A TIME-VARIANT SPIKING NEURON FROM A UNIFIED STATE-DYNAMICS PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) have increasingly incorporated rich neuronal dynamics to enhance temporal processing. However, richer dynamics do not necessarily enable neurons to adapt how information is processed over time. To examine this issue systematically, we introduce a unified state-dynamics framework that characterizes diverse spiking neurons through a common formulation and enables direct comparison of their dynamical designs. The framework reveals a common limitation of existing neurons: although they enhance temporal processing through richer internal states and state evolution, the dynamical rules determining how inputs are integrated, how states evolve, and how they are read out generally remain unchanged throughout temporal processing. Such time-invariant rules limit the ability of neurons to adjust information processing as information relevance changes over time. Inspired by time-varying regulation in biological neurons, we propose the Time-Variant Multi-Compartment Neuron (TV-MCN). TV-MCN builds on multiscale compartment dynamics and makes the rules for input integration, state evolution, and state readout time-varying according to the current input. Specifically, an input-dependent internal time regulates the rate of state evolution, while the current input dynamically modulates input integration and state readout. Extensive experiments show that TV-MCN achieves strong overall performance, with particularly large gains on tasks requiring selective retention and retrieval of information, while ablation studies further support the effectiveness of its time-varying mechanisms.

1 INTRODUCTION

Spiking neural networks (SNNs) are biologically inspired models that communicate through discrete spikes between neurons. Within each neuron, internal states integrate incoming signals and evolve over time, determining how past inputs influence future responses. The design of these state dynamics is therefore central to the temporal processing capability of SNNs. Recent studies have extended neuronal dynamics by introducing heterogeneous time constants to capture multiple retention scales (Zheng et al., 2024; Deng et al., 2025), interacting compartments for modeling structured internal dynamics (Zhang et al., 2024; Chen et al., 2026), and incorporating resonant modes with distinct frequency responses Zhang et al. (2025). These developments enrich the temporal dynamics of spiking neurons beyond the single decay mode of the leaky integrate-and-fire (LIF) neuron.

However, richer dynamics do not necessarily enable neurons to adapt how information is processed over time. In real-world settings, useful information is often interleaved with irrelevant distractors, and information that is important at one moment may become less relevant as new inputs arrive. Neurons therefore are required to determine what information to integrate or discard, what to maintain or forget, and which stored information should be used for the output (Chatham & Badre, 2015; Unger et al., 2016). Yet it remains unclear whether existing neuron designs enable such flexible information processing.

To examine this question systematically, we propose a unified state-dynamics framework that describes spiking-neuron computation through three biologically motivated stages: input integration and state evolution, state readout to form the somatic membrane potential, and spike generation followed by post-spike state recovery (Hodgkin & Huxley, 1952; Magee, 2000; Bean, 2007). Neuron dynamics developed from different biological motivations and mathematical formulations can be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

expressed as different parameterizations of this framework, enabling their dynamical structures and design choices to be directly compared and analyzed within a common formulation.

Using this unified framework, we identify a common limitation of existing designs. Although they enhance temporal processing by increasing the dimensionality or diversifying the temporal evolution of internal states, their dynamical rules determining how inputs are integrated into these states, how the states evolve, and how they are read out generally remain unchanged throughout temporal processing. Consequently, such time-invariant rules limit the ability of neurons to adjust how information is integrated, maintained, and used as the relevance of incoming information changes.

Inspired by the time-varying regulation observed in biological neurons (Schulz et al., 2006), we propose the Time-Variant Multi-Compartment Neuron (TV-MCN). TV-MCN builds on multiscale compartment dynamics that provide diverse intrinsic temporal scales for representing information. On this basis, it makes the dynamical rules for input integration, state evolution, and state readout time-varying according to the current input. Specifically, an input-dependent internal time regulates the rate of state evolution, while the current input dynamically modulates input integration and state readout. Together, these mechanisms enable TV-MCN to adjust how information is integrated, maintained, and used as information relevance changes over time.

We evaluate TV-MCN across a broad range of temporal classification and synthetic sequence benchmarks. The classification benchmarks include Google Speech Commands (GSC), Spiking Heidelberg Digits (SHD) and Spiking Speech Commands (SSC), and DVS128 Gesture (Warden, 2018; Cramer et al., 2022; Amir et al., 2017), while the synthetic tasks include Adding, Selective Copying, 20D STORE–RECALL, MAD ICR and MAD Noisy ICR, and Induction (Hochreiter & Schmidhuber, 1997; Salaj et al., 2021; Poli et al., 2024; Gu & Dao, 2024). TV-MCN achieves strong performance across the benchmarks and substantially outperforms existing spiking neuron models on tasks that require selective retention and retrieval of information. For example, on MAD Noisy ICR (Poli et al., 2024), TV-MCN reaches an accuracy of 99.99%, compared with 85.43% for HetSyn (Deng et al., 2025) and 40.15% for D-RF (Zhang et al., 2025). Further ablation studies confirm the contribution of its time-varying dynamics. Our contributions are summarized as follows:

- We introduce a unified state-dynamics framework that characterizes diverse spiking neurons through a common formulation, revealing that the governing dynamical rules are temporally unchanged in existing designs.
- We propose the Time-Variant Multi-Compartment Neuron (TV-MCN), which introduces explicit time variation into input integration, state evolution, and state readout, enabling adaptive information processing as inputs change over time.
- Experimental results demonstrate that TV-MCN achieves strong overall performance, with particularly large gains on tasks requiring selective retention and retrieval, validating the effectiveness of its time-varying dynamics.

