DOI: 10.34749/3061-1466.2026.52

Recurrent versus parallelizable spiking neural networks: A comparative study

Alexander Mayr, Simon Hitzginger and Robert Legenstein
Institute of Machine Learning and Neural Computation
Graz University of Technology, 8010 Graz, Austria
{alexander.mayr, simon.hitzginger, robert.legenstein}@tugraz.at

Abstract

Spiking neural networks (SNNs) have emerged as a biologically plausible computational paradigm with strong links to real-world brain dynamics. Recently, interest has grown in parallelizable State Space Model (SSM)–inspired architectures, which offer improved scalability compared to recurrent networks. While effective at scale, these models represent a step away from biological realism. In particular, the impact of removing recurrent connections and membrane nonlinearities on the temporal processing capabilities of SNNs remains largely unexplored. In this work, we investigate the impact of these changes to the network dynamics on the temporal processing capabilities of SNNs with a focus on recurrent connectivity. To this end, a suite of sequential tasks was used to systematically compare parallelizable SSM-style networks with recurrent SNNs. The results demonstrate that while parallelizable models perform well on tasks with simple or weak temporal dependencies, they struggle to maintain persistent internal state when complex, state-dependent computation is required. In contrast, recurrent architectures exhibit superior memory retention and robustness under these conditions. These findings suggest fundamental limitations of parallelizable SSM-style approaches for sequence tasks that rely on long-term internal memory, highlighting the continued relevance of recurrence in spiking neural computation as suggested by biology.

1 Introduction

Spiking neural networks (SNNs) provide a biologically motivated framework for neural information processing in which information is transmitted via discrete spike events resembling biological action potentials [Maass, 1997, Gerstner and Kistler, 2002]. Their event-driven dynamics yield sparse activity patterns and enable temporal coding, offering the potential for improved energy efficiency and reduced memory requirements, particularly when deployed on neuromorphic hardware. Recurrence is a fundamental characteristic of biological neural circuits [Vidal-Saez et al., 2024, Larsen and Druckmann, 2022, Douglas and Martin, 2007]. In contrast, many modern machine learning architectures favour non-recurrent designs with simplified linear dynamics due to their favourable parallelisation properties, stable optimization behaviour, and scalability [Gu et al., 2020]. From a neurobiological perspective, however, the ubiquity of recurrent connectivity suggests functional significance rather than architectural redundancy, particularly for temporal information processing.

By operating intrinsically in the temporal domain, SNNs are naturally suited for sequential tasks such as speech recognition and time-series modeling. Incorporating recurrent connectivity further enhances their capacity to retain and update information over time, and recurrent SNN architectures have demonstrated strong performance on sequential learning problems [Bellec et al., 2018, Baronig et al., 2025]. Training recurrent SNNs typically relies on Backpropagation Through Time (BPTT) [Eshraghian et al., 2023], which enables gradient-based optimization but introduces substantial computational challenges. The memory and computational cost of BPTT scale linearly with both

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

This work is licensed under CC BY 4.0, excluding materials attributed to other sources or explicitly excluded.

sequence length and network depth, leading to high memory overhead and gradient instability. Moreover, the sequential nature of temporal backpropagation limits parallelization during training, rendering large-scale or long-sequence training computationally expensive. Recent work has sought to address these limitations by leveraging structured state space models (SSMs). By formulating a Resonate-and-Fire neuron within the HiPPO framework, the S5-RF model was introduced [Huber et al., 2024]. However, these parallelizable formulations deviate from biological neuron models by omitting spike-triggered voltage resets and recurrent network connectivity.

Alternative strategies achieve parallelization in recurrent SNNs by selectively neglecting specific gradient pathways in the computational graph, enabling scalable training of recurrent SNNs [Baronig et al., 2025, Fang et al., 2023]. While such approaches improve computational efficiency, their impact on sequential modeling performance and memory capacity remains insufficiently characterized [Merrill et al., 2024]. In this work, we systematically evaluate parallelizable SNN architectures based on S5-RF, leaky integrate-and-fire (LIF) and adaptive LIF neurons [Baronig et al., 2025, Higuchi et al., 2024] on established sequential benchmarks and compare them to non-parallelizable recurrent models incorporating voltage resets and recurrent connectivity. For tasks with limited temporal structure, we observe that the performance of parallelizable variants is slightly inferior to their recurrent counterparts. To specifically assess the capacity to preserve and update internal state representations, we further introduce a set of tasks which require accurate tracking of latent state transitions [Merrill et al., 2024]. We show that recurrent SNN architectures solve all task variants and generalize robustly beyond training conditions. In contrast, parallelizable non-recurrent models succeed only on simpler variants and fail to generalize beyond training regimes.

