

Stabilizing Spiking Neurons Through Biologically Inspired Polarization

Matthew Lai¹, Longbing Cao²

¹Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia

²School of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia
matthew.t.lai@gmail.com, longbing.cao@mq.edu.au

Abstract

The Leaky Integrate-and-Fire (LIF) neuron model remains a staple in spiking neural networks (SNNs), yet its oversimplified dynamics lead to unstable gradients and limit scalability. We introduce a polarization-aware spiking architecture (POLARA) that models depolarization, repolarization, and hyperpolarization through analytically defined membrane dynamics. POLARA unifies biologically grounded design with stable gradient propagation—formulating both forward and backward paths directly, and applying gradient shaping solely for numerical control, without requiring learnable gates or surrogate tuning. By bounding membrane potentials within realistic voltage ranges, POLARA avoids vanishing and exploding gradients, enabling scalable training in deeper architectures. Experiments show consistent gains over LIF and competitive results against optimized SNNs, positioning POLARA as a principled alternative to surrogate-driven or reset-based designs.

Introduction

Deep neural networks (DNNs) have driven breakthroughs in AI, achieving state-of-the-art results in vision, language, and control (Hinton, Osindero, and Teh 2006; LeCun, Bengio, and Hinton 2015). DNNs rely on continuous activations and global backpropagation, diverging from biological computation. In contrast, the human brain—despite sparse spikes, noisy signals, and local plasticity—learns robustly with fewer neurons and dramatically lower energy.

Spiking neural networks (SNNs) aim to bridge this gap by using discrete, time-localized spikes. Spiking neurons fire sparsely, enabling event-driven computation that captures temporal structure (Lobo et al. 2020; Maass 1997; Wang, Lin, and Dang 2020), offering a path toward energy-efficient, biologically plausible models.

A central challenge lies in training. Spike generation is non-differentiable: the Heaviside step function has zero gradient almost everywhere, blocking gradient-based learning. Surrogate Gradient (SG) methods replace the spike derivative with smooth approximations (Nefcici, Mostafa, and Zenke 2019; Zenke and Vogels 2021), enabling Backpropagation Through Time (BPTT). However, most models

stop at surrogate replacement without addressing the underlying instability in signal propagation.

Generalized LIF (GLIF) (Teeter et al. 2018; Yao et al. 2022) introduces adaptation or conductance modulation, yet relies on simplified abstractions that approximate complex ionic behavior using static thresholds or decay rules. These models omit polarization dynamics—a fundamental membrane process critical for sustained neural signaling and gradient flow. Such simplifications reduce physiological fidelity, degrade gradient propagation, and limit scalability.

We argue that overcoming these limitations requires modeling polarization dynamics directly, rather than compensating with heuristics or resets. In this paper, we propose a novel polarization-aware architecture, POLARA, which stabilizes spiking neurons by treating membrane behavior as a dynamic, structured process that supports stable gradient propagation throughout training.

POLARA captures three core phases—depolarization, repolarization, and hyperpolarization—via structured kernel dynamics grounded in neurophysiology. Unlike LIF and GLIF, which abstract away internal feedback and ionic regulation, POLARA draws from the Hodgkin–Huxley (HH) theory, Generalized Linear Models (GLMs), and spike-timing dependent plasticity (STDP) (Hodgkin and Huxley 1952; Gerstner 1995; Pillow et al. 2008; Bengio et al. 2015; Mozafari et al. 2018), informing both functional shape and temporal dependencies.

Unlike models with resets or trainable decay, POLARA maintains membrane continuity through bounded, interpretable component interactions. This enables sustained gradient flow without artificial gates or external stabilizers. Rather than smoothing over non-differentiability, we define inherently learnable dynamics.

Our contributions offer a biologically grounded and scalable alternative to reset-based, gate-driven, or surrogate-tuned models:

1. **Biophysical kernel dynamics:** Modeling membrane transitions—depolarization, repolarization, and hyperpolarization—with biologically interpretable kernels instead of abstract decay or reset functions.
2. **Gradient-aligned formulation:** Defining bounded gradients that remain consistent with membrane evolution over time, supporting stable learning without relying on heuristic spike approximations.

3. **Provable training stability:** Guaranteeing bounded membrane evolution and sustained gradient flow over long temporal sequences, enabling reliable training of deep and temporally extended architectures.

We present POLARA not as biological mimicry but as a principled reconstruction of membrane dynamics to solve gradient instability at its root. This offers a path toward biologically faithful and scalable spiking architectures.

While POLARA addresses membrane fidelity and gradient stability, biological plausibility also depends on how the model is deployed. Artificially limiting the number of time steps risks collapsing rich temporal dynamics into static thresholding—undermining the fundamental advantages of spike-based computation. An extended version of this paper, including the full appendix and complete proofs, will be made available on arXiv.

Related Work

SNNs aim to capture neural computation using discrete spike-based communication. While they promise biological plausibility and energy efficiency, most modern SNNs remain built upon simplified abstractions—particularly the LIF neuron—and rely on BPTT for training (Li et al. 2024; Singh et al. 2022). These models perform adequately on standard benchmarks, but fall short in two key areas: (1) they fail to reflect critical biological dynamics—particularly polarization and ionic regulation—and (2) they break down in deep or long-timescale learning due to unstable gradients.

