

SiLIF: Structured State Space Model Dynamics and Parametrization for Spiking Neural Networks

Maxime Fabre^{*12} Lyubov Dudchenko^{*3} Younes Bouhadjar¹ Emre Neftci¹⁴

Abstract

Multi-state spiking neurons combine sparse binary activations with rich second-order nonlinear recurrent dynamics, making them a promising alternative to standard deep learning models. However, gradient propagation through these dynamics often leads to instabilities that hinder scalability and performance. Inspired by the stable training and strong performance of state space models (SSMs) on long sequences, we introduce two SSM-inspired Leaky Integrate-and-Fire (SiLIF) neuron models. The first extends a two-state neuron with a learnable discretization timestep and logarithmic reparametrization, while the second additionally incorporates the initialization scheme and structure of complex-state SSMs, enabling oscillatory regimes. Our two SiLIF models achieve new state-of-the-art performance among spiking neuron models on both event-based and raw-audio speech recognition datasets. We further demonstrate a favorable performance-efficiency trade-off compared to SSMs, even surpassing them while using half the computational cost through the use of synaptic delays. Our code is available at <https://github.com/Maxtimmer97/SSM-inspired-LIF>.

1. Introduction

Spiking Neural Networks (SNNs) and their realization in neuromorphic hardware are gaining interest as promising low-power alternatives to artificial neural networks (ANNs) (Yik et al., 2025; Caccavella et al., 2024). In addition to being compact recurrent units with strong temporal encod-

ing capabilities, the biologically inspired binary spiking and the nonlinear feedback mechanism, known as reset, enable sparse event-based communication.¹ Over the past decade, methods leveraging backpropagation through time (BPTT) to train SNNs similarly to recurrent neural networks (RNNs) have significantly improved SNN performance on a variety of benchmarks, thanks to the use of surrogate gradients (Kaiser et al., 2020; Zenke & Ganguli, 2018). Nonetheless, due to vanishing and exploding gradients and saturation caused by binary spikes, the training dynamics of SNNs can remain unstable, limiting their performance (Herranz-Celotti & Rouat, 2022).

A potential avenue for addressing these challenges is through the perspective of state space models (SSMs) (Gu et al., 2022a; Gupta et al., 2022; Orvieto et al., 2023). SSMs have recently demonstrated superior long-sequence modeling performance on various benchmarks, often outperforming traditional RNNs. This performance gain is believed to stem from SSMs’ linear recurrent dynamics (unlike traditional RNNs or SNNs) along with several training optimizations (Orvieto et al., 2023). In fact, dedicated parametrizations and initialization introduced with deep learning SSMs such as the S4 model (Gu et al., 2022a) have been shown to provide robust and backpropagation-friendly recurrent state dynamics (Zucchet & Orvieto, 2024).

This study leverages the strengths of the S4 model to enhance nonlinear recurrent SNNs on spatiotemporal sequences. Specifically, we establish parallels between the Adaptive Leaky Integrate-and-Fire (AdLIF) neuron (Bittar & Garner, 2022), the Resonate and Fire (RF) neuron (Izhikevich, 2001), and SSMs. Building on these parallels, we propose two models: (1) The SSM-Inspired LIF (SiLIF) model, a S4-based enhancement of the AdLIF neuron with timestep training and logarithmic reparametrization. (2) The complex-valued SSM-Inspired LIF (C-SiLIF) model, a novel top-down designed spiking neuron employing the

¹Peter Grünberg Institute, Forschungszentrum Jülich, Jülich, Germany ²Groningen Cognitive Systems and Materials Center (CogniGron), University of Groningen, Groningen, The Netherlands ³ETH Zürich, Zürich, Switzerland ⁴RWTH Aachen University, Aachen, Germany. Correspondence to: Maxime Fabre <m.fabre@fz-juelich.de>, Lyubov Dudchenko <ldudchenko@ethz.ch>.

¹In the following, we refer to such models as reset-based spiking models, in contrast to non-reset models that only offer linear recurrence and non-spiking models that do not employ binary or at least graded spiking.

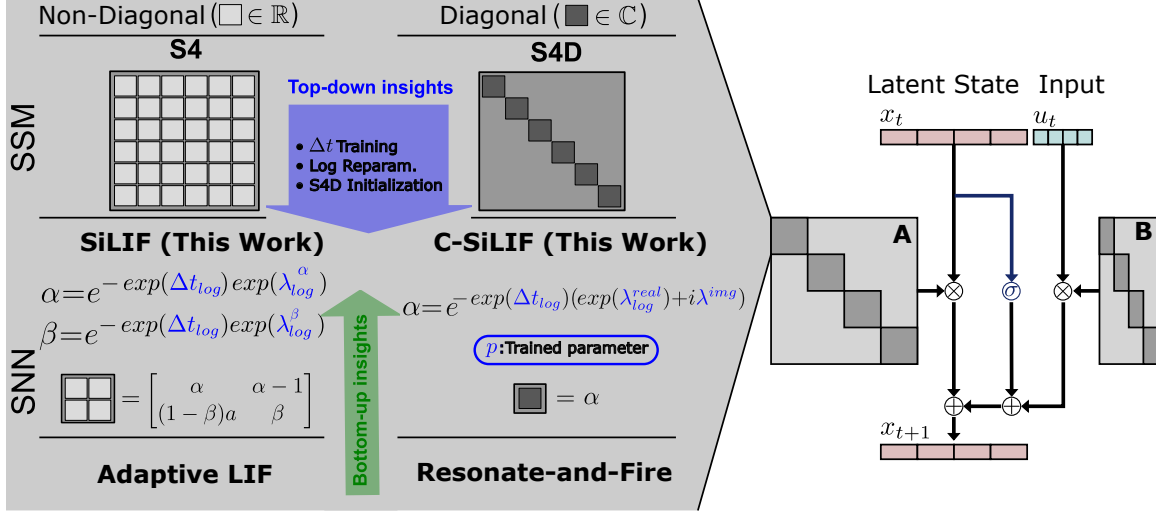


Figure 1. Our proposed method distills features from modern structured state space models (SSMs) to build novel high-performing spiking neural networks (SNNs). The SiLIF model is an enhancement of the adaptive LIF (AdLIF) neuron, achieved by reparametrizing its state transition matrix according to the S4 model. The C-SiLIF additionally exploits the complex representation and specific initialization of the S4D model, drawing a parallel with the resonate-and-fire (RF) neuron. The right part of the figure illustrates the similar structure of SSMs and SNNs, plus the SNN-exclusive reset mechanism.

diagonal S4 (S4D) (Gupta et al., 2022) resonant dynamics, parametrization, and initialization with a matching reset mechanism.

Following previous work, we evaluate these reset-based spiking models on a suite of keyword classification datasets spanning event-based and raw-audio domains. Thanks to their integrated SSM-inspired features and improved training stability, our SiLIF models achieve new state-of-the-art (SOTA) performance across all covered tasks and even outperform SSM models with synaptic delays. Our key contributions are:

1. Establishing theoretical bridges between SSMs and spiking neural networks (SNNs), revealing key insights into their shared and specific training dynamics.
2. Proposing two novel reset-based spiking neuron models that incorporate SSM-derived dynamics and parametrizations.
3. Demonstrating significant accuracy improvements on audio classification tasks while maintaining efficiency, surpassing previous reset-based models.
4. Demonstrating Pareto-optimal accuracy-efficiency balance compared to SSM models across speech recognition tasks, and surpassing them with half the number of synaptic operations (SOPs) - our proxy for MACs in SNN weight matrices - through the insertion of synaptic delays.

2. Related Work

2.1. Spiking Neural Networks for Spoken Word Classification Tasks

Training SNNs has been vastly improved by techniques from deep learning, such as the surrogate gradient method (Nefci et al., 2019), error backpropagation mechanisms (Lee et al., 2016), and ANN-to-SNN conversion (Bu et al., 2022). One promising application for spiking neurons is sequence processing, such as speech recognition.

Inspired by previous neuroscience studies (Brette & Gerstner, 2005; Gerstner et al., 2014; Salaj et al., 2021; Yin et al., 2021), Bittar & Garner (2022) successfully introduced an Adaptive Leaky Integrate-and-Fire (AdLIF) neuron for audio classification. Several works subsequently refined the AdLIF model, either by constraining parameters during training (Deckers et al., 2024), or by employing different discretization approaches (Baronig et al., 2025). However, these only led to marginal improvements on key benchmarks. Our work significantly advances performance on benchmarks by extending the AdLIF model through SSM-inspired training dynamics.

In parallel, the resonate-and-fire (RF) neuron model (Izhikevich, 2001) was also extended in deep SNN works (Alkhamissi et al., 2021; Frady et al., 2022; Higuchi et al., 2024), but their performance remains below the state-of-the-art (SOTA), even when scaled up and employing an SSM structure Huber et al. (2025). Our improved initialization and parametrization, inspired by SSMs, lead to the first RF-style spiking neuron achieving

SOTA performance on audio classification benchmarks. Karilanova et al. (2025) also draw parallels between SSMs and SNNs but do not derive enhancements for neuron models.

Synaptic delays further enhance SNNs’ temporal processing capabilities, achieving SOTA accuracy on event-based datasets using LIF neurons (Hammouamri et al., 2024). Synaptic delays can also be combined with richer models. For instance, Deckers et al. (2024) extend the AdLIF model with learnable delays to improve accuracy on keyword spotting tasks. We note that delay-based models require per-synapse buffers to store incoming spikes and deliver them when the delay has passed, thus requiring additional on-chip memory, registers, and compute. While our models meet or exceed the accuracy of several delay-based models, we show that adding synaptic delays to the SiLIF achieves new SOTA performance.

