles future-dependent credit assignment, and our stochastic-approximation analysis establishes exponential decay of the additional truncation-induced mean-field bias. Synthetic experiments support the predicted horizon-dependent behavior, while five-seed N-MNIST results demonstrate near-BPTT accuracy with substantially lower peak memory and moderate trace overhead. These results support scalable online variational training for probabilistic SNNs.

Acknowledgments and Disclosure of Funding

We thank the Area Chair and the anonymous reviewers for their constructive comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant No. 12402032. The authors declare no competing interests.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7388–7397. IEEE, 2017.

[2] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.

[3] Michel Benaïm. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII*, pages 1–68. Springer, 1999.

[4] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[5] Vivek S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 100. Springer, 2008.

[6] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1555–1564, 2016.

[7] 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, 2023.

[8] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[9] Wenzhe Guo, Mohammed E Fouda, Ahmed M Eltawil, and Khaled Nabil Salama. Efficient training of spiking neural networks with temporally-truncated local backpropagation through time. *Frontiers in neuroscience*, 17:1047008, 2023.

[10] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[11] Hyeryung Jang and Osvaldo Simeone. Multisample online learning for probabilistic spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2034–2044, 2022.

[12] Hyeryung Jang, Osvaldo Simeone, Brian Gardner, and Andre Gruning. An introduction to probabilistic spiking neural networks: Probabilistic models, learning rules, and applications. *IEEE Signal Processing Magazine*, 36(6):64–77, 2019.

[13] Hyeryung Jang, Nicolas Skatchkovsky, and Osvaldo Simeone. Vowel: A local online learning rule for recurrent networks of probabilistic spiking winner-take-all circuits. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4597–4604. IEEE, 2021.

[14] Haiyan Jiang, Giulia De Masi, Huan Xiong, and Bin Gu. Ndot: neuronal dynamics-based online training for spiking neural networks. In *Forty-first International Conference on Machine Learning*, 2024.

[15] Danilo Jimenez Rezende and Wulfram Gerstner. Stochastic variational learning in recurrent spiking networks. *Frontiers in computational neuroscience*, 8:38, 2014.

[16] Hiroshi Kajino. A differentiable point process with its application to spiking neural networks. In *International Conference on Machine Learning*, pages 5226–5235. PMLR, 2021.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Harold Joseph Kushner and Dean S Clark. *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012.

[19] Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13948–13958, 2025.

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

[21] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[22] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

[23] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Towards memory-and time-efficient backpropagation for training spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6166–6176, 2023.

[24] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

[25] James M Murray. Local online learning in recurrent networks with random feedback. *Elife*, 8: e43299, 2019.

[26] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[27] 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.

[28] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243, 2004.

[29] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[30] Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. *arXiv preprint arXiv:2104.03528*, 2021.

[31] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

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

[33] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

[34] Xiangwen Wang, Xianghong Lin, and Xiaochao Dang. Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Networks*, 125:258–280, 2020.

[35] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1761–1771, 2023.

[36] Peixi Wu, Bosong Chai, Hebei Li, Menghua Zheng, Yansong Peng, Zeyu Wang, Xuan Nie, Yueyi Zhang, and Xiaoyan Sun. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21563–21571, 2025.

[37] 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.

[38] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Online training through time for spiking neural networks. *Advances in neural information processing systems*, 35:20717–20730, 2022.

[39] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain sciences*, 12(7):863, 2022.

[40] Chengting Yu, Lei Liu, Gaoang Wang, Erping Li, and Aili Wang. Advancing training efficiency of deep spiking neural networks through rate-based backpropagation. *Advances in Neural Information Processing Systems*, 37:115786–115815, 2024.

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

[42] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

A Proofs and additional experimental results

This appendix provides the proofs of the theoretical results in Section 5 (Sections A.1–A.5) and presents additional experiments (Sections A.6–A.9) that further examine the proposed truncated online credit-assignment rule, its gradient-level behavior, its sensitivity to the SRM basis configuration, and its performance on temporally structured real data. The proofs use standard tools from Markovian stochastic approximation and asymptotic-pseudo-trajectory analysis [18, 5, 3, 4]. Throughout the theoretical analysis and the additional gradient-alignment experiments, we use $R \in \{-1, 0, 1, 2, \dots\} \cup \{\infty\}$ to denote the future-credit horizon: $R = -1$ corresponds to the direct instantaneous signal only, finite R corresponds to explicit truncation at horizon R, and $R = \infty$ denotes the full future-dependent correction. With the filtered-trace recursion derived in Section 4, the full-horizon case $R = \infty$ can be implemented causally without explicitly storing or revisiting the entire future sequence.

Implementation details. All experiments were run on a single NVIDIA GeForce RTX 5070 Ti GPU using a PyTorch implementation, with peak GPU memory measured via PyTorch CUDA memory statistics; standard libraries including PyTorch and NumPy were used throughout. For the neuromorphic datasets N-MNIST and DVS Gesture, asynchronous event streams were discretized into synchronous time frames using SpikingJelly (version 0.0.0.0.14, distributed under the Open-Intelligence Open Source License Version 1.0) [7]. We use the official training/test splits for both datasets. N-MNIST [27] and DVS Gesture [1] are publicly available benchmarks, released under the Creative Commons Attribution-ShareAlike 4.0 and Creative Commons Attribution 4.0 licenses, respectively.