Simplified Neuron Models and Limits of Abstraction

The classical LIF neuron integrates input over time with exponential decay, generating a spike when the membrane potential crosses a fixed threshold:

$$\hat{u}_t^l = \beta u_{t-1}^l + \sum_{j=1}^N w_{ij} \sigma_j^l \quad [\text{leak + integrated}] \quad (1)$$

$$o_t^l = \begin{cases} 1 & \text{if } \hat{u}_t^l \geq \vartheta \\ 0 & \text{otherwise} \end{cases} \quad [\text{Fire}] \quad (2)$$

$$u_t^l = \hat{u}_t^l - \vartheta o_t^l \quad [\text{Reset}] \quad (3)$$

While LIF captures basic spike behavior, it lacks the complexity to represent nonlinear, history-dependent dynamics of depolarization, repolarization, and hyperpolarization observed in real neurons.

Extensions such as GLIF improve biological fidelity by introducing adaptation currents and multiple decay timescales. However, they continue to rely on resets and fixed decay rules, without explicitly encoding temporal dependencies in the forward pass. As a result, gradient instability remains an open challenge, and these models act as heuristic improvements rather than fundamental redesigns.

Backward-Phase Interventions: Surrogate Gradients and Complexity

GLIF-style models do not fully resolve gradient instability—especially in deep networks—due to limited forward-

phase temporal structure. To compensate, recent methods modify the backward pass by introducing learnable surrogate mechanisms. Learnable Surrogate Gradients (LSG)(Lian et al. 2023) adapt gradient shapes during training, while Parametric Surrogate Gradients (PSG)(Wang, Cheng, and Lim 2024) incorporate auxiliary gates inspired by recurrent networks to modulate membrane updates. While effective in some cases, these methods increase architectural complexity and treat gradient shaping as an optimization technique rather than a model design feature.

In contrast, our approach constrains gradient flow analytically—directly shaping the backward signal without relying on learned modulation.

The Missing Piece: Forward-Phase Temporal Dynamics

A key limitation across models is the lack of biologically meaningful temporal dynamics in the forward pass. Real neurons evolve membrane potentials through ionic-channel interactions shaped by recent activity, naturally stabilizing gradients and encoding long-range context. Most models ignore this or approximate it indirectly.

What remains absent is a neuron whose internal dynamics inherently support stable learning—without relying on surrogate parameterization, auxiliary gates, or task-specific tuning. POLARA addresses this by structuring membrane evolution to encode temporal dependencies intrinsically, aligning learning stability with biologically grounded dynamics.

However, even models with internal dynamics risk collapsing to static behavior under short time windows—reducing spiking to simple thresholding.

Preliminaries

To overcome the limitations of LIF-based neurons, POLARA draws on foundational models in computational neuroscience—particularly the GLM, STDP, HH, and the Nernst potential (Hille 2001). These frameworks inspire POLARA’s design, especially in its treatment of membrane dynamics and temporal structure, without directly replicating any one model.

STDP-Inspired Timing Sensitivity

STDP is a biological rule that adjusts synaptic strength based on the relative timing of pre- and post-synaptic spikes (Benigno et al. 2015). Although POLARA does not implement STDP as a learning mechanism, it incorporates its key insight: recent spike history shapes neural responses. Instead of treating membrane updates as temporally isolated, POLARA retains recent activity through temporal kernels, allowing forward-phase membrane evolution to reflect spike timing correlations even without explicit plasticity.

GLM: Temporal Dynamics and Probabilistic Activation

The GLM builds on STDP’s core idea that past activity modulates future behavior. Unlike LIF, which responds only to

instantaneous input with a fixed threshold, GLM integrates spike history and external stimuli over time:

$$u(t) = \int \kappa(s)o(t-s)ds + \int_0^\infty \eta(s)I(t-s)ds + u_{\text{rest}} \quad (4)$$

where $o(t)$ denotes the neuron's own output spike train, κ captures the influence of the neuron's own past spikes, modeling hyperpolarization effects, and η encodes how external input shapes the membrane potential temporally. This time-aware integration enables richer membrane dynamics than LIF.

GLM also models a dynamic, adaptive threshold as a convolution with past spikes:

$$\vartheta(t) = \vartheta_0 + \int \theta_1(s)o(t-s)ds \quad (5)$$

differing from GLIF where adaptation is often manually added and decoupled from membrane updates.

Finally, GLM generates spikes probabilistically:

$$O(\delta) = f(\hat{u}_t - \vartheta_t) \quad (6)$$

capturing biological stochasticity and yielding a differentiable firing rule. While POLARA does not directly adopt GLM, it draws on its principles by using structured temporal kernels to shape membrane evolution based on recent activity, producing smooth, biologically grounded activation when paired with a continuous firing function.

