Probabilistic LIF Neurons Improve Learning in Recurrent Spiking Neural Networks

Sebastian Higuchi
Adaptive AI Lab
Institute for Robotics and Cognitive Systems
University of Lübeck
Ratzeburger Allee 160
23562 Lübeck, Germany
sebastian.higuchi@uni-luebeck.de

Niels A. Kloosterman
Adaptive Brain and Cognition Lab
Department of Psychology
University of Lübeck
Maria Goeppert-Straße 9a
23562 Lübeck, Germany
n.kloosterman@uni-luebeck.de

Stefan Hallermann
Carl-Ludwig-Institute of Physiology
Medical Faculty
University Leipzig
Liebigstraße. 27
04103 Leipzig, Germany
hallermann@medizin.uni-leipzig.de

Sebastian Otte
Adaptive AI Lab
Institute for Robotics and Cognitive Systems
University of Lübeck
Ratzeburger Allee 160
23562 Lübeck, Germany
sebastian.otte@uni-luebeck.de

Abstract

Training recurrent spiking neural networks (SNNs) with leaky integrate-and-fire (LIF) neurons is often slow, particularly during the early phase, when networks must first establish sufficient spike activity to form patterns. Strategies such as low firing thresholds or high-magnitude weight initialization can increase early spiking, but typically introduce instabilities and impair learning. Here we introduce a modification of classical LIF and parameterized LIF (PLIF) neurons, in which spikes are generated probabilistically, including a proper surrogate gradient formulation. The membrane potential parameterizes the instantaneous spike probability, and spikes are sampled as Bernoulli variables at each time step, whereas underlying LIF membrane dynamics remain unchanged. This stochastic activation stabilizes early spike activity and substantially accelerates learning. In two benchmark tasks, these probabilistic LIF networks surprisingly achieve substantially higher classification accuracy than its deterministic LIF baselines. These findings suggest that probabilistic spike generation may provide a promising new perspective for building compact and effective spiking architectures.

1 Introduction

Spiking neural networks (SNNs) based on leaky integrate-and-fire (LIF) neurons are widely studied as a promising framework for energy-efficient and neuromorphic machine learning. Their event-driven computation, temporal dynamics, and compatibility with neuromorphic hardware make them an attractive alternative to conventional artificial neural networks. However, despite significant progress in surrogate gradient methods that enable gradient-based training, learning in SNNs often remains considerably slower and less stable than in conventional deep networks. One persistent challenge lies in the early stages of training, where the network must first establish sufficient spiking activity to propagate information through the system.

The Third Austrian Symposium on AI and Vision (AIROV26).

In deterministic LIF networks, meaningful learning only occurs once spike activity emerges and begins to transmit signals across layers. During the initial phase of training, however, membrane potentials often remain below threshold for extended periods, leading to rare spike activity. As a result, gradient flow vanishes almost entirely, slowing the formation of meaningful representations. A simple workaround is to initialize synaptic weights with unusually large magnitudes or to reduce neuronal thresholds in order to provoke early spikes. These strategies increase activity, but they frequently introduce instability or highly irregular dynamics that hinder the formation of structured network representations.

Stochasticity and noise have long been considered computational resources in networks of spiking neurons, supporting computation, inference, and learning in such systems [5]. Gradient estimators for stochastic binary neurons have been studied in [1], and stochastic neuron implementations have also been demonstrated directly in neuromorphic hardware [7].

Based on biological evidence, we hypothesize that introducing spontaneous spiking into spiking neural networks may promote flexible state exploration during learning, improving adaptation and task performance. To this end, we introduce a stochastic modification of LIF neurons. Instead of generating spikes deterministically when the membrane potential crosses a threshold, we interpret a nonlinear transformation of the membrane potential as the instantaneous spike *probability*. At each simulation step, the spike output is then sampled as a Bernoulli random trial with this probability. Importantly, all underlying LIF membrane dynamics remain unchanged.

This modification provides a controlled mechanism to induce spike activity during the early stages of training. Even when membrane potentials remain below classical threshold levels, neurons still emit spikes, allowing signals to propagate through the network and enabling gradients to shape synaptic structure. As learning progresses, the network gradually organizes its activity patterns and synaptic weights while maintaining stable dynamics.

We evaluate probabilistic LIF networks on two benchmark classification tasks and compare them to the conventional deterministic LIF baseline.

2 Methods

Among the simplest and most commonly used neuron models in SNNs are the leaky integrate-and-fire (LIF) neuron [4] and its derived variants such as the parametric LIF (PLIF) neuron [3]. Nevertheless, their deterministic threshold mechanism can make optimization difficult in the early phase of learning. Membrane potentials remain below threshold, neurons stay silent, spike-based signal propagation is weak, and learning can be substantially delayed.