A.1 Unified setup and assumptions

Let

$$\psi := (\theta, \phi)$$

collect all trainable parameters. Let S_t denote the full Markov state at time t, collecting all variables required to specify the next-step transition and the per-step update. These include the SRM kernel states, membrane potentials, conditional intensities, spikes or relaxed spikes, and the filtered eligibility

states used by the filtered-trace recursion. Let $\mathcal{S}$ denote the state space of the Markov state S_t. For a given horizon R, the online training rule can be written in the unified form

$$\psi_{t+1} = \psi_t - \eta_t \, g^{(R)}(S_t, \psi_t), \tag{31}$$

where $g^{(R)}$ is the corresponding per-step update direction. For each fixed finite R, the full and truncated update directions are compared on a common augmented Markov state containing both filtered traces, driven by the same underlying SRM trajectory at fixed ψ. The invariant distribution π_ψ in this comparison is understood as that of the coupled state, with its dependence on R suppressed. Its marginals recover the stationary distributions of the respective trace recursions.

For each fixed ψ, let P_ψ be the Markov transition kernel of (S_t), and let

$$h^{(R)}(\psi) := \mathbb{E}_{S\sim\pi_\psi}\Big[g^{(R)}(S,\psi)\Big] \tag{32}$$

denote the mean-field drift, where π_ψ is an invariant distribution of P_ψ.

We use the following grouped appendix-level assumptions. Assumptions A.1–A.2 are standard technical conditions for applying the ODE method to stochastic approximation with controlled Markov noise [18, 5], rather than additional algorithmic requirements. Assumption A.3 is specific to the truncation-bias analysis; in the SRM setting, the decay condition follows from the exponentially decaying synaptic traces, and the boundedness conditions are satisfied on compact parameter regions with well-behaved conditional intensities. An additional robustness condition is introduced later only for the final neighborhood-convergence statement.

Assumption A.1 (Step sizes)**.** The step sizes satisfy

$$\eta_t > 0, \qquad \sum_{t=0}^{\infty} \eta_t = \infty, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty, \qquad \eta_t \to 0.$$

Assumption A.2 (Standard regularity for controlled-Markov stochastic approximation)**.** There exists a compact set $\mathcal{K} \subset \mathbb{R}^d$ such that $\psi_t \in \mathcal{K}$ almost surely for all t. For each fixed $\psi \in \mathcal{K}$, the state recursion $S_{t+1} \sim P_\psi(S_t, \cdot)$ defines a controlled Markov chain on $\mathcal{S}$ with a unique invariant distribution π_ψ. Moreover, uniformly on $\mathcal{K}$, the kernel family $\{P_\psi\}$ and the update map $g^{(R)}(S,\psi)$ satisfy the standard regularity conditions used in the ODE method for stochastic approximation with controlled Markov noise; in particular, the mean field

$$h^{(R)}(\psi) := \mathbb{E}_{S\sim\pi_\psi}[g^{(R)}(S,\psi)]$$

is well defined and locally Lipschitz on $\mathcal{K}$, and the associated Markov-noise terms admit the usual Poisson-equation decomposition with uniformly bounded second moments. These regularity conditions are used to define the stationary mean field and establish the asymptotic ODE characterization; the finite-prefix filtered-trace recursion itself is defined independently of this stationary-regime analysis.

Assumption A.3 (Boundedness and decay conditions for the truncation analysis)**.** There exist finite constants $L_{\max}, D_{\max}, W_{\max}, e_{\max} > 0$ such that, for all relevant indices and time steps,

$$|L^t_{\theta,\ell}| \le L_{\max}, \qquad |D^t_{q,\ell}| \le D_{\max},$$

$$|c_{k'} w^{(\theta)}_{\ell j,k'}| \le W_{\max}, \qquad |c_{k'} w^{(\phi)}_{\ell j,k'}| \le W_{\max},$$

and

$$|e^t_{\phi,ji,k}| \le e_{\max}.$$

In addition, define

$$\bar{\alpha} := \max_k \alpha_k.$$

We assume $\bar{\alpha} < 1$.

A.2 A lemma for the filtered-trace recursion

Lemma A.4 (Exact causal re-indexing of the future-credit correction)**.** *Let $\hat{L}^{t,(\infty)}_{\phi,j}$ denote the full future-dependent learning signal, i.e.,*

$$\hat{L}^{t,(\infty)}_{\phi,j} = \log \frac{\lambda^t_{q,j}}{\lambda^t_{p,j}} + \sum_{\ell} \sum_{t'>t} \sum_{k'} \Big[(c_{k'} w^{(\theta)}_{\ell j,k'}) L^{t'}_{\theta,\ell} + (c_{k'} w^{(\phi)}_{\ell j,k'}) D^{t'}_{q,\ell}\Big] \alpha_{k'}^{t'-t-1}. \tag{33}$$