HH: Ionic Dynamics and Natural Boundaries

The HH model describes membrane voltage as a continuous process driven by voltage-gated ion channels rather than discrete thresholds. Ionic currents flow based on the difference between membrane potential and ion-specific reversal potentials, producing dynamic state evolution (Gerstner 1995):

$$I_{\text{TOT}} = C \frac{du}{dt} = (u - E_K) \bar{g}_K n(u, t)^4 + (u - E_{\text{Na}}) \bar{g}_{\text{Na}} m(u, t)^3 h(u, t) + (u - E_L) \bar{g}_L$$

where C is the membrane capacitance, and E_{Na} , E_K , and E_L are reversal potentials computed via the Nernst equation that define ionic flow direction and impose natural voltage constraints.

Figure 1 illustrates how sodium and potassium ions flow oppositely depending on membrane voltage, producing depolarization and repolarization phases. These biophysical principles inspire our use of membrane-sensitive kernels to stabilize voltage dynamics during forward computation.

Such smooth activation functions, when bounded, enable treating spikes as discrete events during forward computation while stabilizing gradient propagation during training—consistent with biological constraints observed in HH and GLM frameworks.

Together, these models establish the necessity of time-aware, biophysically grounded membrane evolution—principles that POLARA reconfigures into a trainable forward pass.

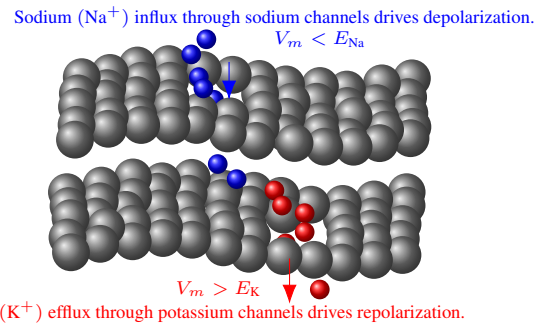


Figure 1: Voltage-gated ion channels modulate membrane potential V_m via sodium and potassium flow relative to reversal potentials E . Sodium influx occurs when $V_m < E_{\text{Na}}$ (depolarization); potassium efflux when $V_m > E_K$ (repolarization). Unlike LIF resets, this process is continuous and biophysically constrained.

The POLARA Methodology

We present the design of our neuron model, translating biological principles into a concrete, trainable system. POLARA decomposes membrane behavior into distinct polarization phases embedded directly in the forward computation. This section details the core components, their interactions, and how they support stable learning.

Forward Computation

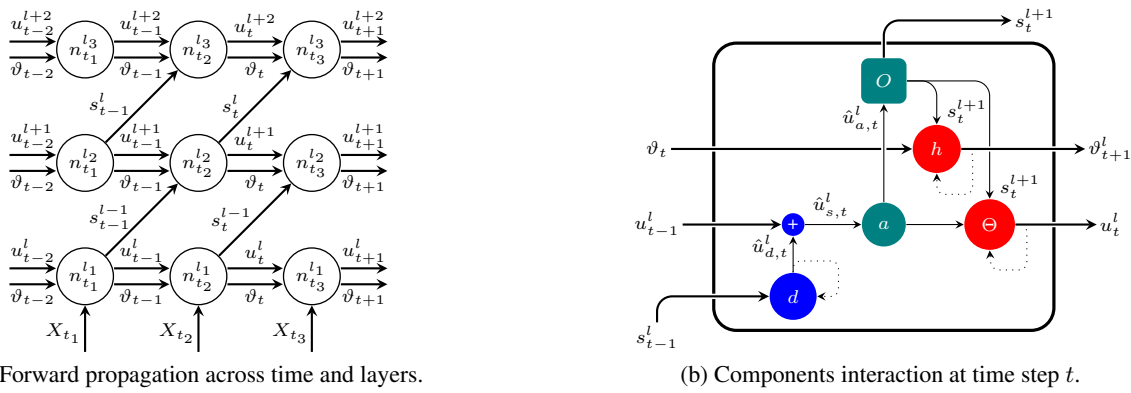
Building on its structured design, POLARA models membrane evolution through five interacting components: membrane decay, stochastic amplification, spike activation, refractory inhibition, and adaptive thresholding. These jointly express the full polarization cycle—depolarization, repolarization, and hyperpolarization—via structured temporal kernels embedded in the forward pass.

This contrasts with models that treat temporal dependencies heuristically or defer complexity to the backward pass via surrogate gradients. Figure 2 illustrates POLARA's architecture: spike propagation across layers, and internal state evolution within neurons over time. Crucially, it enforces natural membrane boundaries, preventing voltage drift and stabilizing forward dynamics.

Membrane Decay To model gradual voltage leakage, POLARA integrates decay over a recent window I , simulating the slow loss of membrane potential observed in biological neurons:

$$\hat{u}_{d,t}^l = u_{t-1}^l \left(1 - \sum_{i \in I} d(i) + \epsilon \right) \quad (7)$$

where $d(i) = Ae^{-\alpha i}$ defines decay contributions with magnitude A and rate α , and a small ϵ to ensure stability. This kernel-based decay models the **depolarization phase**, where the neuron prepares for incoming input without triggering immediate spiking.