2 A Unified Framework for Neuronal Dynamics

In biological neurons, membrane potentials, synaptic conductances, and ion-channel states continuously evolve over time, while synaptic inputs drive these internal states by changing local currents and conductances (Hodgkin & Huxley, 1952). The electrical signals generated by these states propagate and integrate within dendrites and jointly influence the somatic membrane potential (Magee, 2000). Once the firing condition is reached, the neuron generates a spike, followed by post-spike recovery that updates its internal states (Bean, 2007). The biological facts above can be summarized as follows: neuronal internal states are driven by current inputs and evolve over time; these states jointly influence the membrane potential, which in turn determines spike generation.

Based on this biological process, we abstract the spiking neuron as a state-dynamics model in which the internal states carry information and evolve under incoming inputs. As illustrated in Figure 1, neuronal computation is decomposed into three stages: (1) input integration and state evolution, (2) state readout, and (3) spike generation and state recovery. The corresponding dynamics can be

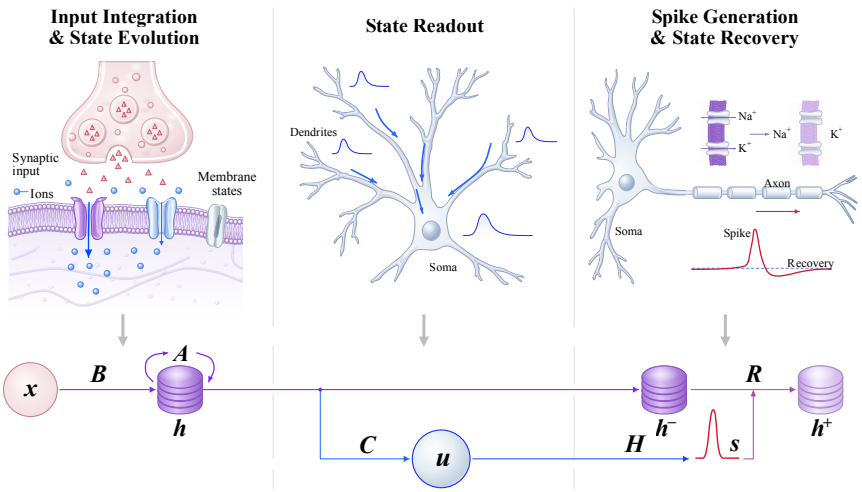


Figure 1: From biological neuronal computation to the unified state-dynamics framework. Left: Synaptic inputs drive internal states, corresponding to input integration through B and state evolution through A. Middle: Dendritic states are read out at the soma to form the membrane potential. Right: Spike generation is followed by ionic currents that drive post-spike state recovery.

written as:

$$\begin{aligned} \dot{h}(t) &= Ah(t) + Bx(t), \\ u(t) &= Ch(t), \\ s(t) &= H(u(t) - \vartheta(t)), \qquad h(t^+) = R\big(h(t^-), s(t)\big). \end{aligned} \tag{1}$$

where $h(t)$ denotes the internal states, $x(t)$ the input, and $u(t)$ the membrane potential. In the first stage, A governs how these states evolve, including decay and oscillation, while B determines how the current input is integrated into states. In the second stage, C specifies how the internal states are read out to form membrane potential. In the final stage, H denotes spike generation and R describes the post-spike state recovery. $h(t^-)$ and $h(t^+)$ denote the states immediately before and after the recovery, respectively.

Under this framework, spiking neurons with different biological motivations and mathematical forms can be viewed as different parameterizations of the internal state h, state evolution A, input integration B, and state readout C, as summarized in Table 1. This unified formulation allows their dynamical structures and design choices to be directly compared and analyzed. It reveals that additional compartments (Zhang et al., 2024; Chen et al., 2026) and synaptic-current states (Deng

Table 1: Parameterizations of spiking neurons under the unified state-dynamics framework.

Neuron	h	A	B	C
LIF (Burkitt, 2006)	u	$-\tau_m^{-1}$	$W\tau_m^{-1}$	1
ad-LIF (Baronig et al., 2025)	$[u, w]^\mathsf{T}$	$\begin{bmatrix} -\tau_u^{-1} & -\tau_u^{-1} \\ a\tau_w^{-1} & -\tau_w^{-1} \end{bmatrix}$	$\mathrm{diag}(\tau_u^{-1}, b\tau_w^{-1})$	e_1^T
TC-LIF (Zhang et al., 2024)	$[U^D, U^S]^\mathsf{T}$	$\begin{bmatrix} 1 & \beta_1 \\ \beta_2 & 1+\beta_1\beta_2 \end{bmatrix}$	$\begin{bmatrix} 1 & -\gamma \\ \beta_2 & -(\beta_2\gamma+V_{\mathrm{th}}) \end{bmatrix}$	e_2^T
DGN (Bai et al., 2026)	$[\mathbf{D}^\mathsf{T}, V]^\mathsf{T}$	$\begin{bmatrix} -\tau_s^{-1} I_N & 0 \\ \mathbf{w}^\mathsf{T} & -(g_l+\mathbf{c}^\mathsf{T}\mathbf{D}) \end{bmatrix}$	$\begin{bmatrix} \tau_s^{-1} I_N \\ 0 \end{bmatrix}$	e_{N+1}^T
DH-LIF (Zheng et al., 2024)	$[\mathbf{i}^\mathsf{T}, u]^\mathsf{T}$	$\begin{bmatrix} \Lambda_\alpha & 0 \\ (1-\beta)\boldsymbol{\alpha}^\mathsf{T} & \beta \end{bmatrix}$	$\begin{bmatrix} I_D - \Lambda_\alpha & 0 \\ (1-\beta)(\mathbf{1}-\boldsymbol{\alpha})^\mathsf{T} & -u_{\mathrm{th}} \end{bmatrix}$	e_{D+1}^T
HetSyn (Deng et al., 2025)	$[\mathbf{I}^\mathsf{T}, J]^\mathsf{T}$	$-\mathrm{diag}(\tau_{s,1}^{-1}, \ldots, \tau_{s,M}^{-1}, \tau_J^{-1})$	$\mathrm{diag}(w_1, \ldots, w_M, \vartheta)$	$[\mathbf{1}_M^\mathsf{T}, -1]$
D-RF (Zhang et al., 2025)	$[z_i]_{i=1}^n$	$\mathrm{diag}(-\tau_i^{-1} + j\omega_i)$	$\boldsymbol{\gamma}$	$\mathbf{c}^\mathsf{T}$
PMSN (Chen et al., 2026)	$[v_i]_{i=1}^n$	$-\mathrm{diag}(\tau_i^{-1}) + A_\beta$	$\boldsymbol{\gamma}$	e_n^T