For any finite horizon T, the cumulative future-credit correction obtained by restricting the sum in (33) *to $t' < T$ admits an exact causal re-indexing through the filtered-trace recursion derived in Section 4. This re-indexing is an algebraic identity and introduces no additional approximation.*

Proof. For notational convenience, we use zero-based indexing. Fix any parameter component $w_{ji,k}^{(\phi)}$ and a finite horizon $T \geq 1$. We consider the unweighted cumulative future-credit correction over $t = 0, \ldots, T-1$, retaining only future contributions within this horizon:

$$\sum_{t=0}^{T-1} \left(\sum_{\ell} \sum_{t'=t+1}^{T-1} \sum_{k'} \left[(c_{k'} w_{\ell j,k'}^{(\theta)}) L_{\theta,\ell}^{t'} + (c_{k'} w_{\ell j,k'}^{(\phi)}) D_{q,\ell}^{t'} \right] \alpha_{k'}^{t'-t-1} \right) e_{\phi,ji,k}^{t}. \tag{34}$$

All sums in (34) are finite, hence their order may be exchanged without any additional condition. Reordering the sums gives

$$\begin{aligned} (34) &= \sum_{t'=0}^{T-1} \sum_{k'} \left(\sum_{\ell} \left[(c_{k'} w_{\ell j,k'}^{(\theta)}) L_{\theta,\ell}^{t'} + (c_{k'} w_{\ell j,k'}^{(\phi)}) D_{q,\ell}^{t'} \right] \right) \sum_{t=0}^{t'-1} \alpha_{k'}^{t'-t-1} e_{\phi,ji,k}^{t} \\ &= \sum_{t'=0}^{T-1} \sum_{k'} F_{j,\mathrm{fb}}^{t',k'} \, \bar{e}_{\phi,ji,k}^{t',k'}, \end{aligned} \tag{35}$$

where $F_{j,\mathrm{fb}}^{t',k'}$ is the feedback coefficient defined in (23), and

$$\bar{e}_{\phi,ji,k}^{t',k'} := \sum_{t=0}^{t'-1} \alpha_{k'}^{t'-t-1} e_{\phi,ji,k}^{t}. \tag{36}$$

We now verify the online recursion for the filtered trace:

$$\begin{aligned} \bar{e}_{\phi,ji,k}^{t+1,k'} &= \sum_{\tau=0}^{t} \alpha_{k'}^{t-\tau} e_{\phi,ji,k}^{\tau} \\ &= \alpha_{k'} \sum_{\tau=0}^{t-1} \alpha_{k'}^{t-1-\tau} e_{\phi,ji,k}^{\tau} + e_{\phi,ji,k}^{t} \\ &= \alpha_{k'} \bar{e}_{\phi,ji,k}^{t,k'} + e_{\phi,ji,k}^{t}. \end{aligned} \tag{37}$$

Therefore, after exchanging the order of summation, each contribution to the cumulative future-dependent correction can be written using the current filtered traces $\bar{e}_{\phi,ji,k}^{t',k'}$ and the instantaneous feedback coefficients $F_{j,\mathrm{fb}}^{t',k'}$.

Since T was arbitrary, the identity holds for the cumulative future-credit correction over any finite horizon. Thus, the filtered-trace recursion provides an exact causal re-indexing without introducing additional algebraic approximation.

□

A.3 Proof of Proposition 5.1

For convenience, we restate a slightly stronger appendix-level version of the proposition.

Proposition A.5 (ODE tracking for the online training rule)**.** *Fix any $R \in \{-1, 0, 1, \ldots\} \cup \{\infty\}$. Under Assumptions A.1 and A.2, the linearly interpolated process associated with*

$$\psi_{t+1} = \psi_t - \eta_t g^{(R)}(S_t, \psi_t)$$

is an asymptotic pseudo-trajectory of the ODE

$$\dot{\psi} = -h^{(R)}(\psi), \qquad h^{(R)}(\psi) = \mathbb{E}_{S \sim \pi_\psi}[g^{(R)}(S, \psi)]. \tag{38}$$

Consequently, the limit set of $\{\psi_t\}$ is contained in the internally chain transitive set of (38)*. If, in addition,* (38) *admits a strict Lyapunov function, then ψ_t converges to the equilibrium set of* (38)*.*

Proof. The online recursion can be written as

$$\psi_{t+1} = \psi_t - \eta_t g^{(R)}(S_t, \psi_t),$$

where S_t is the finite-dimensional state introduced in Section 4 and evolves as a controlled Markov chain with transition kernel P_{ψ_t}.

By Assumption A.1, the step sizes satisfy the standard Robbins–Monro conditions. By Assumption A.2, the iterates remain in a compact set $\mathcal{K}$, and each frozen kernel P_ψ admits a unique invariant distribution π_ψ. The mean field

$$h^{(R)}(\psi) = \mathbb{E}_{S\sim\pi_\psi}[g^{(R)}(S, \psi)]$$

is therefore well defined and locally Lipschitz on $\mathcal{K}$. Assumption A.2 also guarantees the standard Poisson-equation regularity for the controlled Markov noise required by the ODE method. This places the recursion within the standard stochastic-approximation setting with controlled Markov noise [18, 5].