2.2. State Space Models for Spiking and Event-driven Deep Neural Networks

Linear recurrent SSMs emerged as reference models for long sequences (Gu et al., 2022a; 2020; Smith et al., 2023; Orvieto et al., 2023), outperforming vanilla transformer models in the long range arena benchmark (Tay et al., 2021). Gated SSM variants, such as Mamba (Gu & Dao, 2024), further improve SSM performance with selective data-dependent gating (Gu & Dao, 2024; Dao & Gu, 2024; Beck et al., 2024). Such gating mechanisms endow SSMs with LSTM-style input and forget gates, while maintaining their linear recurrency and fast, parallelized training. While these models excel on long sequence tasks, our SiLIF models exhibit a more favorable accuracy-efficiency balance on shorter-sequence speech tasks.

Several works propose to attach neuron dynamics and binary spiking to SSMs to enhance their efficiency (Huang et al., 2024; Bal & Sengupta, 2025; Shen et al., 2025). Notably, SpikingSSM (Shen et al., 2025) appends a LIF neuron to the S4D model (Gupta et al., 2022) and proposes a surrogate network to maintain fast training. The modifications, however, incur accuracy degradation that is not compensated by the efficiency gains, leaving our SiLIF models Pareto-optimal in comparison.

In parallel, conventional deep SSMs were applied to event-based data (Schöne et al., 2024; Zubić et al., 2024). Event-SSM (Schöne et al., 2024) reaches, for instance, new SOTA performance on event-based keyword classification but at a significant computational overhead, resulting in unfavorable accuracy-efficiency trade-offs compared to our SiLIF models.

3. Methods

3.1. State Space Models

State space models are based on the linear latent state dynamics (Eq. 1 below). Each scalar input feature $u \in \mathbb{R}$ is expanded to N dimensions through a complex vector $\mathbf{B} \in \mathbb{C}^N$, yielding the complex state $x \in \mathbb{C}^N$, which is recursively updated through the transition matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$. The readout $y \in \mathbb{R}$ is defined as the real part $\text{Re}(\cdot)$ of the state projection through $\mathbf{C} \in \mathbb{C}^N$ plus the direct signal projection through $\mathbf{D} \in \mathbb{R}$. The discretized form for a timestep Δt is obtained with the zero-order hold (ZOH) method:

$$\begin{aligned} x_t &= \bar{\mathbf{A}}x_{t-1} + \bar{\mathbf{B}}u_t \\ y_t &= \text{Re}(\bar{\mathbf{C}}x_t) + \bar{\mathbf{D}}u_t \end{aligned} \quad (1)$$

where $\bar{\mathbf{A}} = e^{\mathbf{A}\Delta t}$, $\bar{\mathbf{B}} = \mathbf{A}^{-1}(\bar{\mathbf{A}} - \mathbf{I})\mathbf{B}$, $\bar{\mathbf{C}} = \mathbf{C}$, $\bar{\mathbf{D}} = \mathbf{D}$. These dynamics generalize to an input of dimension H by instantiating separate SSM blocks for each input dimension, resulting in block-diagonal matrices $\mathbf{A}$ and $\mathbf{B}$ (Fig. 1). Following Gu et al. (2022a), SSM blocks are then typically placed in successive layers interleaved with multi-layer perceptrons (MLP) and gated linear units (GLU).

3.2. AdLIF Neurons in the SSM Domain

Second-order spiking neurons take inspiration from the dynamics of biological neurons to enable complex and scalable temporal processing (Izhikevich, 2001; Brette & Gerstner, 2005; Gerstner et al., 2014; Salaj et al., 2021; Yin et al., 2021). Our work builds on the Adaptive Leaky Integrate-and-Fire (AdLIF) model (Bittar & Garner, 2022), which achieved competitive performance on speech recognition tasks. The AdLIF neuron follows second-order dynamics, where an adaptation variable w is coupled with the neuron’s membrane potential u :

$$\begin{aligned} u_t &= \alpha u_{t-1} + (1 - \alpha)(I_t - w_{t-1}) - \theta s_{t-1} \\ w_t &= \beta w_{t-1} + a u_{t-1} + b s_{t-1} \\ s_t &= (u_t \geq \theta) \end{aligned} \quad (2)$$

where t is the timestep, I_t is the input signal, s_t the potential binary spike, θ is the spiking threshold, and a and b sub- and above-threshold feedback parameters. The decay constants α and β are defined as $\alpha = e^{-\Delta t/\tau_u}$ and $\beta = e^{-\Delta t/\tau_w}$, where Δt is the discretization timestep and τ_u and τ_w are the continuous-space time constants of the u and w variables. The model is trained in the discrete domain, using the decay constants α and β as learnable parameters, without involving τ_u or τ_w .

In its subthreshold regime, the AdLIF neuron (Eq. 2) can be mapped to a linear SSM block (Eq. 1) identifying the state

$x_t = \begin{bmatrix} u_t \\ w_t \end{bmatrix}$ and using parameters:

$$\bar{\mathbf{A}} = \begin{bmatrix} \alpha & \alpha - 1 \\ a & \beta \end{bmatrix}, \bar{\mathbf{B}} = \begin{bmatrix} 1 - \alpha \\ 0 \end{bmatrix}, \bar{\mathbf{C}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \bar{\mathbf{D}} = 0 \quad (3)$$

and noting $s_t = 0$ in the subthreshold regime. The only remaining differences between AdLIF and SSMs are the SSM’s state expansion and the spike-triggered feedback. This feedback captures the neuron’s reset mechanism, which is applied to the states u and w through θ and b , respectively, making the state dynamics nonlinearly dependent on the neuron membrane potential. This feature plays a crucial role in the sparsity and nonlinear adaptation of SNNs. In the following sections, we present two model variants that merge efficient SSM features with nonlinear adaptive spiking neuron models.

We note that bridges between SNNs and SSMs have already been investigated in previous work (Huber et al., 2025; Karilanova et al., 2025), but without leveraging the associated training dynamics. We study these discrepancies in more detail in the next two paragraphs.

3.3. SiLIF Model: SSM-inspired Parametrization

The SSM-inspired LIF neuron (SiLIF) is derived by integrating SSM features into the AdLIF neuron. Firstly, we note that structured SSMs such as S4 (Gu et al., 2022a) are generally used in their discretized form (Eq. 1), while employing continuous-domain state-space variables $\mathbf{A}$ and $\mathbf{B}$ as trained parameters. Following this approach, rather than directly training the discrete decay factors α and β of the AdLIF (Eq. 2), we train their continuous-domain counterparts $\lambda^\alpha = \tau_u^{-1}$ and $\lambda^\beta = \tau_w^{-1}$.

Unlike most prior SNN studies, which treat the timestep Δt as a fixed dataset-dependent constant, following S4’s approach (Gu et al., 2022a), we define it as a learnable parameter for each SiLIF neuron. This additional flexibility allows the model to cover a wider range of transition dynamics regimes across neurons (Fig. 2).

Additionally, we implement S4’s logarithmic reparametrization to enhance numerical stability and improve optimization smoothness. We train the logarithmic counterparts λ_{log}^α , λ_{log}^β and Δt_{log} , such that $\lambda^\alpha = \exp(\lambda_{log}^\alpha)$, $\lambda^\beta = \exp(\lambda_{log}^\beta)$ and $\Delta t = \exp(\Delta t_{log})$. This transformation reduces the risk of vanishing and exploding gradients, thus improving the training stability of recurrent architectures (Zucchet & Orvieto, 2024). We extend and evaluate this observation on SNNs for the first time here.

Furthermore, since $\exp(\lambda_{log}^\alpha)$, $\exp(\lambda_{log}^\beta)$ and $\exp(\Delta t_{log})$ are strictly positive, the effective decay factors are constrained within a stable regime $0 \leq \alpha, \beta \leq 1$ without requiring explicit clamping, as previously employed in AdLIF

models for stability (Bittar & Garner, 2022). The resulting SiLIF model is described in Alg. 1.

Algorithm 1 SiLIF: an SSM-inspired LIF neuron

Initialize trainable parameters:

$\lambda_{log}^\alpha \sim \mathcal{U}(\log(\lambda_{min}^\alpha), \log(\lambda_{max}^\alpha))$,
 $\lambda_{log}^\beta \sim \mathcal{U}(\log(\lambda_{min}^\beta), \log(\lambda_{max}^\beta))$, $\Delta t_{log} \leftarrow \log(\Delta t_0)$,
 $a \sim \mathcal{U}(a_{min}, a_{max})$, $b \sim \mathcal{U}(b_{min}, b_{max})$

Initialize states:

$u \sim \mathcal{U}(0, 1)$, $w \sim \mathcal{U}(0, 1)$, $s \sim \mathcal{U}(0, 1)$

Forward pass:

$\alpha \leftarrow \exp(-\exp(\lambda_{log}^\alpha) \cdot \exp(\Delta t_{log}))$,

$\beta \leftarrow \exp(-\exp(\lambda_{log}^\beta) \cdot \exp(\Delta t_{log}))$

Clamp $a \in [a_{min}, a_{max}]$, $b \in [b_{min}, b_{max}]$

for t in sequence **do**

$w \leftarrow \beta \cdot w + a \cdot u + b \cdot s$

$u \leftarrow \alpha \cdot (u - s) + (1 - \alpha) \cdot (I_t - w)$

$s \leftarrow (u \geq \theta)$

end for

3.4. C-SiLIF Model: SSM-Inspired Complex State

State space models like S4 use complex-valued parameters and states, $x = x^{real} + ix^{img} \in \mathbb{C}$, which was recently shown to improve parameter efficiency and stability relative to real-valued parametrization (Ran-Milo et al., 2024). A scalar complex transition $x_t = ax_{t-1} + bu_t$ with $a, b \in \mathbb{C}$, corresponds to the following 2x2 real-valued system:

$$\begin{bmatrix} x_t^{real} \\ x_t^{img} \end{bmatrix} = \begin{bmatrix} a^{real} & -a^{img} \\ a^{img} & a^{real} \end{bmatrix} \begin{bmatrix} x_{t-1}^{real} \\ x_{t-1}^{img} \end{bmatrix} + \begin{bmatrix} b^{real} \\ b^{img} \end{bmatrix} u_t \quad (4)$$

This has a similar format to the state transition in the AdLIF and SiLIF models, except that a complex system exhibits a structured, antisymmetric matrix. As demonstrated by Ran-Milo et al. (2024), such a complex parametrization allows for a more compact model with training-friendly, moderate-value parameters. We empirically corroborate the benefits of such parametrization on spiking neurons using event-based datasets (Sec. 4.2).