These findings highlight a trade-off between computational scalability and memory capacity in spiking neural network design. While parallelization substantially improves training efficiency, recurrent connectivity and biologically inspired dynamical mechanisms remain critical for tasks requiring structured temporal reasoning and robust state tracking.

2 Methods

We consider two architecture types: *non-recurrent spiking networks* which are fully parallelisable and *recurrent spiking neural networks* (RSNNs). Both use two hidden layers with task-dependent dimensions d_{in}, d_h, and d_{out}. Architectures share identical depth and differ only in connectivity. Due to the lack of recurrent connections the non-recurrent networks have less connections per layer and therefore a smaller number of weights. To ensure that all compared networks have a similar number of learnable parameters the network size for the non-recurrent networks is increased to double that of the recurrent networks.

All hidden-layer weights were initialised using orthogonal initialisation. Neuron-specific parameters were sampled from task-dependent ranges. To avoid silent neurons, inputs were rescaled as $I_{in} \leftarrow I_{in}\left(1+\frac{4}{d_{in}}\right)$. The output layer consisted of leaky integrator (LI) neurons (see Baronig et al. [2025]) with LeCun-uniform initialisation. In the ECG task LIF and SE-adLIF networks used membrane time constant $\tau = 3$ but otherwise unless stated all configurations used $\tau = 15$.

Four distinct neuron models were evaluated in this study. First, the standard LIF model serves as a well-established reference point. Second, the SE-adLIF and BRF neurons were selected as representative second-order neuron models that reflect the current state of the art in biologically inspired spiking dynamics. Finally, the S5-RF model is a parallelizable SSM-based SNN that replaces recurrent connectivity with structured state-space dynamics. Its hidden state is discretized using the Dirac scheme, which preserves high-frequency components and ensures numerical stability during parallel evaluation [Huber et al., 2024].

The Leaky Integrate-and-Fire (LIF) neuron maintains a membrane potential $u[t]$ with exponential decay and spike-triggered reset,

$$u[t] = \alpha u[t-1] + I[t] - \vartheta z[t-1], \tag{1}$$

$$z[t] = \Theta(u[t] - \vartheta), \tag{2}$$

where $I[t]$ is the input current at time step t, ϑ is the spiking threshold, $z[t] \in \{0, 1\}$ is the spike output at time step t, Θ is the Heaviside step function, and $\alpha \in (0, 1)$ is the decay factor. In terms of a membrane time constant τ_u, α is given by $\alpha = \exp(-dt/\tau_u)$ where dt denotes the discretization time step.

The Symplectic-Euler adaptive LIF (SE-adLIF) extends LIF with an adaptation variable $w[t]$:

$$u[t] = \alpha u[t-1](1 - z[t-1]) + (1-\alpha)(I[t] - w[t-1]), \tag{3}$$
$$w[t] = \beta w[t-1] + (1-\beta)(au[t] + bz[t-1])c_{adapt}, \tag{4}$$

where α and β are decay coefficients for u and w respectively. The coupling between the membrane potential $u[t]$ with the adaptation current $w[t]$ is scaled by the first adaptation coefficient a. The second coefficient b governs the feedback of the output spike into $w[t]$ [Baronig et al., 2025]. c_{adapt} is an additional custom adaptation coefficient for fine tuning purposes. It scales the effect of the previous output spikes, the current membrane potential and the adaptation current. The Balanced Resonate-and-Fire (BRF) neuron maintains complex-valued oscillatory dynamics [Higuchi et al., 2024]. We used the real-valued formulation given in Baronig et al. [2026]. The dynamics are mainly determined by the angular frequency parameter ω and the damping parameter b_{offset}.