(a) Forward propagation across time and layers.

(b) Components interaction at time step t .

Figure 2: (a) At each time step, spike outputs s_t^l feed forward to the next layer, while internal states—membrane potential u_t^l and adaptive threshold ϑ_t^l —propagate temporally within the same layer. (b) Within a single time step, input X , membrane u_t^l , and threshold ϑ_t^l are jointly suppressed and modulated by a , integrated with stochastic amplification, and passed through activation O to produce spike s_t^{l+1} . After spiking, refractory function h inhibits future firing and updates u_{t+1}^l , while ϑ_{t+1}^l updates independently via \ominus . Dotted loops indicate temporal memory over window I , enabling biologically grounded recurrence. Blue, teal, and red denote depolarization, repolarization, and hyperpolarization.

Membrane Input Integration Following decay, external input x_t integrates to update internal state:

$$\hat{u}_{s,t}^l = \hat{u}_{a,t}^l + x_t \quad (8)$$

This additive step contributes to the **depolarization phase**, combining residual potential with new input to update the membrane state.

Stochastic Amplification To capture near-threshold excitability, we introduce stochastic amplification that boosts subthreshold membrane potentials:

$$\hat{u}_{a,t}^l = \begin{cases} \hat{u}_{d,t}^l(1 + a(\delta)), & \delta \leq 0 \\ \hat{u}_{d,t}^l, & \text{otherwise} \end{cases} \quad (9)$$

where $\delta = \hat{u}_{d,t}^l - \vartheta_t$ reflects distance to threshold.

This biologically grounded mechanism enhances responsiveness in the **depolarization phase** without requiring hard thresholds or probabilistic firing.

Spike Activation Spikes are triggered when the membrane potential exceeds threshold, using a smooth activation based on overshoot $\Delta = \hat{u}_{a,t}^l - \vartheta_t$:

$$s_t^l = \begin{cases} \sigma(\hat{u}_{a,t}^l O(\Delta)), & \Delta \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where $O(\Delta) = \min(\Delta(1 + \Delta e^{-\gamma\Delta}), o_{\max})$. This bounded, biophysically grounded function captures near-threshold excitability while preventing runaway growth. It supports gradient flow without surrogate tricks and mimics soft threshold behavior observed in HH and GLM literature. Unlike methods with discontinuous spikes or surrogate-based approximations, our formulation produces continuous, low-saturation activations that preserve differentiability across layers and time—completing the **repolarization phase**.

Refractory Inhibition After spiking, POLARA applies a graded inhibitory trace to enforce post-activation suppression. A time-varying kernel $h(i) = \sum_{j=1}^3 D_j e^{-(i-\mu_j)^2/2\sigma_j^2}$, inspired by GLM, accumulates over window I and modulates the updated state as:

$$u_t^l = \hat{u}_{a,t}^l \left(1 + \sum_{i \in I} h(i) \right) \quad (11)$$

This corresponds to the **hyperpolarization phase**, ensuring membrane continuity while promoting temporal sparsity and refractory suppression.

Adaptive Thresholding To promote sparsity and prevent runaway excitation, POLARA adapts its threshold in response to recent spiking:

$$\vartheta_{t+1}^l = \vartheta_t^l \left(1 + \sum_{i \in I} \Theta(i) \right) \quad (12)$$

where $\Theta(i) = E e^{-\eta i}$ encodes exponential decay over recent steps I , with constants E and η modulating amplitude and decay rate; this post-spike rise enforces a refractory-like effect, suppresses high-frequency firing, and promotes temporal sparsity. Threshold and membrane co-adapt over time, enabling endogenous control.

This completes the **hyperpolarization phase** and closes the polarization cycle in a differentiable, temporally structured loop.

These components do not operate in isolation; they jointly constitute the polarization cycle—spanning depolarization, firing, and hyperpolarization—ensuring both stability and expressiveness in temporal learning.

Algorithm 1 summarizes the forward pass.

Backward Computation and Gradient Analysis

Training SNNs with BPTT demands stable gradient flow across time and layers. While prior models circumvent non-differentiable spikes via SG, POLARA achieves gradient

Algorithm 1: Forward Pass Algorithm (vectorized)

Input: Previous membrane potentials $\{u_{t-1}^l, u_{t-2}^l, \dots, u_{t-\tau}^l\}$, previous spikes s_{t-1}^l , previous thresholds ϑ_{t-1}^l , time window $I = \{i_1, \dots, i_n\}$ input $X = \{x_1, x_2, \dots, x_T\}$