Inspired by this complex constrained approach, we propose the complex-valued SSM-inspired LIF (C-SiLIF). C-SiLIF employs a complex state and follows the same dynamics as the 2x2 system described in Eq. 4. While S4D (Gupta et al., 2022) typically freezes the input projection $b = 1 + i$ and optimizes the output projection $\bar{\mathbf{C}}$ during training, we empirically found that freezing $\bar{\mathbf{C}} = 2$ and optimizing $b \in \mathbb{R}$ yields better performance for our model. As in SiLIF, we incorporate a reset by introducing feedback from spiking events into the neuronal state. Since the output consists of twice the real part of the state variable, half the spike value is fed back into the state space to maintain consistent pre-synaptic dynamics.

Following the same reasoning as for the SiLIF model, we

use heterogeneous parameters, logarithmic reparametrization, and timestep training for the transition parameters α^{real} and α^{img} to enhance the model’s stability and capacity. Additionally, for the C-SiLIF, we utilize S4D-Lin initialization derived from orthogonal polynomial projections for optimal state dynamics (Gu et al., 2022b).

The final version of our complex-valued diagonal SSM-inspired LIF (C-SiLIF) is presented in Alg. 2. While C-SiLIF closely resembles the Resonate and Fire (RF) model (Izhikevich, 2001), the novel parametrization, initialization, and discretization significantly enhance its stability (Zucchet & Orvieto, 2024), dynamical richness (Fig. 2), and performance (Sec. 4.2) compared to prior RF models.

Algorithm 2 C-SiLIF: SiLIF with complex state

Initialize trainable parameters:

$\lambda_{log}^{real} \leftarrow \log(0.5), \lambda_{log}^{img} \leftarrow \pi,$
 $\Delta t_{log} \sim \mathcal{U}(\log(\Delta t_{min}), \log(\Delta t_{max})), b \sim \mathcal{U}(0, 1)$

Initialize states:

$u \sim \mathcal{U}(0, 1), s \sim \mathcal{U}(0, 1)$

Forward pass:

$\alpha \leftarrow \exp((- \exp(\lambda_{log}^{real}) + i \cdot \lambda_{log}^{img}) \cdot \exp(\Delta t_{log}))$

for t **in** sequence **do**

$u \leftarrow \alpha \cdot (u - 0.5 \cdot s) + b \cdot I_t$

$s \leftarrow (2\text{Re}(u) \geq \theta)$

end for

3.5. Neuronal Dynamics Regimes

We now examine the impact of the SSM-inspired features on the dynamics of our C-SiLIF and SiLIF models. We represent the distribution of eigenvalues of the transition matrix $\tilde{A}$ in the complex plane and compare these to the RF model and the cAdLIF (Deckers et al., 2024). The cAdLIF simply constrains the adaptive parameter a to positive values (Eq. 2), which helps stability and yields improved performance. The distributions are obtained after training models on the SSC dataset (see Sec. 4.2) and presented in Fig. 2.

The cAdLIF model displays the most constrained eigenvalues, with mostly real values above 0.5. In fact, cAdLIF and AdLIF use strict constraints on their parameters range, with decays α and β slightly below 1 and the adaptation a within $[-1, 1]$, which mathematically results in predominantly real eigenvalues. The RF model, with its complex-constrained 2x2 system and fixed timestep Δt , yields a very concentrated eigenvalue distribution around 1 with limited imaginary components.

On the other hand, our SiLIF and C-SiLIF models cover a considerably wider spectrum of neuronal dynamics: The SiLIF covers most of the real-positive unit circle due to the absence of clamping on the transition parameters α and β and the Δt training, coinciding with significant performance improvements. Additionally, the complex system defined by the C-SiLIF allows transition parameters to be negative,

leading to an eigenvalue distribution covering the entire unit circle. This allows our models to produce a wider, more flexible range of trained neuronal dynamics, including oscillatory ones which the cAdLIF or AdLIF typically cannot, as shown in the Appendix A.1.

We demonstrate in the Experiments section below that these widened neuronal dynamics arising from the Δt training, the absence of parameter clamping, or the specific complex initialization significantly improve performance on audio classification tasks, establishing new SOTA results with our SSM-inspired models.

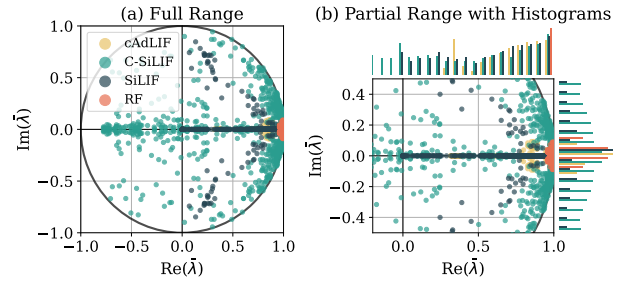


Figure 2. Scatter plot and histogram of state transition matrix eigenvalues for different pre-trained neuron models on the SSC dataset. (a) depicts the full unit circle, while (b) shows a zoomed view with a logarithmic range histogram of the eigenvalues. The range of covered eigenvalues, especially out of the real axis, corresponds to different accessible neuronal dynamics regimes.

4. Experiments

4.1. Experimental Setup

We evaluate our models on keyword classification tasks, which are common benchmarks to evaluate spiking and recurrent neural networks. We use the Spiking Heidelberg Digits (SHD), the Spiking Speech Commands (SSC) (Cramer et al., 2020), and the Google Speech Commands v0.02 (GSC) dataset (Warden, 2018), in accordance with recent research on neuromorphic models (Bittar & Garner, 2022; Deckers et al., 2024; Baronig et al., 2025; Huber et al., 2025; Hammouamri et al., 2024; Schöne et al., 2024; Queant et al., 2025). GSC contains over 100k samples across 35 classes of spoken command words. SSC is an event-based converted version of GSC using a cochlea model (Cramer et al., 2020). The SHD dataset consists of 10k audio samples, equally distributed across 20 classes of German and English spoken digits, and is converted to events similarly to SSC.

Different studies have used varying temporal binning and resolution for the spiking datasets, which has a significant impact on classification performance. Thus, we handle multiple cases to present fair comparisons: by default, we employ bins of 4ms for event-based datasets, and scale

Table 1. Classification accuracy on SSC and GSC datasets. Models are ranked by Top-1 accuracy (highest at the bottom) within non-delay and delay-based categories. Our models are shown in bold. Non-delay and delay (Del.) models are separated by a horizontal line for each task. Models using inter-neuron recurrent connections are indicated as Rec.

Dataset	Method	Resolution	# Params	Rec.	Del.	Top1 Accuracy
SSC	RadLIF (Bittar & Garner, 2022)	14 ms	3.9M	✓		77.40%
	cAdLIF (Deckers et al., 2024)	10 ms	0.35M			77.50%
	S5-RF (Huber et al., 2025)	4 ms	1.7M			78.8%
	SiLIF, 10 ms bins (ours)	10 ms	0.35M			80.11 ± 0.31%
	SE-adLIF (Baronig et al., 2025)	4 ms	1.6M			80.44 ± 0.26%
	C-SiLIF, 4 ms bins (ours)	4 ms	0.35M			81.59 ± 0.31%
	SiLIF, 4 ms bins (ours)	4 ms	0.35M			82.03 ± 0.25%
	d-cAdLIF (Deckers et al., 2024)	10 ms	0.7M		✓	80.23 ± 0.07%
	DCLS-Delays (Hammouamri et al., 2024)	10 ms	1.2M		✓	80.29 ± 0.06%
	SiLIF + Delays (ours)	10 ms	0.7M		✓	80.62 ± 0.16%
GSC	DelRec (Queant et al., 2025)	5.6 ms	0.37M	✓	✓	82.58 ± 0.08%
	RadLIF (Bittar & Garner, 2022)	10 ms	0.83M	✓		94.51%
	cAdLIF (Deckers et al., 2024)	10 ms	0.3M			94.67%
	SiLIF, 512 hidden units (ours)	10 ms	0.3M			95.25 ± 0.12%
	SiLIF, 1024 hidden units (ours)	10 ms	1.1M			95.49 ± 0.09%
	DCLS-Delays (Hammouamri et al., 2024)	10 ms	1.2M		✓	95.29 ± 0.11%
	d-cAdLIF (Deckers et al., 2024)	10 ms	0.61M		✓	95.69 ± 0.03%
	SiLIF + Delays (ours)	10 ms	0.61M		✓	95.88 ± 0.12%

down to 10 *ms* bins for comparison when required. Additionally, the spiking datasets are binned spatially from 700 to 140 input channels, matching recent studies. For GSC, the samples are fixed to 10 *ms* time bins, obtained from a Mel filterbank with 40 Mel filters, matching the setup used in all similar studies. Finally, we note that the SHD dataset lacks a separate test and validation set, which led previous studies to report the model’s validation accuracy at the best epoch. We follow the same approach here for fair comparison.