To isolate the contribution of nonlinear state interactions, parallelisable variants were constructed by removing spike-dependent reset and recurrent dependencies. The resulting dynamics form linear time-invariant (LTI) recurrences while retaining the spiking nonlinearity for readout. For example, in the SE-adLIF model, spike-dependent interactions in both membrane and adaptation dynamics were removed. The resulting updates are

$$u[t] = \alpha u[t-1] + (1-\alpha)\big(I[t] - w[t-1]\big), \tag{5}$$
$$w[t] = \beta w[t-1] + (1-\beta)\, au[t], \tag{6}$$
$$z[t] = \Theta(u[t] - \vartheta), \tag{7}$$

The dynamics reduce to a linear coupled system driven by input current.

The SNNs were trained using back propagation through time (BPTT) and the ADAM optimizer [Kingma and Ba, 2014] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Due to the non-differentiability of the output spikes surrogate gradients were used. The BRF models used a double Gaussian surrogate while LIF and SE-adLIF used a Heaviside surrogate implemented with Slayer [Shrestha and Orchard, 2018]. Neuron parameters were optimized alongside with network weights. A cosine annealing learning rate schedule was used [Eshraghian et al., 2022]. This allows for bigger steps in the beginning whilst retaining the ability to fine-tune in later epochs. All tasks were trained for 50 epochs. The learning rates η were initialised for the first epoch at 0.05 for the BRF, 0.02 for the LIF, and 0.01 for the SE-adLIF models. For the loss, the *per-timestep cross entropy* function was used for the ECG, SHD, and state tracking task, while the *sum of softmax* was used for SMNIST. To determine test accuracies, class labels were predicted using *per timestep* prediction in the ECG and state tracking tasks while *mean over sequence* was used for SHD and *sum of softmax* prediction was used for SMNIST.

Unless stated otherwise, neuron model parameters were initialised by sampling independently from predefined ranges. For the LIF neuron, the initial membrane time constants were sampled from a uniform distribution, $\tau_u \sim U(0.5, 25)$, while the firing threshold was fixed to $\vartheta = 1.0$. For the SE-adLIF neuron, the initial membrane time constants were sampled from $\tau_u \sim \mathcal{U}(5, 25)$ and the adaptation time constants from $\tau_w \sim \mathcal{U}(60, 300)$. The subthreshold adaptation strength and spike-triggered adaptation strength were initialised as $a \sim \mathcal{U}(0.0, 1.0)$ and $b \sim \mathcal{U}(0.0, 1.0)$, respectively. The firing threshold was fixed to $\vartheta = 1.0$, and the adaptation scaling coefficient was set to $c_{\text{adapt}} = 120$. For the BRF neuron model, the intrinsic oscillatory dynamics are determined by the angular frequency parameter ω and the damping parameter b_{offset}. The natural frequency was initialised from $\omega \sim \mathcal{U}(3.0, 5.0)$, while the damping parameter was sampled from $b_{\text{offset}} \sim \mathcal{U}(0.1, 10.0)$. Other neuron parameters were fixed and chosen as in Baronig et al. [2026]. The S5-RF [Huber et al., 2024] was initialised with a structured state-space architecture composed of multiple stacked S5 blocks. The hidden state dimension was set to 256, and the model consists of 32 S5 blocks organised across 2 layers. The state-space discretisation follows the Dirac scheme. During training, the learning rate was scheduled with a cosine decay starting with learning rate $\eta = 0.02$. For a more detailed description of this model please refer to the original paper [Huber et al., 2024].

We introduce a synthetic benchmark for sequential state inference based on a discrete-time *Moore machine* [Gill, 1965]. The system maintains a hidden state $s_t \in \{0, \dots, n-1\}$ that evolves deterministically according to the previous state and the current action a_t. At each timestep, the SNN observes only the action and must output the hidden state. Performance was evaluated exclusively on the final state of the sequence. Action sequences were randomly sampled for training and testing.

Table 1: **Results on the ECG task.** Comparison of recurrent and non-recurrent architectures constructed from SE-adLIF, BRF, and LIF neurons over 10 runs. Hidden-layer dimensionality was adjusted to achieve comparable parameter counts. Reported values correspond to test-set classification accuracy (mean ± standard deviation over 10 runs). A check in the column "Rec." indicates a recurrent network model while a cross indicates a parallelizeable non-recurrent model.