Functions: d, a, O, h, Θ

Output: Updated membrane potentials u_t^l , spikes s_t^l , threshold ϑ_t^l

```

1: for  $l = 1$  to  $L$  do
2:   for  $t = 1$  to  $T$  do
3:      $\hat{u}_{d,t}^l \leftarrow u_{t-i}^l(1 - \sum_i d(i))$  # Decay (vectorized)
4:      $\hat{u}_{s,t}^l \leftarrow u_{d,t}^l + x_t$  # integration
5:      $\delta \leftarrow \hat{u}_{s,t}^l - \vartheta_{t-1}^l$  # Compute deviation from threshold
6:      $\hat{u}_{a,t}^l \leftarrow \hat{u}_{s,t}^l(1 + a(\delta))$  # Stochastic amplification near threshold
7:      $\Delta \leftarrow \hat{u}_{a,t}^l - \vartheta_{t-1}^l$  # overshoot
8:      $s_t^l \leftarrow \sigma(\hat{u}_{a,t}^l O(\Delta))$  # Spike activation
9:      $\hat{u}_t^l \leftarrow \hat{u}_{a,t}^l(1 + \sum_i h(i))$  # Refractory inhibition (vectorized)
10:     $\vartheta_t^l \leftarrow \vartheta_{t-1}^l(1 + \sum_i \Theta(i))$  # Adaptive threshold update (vectorized)
11:   end for
12: end for
13: return  $u_t^l, s_t^l, \vartheta_t^l$ 

```

stability inherently through bounded, biologically grounded dynamics.

We analyze the full computation path and show that the activation function O has a bounded derivative. This guarantees that gradients neither vanish nor explode, enabling reliable learning across long temporal sequences.

Gradient Through Activation The spike output gradient s_t^l with respect to the pre-spike membrane potential $\hat{u}_{a,t}^l$ is:

$$\frac{\partial s_t^l}{\partial \hat{u}_{a,t}^l} = \begin{cases} O(\Delta) + \hat{u}_{a,t}^l \frac{\partial O(\Delta)}{\partial \hat{u}_{a,t}^l}, & \Delta \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

with

$$\frac{\partial O(\Delta)}{\partial \hat{u}_{a,t}^l} = (1 - \Delta)e^{-\gamma\Delta} \frac{\partial \Delta}{\partial \hat{u}_{a,t}^l}$$

This smooth, bounded derivative ensures stable backpropagation without requiring SG. Figure 3 contrasts typical SG curves (a) with our bounded derivative (b).

Gradient Through Stochastic Amplification The derivative of the amplified membrane voltage with respect to its decayed input is:

$$\frac{\partial \hat{u}_{a,t}^l}{\partial \hat{u}_{d,t}^l} = \begin{cases} 1 + a(\delta) + \hat{u}_{d,t}^l \frac{\partial a(\delta)}{\partial \hat{u}_{d,t}^l}, & \delta \leq 0 \\ 1, & \text{otherwise} \end{cases} \quad (14)$$

where

$$\frac{\partial a(\delta)}{\partial \hat{u}_{d,t}^l} = B\beta e^{\beta\delta}$$

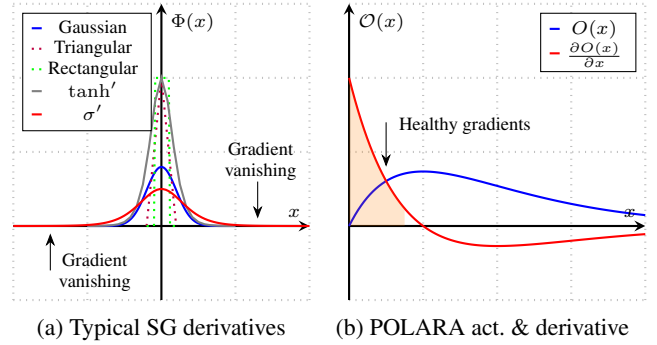


Figure 3: (a) Various SG functions approximate a step but vanish beyond a narrow range, limiting credit assignment. (b) Our activation O maintains bounded, non-zero gradients, avoiding explosion and enabling stable backpropagation.

This term increases with δ but remains bounded by the slope parameter β , ensuring stable gradient propagation when amplification is active.

Gradients Through Input and Membrane Decay The remaining gradients are linear and bounded:

$$\frac{\partial \hat{u}_{s,t}^l}{\partial \hat{u}_{a,t}^l} = 1 \quad (15)$$

$$\frac{\partial \hat{u}_{a,t}^l}{\partial \hat{u}_{d,t}^l} = \frac{\partial \hat{u}_{a,t}^l}{\partial \hat{u}_{d,t}^l} \left(1 - \sum_i d(i) + \epsilon \right) \quad (16)$$

where the decay sum is strictly less than 1, mirroring biological leakage.

Together, these components ensure stable gradients across time, enabling reliable learning over long sequences.

The bounded derivative of O ensures gradients remain stable over time. **Appendix B** proves the total gradient product over any sequence length T is strictly bounded. Hence, BPTT remains stable over long horizons.

Corollary 1. *Given Theorem in Appendix B, the product of gradients over any sequence of length T remains bounded away from zero and infinity. Hence, BPTT avoids both vanishing and exploding gradients across long time scales.*

Experiments

We begin by evaluating POLARA on MNIST, FashionMNIST, and EMNIST-byclass—standard benchmarks used to diagnose learning stability in compact, shallow networks. These datasets provide a controlled setting to assess POLARA’s ability to learn from sparse inputs, and allow direct comparison to a standard LIF model under identical conditions.