Here, τ denotes time constants, ω oscillation frequencies, and A_β inter-compartment coupling. W, $\boldsymbol{\gamma}$, and related coefficients specify input integration; e_k^T selects the k-th state, and $\mathbf{c}^\mathsf{T}$ defines a weighted state readout. Other symbols denote model-specific internal states and parameters as defined in the cited works.

et al., 2025) expand the dimensionality of h, while heterogeneous time constants (Zheng et al., 2024; Deng et al., 2025), state coupling (Zhang et al., 2024; Chen et al., 2026), and resonant dynamics (Zhang et al., 2025) enrich the state evolution expressed by the structure of A. Differences in input integration and state readout are reflected in how B distributes inputs across these states and how C reads out either a single state or a weighted combination of states (Chen et al., 2026; Zhang et al., 2025). Further analyses of representative neuron models under this framework are provided in Appendix A.

The framework further highlights a key distinction between biological neurons and existing spiking neuron designs. Biological neuronal dynamics are continuously regulated during information processing, whereas existing designs generally rely on fixed dynamical rules. In biological neurons, these dynamical processes are regulated through multiple mechanisms. Membrane conductance changes affect the effective membrane time constant and state-evolution rate (Cavallari et al., 2014), NMDA receptors modulate input integration (Li & Gulledge, 2021), and dendritic or somatic inhibition regulates state contributions to the somatic response (Jadi et al., 2012). By contrast, most of the representative spiking neurons above use coefficients A, B, and C that remain unchanged, leaving input integration, state evolution, and state readout without time-variant regulation.

3 Time-Variant Multi-Compartment Neuron

Motivated by the dynamic regulation observed in biological neurons, we redesign neuronal dynamics within the unified framework and propose the Time-Variant Multi-Compartment Neuron (TV-MCN). The core idea is to move beyond fixed dynamical rules by allowing neuronal dynamics to vary over time. Accordingly, we introduce time-varying regulation into input integration, state evolution, and state readout. TV-MCN combines this regulation with multiple compartments exhibiting distinct decay rates and oscillation frequencies, providing a flexible basis for adapting how information is integrated, maintained, and used.

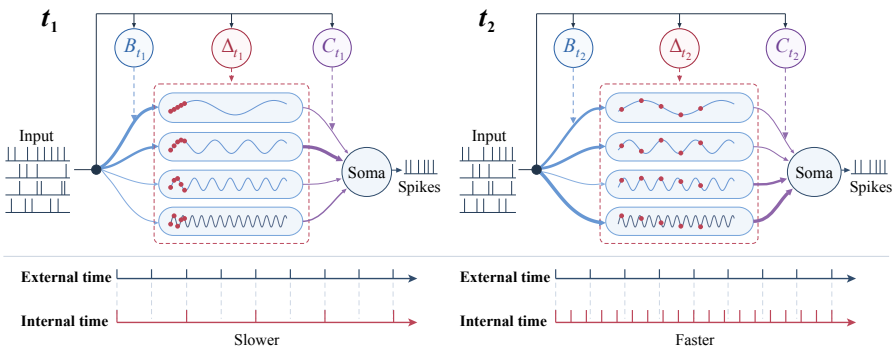


Figure 2: Input-dependent dynamics of TV-MCN at t_1 (Left) and t_2 (Right). Blue and purple arrow widths indicate input-integration and state-readout strengths regulated by B_t and C_t, respectively. Blue curves illustrate compartment dynamics, with red dots marking states at equal external-time intervals. The smaller Δ_{t_1} slows internal time and state evolution, whereas the larger Δ_{t_2} accelerates them.