Model	Rec.	Layer Dim.	#Params	#Runs	Test Acc. [%]
SE-adLIF	✓	128	52,0k	10	85.27 ± 0.42
	✗	256	71,2k	10	84.29 ± 0.32
BRF	✓	128	51,2k	10	85.81 ± 0.48
	✗	256	69,6k	10	84.37 ± 0.74
LIF	✓	128	51,5k	10	78.8 ± 2.73
	✗	256	70,2k	10	82.26 ± 0.53

The initial state was fixed to $s_0 = \frac{n}{2}$ (assuming even n). State evolution is defined recursively as $s_t = f_{a_t}(s_{t-1})$, where f_{a_t} denotes the transition induced by action a_t. The available actions are *up*, *down*, *stay*, and *mirror*. We consider two variants of the task, one where the state is kept when the maximum/minimum state is exceeded ("no overflow") and one with modular arithmetic at the state boundaries ("overflow"). Their transition functions are defined as

$$f_{\text{up}}(s) = \begin{cases} (s+1) \bmod n, & \text{overflow} \\ \min(n-1, s+1), & \text{no overflow} \end{cases}$$

$$f_{\text{down}}(s) = \begin{cases} (s-1) \bmod n, & \text{overflow} \\ \max(0, s-1), & \text{no overflow} \end{cases}$$

$$f_{\text{stay}}(s) = s, \qquad f_{\text{mirror}}(s) = |s - (n-1)|.$$

The *mirror* action introduces explicit state-dependent nonlinearity, while overflow corresponds to modular arithmetic at the state boundaries. Task difficulty can be controlled via the sequence length and the inclusion of nonlinear transitions. Solving the task requires iterative state updates and therefore tests a model's ability to hold internal state.

3 Results

3.1 Testing recurrent and non-recurrent SNNs on sequential benchmark tasks

The ECG [Laguna et al., 1997] and SMNIST [Bellec et al., 2018] datasets are standard benchmarks for evaluating sequential learning models. Performance on these tasks is often used as a primary criterion for assessing the effectiveness of newly proposed architectures. The networks evaluated on the benchmark tasks consisted of two hidden layers, with the number of neurons varying by task. We considered standard recurrent SNNs consisting of LIF, SE-adLIF, or BRF neurons. For each of these networks, we also considered a parallelizeable variant without state reset and without recurrent connections (see *Methods*) denoted by the cross in the Recurrent table column. The S5-RF model was not evaluated on this task. The non-recurrent networks had double the amount of hidden neurons per layer as to offset the learnable parameter increase from the recurrent connection weights and achieve comparable parameter counts. All models received identical input representations.

ECG On the ECG dataset (Table 1), both recurrent and non-recurrent architectures achieved comparable performance, with accuracies in the 85% range. This observation is consistent with results reported for current state-of-the-art models on this task [Higuchi et al., 2024, Yin et al., 2021]. The LIF model performed the worst only reaching 78.8% in recurrent configuration. Interestingly it reached a higher accuracy of 82% as a non-recurrent network. This is unique in our results and could be further investigated. Otherwise, the performance of the recurrent SNNs performed slightly better than their parallelizable variants.

SMNIST All evaluated models performed well above chance level on the SMNIST task (Table 2), indicating that both recurrent and non-recurrent architectures are capable of learning the sequential

Table 2: **Results on the SMNIST task.** Test accuracy of recurrent and non-recurrent architectures using SE-adLIF, BRF, and LIF neuron models over ten runs (mean ± standard deviation). Hidden-layer dimensionality was adjusted to obtain comparable parameter counts across architectures. S5-RF accuracy are taken from Huber et al. [2024].