Our implementation builds on the SpArch² code by Bittar & Garner (2022). The corresponding network architecture has since been employed by most works in this field (Deckers et al., 2024; Hammouamri et al., 2024; Huber et al., 2025). The network consists of two non-recurrent hidden layers with batch normalization and dropout. The layer width is fixed to 512 units unless otherwise specified. The output layer is fixed to a non-firing Leaky Integrate (LI) neuron whose outputs are passed through a softmax and summed over time. For our SiLIF and C-SiLIF models, the trainable neuron parameters and their initializations are listed in Algorithms 1 and 2, respectively. The spiking threshold is fixed at 1. We use a cross-entropy loss with an Adam optimizer and a Plateau scheduler. We select hyperparameters for our models based on grid searches of the learning rate, dropout, and scheduler (see Appendix A.3 for details). Details of

other parameters, such as initialization ranges, can be found directly in our code.

In addition to the base SiLIF model introduced here, we implement a variant with learnable synaptic delays, termed "SiLIF + Delays", in which each synapse can delay incoming spikes by a certain number of timesteps before propagating them through the network. This variant is implemented separately following the DCLS-Delay framework and its proposed training pipeline (Hammouamri et al., 2024) (see Appendix A.2 for details). Our reported results are averaged over 5 seeds for SHD, and 10 seeds for SSC and GSC. The average μ and the standard deviation σ are reported as $\mu \pm \sigma\%$.

Altogether this project required about 5000 GPU hours on a Nvidia RTX 4090 with 24 GB memory with a shared 377 GB AMD EPYC 9384X 32-Core CPU on an internal cluster, also considering initial exploratory runs. Each run takes an average of 2 hours for the SHD dataset, 15 hours for SSC, and 5 hours for GSC. This allowed for a detailed ablation study and reproduction of previous results with grid search analysis, targeting the most detailed and representative results possible.

4.2. Results on Audio Keyword Classification Tasks

We present results of our C-SiLIF, SiLIF, and SiLIF + Delays models on the SSC and GSC datasets in Table 1 and compare these to the previous state-of-the-art (SOTA) re-

²<https://github.com/idiap/sparch>, released under BSD-3-Clause License.

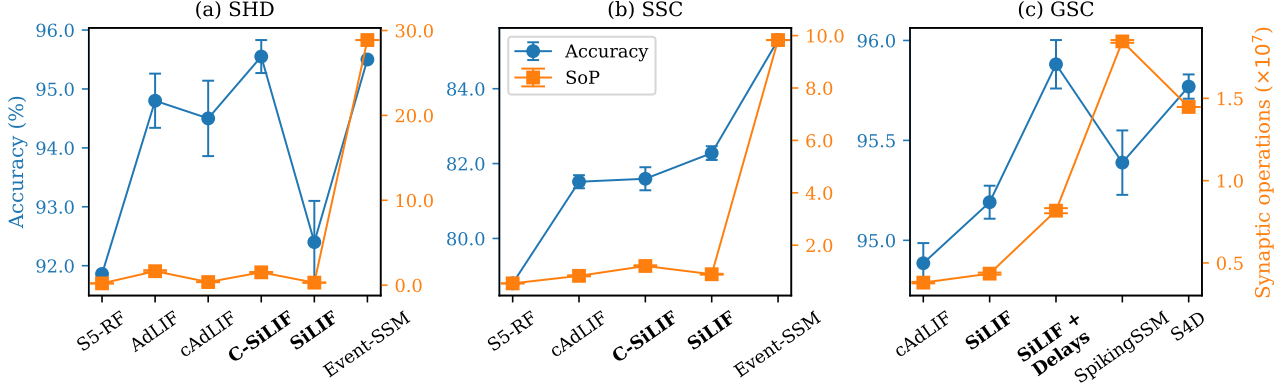


Figure 3. Test accuracy and synaptic operations (SOP) with standard deviation for different models on the 3 audio datasets SHD, SSC, and GSC from left to right. Our models are shown in bold.

sults obtained with reset-based spiking neuron models. For conciseness, the results for the smaller SHD dataset are presented in the Appendix A.4. Along with accuracy, we report the total number of parameters and indicate which models have recurrent connections (Rec.) and which ones have delays (Del.). These two specific features typically improve performance but come at the cost of extra memory or computation, especially when implemented on neuromorphic hardware.

As presented in Table 1, our models achieve significant improvements over previous SOTA models in all categories. The SiLIF model establishes new SOTA performances for non-delay models on both SSC and GSC by clear margins, while the version with synaptic delays establishes a new general SOTA for GSC with spiking neural networks. Our SiLIF + Delays model also achieves a new best performance for SSC at a 10 ms resolution, while our models are only surpassed by the recent work from Queant et al. (2025), which requires an increased resolution of 5.6 ms along with costly recurrent connections.

4.3. Computational Cost vs. Accuracy Analysis

A key motivation for spiking neurons is their potential for reduced latency and energy consumption on dedicated event-based hardware (Mead, 1990; Indiveri et al., 2011; Gao et al., 2018; Merolla et al., 2014; Pei et al., 2019), due to their spike-induced sparsity. Here, we assess model efficiency by reporting the number of synaptic operations (SOPs) per sample. The SOPs are computed here as the number of times a non-zero activity passes through a synaptic weight.

We report accuracy and SOPs from reproduced experiments on the AdLIF (Bittar & Garner, 2022), cAdLIF (Deckers et al., 2024), and our C-SiLIF, SiLIF, and SiLIF + Delays models. All architectures consist of 2 hidden layers of 512 neurons and are trained without activity regularization loss

or any other sparsity-enhancing method. Each reported performance is obtained with 5 random seeds after a grid search.

In our efficiency comparisons with delay-based models, we assume that storing and loading delayed spikes is as costly as accumulating weights. This leads to doubling the number of synaptic operations (SOPs).

Additionally, we compare these models to other SSM-related models. On the spiking datasets SHD and SSC, we also reported results from the S5-RF (Huber et al., 2025) and Event-SSM (Schöne et al., 2024). The S5-RF is a SSM-inspired spiking neuron model employing the HIPPO initialization but no further reparametrization, which is a key feature of our work. Huber et al. (2025) report SOPs per timestep; we thus scale them by the number of timesteps per sample to match our metric. The Event-SSM is a linear model made to process event streams, but without any spiking or reset mechanism, i.e., without any sparsity. Their SOP value computation is detailed in Appendix A.6.

On the GSC task, we report results for the standard S4D (Gupta et al., 2022), the recent SpikingSSM model (Shen et al., 2025), a sparsified S4D model with appended LIF neurons, and Mamba (Gu & Dao, 2024), an enhanced version of S4D with selective gating. While Shen et al. (2025) and Ding et al. (2025) already report results for these models on GSC, the respective use of raw-audio data at very high resolution and of bidirectional models results in orders of magnitude higher SOPs. We thus retrain SpikingSSM, S4D, and Mamba using our training pipeline to ensure a fair comparison (see Appendix A.7 for details). While we evaluate all three across multiple computational scales (Fig. 4), only representative configurations of S4D and SpikingSSM are reported in Fig. 3, as Mamba is consistently dominated in the accuracy-efficiency trade-off.

Fig. 3 summarizes the resulting accuracy and computational

Table 2. Test accuracy on the SHD dataset with ablation of SSM-inspired features of the C-SiLIF model. The configurations represent the best accuracy hyperparameters obtained from a 5-seed sweep for each model with 2 hidden layers of 512 neurons.

Model	Accuracy	Configuration
C-SiLIF	$95.6 \pm 0.3\%$	lr=0.005, dropout=0.1
C-SiLIF w/o half reset and input gating	$94.9 \pm 0.3\%$	lr=0.005, dropout=0.1
C-SiLIF w/o half reset, input gating, and S4D-Lin init.	$93.3 \pm 0.5\%$	lr=0.001, dropout=0.5
C-SiLIF w/o half reset, input gating, S4D-Lin init. and log. reparametrization	$90.2 \pm 0.6\%$	lr=0.001, dropout=0.1
C-SiLIF with Δt constant (uniform distribution)	$95.1 \pm 0.7\%$	lr=0.01, dropout=0.5
C-SiLIF with Δt constant (single value)	$89.3 \pm 1.2\%$	lr=0.005, dropout=0.1

efficiency across the considered models. Since the SHD and SSC experiments here are conducted at a 4 ms resolution, we confirm that our C-SiLIF and SiLIF models, respectively, outperform all reset-based models reproduced. Our C-SiLIF is Pareto-optimal in accuracy-efficiency on the SHD task, while SiLIF and SiLIF + Delays are for the SSC and GSC tasks. Despite achieving top accuracy on the SSC task, the Event-SSM requires nearly an order of magnitude more operations and relies on dense real-valued activations. On the GSC dataset, despite its sparsity, SpikingSSM is outperformed by S4D at iso-compute. Both these linear recurrent models are outperformed by our SiLIF + Delays model at half the computational cost, while the standard SiLIF remains Pareto-optimal at a third of S4D’s budget.