In the conventional formulation, internal states evolve directly with external time t:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = Ah + Bx. \tag{2}$$

Here, neuronal dynamics evolve at a fixed rate as external time advances. However, different inputs may require different rates of state evolution. Irrelevant information can be attenuated more rapidly by accelerating state evolution, whereas useful information can be preserved for longer by slowing the rate down. To enable such adaptive temporal progression, we introduce an internal time τ and formulate the state dynamics with respect to it:

$$\frac{\mathrm{d}h}{\mathrm{d}\tau} = Ah + B_t x. \tag{3}$$

We then introduce a progression coefficient Δ_t that determines how fast internal time advances relative to external time:

$$\frac{\mathrm{d}\tau}{\mathrm{d}t} = \Delta_t, \quad \Delta_t > 0. \tag{4}$$

Accordingly, over an external-time interval $\mathrm{d}t$, internal time advances by

$$\mathrm{d}\tau = \Delta_t \, \mathrm{d}t. \tag{5}$$

A small Δ_t slows the progression of internal time and thus the evolution of internal states, allowing information to be preserved for longer. In contrast, a large Δ_t accelerates state evolution, allowing stored information to change or decay more rapidly. We let the current input regulate this progression coefficient:

$$q_t = \mathrm{SiLU}(W_q \mathbf{x}_t), \quad \Delta_t = \mathrm{softplus}(b_\Delta + W_\Delta q_t). \tag{6}$$

where $\mathbf{x}_t \in \mathbb{R}^M$ denotes the current input, W_q and W_Δ are learnable projection weights, and b_Δ is a learnable bias. The neuron can therefore dynamically adjust the progression of its internal dynamics according to the current input. By the chain rule, the state dynamics with respect to external time become

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{\mathrm{d}\tau}{\mathrm{d}t}\frac{\mathrm{d}h}{\mathrm{d}\tau} = \underbrace{\Delta_t A}_{A} h + \underbrace{\Delta_t B_t}_{B} x. \tag{7}$$

Thus, Δ_t directly regulates the rate at which the internal states evolve with external time.

We next specify the intrinsic dynamics of TV-MCN. The internal state $h(t) \in \mathbb{C}^N$ consists of N independently evolving compartments, parameterized by a set of learnable complex-valued modes:

$$A = \mathrm{diag}(-e^{a_1} + j\omega_1, \ldots, -e^{a_N} + j\omega_N), \tag{8}$$

where a_i and ω_i are learnable parameters. For the i-th compartment, the real part $-e^{a_i}$ determines its decay rate of membrane potential, while the imaginary part ω_i determines its oscillation frequency. Different compartments therefore exhibit distinct decay scales and frequency responses.

Beyond state evolution, B and C in the unified framework determine how the current input is integrated into the internal states and how these states are read out, respectively. In TV-MCN, both processes are dynamically regulated by the current input:

$$B_t = W_B \mathbf{x}_t, \quad C_t = W_C \mathbf{x}_t. \tag{9}$$

The input-dependent B_t allows the neuron to dynamically regulate which information is written into its internal states, while C_t determines which stored information contributes to the output.

With the internal-time modulation and input-dependent B_t and C_t, the full form of TV-MCN is

$$\underbrace{\begin{bmatrix} \dot{h}_1(t) \\ \dot{h}_2(t) \\ \vdots \\ \dot{h}_N(t) \end{bmatrix}}_{\dot{h}(t)} = \Delta_t \underbrace{\begin{bmatrix} -e^{a_1} + j\omega_1 & 0 & \cdots & 0 \\ 0 & -e^{a_2} + j\omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -e^{a_N} + j\omega_N \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h_1(t) \\ h_2(t) \\ \vdots \\ h_N(t) \end{bmatrix}}_{h(t)} + \Delta_t \underbrace{\begin{bmatrix} b_{1,t} \\ b_{2,t} \\ \vdots \\ b_{N,t} \end{bmatrix}}_{B_t} x(t).$$

$$u(t) = \underbrace{\begin{bmatrix} c_{1,t} \\ c_{2,t} \\ \vdots \\ c_{N,t} \end{bmatrix}^{\mathsf{T}}}_{C_t} \underbrace{\begin{bmatrix} h_1(t) \\ h_2(t) \\ \vdots \\ h_N(t) \end{bmatrix}}_{h(t)}, \tag{10}$$

$$s(t) = H(u(t) - \vartheta), \qquad h(t^+) = R\big(h(t^-), s(t)\big).$$

where Δ_t, B_t, and C_t regulate the rate of state evolution, input integration, and state readout, respectively. We then use the real part of the membrane potential for threshold-based spike generation and apply a soft reset to the somatic membrane potential after firing. For spike generation, we take the real part of the membrane potential for threshold comparison and apply a soft reset after firing. Figure 2 provides an overview of these input-dependent dynamics.

To implement TV-MCN in discrete time, we adopt zero-order hold (ZOH) discretization (Gu et al., 2022), assuming that the input and the dynamic parameters generated from it remain constant within each time step. Normalizing the external time-step length to 1 gives

$$\bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t = A^{-1}\left[\exp(\Delta_t A) - I\right] B_t. \tag{11}$$

Therefore, the discrete state update of TV-MCN can be written as

$$\begin{aligned} h_t &= \bar{A}_t h_{t-1} + \bar{B}_t x_t, \\ u_t &= C_t h_t, \\ s_t &= H(u_t - \vartheta), \qquad h_{t^+} = R(h_{t^-}, s_t). \end{aligned} \tag{12}$$

4 Experimental Results

We evaluate TV-MCN on two categories of benchmarks: temporal classification and synthetic sequence tasks. For temporal classification, we use Google Speech Commands (GSC) (Warden, 2018), Spiking Heidelberg Digits (SHD) and Spiking Speech Commands (SSC) (Cramer et al., 2022), and DVS128 Gesture (Amir et al., 2017), covering speech and event-based vision. For synthetic sequence tasks, we evaluate Adding (Hochreiter & Schmidhuber, 1997), Selective Copying (Gu & Dao, 2024), 20D STORE–RECALL (Salaj et al., 2021), MAD ICR and MAD Noisy ICR (Poli et al., 2024), and Induction (Gu & Dao, 2024). These tasks assess accumulation, selective retention under distraction, instruction-dependent storage and retrieval, associative recall with or without irrelevant inputs, and the use of previously established temporal associations. Adding is evaluated using mean squared error (MSE), while the remaining tasks use classification or recall accuracy. Detailed experimental settings are provided in Appendix B.