Architectural details, ablations, and FLOP-based efficiency analyses follow.

Small-Scale Evaluation: MNIST Variants

We train over 40 epochs using 128 time steps and a temporal window of size 6, chosen to reflect the short-term memory

in STDP-like rules, where spike influence fades after a few steps. Optimization is done with SGD at learning rate 0.1, with a step size 2 scheduler. The model is deliberately simple: one hidden layer of 100 neurons for MNIST and FashionMNIST, and 1024 for EMNIST-byclass.

As a baseline, we compare against the LIF model from the `spytorch` framework¹, which employs a well-tuned surrogate gradient known to improve training stability.

Figure 4 (a) shows that POLARA achieves reliable learning across all MNIST variants, despite the architecture’s simplicity. This demonstrates that POLARA stabilizes gradient flow without sacrificing performance. Table 1 confirms that POLARA outperforms the LIF baseline, indicating that biologically grounded dynamics can improve both stability and accuracy.

Large-Scale Evaluation: CIFAR Benchmarks

We evaluate POLARA on CIFAR-10 and CIFAR-100—static image classification tasks with greater class diversity, deeper architectures, and longer sequences.

POLARA is integrated with a ResNet-38 backbone (Liu et al. 2021). Training runs for 70 epochs over 50 time steps using a temporal window of size 6. The learning rate begins at 0.0628 and decays by 90% at epochs 50 and 60.

Gradient Stabilization CIFAR tasks introduce higher input intensities and deeper models, increasing the risk of gradient explosion. To counter this, we apply a biologically inspired gradient mask that bounds gradients during backpropagation without altering forward spike dynamics.

Specifically, the gradient of the spike activation with respect to the membrane overshoot is defined as:

$$\frac{\partial s_t^l}{\partial \hat{u}_{h,t}^l} = \begin{cases} \alpha, & \text{if } |\Delta| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

where $\Delta = u_{h,t}^l - \vartheta_t$ is the overshoot and $\alpha = 0.3$ is manually selected to enforce bounded gradients across time and depth.

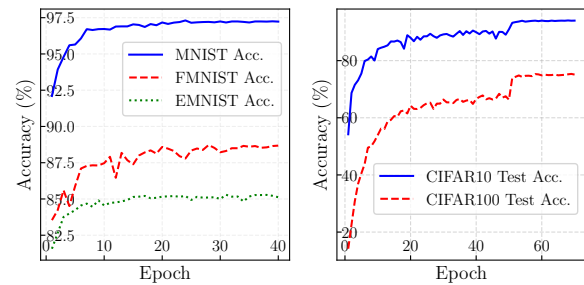
This gradient gating mechanism operates only during training and is analytically defined—offering stability without requiring arbitrary approximations.

Remark 1. *Unlike conventional SG methods that retroactively patch spike derivatives, POLARA embeds boundedness directly into the gradient formulation, echoing Hodgkin and Huxley’s vision of controlled excitability.*

Temporal Robustness and Gradient Stability Figure 4(b) shows that POLARA reliably learns on CIFAR benchmarks despite operating over 50 time steps and a 38-layer backbone. In contrast to recent models that favor shorter time horizons (2–6 steps) and reduced depth to simplify optimization, POLARA maintains stable learning under more biologically realistic conditions (Table 1).

This is further supported by **Appendix E Figure (a)**, which confirms that gradients remain well-controlled across time and depth throughout training. Meanwhile, **Appendix E Figure (b)** illustrates that longer sequences reveal

richer polarization cycles, with multiple spikes per neuron—temporal dynamics that are entirely suppressed in short-horizon models, where neurons effectively behave as single-bit gates.



(a) MNIST Variants Acc. (b) CIFAR10/100 Acc.

Figure 4: (a) Accuracy on MNIST, FashionMNIST, and EMNIST-byclass with a single-layer POLARA model, showing consistent performance across variants. (b) Accuracy on CIFAR-10 and CIFAR-100 with a ResNet-38 backbone, demonstrating POLARA’s scalability to high-complexity benchmarks.

Ablation Studies: Understanding Key Components

We conduct forward ablation studies to isolate the effect of POLARA’s core biologically inspired components: membrane decay, stochastic amplification, spike shaping, refractory suppression, and adaptive thresholding. Each contributes to stable learning and realistic polarization cycles.

We use a fully connected network with one hidden layer, trained on MNIST for 40 epochs. In each ablation, one component is removed or replaced while all others remain intact. Results are shown in Table 2.

- **Membrane Decay & Amplification:** Replacing both with a static leak causes severe accuracy loss, confirming their role in regulating depolarization and maintaining stability.
- **Spike Activation & Gradient Control:** Substituting the shaped spike with a step function and surrogate gradient reduces accuracy, underscoring the value of bounded, differentiable spike shaping.
- **Refractory Suppression:** Removing this inhibition degrades performance and destabilizes spiking, highlighting its role in enforcing polarization cycles.
- **Adaptive Thresholding:** Replacing the adaptive threshold with a fixed value reduces accuracy, disrupting regulation and weakening temporal learning.