Further on, Fig. 4 shows the scaling curves of S4D, SpikingSSM, and Mamba next to our SiLIF model, scaled from 512 to 1024 hidden units and with added synaptic delays. We observe poor performance from Mamba, which is outperformed by all other models, potentially due to its optimization for large-scale models and long-sequence tasks. While SpikingSSM benefits from sparsity, the associated accuracy degradation makes it less competitive than S4D and SiLIF. Our SiLIF + Delays model shows great performance and is only marginally surpassed by the large S4D configuration, while requiring four times fewer operations. Altogether, these results firmly establish the competitive edge of our reset-based spiking models over SOTA SNN and SSM baselines for audio tasks on neuromorphic hardware.

4.4. Ablation Study

We conduct an ablation study of the C-SiLIF model, considering that it includes all SiLIF features plus the complex-value constraint and S4D initialization. Each ablation configuration is optimized with a hyperparameter grid search to reach optimized performance for this configuration. We report the best obtained performance along with the corresponding hyperparameter configuration in Table 2.

The ablation mostly highlights the importance of the logarithmic reparametrization and the timestep Δt , which

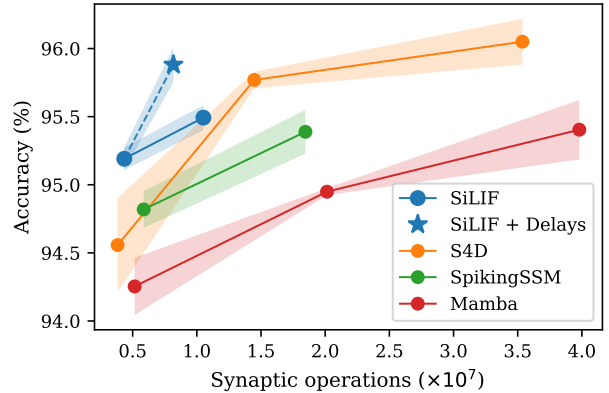


Figure 4. Test accuracy per synaptic operations (SOPs) on the GSC dataset for our SiLIF model and SOTA SSM models at different scales. The standard deviation is represented as a shaded area.

are common features to our two models. Removing reparametrization from lines 3-4 of Table 2 results in a performance drop of more than 3%. Additionally, we show for the first time the importance of a heterogeneous, or even trained Δt for reset-based spiking neurons, especially as the use of a single homogeneous Δt on the last line leads to more than 6% performance drop. As seen in Fig. 2, this directly impacts the range of transition parameters and thus dynamic regimes covered by the model. This underscores the correlation between dynamic regimes and task performance. We also study the impact of these different features in an incremental fashion in Appendix A.8.

5. Conclusion

This work presents a theoretical bridge between state space models (SSMs) and spiking neurons and proposes two novel spiking neuron models integrating SSM-inspired dynamics and parametrization. In particular, we show the importance of heterogeneous timestep parameters Δt along with the use of logarithmic reparametrization, which leads to a wider range of dynamical regimes and, eventually, to improved performance. Our models achieve new state-of-the-art (SOTA) performances across the board on speech recognition tasks,

both for delay and non-delay models, with significant margins and reduced scale. The SiLIF benefits also extend to non-reset models, where our model is Pareto-optimal in performance-efficiency with respect to other SSM-SNN hybrids and even modern SSM models, opening a new avenue for neuromorphic models and systems at the edge. Future work could investigate the use of SSM state expansion in spiking neurons or apply our method to other network architectures or sensory processing tasks.

Acknowledgments

This work was sponsored by the Federal Ministry of Education, Germany (project NEUROTEC-II grant no. 16ME0398K and 16ME0399), by Neurosys as part of the initiative "Cluster4Future" is funded by the Federal Ministry of Education and Research BMBF (03ZU1106CB) and by the Federal Ministry of Education and Research (BMBF) under grant no. 01IS22094E WEST-AI. The authors also extend their gratitude to Professor Elisabetta Chicca for her general feedback on analog hardware implications and Sanja Karilanova for our insightful discussions on the link between state space models and spiking neurons.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

- AlKhamissi, B., ElNokrashy, M., and Bernal-Casas, D. Deep spiking neural networks with resonate-and-fire neurons, 2021. URL <https://arxiv.org/abs/2109.08234>.
- Bal, M. and Sengupta, A. P-spikessm: Harnessing probabilistic spiking state space models for long-range dependency tasks. In *International Conference on Learning Representations*, 2025. URL <https://openreview.net/pdf?id=35gFlnqsgw>.
- Baronig, M., Ferrand, R., Sabathiel, S., and Legenstein, R. Advancing spatio-temporal processing in spiking neural networks through adaptation. *Nature Communications*, 2025. URL <https://www.nature.com/articles/s41467-025-60878-z>.
- Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37:107547–107603, 2024.
- Bittar, A. and Garner, P. N. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16, 2022. ISSN 1662453X. doi: 10.3389/fnins.2022.865897.
- Brette, R. and Gerstner, W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94, 2005. ISSN 00223077. doi: 10.1152/jn.00686.2005.
- Bu, T., Fang, W., Ding, J., Dai, P. L., Yu, Z., and Huang, T. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. In *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.
- Caccavella, C., Paredes-Vallés, F., Cannici, M., and Khacef, L. Low-power event-based face detection with asynchronous neuromorphic hardware. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2024. doi: 10.1109/IJCNN60899.2024.10650843.
- Cramer, B., Stradmann, Y., Schemmel, J., and Zenke, F. 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.
- Dao, T. and Gu, A. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, pp. 10041–10071. PMLR, 2024.

- Deckers, L., Van Damme, L., Tsang, I. J., Van Leekwijck, W., and Latré, S. Co-learning synaptic delays, weights and adaptation in spiking neural networks. *Frontiers in Neuroscience*, 18, 2024. doi: 10.3389/fnins.2024.1360300.
- Ding, H., Dong, W., and Mao, Q. Keyword mamba: Spoken keyword spotting with state space models. *Computer Speech & Language*, pp. 101909, 2025.
- Fraday, E. P., Sanborn, S., Shrestha, S. B., Rubin, D. B. D., Orchard, G., Sommer, F. T., and Davies, M. Efficient neuromorphic signal processing with resonator neurons. *Journal of Signal Processing Systems*, 94, 2022. ISSN 19398115. doi: 10.1007/s11265-022-01772-5.
- Gao, C., Neil, D., Ceolini, E., Liu, S., and Delbruck, T. Deltarnn: A powerefficient recurrent neural network accelerator. In *Proceedings of the 2018 ACM/SIGDA International Symposium on FieldProgrammable Gate Arrays*, FPGA '18, pp. 21–30, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356145. doi: 10.1145/3174243.3174261. URL <https://doi.org/10.1145/3174243.3174261>.
- Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.
- Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL <https://arxiv.org/abs/2008.07669>.
- Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *ICLR 2022 - 10th International Conference on Learning Representations*, 2022a.
- Gu, A., Gupta, A., Goel, K., and Ré, C. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, volume 35, 2022b.
- Gupta, A., Gu, A., and Berant, J. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- Hammouamri, I., Khalfaoui-Hassani, I., and Masquelier, T. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=4r2ybzJnmN>.
- Hassani, I. K., Pellegrini, T., and Masquelier, T. Dilated convolution with learnable spacings. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=Q3-1vRh3HOA>.
- Herranz-Celotti, L. and Rouat, J. Stabilizing spiking neuron training. *arXiv preprint arXiv:2202.00282*, 2022.
- Higuchi, S., Kairat, S., Bohte, S. M., and Otte, S. Balanced resonate-and-fire neurons, 2024. URL <https://arxiv.org/abs/2402.14603>.
- Huang, Y., Liu, Z., Feng, C., Lin, X., Ren, H., Fu, H., Zhou, Y., Xing, H., and Cheng, B. Prf: Parallel resonate and fire neuron for long sequence learning in spiking neural networks, 2024. URL <https://arxiv.org/abs/2410.03530>.
- Huber, T. E., Lecomte, J., Polovnikov, B., and von Arnim, A. Scaling up resonate-and-fire networks for fast deep learning. In Del Bue, A., Canton, C., Pont-Tuset, J., and Tommasi, T. (eds.), *Computer Vision – ECCV 2024 Workshops*, pp. 241–258, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-92460-6.
- Indiveri, G., LinaresBarranco, B., Hamilton, T., van Schaik, A., EtienneCummings, R., Delbruck, T., Liu, S., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., SerranoGotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5:123, 2011. ISSN 1662453X. doi: 10.3389/fnins.2011.00073.
- Izhikevich, E. M. Resonate-and-fire neurons. *Neural Networks*, 14, 2001. ISSN 08936080. doi: 10.1016/S0893-6080(01)00078-8.
- Kaiser, J., Mostafa, H., and Neftci, E. Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14:424, 2020.
- Karilanova, S., Fabre, M., Neftci, E., and Özçelikkale, A. Zero-shot temporal resolution domain adaptation for spiking neural networks. *Neural Networks*, pp. 108483, 2025. URL <https://www.sciencedirect.com/science/article/pii/S0893608025013644>.
- Lee, J. H., Delbruck, T., and Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10, 2016. ISSN 1662453X. doi: 10.3389/fnins.2016.00508.