Model	Rec.	Layer Dim.	#Params	Test Acc. [%]
SE-adLIF	✓	256	202,5k	98.9 ± 0.06
	✗	512	273,9k	94.69 ± 0.31
BRF	✓	256	201,0k	98.48 ± 0.05
	✗	512	270,9k	97.92 ± 0.07
LIF	✓	256	201,5k	88.54 ± 0.87
	✗	512	271,9k	81.82 ± 0.35
S5-RF	✗	128	36.3k	98.89

digit classification problem. Nevertheless, clear performance differences emerged across architectural choices. Recurrent networks consistently achieved higher accuracy than their non-recurrent counterparts, with the recurrent SE-adLIF model reaching the best overall performance at 98.9%. Among the non-recurrent architectures, the BRF model performed strongest, achieving an accuracy of 97.92%, which is slightly lower than that of the recurrent configuration. In contrast, the non-recurrent SE-adLIF and LIF models exhibited more substantial performance degradations, particularly for LIF neurons, which showed a pronounced sensitivity to the absence of recurrence.

3.2 Testing recurrent and non-recurrent SNNs on state tracking tasks

As shown in the previous section, recurrent SNNs perform slightly better than their fully parallelisable counterparts on the ECG, and SMNIST benchmarks. Despite their widespread use, these benchmark tasks can however potentially be solved through pattern recognition over temporal input sequences. In contrast, we propose variants of a state-tracking task (see Section 2), which are in the same spirit as the well-known Shell Game, but with extended complexity. In these tasks, the network observes a sequence of actions, which manipulate a hidden state. The task for the network is to predict the final hidden state. This requires the model to maintain and update information about a hidden state, as no simple input patterns are available that could be memorized. Moreover, when the network is able to learn the effect of actions on this state, which enables generalisation across varying sequence lengths.

Such capabilities are readily observed in biological neural systems, raising the question of whether non-recurrent architectures can achieve comparable behaviour. Specifically, it was argued that these models do not represent and manipulate internal state, and their apparent success on conventional benchmarks arises from learning sufficiently rich input–output correlations that emulate memory without explicitly maintaining it [Merrill et al., 2024].

We tested the architectures on two variations of the state tracking task, each with 6 hidden states and input sequences of length 20. We classify the two variants of the state tracking tasks as *easy* (task configuration with *no mirror* and *no overflow*; see Section 2), when there are limited state dependencies, and *hard*, which incorporates the *mirror* and *overflow* mechanics.

We observed a clear difference in the training and validation accuracies between recurrent and non-recurrent architectures across task difficulty as illustrated in Table 3. Non-recurrent models were able to solve the *easy* task variants, i.e. tasks without *overflow* or *mirror* actions, achieving test accuracies comparable to those of recurrent models. The main exception was the non-recurrent LIF model, which reached only 60% test accuracy, indicating limitations even in low-complexity settings. As an additional point of comparison, we included the state-of-the-art parallelisable S5-RF network [Huber et al., 2024]. This model combines structured state-space (S5) dynamics with spiking nonlinearities, enabling efficient parallel evaluation while retaining sensitivity to temporal structure. The model has previously demonstrated strong performance on classical sequential benchmarks, and we therefore evaluated whether it could also solve the state-tracking task and whether its performance differed significantly from that of the proposed parallelisable network architectures. The model achieved good, but not top-performance on the *easy* task variant.

However, in the *hard* task variant, test accuracies of the non-recurrent models decreased to below 20%, only marginally above the chance level of 16.66%, while the recurrent models retained good

Table 3: **Accuracy comparison on different versions of the state tracking task.** The final hidden state of Moore machines with $n = 6$ hidden states and state-transition dynamics of varying difficulty had to predicted. Training and test sequences had both a length of 20. Test accuracies are shown for 10 runs (mean $\pm$ standard deviation).

Type	Model	Rec.	Layer Dim.	#Params	Test Acc. [%]
No Mirror No Overflow (easy)	SE-adLIF	✓	128	51,9k	99.83 ± 0.13
	BRF	✓	128	51,1k	97.17 ± 0.52
	LIF	✓	128	51,3k	99.07 ± 1.5
	SE-adLIF	✗	256	70,9k	91.66 ± 0.75
	BRF	✗	256	69,4k	94.78 ± 0.77
	LIF	✗	256	69,9k	60.13 ± 1.43
	S5-RF	✗	256	137,2k	96.55 ± 0.78
Mirror Overflow (hard)	SE-adLIF	✓	128	52,0k	99.98 ± 0.02
	BRF	✓	128	51,2k	89.66 ± 2.07
	LIF	✓	128	51,5k	96.09 ± 7.51
	SE-adLIF	✗	256	71,2k	18.11 ± 0.5
	BRF	✗	256	69,6k	19.96 ± 1.85
	LIF	✗	256	70,2k	18.22 ± 0.63
	S5-RF	✗	256	137,7k	21.72 ± 5.83