4.1 Comparison with Existing Spiking Neurons

We compare LIF (Burkitt, 2006), ad-LIF (Baronig et al., 2025), TC-LIF (Zhang et al., 2024), DH-LIF (Zheng et al., 2024), HetSyn (Deng et al., 2025), DGN (Bai et al., 2026), D-RF (Zhang et al., 2025), PMSN (Chen et al., 2026), and TV-MCN using the same fully connected network architecture.

Tables 2 and 3 compare TV-MCN with existing spiking neurons on temporal classification and synthetic sequence tasks, respectively. On temporal classification, TV-MCN achieves the best accuracy on GSC, SSC, and DVS128 Gesture, reaching 90.56%, 70.98%, and 93.75%, respectively, while achieving 88.30% on SHD, close to the best-performing HetSyn. The competitive performance of existing neurons suggests that conventional temporal classification can already be handled effectively by rich neuronal dynamics, while TV-MCN further improves performance on most benchmarks. A much larger advantage emerges on synthetic sequence tasks that require selective integration, retention, and retrieval of information. TV-MCN achieves the best performance on all six tasks, with particularly large gains on Selective Copying, MAD ICR, MAD Noisy ICR, and Induction. Although DH-LIF also performs strongly on STORE–RECALL, reaching 99.27% compared with 99.92% for TV-MCN, this strong performance does not extend to the other synthetic tasks. More generally, existing neurons tend to perform well on individual tasks but do not maintain consistently strong performance across different information-processing requirements. In contrast, TV-MCN performs consistently well across all six benchmarks, demonstrating stronger capability in handling different forms of temporal information processing.

We next vary sequence length and interference strength to examine whether the learned temporal processing behavior generalizes when information must be retained for different durations or withstand stronger interference. On Induction, we train each model at a fixed sequence length and evaluate them on sequences of different lengths. This setting tests whether the ability to retain and retrieve relevant historical information according to the current input generalizes beyond the training length. As shown in Figure 3 (Left), PMSN and D-RF perform accurately at the training length but deteriorate substantially on shorter or longer sequences. TV-MCN maintains 96.88–98.83% accuracy throughout the tested range, demonstrating substantially stronger length generalization. On MAD Noisy ICR, we vary the interference fraction ρ to evaluate robustness to increasing interference. Each model is trained at $\rho = 0.2$ and evaluated at $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ without further training. This setting tests whether relevant information can still be retained and retrieved as the

Table 2: Comparison of spiking neurons on temporal classification tasks. Accuracies are reported in %. **Bold** and underlining indicate the best and second-best results, respectively.

Neuron	GSC	SSC	SHD	DVS128 Gesture
LIF (Burkitt, 2006)	78.97	53.98	74.73	87.85
ad-LIF (Baronig et al., 2025)	46.51	47.42	71.51	89.24
TC-LIF (Zhang et al., 2024)	82.31	55.59	69.43	85.76
DH-LIF (Zheng et al., 2024)	82.47	61.04	82.60	88.89
HetSyn (Deng et al., 2025)	90.00	66.76	**88.60**	89.58
DGN (Bai et al., 2026)	71.73	63.88	73.85	84.72
D-RF (Zhang et al., 2025)	87.44	61.85	86.13	92.01
PMSN (Chen et al., 2026)	87.31	62.16	84.89	90.62
TV-MCN	**90.56**	**70.98**	88.30	**93.75**

Table 3: Comparison of spiking neurons on synthetic sequence tasks. Accuracies are reported in %, except for Adding, which uses MSE. **Bold** and underlining indicate the best and second-best results, respectively.

Neuron	Adding MSE ($\times 10^{-1}$)	Sel. Copy. Acc.	MAD ICR Acc.	MAD Noisy Acc.	STORE–RECALL Bit	Induction Acc.
LIF (Burkitt, 2006)	1.6680	7.17	42.21	38.93	20.05	7.25
ad-LIF (Baronig et al., 2025)	1.6670	7.33	13.92	12.74	26.03	6.98
TC-LIF (Zhang et al., 2024)	1.6720	11.55	26.39	24.21	21.25	6.42
DH-LIF (Zheng et al., 2024)	1.6680	7.09	30.83	30.89	99.27	21.15
HetSyn (Deng et al., 2025)	1.6680	7.02	97.44	85.43	20.00	6.19
DGN (Bai et al., 2026)	1.6720	7.17	31.26	31.04	20.00	8.93
D-RF (Zhang et al., 2025)	0.0161	19.24	47.57	40.15	98.76	21.04
PMSN (Chen et al., 2026)	0.1443	9.58	39.75	34.77	90.13	20.26
TV-MCN	**0.0022**	**93.46**	**100.00**	**99.99**	**99.92**	**97.71**