Therefore, the classical ODE method implies that its linearly interpolated trajectory is an asymptotic pseudo-trajectory of (38). The statement on internally chain transitive limit sets then follows from standard asymptotic-pseudo-trajectory theory. If (38) further admits a strict Lyapunov function, this limit set reduces to the equilibrium set, which yields the convergence claim. □

A.4 Proof of Theorem 5.2

We first prove a stronger statement, from which Theorem 5.2 in the main text follows immediately after taking expectation with respect to π_ψ and then the supremum over $\psi \in \mathcal{K}$. The noncausal learning-signal representation is used to bound the omitted future-credit tail. To establish the corresponding bound for the per-step online update directions, we use the causal filtered-trace realization obtained by re-indexing the cumulative correction.

Theorem A.6 (Exponential decay of the truncation bias)**.** *For finite $R \geq -1$, define the truncated learning signal by*

$$\hat{L}^{t,(R)}_{\phi,j} = \log \frac{\lambda^t_{q,j}}{\lambda^t_{p,j}} + \sum_\ell \sum_{t<t'\leq t+R+1} \sum_{k'} \Big[(c_{k'} w^{(\theta)}_{\ell j,k'}) L^{t'}_{\theta,\ell} + (c_{k'} w^{(\phi)}_{\ell j,k'}) D^{t'}_{q,\ell}\Big] \alpha^{t'-t-1}_{k'}. \tag{39}$$

Let $\hat{L}^{t,(\infty)}_{\phi,j}$ be defined by (33)*, and let $R = -1$ denote the direct-only case*

$$\hat{L}^{t,(-1)}_{\phi,j} = \log \frac{\lambda^t_{q,j}}{\lambda^t_{p,j}}.$$

Then there exist constants $C_L, C_g > 0$ such that, for every $R \in \{-1, 0, 1, 2, \dots\}$ and all relevant indices j, t,

$$\big|\hat{L}^{t,(\infty)}_{\phi,j} - \hat{L}^{t,(R)}_{\phi,j}\big| \leq C_L \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}, \tag{40}$$

$$\|g^{(\infty)}(S, \psi) - g^{(R)}(S, \psi)\| \leq C_g \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}, \qquad \forall (S, \psi) \in \mathcal{S} \times \mathcal{K}. \tag{41}$$

Proof. By Lemma A.4, the filtered-trace recursion gives an exact causal re-indexing of the cumulative update induced by the full future-dependent signal $\hat{L}^{t,(\infty)}_{\phi,j}$. Under Assumption A.3, the series defining $\hat{L}^{t,(\infty)}_{\phi,j}$ in (33) converges absolutely, since each summand is bounded by $(L_{\max} + D_{\max}) W_{\max} \alpha^{t'-t-1}_{k'}$ and $\bar{\alpha} < 1$. Hence, for any finite $R \geq -1$, the difference between the full and truncated rules is precisely the omitted future tail, where the retained future-dependent sum is empty when $R = -1$.

Subtracting (39) from (33), we obtain

$$\hat{L}^{t,(\infty)}_{\phi,j} - \hat{L}^{t,(R)}_{\phi,j} = \sum_\ell \sum_{t'>t+R+1} \sum_{k'} \Big[(c_{k'} w^{(\theta)}_{\ell j,k'}) L^{t'}_{\theta,\ell} + (c_{k'} w^{(\phi)}_{\ell j,k'}) D^{t'}_{q,\ell}\Big] \alpha^{t'-t-1}_{k'}. \tag{42}$$

Taking absolute values and using Assumption A.3 gives

$$\begin{aligned}\big|\hat{L}^{t,(\infty)}_{\phi,j} - \hat{L}^{t,(R)}_{\phi,j}\big| &\le \sum_{\ell} \sum_{t'>t+R+1} \sum_{k'} \Big(|c_{k'} w^{(\theta)}_{\ell j,k'}|\,|L^{t'}_{\theta,\ell}| + |c_{k'} w^{(\phi)}_{\ell j,k'}|\,|D^{t'}_{q,\ell}|\Big)\alpha_{k'}^{t'-t-1} \\ &\le (L_{\max} + D_{\max}) W_{\max} \sum_{\ell}\sum_{k'} \sum_{t'>t+R+1} \alpha_{k'}^{t'-t-1}. \end{aligned} \tag{43}$$

Let $m = t' - t - 1$. Then $t' > t + R + 1$ is equivalent to $m \ge R + 1$. For all $R \ge -1$,

$$\sum_{m=R+1}^{\infty} \alpha_{k'}^{m} = \frac{\alpha_{k'}^{R+1}}{1-\alpha_{k'}} \le \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.$$

because $\alpha_{k'} \le \bar{\alpha} < 1$ by Assumption A.3. Substituting this into (43) yields

$$\big|\hat{L}^{t,(\infty)}_{\phi,j} - \hat{L}^{t,(R)}_{\phi,j}\big| \le \Big(\sum_{\ell}\sum_{k'} 1\Big)(L_{\max} + D_{\max}) W_{\max} \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.$$

Since the index sets for ℓ and k' are finite, the factor $\sum_{\ell}\sum_{k'} 1$ is finite. Hence (40) holds with

$$C_L := \Big(\sum_{\ell}\sum_{k'} 1\Big)(L_{\max} + D_{\max}) W_{\max}.$$

We next prove (41) using the causal filtered-trace representation. For each finite $R \ge -1$, define the truncated filtered trace

$$\bar{e}^{t,k',(R)}_{\phi,ji,k} := \sum_{\tau=\max\{1,t-R-1\}}^{t-1} \alpha_{k'}^{t-\tau-1} e^{\tau}_{\phi,ji,k}, \tag{44}$$

where an empty sum is zero. In particular, $R = -1$ gives a zero trace, while $R = \infty$ recovers the full filtered trace in (22).