These results confirm that POLARA’s components act in coordination. Each plays a distinct role in enabling polarization dynamics and gradient stability—removing any one impairs performance.

FLOP Analysis of POLARA Neurons

Conventional profiling tools such as `ptflops` and `fvcore` overlook neuron-internal operations like decay and

¹<https://github.com/fzenke/spytorch>

Model	Method	Architecture	MNIST	F-MNIST	EMNIST	CIFAR-10	CIFAR-100
POLARA (Ours)	Direct Training	1-layer	97.49% (128)	88.78% (128)	85.12% (128)	—	—
LIF Neuron	Direct Training	1-layer	96.39% (128)	85.00% (128)	83.18% (128)	—	—
POLARA (Ours)	Direct Training	ResNet-38	—	—	—	94.00% (50)	75.12% (50)
(Deng and Gu 2021)	ANN-to-SNN	ResNet-20	—	—	—	92.41% (16)	67.73% (32)
TSC (Han and Roy 2020)	ANN-to-SNN	VGG16	—	—	—	92.79% (64)	69.86% (64)
TET (Deng et al. 2022)	Direct Training	CNN	—	—	—	94.50% (6)	74.62% (4)
GLIF (Yao et al. 2022)	Direct Training	Resnet-18	—	—	—	94.88% (6)	77.28% (6)
LSG (Lian et al. 2023)	Direct Training	ResNet-19	—	—	—	94.41% (2)	76.22% (2)
Recdis (Guo et al. 2022)	Direct Training	ResNet-19	—	—	—	95.55% (6)	74.10% (4)
PSG (Wang, Cheng, and Lim 2024)	Direct Training	ResNet-19	—	—	—	95.35% (6)	75.98% (4)
PLIF (Fang et al. 2021)	Direct Training	CifarNet	—	—	—	93.50% (8)	—
LTMD (Wang, Cheng, and Lim 2022)	Direct Training	DenseNet	—	—	—	94.19% (4)	—
Dspike (Li et al. 2021)	Direct Training	ResNet-18	—	—	—	94.25% (6)	73.49% (4)
AGL (Jiang et al. 2025)	Direct Training	ResNet-19	—	—	—	96.18% (2)	78.84% (2)

Table 1: Classification accuracy (%) on MNIST and CIFAR benchmarks. Time steps used per model are shown in parentheses. POLARA is trained from scratch with a deeper ResNet and more time steps, without ANN–SNN conversion, surrogate tuning, pretraining, or extended training schedules, which simplify optimization.

Component Removed or Replaced	Accuracy (%)
Membrane Decay and Stochastic Amplification	45.18%
Spike Activation and Gradient Control	96.80%
Refractory Suppression	94.24%
Adaptive Thresholding	94.25%
POLARA (Baseline)	97.49%

Table 2: Ablation results on MNIST using a 1-hidden-layer model (40 epochs). Accuracy drops significantly when core forward components are removed or replaced.

adaptive thresholds. We instead compute FLOPs manually from forward equations, following conventions in Spiking-Jelly and Wu *et al.* (Wu et al. 2018).

We count only runtime, value-dependent operations. Constant initializations are excluded, as they are cached or shared across steps.

FLOPs are reported per neuron and timestep in Table 3. While higher than in minimal models, these costs reflect deliberate biological modeling. Total cost scales linearly with time, neurons, and depth, and provides a conservative upper bound.

Component	FLOPs	Description
Decay	$3k+3$	$u \leftarrow u(1 - \sum_i A e^{-\alpha_i} + \epsilon)$
Integration	1	$u \leftarrow u + x$
Stochastic Boost	6	$u \leftarrow u(1 + B e^{\beta(u-\vartheta)})$
Activation	9	$s \leftarrow \sigma(u(1 + \Delta(1 + \Delta e^{-\gamma\Delta})))$
Refractory	$7nk + k + 2$	$u \leftarrow u(1 + \sum_i \sum_{j=1}^n D_j e^{-(i-\mu_j)^2/2\sigma_j^2})$
Adaptive Threshold	$3k+2$	$\vartheta_i \leftarrow \vartheta_{i-1}(1 + \sum_i E e^{-\eta_i})$

Table 3: FLOPs per internal neuron operation per timestep. Constants are reused across steps. k = temporal window size; n = number of Gaussians (typically 3).

Remark 2. *These estimates assume dense computation. On neuromorphic hardware, event-driven execution may lower*

cost, as silent neurons often incur negligible compute. POLARA anticipates this regime, where biologically grounded dynamics become advantageous.

Conclusion

We introduced POLARA, a biologically grounded neuron model that captures core polarization phases—depolarization, repolarization, hyperpolarization, refractory suppression, and adaptive thresholding—through structured membrane dynamics. These components interact continuously without resets, preserving stable signal flow across both time and depth.

By bounding membrane evolution in both forward and backward passes, POLARA enables gradient-stable training in deep architectures with long temporal horizons. Unlike prior models that depend on resets or heuristic stabilization tricks, POLARA maintains consistent internal dynamics across all layers and timesteps.