To overcome this problem, we introduce a minimal probabilistic extension of LIF neurons that we name PropLIF and PropPLIF, respectively. The central idea is to increase spiking by probabilistic spikes, while membrane dynamics remain unchanged. Neurons thus emit spontaneous spikes in the subthreshold regime, thereby supporting early activity propagation while preserving the simplicity of classical LIF dynamics.

2.1 Deterministic LIF dynamics

The membrane potential in LIF neurons evolves according to $u_t = \alpha u_{t-1} + (1-\alpha)I_t - \theta z_{t-1}$, where u_t denotes the membrane potential at time step t, I_t is the synaptic input current, θ is the firing threshold, and z_{t-1} is the spike emitted at the previous time step. The leakage factor is defined as $\alpha = \exp(-dt/\tau_m)$, with simulation time step dt and membrane time constant τ_m. Whenever a spike occurs, the membrane potential is reset by subtracting the threshold term.

In the standard LIF model, the membrane time constant τ_m is fixed. In the PLIF model, τ_m is learned jointly with the synaptic weights [3]. Apart from this, both models share the same membrane dynamics.

2.2 Probabilistic spike generation

Conventionally, spike generation is deterministic, and occurs only when the membrane potential exceeds threshold θ. We extend this mechanism by a probabilistic firing rule.

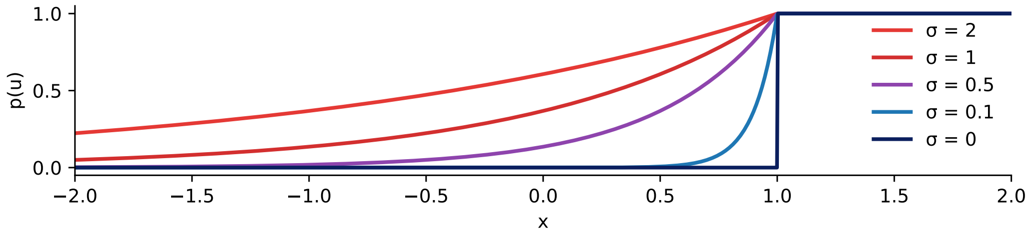

Figure 1: Probability function $p(u) = \exp(\min(0, \frac{u-\theta}{\sigma\theta}))$ for $\theta = 1$ and different values of σ. Larger σ broadens the probabilistic region below the threshold, while $s = 0$ corresponds to the limiting deterministic step-function. Since the resting potential of u is zero, there is a nonzero chance the neuron spikes while resting.

Given membrane potential u_t, we define the instantaneous spike probability as

$$p(u_t) = \exp\left(\min\left(0, -\frac{u_t - \theta}{\sigma\theta}\right)\right), \tag{1}$$

where $\sigma > 0$ controls the width of the probabilistic region below threshold. For $u_t \geq \theta$, the firing probability is 1, whereas for $u_t < \theta$ it decreases exponentially with distance from threshold. Larger values of σ broaden the subthreshold region where spontaneous spikes occur. See Figure 1 for reference.

A spike at time step t is then sampled as $z_t^{(p)} \sim \text{Bernoulli}(p(u_t))$. This formulation preserves the standard membrane update, but extends hard thresholding by a stochastic activation mechanism. As a result, neurons become active even when their membrane potentials are below the deterministic firing threshold, which helps to establish early spike propagation during learning. Figure 2 visualizes this increase in activity.

2.3 Surrogate-gradient formulation

Bernoulli sampling is not differentiable, thus training requires a surrogate gradient approximation. The forward pass uses sampled spikes, while the backward pass propagates gradients through a surrogate function [6]:

$$m_t = u_t \, \text{sg}(g(u_t)) \tag{2}$$

$$z_t = m_t - \text{sg}(m_t) + z_t^{(p)} \tag{3}$$

where $z_t^{(p)} \sim \text{Bernoulli}(p(u_t))$, g is a surrogate gradient function, and $\text{sg}(\cdot)$ denotes the stop-gradient operator.

For the probabilistic neurons as well as the deterministic threshold-based baselines, we use the same surrogate gradient framework as in prior work and approximate the derivative of the Heaviside step function with a double-Gaussian surrogate [8]. In case of probabilistic neurons, we additionally mix this surrogate gradient function with $\frac{d}{du_t}p(u_t)$ to provide better gradients for large negative membrane potentials.

3 Results

We evaluated our PropLIF and PropPLIF neurons within recurrent SNNs on the Sequential-MNIST (S-MNIST) and the Spiking Heidelberg Digits (SHD) benchmark. The models use recurrent architectures with a single hidden layer of probabilistic spiking neurons and a leaky integrator readout. For S-MNIST, we used a network of size $(1, 256^{\mathrm{R}}, 10)$, and for SHD, a network of size $(700, 128^{R}, 20$, where $\cdot^{\mathrm{R}}$ denotes the recurrent hidden layer. All models are trained with backpropagation through time using a negative log-likelihood objective and selected by early stopping on the validation loss. In the probabilistic variants, spike emission is controlled by the parameter σ, which determines the width of the subthreshold probabilistic firing regime; the values used in Table 1 are dataset- and model-specific.