performance, with the SE-adLIF showing close to optimal text accuracy of 99.98%. The non-recurrent and parallelisable S5-RF model followed a similar trend to the other non-recurrent architectures. Its accuracy degraded substantially when *mirror* and *overflow* mechanics were introduced. This suggests that, despite its enhanced temporal dynamics, balanced resonance alone is insufficient to replace explicit recurrence when learning non-linear and state-dependent transition rules.

Overall, these results indicate that non-recurrent models tend to rely on memorisation of short action patterns rather than learning a generalisable state-transition function. In contrast, recurrent SNNs were able to learn also complex variants of the state-tracking task.

3.3 Testing the generalisation capabilities of recurrent and non-recurrent SNNs

Biological systems are highly adept at learning the effects of actions rather than merely internalising fixed input sequences. To assess whether the investigated SNNs truly learn action-induced state transitions, or instead rely on memorising finite input–output patterns, the test sequence length of the state-tracking task was increased from 20 to 100 input actions, while keeping the training data unchanged. If a model successfully learns the underlying effects of actions on the internal state, this increase in sequence length should not significantly affect its test accuracy. In principle, models that capture the true state-transition dynamics should be able to predict arbitrarily long sequences. Increasing the sequence length therefore primarily probes the generalisation capabilities of the models.

Our results on the easy and hard variant of the state tracking task are summarized in Table 4. We found that recurrent SE-adLIF models exhibited the strongest generalisation behaviour, with accuracy drops of only approximately 1-2% between the easy and hard task variants. The recurrent LIF model was similarly robust with a steeper drop. The BRF network was generalizing in the *easy* task variant, but failed in the *hard* variant. These results indicate that recurrent architectures are able to maintain an internal state and update it consistently in response to extended action sequences.

In contrast, non-recurrent models struggled to generalise to longer sequences. Their accuracy dropped across both task variants. In the *hard* variant, the accuracy of non-recurrent architectures consistently degraded to chance-level, suggesting that they lack the internal state memory required to reliably track the hidden state. The S5-RF model displays a similar sensitivity to increased sequence length, further supporting the conclusion that resonance-based temporal dynamics alone do not provide sufficient inductive bias for modelling long-horizon, nonlinear state transitions without explicit recurrence.

Table 4: **Generalization to increased test sequence length on Moore machine modelling accuracy.** Test accuracy for a sequence length of 100 input actions, with results for the original sequence length of 20 shown in parentheses.

Type	Model	Rec.	Layer Dim.	#Params	Test Seq.	Test Acc. [%]
No Mirror No Overflow (easy)	SE-adLIF	✓	128	51,9k	100 (20)	98.54 (99.94)
	BRF	✓	128	51,1k	100 (20)	94.11 (96.70)
	LIF	✓	128	51,3k	100 (20)	99.69 (99.88)
	SE-adLIF	✗	256	70,9k	100 (20)	76.01 (92.25)
	BRF	✗	256	69,4k	100 (20)	64.63 (96.73)
	LIF	✗	256	69,9k	100 (20)	54.41 (60.79)
	S5-RF	✗	256	137,2k	100 (20)	71.37 (96.82)
Mirror Overflow (hard)	SE-adLIF	✓	128	52,0k	100 (20)	99.24 (99.97)
	BRF	✓	128	51,2k	100 (20)	27.89 (91.36)
	LIF	✓	128	51,5k	100 (20)	93.16 (99.18)
	SE-adLIF	✗	256	71,2k	100 (20)	17.55 (19.50)
	BRF	✗	256	69,6k	100 (20)	17.76 (28.99)
	LIF	✗	256	70,2k	100 (20)	17.55 (19.13)
	S5-RF	✗	256	137,7k	100 (20)	17.03 (21.36)