By exchanging the order of summation as in Lemma A.4, the truncated future-credit correction admits the causal representation

$$\sum_{k'} F^{t,k'}_{j,\mathrm{fb}}\, \bar{e}^{t,k',(R)}_{\phi,ji,k}.$$

The difference between the full and truncated filtered traces is the omitted past tail:

$$\begin{aligned}&\Big|\bar{e}^{t,k',(\infty)}_{\phi,ji,k} - \bar{e}^{t,k',(R)}_{\phi,ji,k}\Big| \\ &\quad\le e_{\max} \sum_{m=R+1}^{\infty} \alpha_{k'}^{m} \le e_{\max} \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}. \end{aligned} \tag{45}$$

By Assumption A.3, the feedback coefficient defined in (23) satisfies

$$|F^{t,k'}_{j,\mathrm{fb}}| \le \Big(\sum_{\ell} 1\Big)(L_{\max} + D_{\max}) W_{\max}.$$

The θ-update and the instantaneous terms of the ϕ-update are identical for all horizons. Hence, for each affected ϕ-parameter coordinate, the difference between the two causal update directions is

$$\sum_{k'} F^{t,k'}_{j,\mathrm{fb}} \left(\bar{e}^{t,k',(\infty)}_{\phi,ji,k} - \bar{e}^{t,k',(R)}_{\phi,ji,k}\right).$$

Using (45), its absolute value is bounded by

$$e_{\max} C_L \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.$$

Since the parameter vector is finite dimensional, there exists a norm-equivalence constant $C_{\mathrm{coord}} > 0$ such that

$$\|g^{(\infty)}(S,\psi) - g^{(R)}(S,\psi)\| \le C_{\mathrm{coord}} e_{\max} C_L \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.$$

This proves (41) with $C_g := C_{\mathrm{coord}} e_{\max} C_L$. □

A.5 Proof of Corollary 5.3

We establish the mean-field perturbation bound and the corresponding differential-inclusion characterization stated in Corollary 5.3.

Corollary A.7 (Perturbation of full-horizon dynamics)**.** *Under Assumptions A.1, A.2, and A.3, define*

$$h^{(R)}(\psi) = \mathbb{E}_{S\sim\pi_\psi}[g^{(R)}(S,\psi)], \qquad h^{(\infty)}(\psi) = \mathbb{E}_{S\sim\pi_\psi}[g^{(\infty)}(S,\psi)],$$

and

$$\varepsilon_R := \sup_{\psi\in\mathcal{K}} \|h^{(R)}(\psi) - h^{(\infty)}(\psi)\|.$$

Then there exists a constant $C_h > 0$, independent of R, such that for every $R \in \{-1, 0, 1, 2, \dots\}$,

$$\varepsilon_R \le C_h \, \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}. \tag{46}$$

Moreover, the linearly interpolated trajectory of the online algorithm is an asymptotic pseudo-trajectory of the differential inclusion

$$\dot{\psi} \in -h^{(\infty)}(\psi) + \varepsilon_R \mathbb{B}, \tag{47}$$

where $\mathbb{B} := \{v : \|v\| \le 1\}$. Consequently, the limit set of $\{\psi_t\}$ is internally chain transitive for (47)*.*

Proof. By Theorem A.6, for every $(S,\psi) \in \mathcal{S}\times\mathcal{K}$,

$$\|g^{(R)}(S,\psi) - g^{(\infty)}(S,\psi)\| \le C_g \, \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.$$

Taking expectation with respect to π_ψ and using Jensen's inequality, we obtain

$$\begin{aligned}
&\|h^{(R)}(\psi) - h^{(\infty)}(\psi)\| \\
&= \left\| \mathbb{E}_{S\sim\pi_\psi}\left[g^{(R)}(S,\psi) - g^{(\infty)}(S,\psi) \right] \right\| \\
&\le \mathbb{E}_{S\sim\pi_\psi} \|g^{(R)}(S,\psi) - g^{(\infty)}(S,\psi)\| \\
&\le C_g \, \frac{\bar{\alpha}^{R+1}}{1-\bar{\alpha}}.
\end{aligned} \tag{48}$$

Taking the supremum over $\psi \in \mathcal{K}$ proves (46) with $C_h := C_g$.

Now define the set-valued map

$$F_R(\psi) := -h^{(\infty)}(\psi) + \varepsilon_R \mathbb{B}.$$

Since $h^{(\infty)}$ is continuous on the compact set $\mathcal{K}$, the map F_R has nonempty compact convex values and is upper semicontinuous on $\mathcal{K}$.