- Mead, C. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):162936, 1990.
- Merolla, P. A., Arthur, J. V., AlvarezIcaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al. A million spikingneuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668673, 2014.
- Neftci, E. O., Mostafa, H., and Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.
- Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023. URL <https://proceedings.mlr.press/v202/orvieto23a/orvieto23a.pdf>.
- Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106111, 2019.
- Queant, A., Rançon, U., Cottureau, B. R., and Masquelier, T. Delrec: learning delays in recurrent spiking neural networks. *arXiv preprint arXiv:2509.24852*, 2025.
- Ran-Milo, Y., Lumbroso, E., Cohen-Karlik, E., Giryas, R., Globerson, A., and Cohen, N. Provable benefits of complex parameterizations for structured state space models. *Advances in Neural Information Processing Systems*, 37: 115906–115939, 2024.
- Salaj, D., Subramoney, A., Kraisnikovic, C., Bellec, G., Legenstein, R., and Maass, W. Spike frequency adaptation supports network computations on temporally dispersed information. *eLife*, 10, 2021. ISSN 2050084X. doi: 10.7554/eLife.65459.
- Schöne, M., Sushma, N. M., Zhuge, J., Mayr, C., Subramoney, A., and Kappel, D. Scalable event-by-event processing of neuromorphic sensory signals with deep state-space models. In *International Conference on Neuromorphic Systems*, 2024. URL <https://ieeexplore.ieee.org/document/10766552>.
- Shen, S., Wang, C., Huang, R., Zhong, Y., Guo, Q., Lu, Z., Zhang, J., and Leng, L. Spikingssms: Learning long sequences with sparse and parallel spiking state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20380–20388, 2025.
- Smith, J. T. H., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling, 2023. URL <https://openreview.net/pdf?id=Ai8Hw3AXqks>.
- Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021.
- Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.
- Yik, J., den Berghe, K. V., den Blanken, D., Bouhadjar, Y., Fabre, M., Hueber, P., Ke, W., Khoei, M. A., Kleyko, D., Pacik-Nelson, N., Pierro, A., Stratmann, P., Sun, P.-S. V., Tang, G., Wang, S., Zhou, B., Ahmed, S. H., Joseph, G. V., Leto, B., Micheli, A., Mishra, A. K., Lenz, G., Sun, T., Ahmed, Z., Akl, M., Anderson, B., Andreou, A. G., Bartolozzi, C., Basu, A., Bogdan, P., Bohte, S., Buckley, S., Cauwenberghs, G., Chicca, E., Corradi, F., de Croon, G., Danielescu, A., Daram, A., Davies, M., Demirag, Y., Eshraghian, J., Fischer, T., Forest, J., Fra, V., Furber, S., Furlong, P. M., Gilpin, W., Gilra, A., Gonzalez, H. A., Indiveri, G., Joshi, S., Karia, V., Khacef, L., Knight, J. C., Kriener, L., Kubendran, R., Kudithipudi, D., Liu, S.-C., Liu, Y.-H., Ma, H., Manohar, R., Margarit-Taulé, J. M., Mayr, C., Michmizos, K., Muir, D. R., Neftci, E., Nowotny, T., Ottati, F., Ozelikkale, A., Panda, P., Park, J., Payvand, M., Pehle, C., Petrovici, M. A., Posch, C., Renner, A., Sandamirskaya, Y., Schaefer, C. J., van Schaik, A., Schemmel, J., Schmidgall, S., Schuman, C., sun Seo, J., Sheik, S., Shrestha, S. B., Sifalakis, M., Sironi, A., Stewart, K., Stewart, M., Stewart, T. C., Timcheck, J., Tömen, N., Urgese, G., Verhelst, M., Vineyard, C. M., Vogginger, B., Yousefzadeh, A., Zohora, F. T., Frenkel, C., and Reddi, V. J. Neurobench: A framework for benchmarking neuromorphic computing algorithms and systems, 2025. URL <https://arxiv.org/abs/2304.04640>.
- Yin, B., Corradi, F., and Bohté, S. M. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3, 2021. ISSN 25225839. doi: 10.1038/s42256-021-00397-w.
- Zenke, F. and Ganguli, S. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.
- Zubić, N., Gehrig, M., and Scaramuzza, D. State space models for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

URL https://openaccess.thecvf.com/content/CVPR2024/papers/Zubic_State_Space_Models_for_Event_Cameras_CVPR_2024_paper.pdf.

Zucchet, N. and Orvieto, A. Recurrent neural networks: vanishing and exploding gradients are not the end of the story. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fbb07254ef01868967dc891ea3fa6c13-Paper-Conference.pdf.

A. Appendix

A.1. Link between Eigenvalues and Neuronal Dynamics Regimes

We give additional insights into the importance of the transition matrix eigenvalues of spiking neurons and their link to the dynamical regimes of the system.

The behavior of neurons with second order dynamics, including AdLIF, cAdLIF, C-SiLIF and SiLIF, can be classified as resonator, integrator, or mixed, depending on the ratio of the membrane and adaptation time constants ($\tau_m/\tau_w = \beta/\alpha$) and the coupling constant (a) (see Equation 2). These parameters determine whether the system undergoes a Hopf or saddle-node bifurcation, which is determinant for a periodic system’s stability. These parameters are thus typically clamped within specific ranges to control neuronal behavior (Bittar & Garner, 2022).

As seen in Figure 5, cAdLIF and SiLIF models exhibit both integrator and resonator regimes. The resonator regime corresponds to complex eigenvalues with a non-zero imaginary part, while the integrator regime is associated with purely positive real eigenvalues, as illustrated in Figure 2. Since α and β are not explicitly clamped for the SiLIF model, oscillatory dynamics are more thoroughly explored by the neuron, as there is no artificial constraint on the β/α ratio. This leads to more parameter exploration flexibility during training and richer internal dynamics. Nevertheless, due to the exponential reparametrization, the SiLIF model always remains in the stable regime.

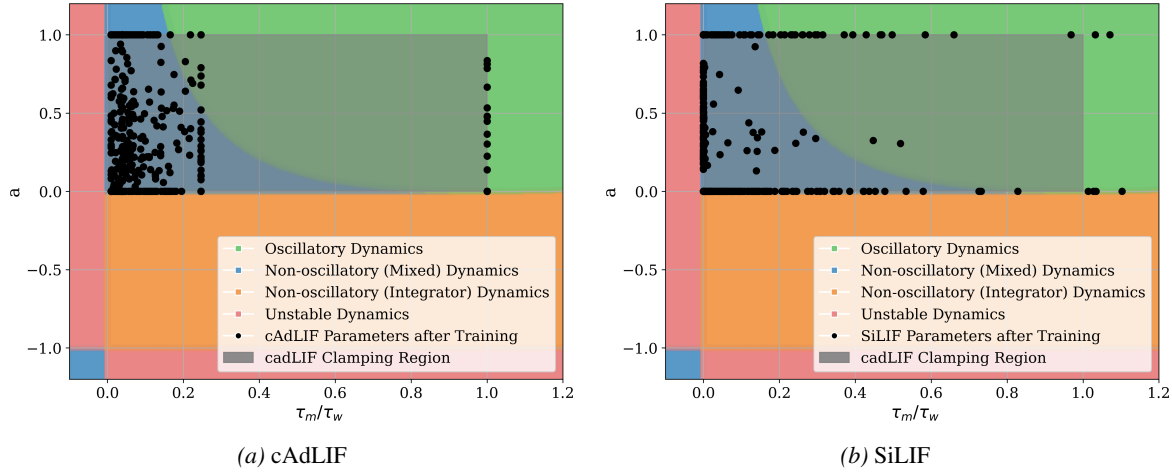


Figure 5. Neuronal dynamics of the (a) cAdLIF and (b) SiLIF models pre-trained on the SSC task. Each dot corresponds to the obtained training parameters for one neuron of the model. The position of each dot directly correlates to the corresponding neuron’s regime.

A.2. Experimental Details of the SiLIF Model with Synaptic Delays

A.2.1. SYNAPTIC DELAYS MODEL DESCRIPTION

We adopt the convolution-based formulation of synaptic delays proposed in (Hammouamri et al., 2024). Each individual synapse ($j \rightarrow i$) is assigned its own learnable delay kernel, allowing different transmission latencies for each pre-post connection. The synaptic current received by neuron i is written as

$$I_i(t) = \sum_{j=1}^N w_{ij} S_j(t - d_{ij}) = \sum_{j=1}^N k_{ij} * S_j, \quad (5)$$

where $S_j \in \{0, 1\}$ is the spike or non-spike from neuron j , w_{ij} the synaptic weight, d_{ij} the synaptic delay, k_{ij} the corresponding kernel, and $*$ denotes discrete-time convolution. Delays are learned using the one-dimensional DCLS parametrization of (Hassani et al., 2023). In this formulation, each kernel k_{ij} is modeled as a Gaussian with standard deviation σ , centered at $T_d - d_{ij} - 1$, where T_d is the maximum allowable delay. The kernel length is set to $T_d + 1$. Consequently, each synapse is parameterized by a weight w_{ij} , a delay d_{ij} , and a standard deviation σ . During training, all standard deviations σ are initialized to $T_d/2$ and progressively annealed to 0.5 over the first quarter of the training epochs. At test time, the learned delays are discretized by rounding to the nearest integer, following (Hammouamri et al., 2024).