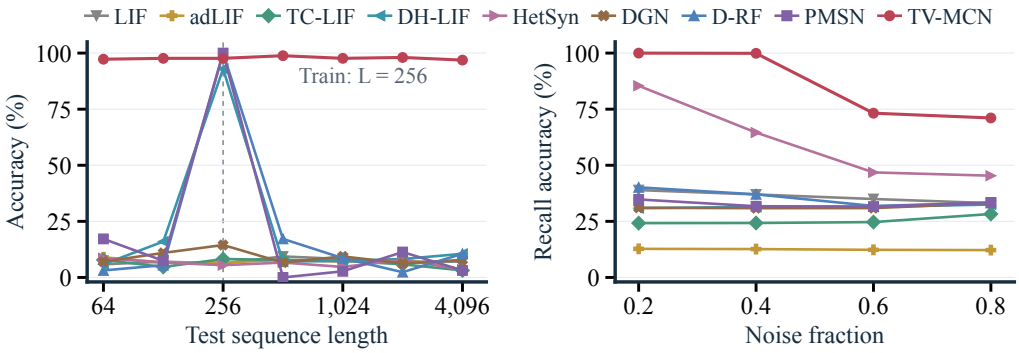


Figure 3: Generalization to sequence length and interference. Left: Length generalization on Induction. The same trained model is evaluated at different sequence lengths, with the dashed line indicating the training length. Right: Generalization across noise fractions ρ on MAD Noisy ICR. Models trained at $\rho = 0.2$ are evaluated at all four noise fractions without retraining.

amount of irrelevant input increases beyond the training condition. As shown in Figure 3 (Right), TV-MCN achieves 99.99%, 99.99%, 99.97%, and 99.90% token accuracy, respectively. At $\rho = 0.8$, the strongest baseline, HetSyn, reaches 79.33%. These results indicate that TV-MCN maintains high recall accuracy under substantially stronger interference.

4.2 Comparison with State-of-the-Art Methods

In this section, we compare TV-MCN with representative full architectures of recent methods and report both parameter counts and computational costs to examine whether its performance gains

Table 4: Comparison with state-of-the-art methods on synthetic sequence tasks. **Bold** and underlining indicate the best and second-best results, respectively.

Task	Model	Params (M)	Est. FLOPs (G/seq)	Performance
MAD ICR Acc. (%)	PMSN (Chen et al., 2026)	0.291	0.098	73.97
	D-RF (Zhang et al., 2025)	3.298	1.059	99.87
	SpikingSSMs (Shen et al., 2025)	35.804	10.491	98.98
	S5-RF (Huber et al., 2024)	1.612	0.887	39.18
	Spiking LMUFormer (Liu et al., 2024)	0.802	0.207	57.65
	TV-MCN	**0.062**	**0.026**	**100.00**
MAD Noisy ICR Acc. (%)	PMSN (Chen et al., 2026)	0.294	0.098	68.16
	D-RF (Zhang et al., 2025)	3.331	1.068	99.85
	SpikingSSMs (Shen et al., 2025)	35.836	10.495	96.96
	S5-RF (Huber et al., 2024)	1.645	0.904	30.78
	Spiking LMUFormer (Liu et al., 2024)	0.811	0.208	50.37
	TV-MCN	**0.066**	**0.026**	**99.99**
Selective Copying Acc. (%)	PMSN (Chen et al., 2026)	0.291	3.500	19.10
	D-RF (Zhang et al., 2025)	3.298	35.244	18.68
	SpikingSSMs (Shen et al., 2025)	35.804	342.255	**97.83**
	S5-RF (Huber et al., 2024)	1.612	29.864	46.68
	Spiking LMUFormer (Liu et al., 2024)	0.802	6.723	7.26
	TV-MCN	**0.062**	**0.827**	95.41
20D STORE–RECALL Bit Acc. (%)	PMSN (Chen et al., 2026)	0.300	1.724	97.83
	D-RF (Zhang et al., 2025)	3.409	17.547	**100.00**
	SpikingSSMs (Shen et al., 2025)	35.880	166.997	99.99
	S5-RF (Huber et al., 2024)	1.690	15.084	98.04
	Spiking LMUFormer (Liu et al., 2024)	0.822	3.377	20.00
	TV-MCN	**0.072**	**0.457**	99.99
Induction Acc. (%)	PMSN (Chen et al., 2026)	0.291	0.972	22.43
	D-RF (Zhang et al., 2025)	3.298	9.924	17.47
	SpikingSSMs (Shen et al., 2025)	35.804	96.697	21.54
	S5-RF (Huber et al., 2024)	1.612	8.391	59.10
	Spiking LMUFormer (Liu et al., 2024)	0.802	1.901	6.92
	TV-MCN	**0.062**	**0.233**	**100.00**
Adding MSE ($\times 10^{-2}$)	PMSN (Chen et al., 2026)	0.288	0.818	0.1210
	D-RF (Zhang et al., 2025)	3.268	8.434	0.0202
	SpikingSSMs (Shen et al., 2025)	35.773	83.053	0.0275
	S5-RF (Huber et al., 2024)	1.582	7.044	0.5470
	Spiking LMUFormer (Liu et al., 2024)	0.795	1.630	8.1600
	TV-MCN	**0.058**	**0.196**	**0.0181**

can be attributed simply to increased model size or computation. The compared methods include PMSN (Chen et al., 2026), D-RF (Zhang et al., 2025), SpikingSSMs (Shen et al., 2025), S5-RF (Huber et al., 2024), and Spiking LMUFormer (Liu et al., 2024). TV-MCN adopts a compact two-layer convolution-augmented spiking sequence network, with architectural details provided in Appendix B. For each method, the core network architecture is kept unchanged across tasks, with only the input and output layers adapted to the task.