Moreover, by definition of ε_R,

$$\|h^{(R)}(\psi) - h^{(\infty)}(\psi)\| \le \varepsilon_R \qquad \forall \psi \in \mathcal{K},$$

which implies

$$-h^{(R)}(\psi) \in F_R(\psi) \qquad \forall \psi \in \mathcal{K}.$$

Therefore, every solution of the ODE

$$\dot{\psi} = -h^{(R)}(\psi)$$

is also a solution of the differential inclusion (47).

By Proposition A.5, the linearly interpolated online trajectory is an asymptotic pseudo-trajectory of the ODE with vector field $-h^{(R)}$. Since every solution of this ODE is also a solution of the differential inclusion (47), the interpolation is also an asymptotic pseudo-trajectory of (47).

The standard asymptotic-pseudo-trajectory theory for differential inclusions then implies that its limit set is internally chain transitive for (47); see, e.g., [3, 4]. □

A.6 Alignment with the BPTT inference-parameter gradient

Experimental setup: We use the same synthetic SRM configuration as in Section 6.1, except that the sequence length is fixed at $T = 20$. We report results averaged over 8 independent random seeds. To prevent trivial gradient evaluations at a completely random or zero-initialized state, the spiking neural network is pre-trained for 5 epochs. Subsequently, the network parameters and the random noise variables are strictly frozen. The approximated online gradients $\hat{g}_\phi^{(R)}$, evaluated at truncation horizons $R \in \{-1, 0, 1, 2, 3, 4\}$, are benchmarked against the BPTT inference-parameter gradient g_ϕ^{BPTT}. This diagnostic captures the overall online–BPTT ϕ-gradient discrepancy, including the local gradient approximations and finite-horizon truncation. The approximation quality is quantified using two metrics: the Cosine Similarity $(\hat{g}_\phi^{(R)} \cdot g_\phi^{\mathrm{BPTT}})/(\|\hat{g}_\phi^{(R)}\|_2 \|g_\phi^{\mathrm{BPTT}}\|_2)$ to assess gradient directional alignment, and the Relative Error $\|\hat{g}_\phi^{(R)} - g_\phi^{\mathrm{BPTT}}\|_2 / \|g_\phi^{\mathrm{BPTT}}\|_2$ to measure the normalized magnitude of deviation. Higher cosine similarity indicates better directional alignment, whereas lower relative error indicates a more accurate approximation in norm.

Table 2: Cosine similarity and relative error between the truncated online ϕ-gradient and the BPTT ϕ-gradient under different truncation horizons R.

	$R = -1$	$R = 0$	$R = 1$	$R = 2$	$R = 3$	$R = 4$
Cosine Similarity	0.9496	0.9774	0.9888	0.9916	0.9924	0.9925
Relative Error	0.2733	0.1859	0.1393	0.1174	0.1097	0.1091

Results. Table 2 and Figure 2 (a) show that increasing the truncation horizon consistently improves the alignment between the online ϕ-gradient and the BPTT ϕ-gradient. The direct-only case $R = -1$ already produces a broadly aligned direction, with a cosine similarity of 0.9496, but its relative error remains relatively large. Adding the future-dependent correction substantially improves the approximation: moving from $R = -1$ to $R = 0$ increases the cosine similarity from 0.9496 to 0.9774 and reduces the relative error from 0.2733 to 0.1859. Further increasing R continues to improve the approximation, with the relative error decreasing to about 0.11 and the cosine similarity exceeding 0.99 when $R \geq 2$.

These results support the role of the truncated future-credit term in the proposed online inference update. A small horizon already captures most of the BPTT ϕ-gradient direction, while larger horizons mainly refine the gradient magnitude and reduce the residual approximation error. The improvement saturates around $R = 3$ and $R = 4$, suggesting that most useful future-dependent credit is concentrated within a short temporal window under the default synaptic decay setting. The remaining discrepancy after saturation is expected because the comparison is made against the full BPTT ϕ-gradient, while the online rule also incorporates local gradient approximations. This behavior is consistent with the theoretical view that the truncation error is controlled by the exponentially decaying SRM trace.

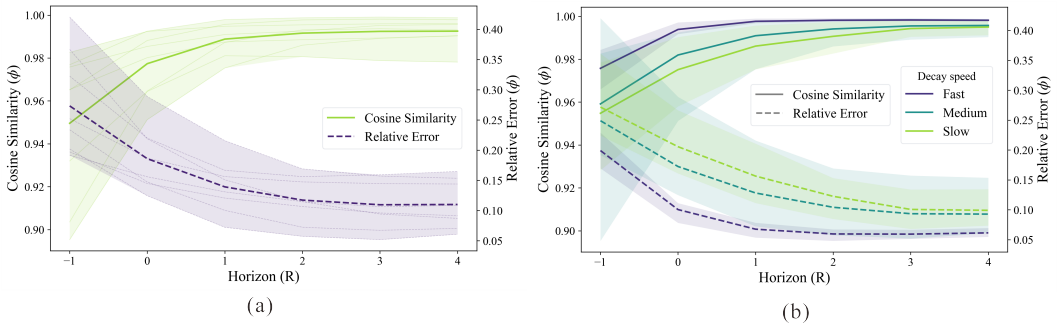

Figure 2: Gradient approximation under different truncation horizons R (shaded regions indicate $\pm$ SD across runs). (a) Cosine similarity and relative error between the truncated online ϕ-gradient and the BPTT ϕ-gradient as R varies. (b) Cosine similarity and relative error between the truncated online ϕ-gradient and the BPTT ϕ-gradient under fast, medium, and slow synaptic decay settings.