A.2.2. COMPUTATIONAL COST OF SYNAPTIC DELAYS

As mentioned in the main text, we note that delay-based models incur additional memory and computational costs when implemented on neuromorphic hardware. Indeed, synaptic delays require buffering incoming spikes for each synapse and delivering them later when the delay has passed. This mostly requires additional dynamic memory, but also some compute to store and load delayed spikes into buffers. Here, we consider the worst case scenario for delay-based models, where the storing and loading of delayed spikes is as costly as the accumulation of weights. This leads to doubling the number of synaptic operations (SOPs). Therefore, in Figures 3 and 4, the reported SOPs for the SiLIF + Delays model are twice the number of effective SOPs for the synaptic weight accumulations.

A.2.3. EXPERIMENTAL SETUP FOR THE SiLIF + DELAYS

We perform the experiments of the SiLIF model with synaptic delays using the code base by Hammouamri et al. (2024) provided in <https://github.com/Thvntos/SNN-delays>. The network architecture consists of 2 layers, each composed of a DCLS module, batch normalization, and a SiLIF layer with 512 neurons. The membrane potential of the leaky integrator is used for the readout mechanism. Key architectural hyperparameters are reported in Table 3. The preprocessing of the SSC and GSC datasets closely follows the procedure described in Section 4.1. In particular, for GSC, we use 40 Mel-frequency channels, whereas the original work by Hammouamri et al. (2024) setup uses 140 Mel-frequency channels. After a hyperparameter search, detailed in Table 3, we find that the configurations yielding the best performance for synaptic delays closely match those reported in (Hammouamri et al., 2024). In particular, we employ the Adam optimizer with separate parameter groups, using base learning rates of 10^{-3} for the synaptic weights and 10^{-1} for the delay parameters. A one-cycle learning rate scheduler is applied to the synaptic weights, while a cosine annealing schedule is used for the delay learning rates.

Table 3. Learning hyperparameter search spaces for the SiLIF + Delays model. Bold numbers indicate the selected hyperparameters. For the one-cycle scheduler, the maximum learning rate is set to $5 \times$ the base learning rate, and the total number of steps in a cycle is set to the number of training epochs; for cosine annealing, the maximum number of iterations is also set to the number of training epochs.

Hyperparameter	GSC/SSC
T_d	{5, 11 , 15, 31}
lr_w (weights)	{ 10^{-2} , 10^{-3} , 10^{-4} }
lr_d (delays)	{ 10^{-1} , 10^{-2} , 10^{-3} }
scheduler (weights)	{ one-cycle , cosine, none}
scheduler (delays)	{one-cycle, cosine , none}
dropout	0.3
epochs	100
batch size	512

A.3. Hyperparameters Search

We detail here the hyperparameters that are searched for each of the three audio datasets for our non-delay C-SiLIF and SiLIF models. These grid searches are performed both for our C-SiLIF and SiLIF models and for all reproduced results on the AdLIF, cAdLIF, and RF models to ensure the most robust results in terms of training optimization. The only exception is for the AdLIF and cAdLIF results on GSC for Figure 3 where configurations from corresponding papers are kept considering that the dataset and resolution are the same.

The final obtained hyperparameters are presented in Table 5. All further fixed configurations are provided in our code base.

A.4. SHD SOTA Comparison

We report in Table 6 the extra SOTA comparison on the SHD task. On the SHD task, our C-SiLIF model outperforms previous state-of-the-art accuracy for networks below 50k parameters with 95.06% accuracy. At a larger scale, it nears the current state-of-the-art obtained for the Simplectic-Euler (SE)-adLIF model (Baronig et al., 2025), within its margin of error. We note that this task is less relevant than the SSC and GSC datasets due to its limited number of samples and the absence of separate validation and test sets, leading to an over-fitted hyperparameter search.

Table 4. Hyperparameter grid search space per dataset for SiLIF and C-SiLIF.

Hyperparameter	SHD	SSC	GSC
lr	$\{10^{-2}, 5 \cdot 10^{-3}, 10^{-3}\}$	$\{10^{-2}, 5 \cdot 10^{-3}, 10^{-3}\}$	$\{10^{-2}, 5 \cdot 10^{-3}, 10^{-3}\}$
dropout	$\{0.1, 0.25, 0.5\}$	$\{0.1, 0.25, 0.5\}$	$\{0.1, 0.25, 0.5\}$
schedule_patience	$\{5, 10^*\}$	$\{5, 10\}$	5
schedule_factor	$\{0.7, 0.9^*\}$	0.7	0.7
C-SiLIF $\Delta t_{\max}$	0.5	5	5
batch size	128	256	32
epochs	100	100	100

* Fixed for ablation study (Table 2).

Table 5. Experimental setups for our models on different datasets.

Dataset	Model	Resolution	#Hidden Size	lr	Dropout	Scheduler (Patience/Factor)
SHD	C-SiLIF	4 ms	128	10^{-2}	0.5	5/0.9
		4 ms	512	$5 \cdot 10^{-3}$	0.1	10/0.9
SSC	C-SiLIF	10 ms/4 ms	512	$5 \cdot 10^{-3}/10^{-2}$	0.5	10/0.7
	SiLIF	10 ms/4 ms	512	10^{-2}	0.5	10/0.7
GSC	C-SiLIF	10 ms	512	10^{-3}	0.25	5/0.7
	SiLIF	10 ms	512/1024	10^{-3}	0.5	5/0.7

Table 6. Classification accuracy on SHD dataset. Models are ranked by Top-1 accuracy (highest at the bottom) within non-delay and delay-based categories. Our models are shown in bold.

Method	Resolution	# Params	Rec.	Del.	Top1 Accuracy
S5-RF (Huber et al., 2025)	4 ms	0.2M			91.86%
cAdLIF (Deckers et al., 2024)	10 ms	38.7k			94.19%
SE-adLIF, 1 layer (Baronig et al., 2025)	4 ms	37.5k			$94.59 \pm 0.27\%$
RadLIF (Bittar & Garner, 2022)	14 ms	3.9M	✓		94.62%
C-SiLIF, 128 hidden units (ours)	4 ms	38.7k			$95.06 \pm 0.37\%$
C-SiLIF, 512 hidden units (ours)	4 ms	0.35M			$95.55 \pm 0.28\%$
SE-adLIF, 2 layer (Baronig et al., 2025)	4 ms	0.45M			$95.81 \pm 0.56\%$
d-cAdLIF (Deckers et al., 2024)	10 ms	76k		✓	$94.85 \pm 0.64\%$
DCLS-Delays (Hammouamri et al., 2024)	10 ms	0.2M		✓	$95.07 \pm 0.24\%$

A.5. Sparsity Analysis

In addition to the synaptic operations (SOPs) shown in Figure 3, we report the specific sparsity values for different SNN models in Table 7 for reference. The sparsity corresponds to the average amount of non-spiking timesteps per neuron produced with respect to the amount of total timesteps per sample. These high levels of sparsity on all models lead to a significant reduction of SOPs and contribute to making SNNs highly efficient for neuromorphic implementations.

A.6. Event-SSM Synaptic Operations Computation

We detail here the number of synaptic operations (SOPs) required per sample for the Event-SSM model (Schöne et al., 2024). This model achieves state-of-the-art performance on the SSC dataset but relies on projecting each event to a real-value vector and processing them without any sparsity, thus requiring a significant number of SOPs. We proceed through the SOPs computation for the smaller Event-SSM on SSC with 64 hidden states, as reported in Figure 3.

First, the Event-SSM relies entirely on event-based processing and lacks a sparsity mechanism, meaning each event triggers

Table 7. Test sparsity at best validation epoch for main models on SHD and GSC datasets, all with 2 hidden layers of 512 units. Standard deviation is reported with $\pm$.

Dataset	AdLIF	cAdLIF	C-SiLIF	SiLIF
SHD, 4ms resolution	88.57 $\pm$ 0.56%	97.80 $\pm$ 0.13%	80.25 $\pm$ 0.33%	98.66 $\pm$ 0.09%
SSC, 4ms resolution	N/A	93.65 $\pm$ 0.11%	90.37 $\pm$ 0.28%	93.35 $\pm$ 0.20%
GSC, 10ms resolution	93.39 $\pm$ 0.65%	93.03 $\pm$ 0.02%	85.21 $\pm$ 0.47%	91.71 $\pm$ 0.20%

the same number of fixed operations throughout the system. The first operation each event goes through is an embedding layer, which projects it to a real-value vector of size 64. Then this vector goes through a first event-based SSM at its full resolution. The embedding is projected through the $\bar{\mathbf{B}}$ matrix, which is square here, meaning a cost of 64×64 SOPs/event. The state update is made through the diagonal matrix $\bar{\mathbf{A}}$, thus negligible in SOP, but the state is then projected further through the square matrix $\bar{\mathbf{C}}$ for again 64×64 SOPs/event. Considering the SSC dataset presents an average number of 8000 events per sample, as reported by Schöne et al. (2024), the average SOP cost per sample for this first SSM block is already of $SOP_{\text{first block}} = 8000 \times (64 \times 64 + 64 \times 64) = 65.5M$.

The model then uses a pooling operation that divides the number of events passing through the network by 8, resulting in 1000 events per sample in the second block. This block consists of two dense projections of 64×64 SOP/event and 3 more SSMs, for a total of $6 \times 64 \times 64$ SOP/event before the next event pooling. Altogether $SOP_{\text{second block}} = 1000 \times (2 \times 64 \times 64 + 6 \times 64 \times 64) = 32.8M$. The final pooling reduces the number of events to an average of 125 per sample in the final block, making it negligible in terms of SOP.