Table 4 reports performance, parameter counts, and FLOPs per sequence on the six synthetic sequence tasks. Details of the calculation procedure are provided in Appendix B.5. TV-MCN achieves strong performance across the synthetic sequence tasks while using the fewest parameters and the lowest FLOPs among the compared methods. For example, on Induction, TV-MCN achieves 100.00% accuracy with 0.062 M parameters and 0.233 G estimated FLOPs per sequence, compared with 17.47% for D-RF and 59.10% for S5-RF, both of which require substantially more parameters and computation. These results demonstrate a favorable performance–cost trade-off for TV-MCN across the evaluated tasks. In particular, its performance gains cannot be simply attributed to a larger model or greater computation.

4.3 Ablation Studies

To examine the role of time-variant regulation in selective retention and retrieval, we separately remove the two modulation mechanisms of TV-MCN on Selective Copying while retaining its multiscale state structure. Fixing $\Delta_t = 1$ removes input-dependent control of the state-evolution rate, while replacing B_t, C_t with learnable static parameters B_0, C_0 removes dynamic regulation of input integration and state readout. The network architecture, state size, and training budget remain unchanged.

Table 5: Component ablation on Selective Copying. Exact accuracy requires all target tokens to be correct. **Bold** indicates the best results.

Model Variant	Token Acc. (%)	Exact Acc. (%)
TV-MCN	**93.46**	**37.72**
w/o internal time modulation	7.20	0.00
w/o dynamic B/C coupling	61.24	0.02

Table 5 shows that removing internal-time modulation reduces token accuracy from 93.46% to 7.20%, while static B/C reduces it to 61.24%. Exact accuracy drops from 37.72% to 0.00% and 0.02%, respectively. Although static B/C retains partial token-level recall, it rarely recovers the complete target sequence. Both variants retain the multiscale state structure but lose substantial recall performance.

Figure 4 (left) illustrates these recall differences for the same input. To further test retention under interference, we insert additional irrelevant inputs before recall without retraining, while preserving the target content and order; the results are shown in Figure 4 (right). With 2048 additional irrelevant inputs, token accuracy decreases by only 0.88 percentage points for the full model, compared with 9.33 for the variant with static B/C, while the variant without internal-time modulation remains at 6.45%. The full model therefore combines more accurate recall with greater resistance to extended interference. Together, these results support the role of time-variant state evolution, input integration, and state readout in selective retention and retrieval. Further analysis of instruction-dependent state evolution is provided in Appendix D.2.

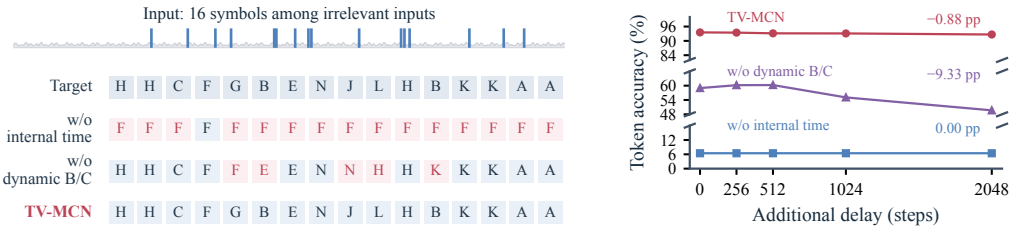


Figure 4: Recall behavior after component removal. Left: Target and outputs of the three variants for the same input; red indicates errors. Right: Token accuracy after inserting additional irrelevant inputs before recall for the same samples.

5 Conclusion

We introduced a biologically inspired unified state-dynamics framework for analyzing and comparing spiking neuron designs. The framework reveals that although existing designs enrich neuronal dynamics, the rules determining input integration, state evolution, and state readout generally remain unchanged during temporal processing. To address this limitation, TV-MCN introduces time-variant regulation into multiscale compartment dynamics: input-dependent internal time regulates state evolution, while dynamic B_t and C_t regulate input integration and state readout. Experiments show competitive performance on temporal classification and substantial gains on synthetic sequence tasks requiring selective retention and retrieval. Generalization tests and ablations further support these time-varying mechanisms. Together, the results highlight time-variant neuronal dynamics as an effective approach for adjusting how information is integrated, maintained, and used over time.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of the manuscript. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael DeBole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. doi: 10.1109/CVPR.2017.781.

Baiqianyi Bai, Haiteng Wang, and Qiang Yu. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. In *International Conference on Learning Representations*, 2026.

Martin Baronig, Romain Ferrand, Simon Sabathiel, and Robert Legenstein. Advancing spatio-temporal processing through adaptation in spiking neural networks. *Nature Communications*, 16:5776, 2025. doi: 10.1038/s41467-025-60878-z.

Bruce P. Bean. The action potential in mammalian central neurons. *Nature Reviews Neuroscience*, 8:451–465, 2007. doi: 10.1038/nrn2148.

Anthony N. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95:1–19, 2006. doi: 10.1007/s00422-006-0068-6.