Ablation studies confirm that each component contributes meaningfully to training stability and signal fidelity. POLARA thus offers a scalable foundation for biologically structured computation—capable of supporting deep learning without compromising learnability.

While POLARA already departs from conventional designs by modeling rich internal dynamics within each neuron, we believe this is only the beginning. These dynamics suggest that learning need not rely solely on global error signals or reset-based simplifications. Instead, future architectures may draw on neuron-level behavior—such as spike timing and internal modulation—to shape synaptic connectivity and guide more local, biologically grounded learning. If each neuron carries more computational structure, even smaller networks may become viable, shifting how we think about scale and learning in biologically inspired systems.

Acknowledgments

The author is grateful to those who pushed this work forward through both support and challenge.

References

- Bengio, Y.; Mesnard, T.; Fischer, A.; Zhang, S.; and Wu, Y. 2015. STDP as presynaptic activity times rate of change of postsynaptic activity. *arXiv preprint arXiv:1509.05936*.
- Deng, S.; and Gu, S. 2021. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*.
- Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*.
- Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2021. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2661–2671.
- Gerstner, W. 1995. Time structure of the activity in neural network models. *Physical review E*, 51(1): 738.
- Guo, Y.; Tong, X.; Chen, Y.; Zhang, L.; Liu, X.; Ma, Z.; and Huang, X. 2022. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 326–335.
- Han, B.; and Roy, K. 2020. Deep spiking neural network: Energy efficiency through time based coding. In *European conference on computer vision*, 388–404. Springer.
- Hille, B. 2001. Ion Channels of Excitable Membranes Third Edition. (*No Title*).
- Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7): 1527–1554.
- Hodgkin, A. L.; and Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4): 500.
- Jiang, J.; Wang, L.; Jiang, R.; Fan, J.; and Yan, R. 2025. Adaptive Gradient Learning for Spiking Neural Networks by Exploiting Membrane Potential Dynamics. *arXiv preprint arXiv:2505.11863*.
- LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*, 521(7553): 436–444.
- Li, Y.; Guo, Y.; Zhang, S.; Deng, S.; Hai, Y.; and Gu, S. 2021. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in neural information processing systems*, 34: 23426–23439.
- Li, Y.; Zhao, F.; Zhao, D.; and Zeng, Y. 2024. Directly training temporal Spiking Neural Network with sparse surrogate gradient. *Neural Networks*, 179: 106499.
- Lian, S.; Shen, J.; Liu, Q.; Wang, Z.; Yan, R.; and Tang, H. 2023. Learnable Surrogate Gradient for Direct Training Spiking Neural Networks. In *IJCAI*, 3002–3010.
- Liu, Z.; Hu, H.; Xu, K.; et al. 2021. Spiking ResNet: Deep Spiking Neural Network with Residual Connections. *arXiv preprint arXiv:2102.04159*.
- Lobo, J. L.; Del Ser, J.; Bifet, A.; and Kasabov, N. 2020. Spiking neural networks and online learning: An overview and perspectives. *Neural Networks*, 121: 88–100.
- Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671.
- Mozafari, M.; Kheradpisheh, S. R.; Masquelier, T.; Nowzari-Dalini, A.; and Ganjtabesh, M. 2018. First-spike-based visual categorization using reward-modulated STDP. *IEEE transactions on neural networks and learning systems*, 29(12): 6178–6190.
- Neftci, E. O.; Mostafa, H.; and Zenke, F. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6): 51–63.
- Pillow, J. W.; Shlens, J.; Paninski, L.; Sher, A.; Litke, A. M.; Chichilnisky, E.; and Simoncelli, E. P. 2008. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207): 995–999.
- Singh, S.; Sarma, A.; Lu, S.; Sengupta, A.; Kandemir, M. T.; Neftci, E.; Narayanan, V.; and Das, C. R. 2022. Skipper: Enabling efficient snn training through activation-checkpointing and time-skipping. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 565–581. IEEE.
- Teeter, C.; Iyer, R.; Menon, V.; Gouwens, N.; Feng, D.; Berg, J.; Szafer, A.; Cain, N.; Zeng, H.; Hawrylycz, M.; et al. 2018. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature communications*, 9(1): 709.
- Wang, S.; Cheng, T. H.; and Lim, M.-H. 2022. LTMD: learning improvement of spiking neural networks with learnable thresholding neurons and moderate dropout. *Advances in Neural Information Processing Systems*, 35: 28350–28362.
- Wang, S.; Cheng, T. H.; and Lim, M.-H. 2024. Potential distribution adjustment and parametric surrogate gradient in spiking neural networks. *Neurocomputing*, 129189.
- Wang, X.; Lin, X.; and Dang, X. 2020. Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Networks*, 125: 258–280.
- Wu, Y.; Deng, L.; Li, G.; Zhu, J.; and Shi, L. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12: 331.
- Yao, X.; Li, F.; Mo, Z.; and Cheng, J. 2022. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 32160–32171.
- Zenke, F.; and Vogels, T. P. 2021. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, 33(4): 899–925.