Altogether, the 64 state Event-SSM model requires $SOP_{\text{total}} \leq SOP_{\text{first block}} + SOP_{\text{second block}}$ which corresponds to 98.3M SOPs/sample. One can simply adapt this computation to the 128-state SHD version of the Event-SSM model, leading to an even higher 288.8M SOPs/sample. These numbers are around 10 times higher than the measured values for our SNN models (Figure 3). Additionally, we note that the Event-SSM relies on full precision real-value number for all its activations, meaning that each synaptic operation requires a full precision multiply-and-accumulate (MAC) operation, whereas our binary spiking models can handle simple accumulate (AC) operations as synaptic values are either summed when there’s a spike or else omitted. Overall, it appears that the Event-SSM is significantly more computationally intensive, especially on event-based neuromorphic hardware.

A.7. SSM Models Optimization

A.7.1. S4D AND SPIKINGSSM FOR KEYWORD CLASSIFICATION

We detail here the method to obtain results on the S4D (Gupta et al., 2022) and SpikingSSM (Shen et al., 2025) presented in Figure 3 in the main text. As mentioned above, (Shen et al., 2025) already report results for these models. But these are obtained by processing the raw-audio data at a resolution of 0.06 ms (vs 10 ms to 4 ms for our models). This leads to excessively high SOPs due to the repetition of operations at a very high frequency, leading to 3.1×10^9 SOPs for SpikingSSM and 5.7×10^9 SOPs for S4D, which is at least 1000 times more than our spiking models. Therefore, we do not report the results presented by Shen et al. (2025) and rather retrain and optimize these models in our standardized setup with a MFCC preprocessing at a resolution of 10 ms for GSC.

A.7.2. MAMBA FOR KEYWORD CLASSIFICATION

To these SSM models, we also add Mamba (Gu & Dao, 2024), a selective gated extension of S4D, which became a new standard for efficient large language models. Mamba modifies the SSM parameters A , B , and C to be data-dependent such that writing or forgetting data for the state can be selective based on the input. This modification has opened an avenue for SSM-like models to tackle natural language processing tasks at scale, but the suitability of Mamba for shorter sequence edge tasks is still to be proven. Ding et al. (2025) recently proposed a bidirectional implementation of Mamba for keyword spotting, reaching high accuracies on the GSC dataset.

However, the use of bidirectional models prevents streamlined data processing, timestep by timestep, leading to a significant increase in computational costs. Indeed, the whole sequence needs to be buffered before it is sent to the bidirectional Mamba model, and the computation on the sequence needs to be repeated for each prediction, instead of only processing the latest timestep for standard forward models. This leads to 100 times more compute for GSC at 10 ms resolution, doubled further

due to the two paths of bidirectional models. The accuracy reported by Ding et al. (2025) is thus not comparable to our SNN models, and we retrain and optimize Mamba on GSC in a forward mode here.

A.7.3. EXPERIMENTAL SETUP

To obtain the fairest results possible, as presented in Figure 3 and 4, we explore the S4D, SpikingSSM, and Mamba models at different scales. We first re-use the model sizing proposed in (Shen et al., 2025) with 6 hidden layers of model size 128 and state size of 32 for the SSM. This leads to SOPs between 2.0×10^7 and 4.0×10^7 , which is around 10 times more than our models. Therefore, we also scale the models down to a medium size with 4 layers and 32 state size, and an extra small size for the S4D and Mamba with 3 layers of model size 64 and state size 32. For each of these scales, we conduct a grid search on the learning rate, dropout level, the Δt_{max} initialization, and the normalization type between layernorm and batchnorm. We also maintain the special S4D optimizer setup with a scaled-down learning rate for the SSM parameters A , B , and C (Equation 1). The swept hyperparameters and sizes per model are reported in Table 8, and the best obtained configurations for each model and scale are reported in Table 9.

Table 8. Hyperparameter grid search space for the SSM models on the GSC dataset.

Hyperparameter	S4D (Gupta et al., 2022)	SpikingSSM (Shen et al., 2025)	Mamba (Gu & Dao, 2024)
lr	$\{5 \cdot 10^{-2}, 5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}\}$	$\{10^{-2}, 5 \cdot 10^{-3}, 10^{-3}\}$	$\{10^{-2}, 5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}\}$
dropout	$\{0.1, 0.2\}$	$\{0.1, 0.2\}$	$\{0.1, 0.2\}$
normalization	batch, layer	batch, layer	batch, layer
Δt_{max}	$\{0.1, 5.0\}$	$\{0.1, 5.0\}$	$\{0.1, 5.0\}$
layers	$\{3, 4, 6\}$	$\{4, 6\}$	$\{3, 4, 6\}$
hidden units	$\{64, 128, 128\}$	128	$\{64, 128, 128\}$
state size	$\{32, 32, 64\}$	$\{32, 64\}$	$\{32, 32, 64\}$
batch size	32	32	32
epochs	100	100	100

Table 9. Obtained best hyperparameter configurations and scale for the SSM models on the GSC dataset.

Model	Scale (Layers / Hidden / State size)	lr	Dropout	Normalization	Δt_{max}
S4D (Gupta et al., 2022)	6 / 128 / 64	$5 \cdot 10^{-3}$	0.2	BatchNorm	0.1
	4 / 128 / 32	10^{-2}	0.2	BatchNorm	5.0
	3 / 64 / 32	10^{-2}	0.1	BatchNorm	5.0
SpikingSSM (Shen et al., 2025)	6 / 128 / 64	$5 \cdot 10^{-3}$	0.1	BatchNorm	5.0
	4 / 128 / 32	$5 \cdot 10^{-3}$	0.1	BatchNorm	5.0
Mamba (Gu & Dao, 2024)	6 / 128 / 64	10^{-3}	0.2	BatchNorm	0.1
	4 / 128 / 32	$5 \cdot 10^{-3}$	0.2	LayerNorm	0.1
	3 / 64 / 32	$5 \cdot 10^{-3}$	0.2	BatchNorm	5.0

A.8. Incremental C-SiLIF Features Analysis

We conduct an additional study of our C-SiLIF model, evaluating its performance as we incrementally add SSM-inspired features, from the Resonate and Fire (RF) neuron all the way to the C-SiLIF. We use networks of 2 hidden layers of 512 units and perform a grid search for each intermediate version. The measured average and standard deviation accuracy on the SHD dataset are reported in Figure 6.

As described by Izhikevich (2001), the RF neuron follows these neuronal dynamics:

$$\begin{aligned} u_t &= u_{t-1} + \Delta t((\alpha^{real} + i\alpha^{img})u_{t-1} + I_t) - \theta s_{t-1} \\ s_t &= (Re(u_t) \geq \theta), \end{aligned} \quad (6)$$

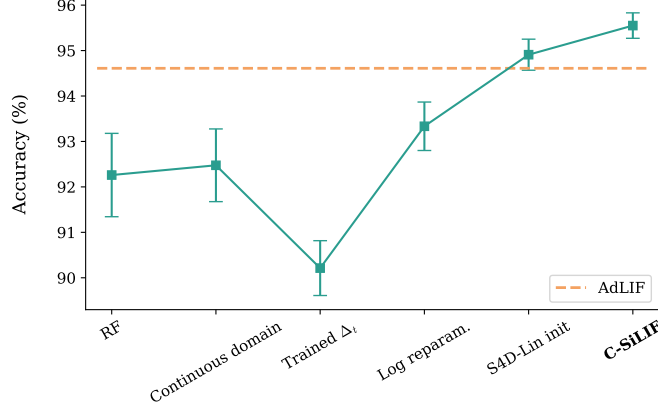


Figure 6. Impact of incremental SSM-imported features on performance on the SHD dataset from the Resonate and Fire (RF) to our proposed C-SiLIF model. Each model’s performance is obtained after a grid hyperparameter sweep for 2 hidden layers of 512 neurons. Error bars correspond to the standard deviation on 5 seed runs.

where α^{real} and α^{img} are the trainable parameters, obtained directly from the discrete form. As for the gap between the AdLIF and our SiLIF model, a first difference with the C-SiLIF is the parametrization of the model, as the C-SiLIF model focus on training continuous state transition parameters. The first incremental feature on top of the base RF model, called *Continual domain* is thus to train λ^{real} and λ^{img} which are then converted back to the discrete transition variables $\alpha = \exp((- \lambda^{real} + i \lambda^{img}) \Delta_t)$ such that $\alpha^{real} = \text{Re}(\alpha)$ and $\alpha^{img} = \text{Im}(\alpha)$. We note that this step slightly improves the model’s performance.

The next step consists of making the timestep Δ_t a trainable parameter. Despite this feature having been identified as a central element of the success of our models (see ablation results in Table 2), this incremental step hinders the model’s performance. But as soon as the logarithmic reparametrization is included again, where λ^{real} and Δ_t are trained in the logarithmic domain, the model outperforms the previous *Continual domain* configuration. We thus hypothesize that these two features work in strong synergy and should be considered as one common method.

Further on, the model’s parameter initialization is switched from the standard uniform distribution and range defined by Higuchi et al. (2024) in their modern study of the RF model, to the *S4D-Lin initialization* from the S4 model (Gu et al., 2022a). This leads to a significant jump in accuracy, reaching a level above our own optimized reproduction of the AdLIF model. This reinforces the idea that the specific S4 initialization has been optimized effectively for this specific model parametrization, and thus serves our C-SiLIF model.

Finally, our proposed half reset and input gating b (see Algorithm 2) complete the C-SiLIF model and improve the final performance further. This all demonstrates the positive impact of the S4-inspired features along with their strong synergy for SNN models.