A.7 Effect of synaptic decay on truncated credit assignment

Experimental setup: To further investigate how the intrinsic temporal dynamics of spiking neurons interact with the analytical truncation horizon R, under the same experimental setup as in Appendix A.6, we categorize the neuronal memory properties into three distinct decay settings governed by the configuration parameters ρ. Here, ρ designates the continuous-time synaptic time constants relative to the integration step size, effectively dictating the temporal decay factor α of the discrete synaptic traces. Specifically, we evaluate:

- **Fast decay setting:** We set the time constants to $\rho = \{1, 2\}$, yielding an effective maximum decay factor of $\bar{\alpha} \approx 0.607$, which causes the influence of historical spikes to vanish rapidly.
- **Medium decay baseline:** The configuration of $\rho = \{2, 5\}$, corresponding to $\bar{\alpha} \approx 0.819$, is utilized as the standard reference representing typical memory retention.
- **Slow decay setting:** By increasing the time constants to $\rho = \{5, 10\}$, we attain $\bar{\alpha} \approx 0.905$, a setting that retains longer temporal dependencies of past spikes.

Here $\bar{\alpha} = \max_k \alpha_k$ denotes the largest discrete synaptic decay factor. Across all three settings, we measure the cosine similarity and relative error of the approximated ϕ-gradient towards the BPTT ground truth, evaluated comprehensively across a sweeping domain of R. Results are averaged over 5 independent random seeds.

Table 3: Relative error between the truncated online ϕ-gradient and the BPTT ϕ-gradient under different synaptic decay settings.

Setting	ρ	$\bar{\alpha}$	$R=-1$	$R=0$	$R=1$	$R=2$	$R=3$	$R=4$
Fast	$\{1,2\}$	0.607	0.1990	0.1003	0.0675	0.0595	0.0592	0.0587
Medium	$\{2,5\}$	0.819	0.2490	0.1723	0.1281	0.1041	0.0934	0.0927
Slow	$\{5,10\}$	0.905	0.2718	0.2055	0.1563	0.1225	0.1005	0.0990

Results. Table 3 and Figure 2 (b) show that synaptic decay has a clear effect on the quality of truncated credit assignment. For all three decay settings, increasing R monotonically reduces the relative error between the truncated online ϕ-gradient and the BPTT ϕ-gradient. This confirms that retaining a longer future-credit window gives a more accurate approximation to the BPTT ϕ-gradient.

The improvement rate, however, depends strongly on the synaptic decay factor $\bar{\alpha}$. When the synaptic trace decays quickly, the error drops rapidly from 0.1990 at $R = -1$ to 0.0595 at $R = 2$, and then only decreases slightly to 0.0587 at $R = 4$. This indicates that a short truncation horizon is already sufficient when future dependencies vanish quickly. In contrast, under the slow-decay setting, the error decreases more gradually, from 0.2718 at $R = -1$ to 0.1225 at $R = 2$ and further to 0.0990 at $R = 4$. Thus, slower synaptic decay requires a longer truncation horizon to recover a comparable level of gradient accuracy.

This trend directly supports the truncation-bias analysis. A larger $\bar{\alpha}$ means that past spikes influence future intensities over a longer temporal range, so truncating the future-dependent terms leaves a larger residual discrepancy. Conversely, when the synaptic trace decays rapidly, the omitted tail becomes negligible after only a few steps. Overall, the results empirically validate the theoretical prediction that the truncation error decreases with R, while its decay rate is governed by the intrinsic synaptic memory.

A.8 Sensitivity to basis size and synaptic time constants

We further evaluate the sensitivity of the proposed method to the number of exponential basis functions B and their corresponding synaptic time constants $\{\rho_k\}$. All experiments are conducted on N-MNIST with $R = 1$ under the same training configuration as in Section 6, while only B and $\{\rho_k\}$ are varied.

As shown in Table 4, all evaluated basis configurations achieve accuracies above 98%, ranging from 98.01% to 98.43%. These results suggest that a compact exponential basis is sufficient for N-MNIST under the current setting. The slight performance degradation with larger basis configurations may be attributed to potential overfitting. Nevertheless, theoretically, increasing the number of basis functions

Table 4: Sensitivity to the number of exponential basis functions B and synaptic time constants $\{\rho_k\}$ on N-MNIST with $R = 1$.

B	$\{\rho_k\}$	Accuracy (%)
1	$\{3.16\}$	98.43
2	$\{1, 2.5\}$	98.29
2	$\{2, 5\}$	98.27
2	$\{4, 10\}$	98.09
3	$\{2, 3.16, 5\}$	98.01

B and adjusting the synaptic time constants $\{\rho_k\}$ can potentially capture richer temporal dynamics, which may be beneficial for more complex datasets. Therefore, B and $\{\rho_k\}$ can be selected according to the characteristics of the dataset and the available computational resources.