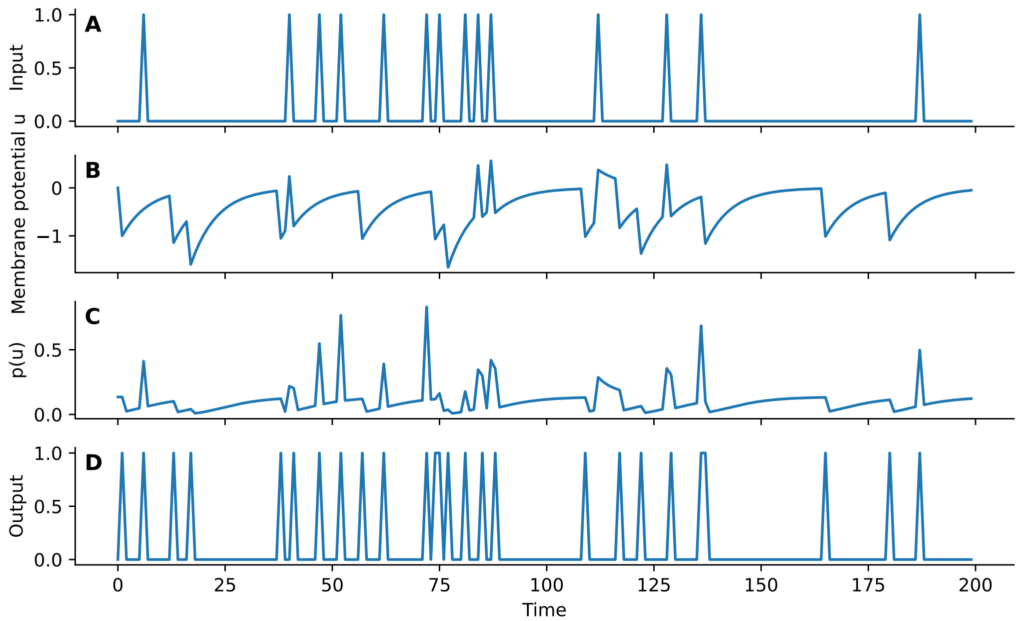

Figure 2: Input response of a ProbLIF neuron with $\sigma = 0.5$ over time. A: Random input. B: The membrane potential u evolves based on the Input and is reduced by $\theta = 1$ whenever an output spike occurs. C: Probability of a spike occurring. D: Stochastic spiking activity of the neuron.

4 Conclusion

We introduced probabilistic extensions of LIF and PLIF neurons—PropLIF and PropPLIF—which preserve classical LIF membrane dynamics while replacing deterministic thresholding with stochastic spike generation. This modification consistently improved learning, yielding faster convergence and higher accuracy, in some cases with substantially fewer parameters. By allowing spontaneous subthreshold spikes, the proposed neurons avoid the silent early-training regime of deterministic SNNs and sustain activity during learning.

Overall, probabilistic spike generation provides a promising new perspective for compact and effective SNNs. Future work should test this principle in other neuron models and explore its broader role as a training mechanism for spiking networks.

Table 1: Comparison of model performance on S-MNIST and SHD datasets. Here, * denotes our reproduced results using publicly available code, and $^{\mathrm{R}}$ denotes a fully recurrent layer. All other layers are exclusively feedforward.

Dataset	Method	Neurons	Parameters (k)	Accuracy (%)
S-MNIST	PLIF [3]	1,64,256$^{\mathrm{R}}$,256,10	112.2/155.1	90.93/91.79
	LIF [9]	1,64,256$^{\mathrm{R}}$,256,10	112.2/155.1	74.91/89.28
	ProbLIF ($\sigma = 1.2$) (ours)	1,256$^{\mathrm{R}}$,10	68.36	95.78
	ProbPLIF ($\sigma = 1$) (ours)	1,256$^{\mathrm{R}}$,10	68.62	**97.09**
SHD	LIF [2]	700,128$^{\mathrm{R}}$,20	108.80	71.40
	PLIF*	700,128$^{\mathrm{R}}$,20	108.69	76.15
	ProbLIF ($\sigma = 1$) (ours)	700,128$^{\mathrm{R}}$,20	108.54	79.99
	ProbPLIF ($\sigma = 1.3$) (ours)	700,128$^{\mathrm{R}}$,20	108.69	**87.72**

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[2] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.

[3] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks, 2021.

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

[5] Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

[6] Sebastian Otte. Flexible and efficient surrogate gradient modeling with forward gradient injection. In *First Austrian Symposium on AI, Robotics, and Vision*. innsbruck university press, 2024.

[7] Tomas Tuma, Angeliki Pantazi, Manuel Le Gallo, Abu Sebastian, and Evangelos Eleftheriou. Stochastic phase-change neurons. *Nature Nanotechnology*, 11:693–699, 2016.

[8] Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

[9] Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling, 2024.