4 Discussion

This work investigated the role of recurrent connections in neural networks and found a clear task-dependent distinction. On conventional sequence benchmarks (ECG, SMNIST), non-recurrent and fully parallelisable architectures achieved accuracies comparable to recurrent models, supporting the view that many standard benchmarks can be solved through pattern recognition rather than long-horizon stateful computation. In such settings, enhanced single-neuron dynamics including adaptation or resonance can approximate temporal processing without explicit recurrence. However, in Moore machine-inspired state-tracking tasks that require maintaining and updating a latent internal state according to action-conditioned rules, recurrent architectures demonstrated a consistent and substantial advantage. They remained robust under increased temporal complexity, longer sequences, and expanded state spaces, while non-recurrent models often degraded to chance-level performance. Although one recurrent variant showed a task-specific generalisation failure, other recurrent models successfully captured the required state-dependent transitions, reinforcing the conclusion that explicit recurrence provides a crucial inductive bias for learning stable, generalisable internal memory.

Mechanistically, the results support the view that recurrence enables networks to implement evolving dynamical systems rather than static input–output mappings. Recurrent models appeared to learn generalisable transition operators and maintained structured internal activity during difficult tasks, whereas non-recurrent models relied more on short-range heuristics that failed under longer horizons or complex state based actions. These findings carry implications for brain-inspired modelling: while parallelisable feed-forward SNNs offer efficiency advantages and remain competitive on pattern-based benchmarks, biologically plausible cognition as characterised by persistent activity, feedback, and latent-state inference, appears to fundamentally depend on recurrence. Thus, recurrence remains essential for modelling brain-like computation.

Acknowledgments and Disclosure of Funding

This research was funded in whole or in part by the Austrian Science Fund (FWF) [10.55776/COE12] (AM, SH, RL), and by NSF EFRI grant #2318152 (RL).

References

M. Baronig, R. Ferrand, S. Sabathiel, and R. Legenstein. Advancing spatio-temporal processing through adaptation in spiking neural networks. *Nature Communications*, 16(1), July 2025.

M. Baronig, Y. Bahariasl, O. Özdenizci, and R. Legenstein. A scalable hybrid training approach for recurrent spiking neural networks. *Neuromorphic Computing and Engineering*, 6(1):014017, 2026.

G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

R. J. Douglas and K. A. Martin. Recurrent neuronal circuits in the neocortex. *Current Biology*, 17 (13):R496–R500, July 2007.

J. K. Eshraghian, C. Lammie, M. R. Azghadi, and W. D. Lu. Navigating local minima in quantized spiking neural networks. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 352–355. IEEE, 2022.

J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

W. Fang, Z. Yu, Z. Zhou, D. Chen, Y. Chen, Z. Ma, T. Masquelier, and Y. Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53674–53687. Curran Associates, Inc., 2023.

W. Gerstner and W. M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Aug. 2002.

A. Gill. On the bound to the memory of a sequential machine. *IEEE Transactions on Electronic Computers*, EC-14(3):464–466, June 1965.

A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré. Hippo: Recurrent memory with optimal polynomial projections. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487. Curran Associates, Inc., 2020.

S. Higuchi, S. Kairat, S. Bohté, and S. Otte. Balanced resonate-and-fire neurons. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 18305–18323. PMLR, 21–27 Jul 2024.

T. E. Huber, J. Lecomte, B. Polovnikov, and A. von Arnim. Scaling up resonate-and-fire networks for fast deep learning. In *European Conference on Computer Vision*, pages 241–258. Springer, 2024.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

P. Laguna, R. Mark, A. Goldberg, and G. Moody. A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg. In *Computers in Cardiology 1997*, pages 673–676, 1997.

B. W. Larsen and S. Druckmann. Towards a more general understanding of the algorithmic utility of recurrent connections. *PLOS Computational Biology*, 18(6):1–33, 06 2022.

W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, Dec. 1997.

W. Merrill, J. Petty, and A. Sabharwal. The illusion of state in state-space models. *arXiv preprint arXiv:2404.08819*, 2024.

S. B. Shrestha and G. Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

M. S. Vidal-Saez, O. Vilarroya, and J. Garcia-Ojalvo. Biological computation through recurrence. *Biochemical and Biophysical Research Communications*, 728:150301, Oct. 2024.

B. Yin, F. Corradi, and S. M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, Oct. 2021.