A.9 Additional evaluation on DVS Gesture

We further evaluate the proposed online method on the DVS Gesture dataset [1], which contains event-based recordings of dynamic human gestures. This dataset provides a complementary setting to N-MNIST for examining the effect of future-dependent credit assignment on temporally structured data.

Experimental setup. We use the official training and test splits of DVS Gesture and discretize each event stream into synchronous time frames using SpikingJelly [7]. The resulting input sequence contains 20 timesteps. We use the same probabilistic SRM classifier family as in the N-MNIST experiments, consisting of an input projection, a recurrent SRM hidden layer, and an output linear classifier. The hidden dimension is 256, and the batch size is 64.

Unless otherwise stated, all methods use the same architecture, probabilistic model, ELBO objective, optimizer, and training configuration. We use Adam with a learning rate of 10^{-3} and weight decay 10^{-5}. All DVS Gesture models are trained for 100 epochs. The exponential SRM basis uses $B = 2$ components with time constants $\{\rho_k\} = \{2, 5\}$. We compare the score-function REINFORCE baseline, the proposed online method with $R \in \{-1, 0, 1, 2\}$, and full-sequence BPTT.

Table 5: Comparison of different online credit-assignment horizons, REINFORCE, and BPTT on DVS Gesture under the same architecture and training configuration. Accuracy is reported from a single run per method.

Method	Accuracy (%) ↑	Peak memory (MB) ↓
REINFORCE	51.73	57.4
$R = -1$	90.28	105.2
$R = 0$	90.97	107.1
$R = 1$	91.67	109.3
$R = 2$	92.71	111.4
BPTT	93.40	147.5

Results. Table 5 presents the classification accuracy and peak GPU memory on DVS Gesture. Increasing the credit-assignment horizon from $R = -1$ to $R = 2$ consistently improves accuracy from 90.28% to 92.71%, with only a modest increase in peak memory from 105.2 MB to 111.4 MB. This trend suggests that incorporating future-dependent credit corrections can improve predictive performance on temporally structured event-based tasks.

At $R = 2$, the proposed online method achieves 92.71% accuracy, approaching the 93.40% achieved by BPTT, while reducing peak GPU memory by 24.5%. In contrast, REINFORCE achieves only 51.73% accuracy despite its lower memory consumption. These results illustrate the accuracy–memory trade-off associated with different credit-assignment horizons and provide complementary empirical support for the proposed filtered-trace approach.

A.10 Limitations

This work focuses on online variational training for probabilistic SNNs under the discrete-time SRM approximation with exponentially decaying synaptic traces. The theoretical analysis relies on standard stochastic-approximation regularity and boundedness assumptions. The exponential bias bound characterizes the additional error introduced by finite-horizon truncation within the proposed online estimator family. Empirically, the synthetic experiments examine the truncation-bias behavior, while N-MNIST and DVS Gesture evaluate predictive performance and memory efficiency using relatively compact SRM architectures. Extending the framework to larger architectures, other neuron models, and more diverse or nonstationary settings remains for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are supported by the proposed online variational training algorithm, the theoretical truncation analysis, and the experimental evaluations presented in Sections 4–6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix A.10, including the theoretical assumptions, the scope of the truncation-bias analysis, and the experimental evaluation settings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions are stated with the theoretical results, and complete proofs are provided in the Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 6.1 and Appendices A.6–A.9 provide the data preparation procedures, model configurations, training hyperparameters, truncation horizons, and evaluation settings needed to reproduce the reported experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The implementation code is not publicly released with this paper. The algorithmic derivations and experimental configurations are provided to facilitate reproducibility. The real-data experiments use the publicly available N-MNIST [27] and DVS Gesture [1] datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup specifies the datasets, data splits, model architectures, training hyperparameters, optimizers, and evaluation metrics in Section 6.1 and Appendices A.6–A.9.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main N-MNIST classification results in Section 6.3 are reported as mean$\pm$standard deviation over five matched random seeds. Additional gradient diagnostics in Appendices A.6–A.7 are evaluated over multiple independent seeds, with variability indicated in Figure 2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: All experiments were conducted on a single NVIDIA GeForce RTX 5070 Ti GPU, as described in Appendix A. The paper reports per-epoch runtime and peak GPU memory for N-MNIST in Table 1, as well as peak GPU memory for DVS Gesture in Table 5; a complete per-run and aggregate compute breakdown is not included.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses synthetic data and existing public neuromorphic datasets. No new human-subject experiments or private data collection were conducted, and the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on foundational algorithmic and theoretical aspects of probabilistic SNN training and does not include a separate broader-impact discussion because there is no direct deployment or application-specific societal claim.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, generative models, or other assets that pose substantial misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: The paper cites the original N-MNIST [27] and DVS Gesture [1] datasets, as well as the SpikingJelly framework [7]. The dataset licenses, together with the SpikingJelly version and its license, are specified in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce or release a new dataset, benchmark, pretrained model, or software package as a new asset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core method, theory, or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.