Stefano Cavallari, Stefano Panzeri, and Alberto Mazzoni. Comparison of the dynamics of neural interactions between current-based and conductance-based integrate-and-fire recurrent networks. *Frontiers in Neural Circuits*, 8:12, 2014. doi: 10.3389/fncir.2014.00012.

Christopher H. Chatham and David Badre. Multiple gates on working memory. *Current Opinion in Behavioral Sciences*, 1:23–31, 2015. doi: 10.1016/j.cobeha.2014.08.001.

X. Chen, J. Wu, C. Ma, Y. Yan, H. Liu, Y. Wu, and K. C. Tan. PMSN: A parallel multi-compartment spiking neuron for multiscale temporal processing. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2026. doi: 10.1109/TNNLS.2026.3708510.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2022. doi: 10.1109/TNNLS.2020.3044364.

Z. Deng, Z. Liu, J. Wang, S. Chen, X. Wei, and Q. Yu. HetSyn: Versatile timescale integration in spiking neural networks via heterogeneous synapses. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Wulfram Gerstner. Time structure of the activity in neural network models. *Physical Review E*, 51 (1):738–758, 1995. doi: 10.1103/PhysRevE.51.738.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014. doi: 10.1017/CBO9781107447615.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Alan L. Hodgkin and Andrew F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952. doi: 10.1113/jphysiol.1952.sp004764.

Thomas E. Huber, Jules Lecomte, Borislav Polovnikov, and Axel von Arnim. Scaling up resonate-and-fire networks for fast deep learning. In *Computer Vision – ECCV 2024 Workshops*, volume 15059 of *Lecture Notes in Computer Science*, pp. 241–258, 2024. doi: 10.1007/978-3-031-92460-6_15.

Eugene M. Izhikevich. Resonate-and-fire neurons. *Neural Networks*, 14(6–7):883–894, 2001. doi: 10.1016/S0893-6080(01)00078-8.

Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003. doi: 10.1109/TNN.2003.820440.

Monika Jadi, Alon Polsky, Jackie Schiller, and Bartlett W. Mel. Location-dependent effects of inhibition on local spiking in pyramidal neuron dendrites. *PLOS Computational Biology*, 8(6): e1002550, 2012. doi: 10.1371/journal.pcbi.1002550.

Chenguang Li and Allan T. Gulledge. NMDA receptors enhance the fidelity of synaptic integration. *eNeuro*, 8(2):ENEURO.0396–20.2020, 2021. doi: 10.1523/ENEURO.0396-20.2020.

Zeyu Liu, Gourav Datta, Anni Li, and Peter Anthony Beerel. Lmuformer: Low complexity yet powerful spiking model with legendre memory units. In *International Conference on Learning Representations*, 2024.

Jeffrey C. Magee. Dendritic integration of excitatory synaptic input. *Nature Reviews Neuroscience*, 1:181–190, 2000. doi: 10.1038/35044552.

Michael Poli, Armin W. Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Ré, Ce Zhang, and Stefano Massaroli. Mechanistic design and scaling of hybrid architectures. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40908–40950, 2024.

Darjan Salaj, Anand Subramoney, Ceca Kraisnikovic, Guillaume Bellec, Robert Legenstein, and Wolfgang Maass. Spike frequency adaptation supports network computations on temporally dispersed information. *eLife*, 10:e65459, 2021. doi: 10.7554/eLife.65459.

David J. Schulz, Richard A. Baines, Chris M. Hempel, Lingjun Li, Birgit Liss, and Hiroaki Misonou. Cellular excitability and the regulation of functional neuronal identity: From gene expression to neuromodulation. *Journal of Neuroscience*, 26(41):10362–10367, 2006. doi: 10.1523/JNEUROSCI.3194-06.2006.

Shuaijie Shen, Chao Wang, Renzhuo Huang, Yan Zhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, and Luziwei Leng. 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. doi: 10.1609/aaai.v39i19.34245.

Katherine Unger, Lauren Ackerman, Christopher H. Chatham, Dima Amso, and David Badre. Working memory gating mechanisms explain developmental change in rule-guided behavior. *Cognition*, 155:8–22, 2016. doi: 10.1016/j.cognition.2016.05.020.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

D. Zhang, M. Zhang, S. Wang, J. Wang, W. Wei, Z. Ma, G. Wang, Y. Yang, and H. Li. Dendritic resonate-and-fire neuron for effective and efficient long sequence modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

S. Zhang, Q. Yang, C. Ma, J. Wu, H. Li, and K. C. Tan. TC-LIF: A two-compartment spiking neuron model for long-term sequential modelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16838–16847, 2024. doi: 10.1609/aaai.v38i15.29625.

H. Zheng, Z. Zheng, R. Hu, B. Xiao, Y. Wu, F. Yu, X. Liu, G. Li, and L. Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15:277, 2024. doi: 10.1038/s41467-023-44614-z.

A Complete Framework Formulations of Representative Neurons

The internal dynamics of spiking neurons can be summarized by the following framework from a state-space perspective:

$$\begin{aligned}
&\dot{h}(t) = Ah(t) + Bx(t) \\
&u(t) = Ch(t) \\
&s(t) = H\left(u\left(t\right) - \vartheta(t)\right), \qquad h\left(t^{+}\right) = R\left(h\left(t^{-}\right), s(t)\right).
\end{aligned}$$

For a given input, we classify neuron models into those with linear state evolution and those with nonlinear state evolution, according to whether state evolution is linear in the current state h, that is, whether the state-evolution matrix A depends on h itself. Below, we start from each model's original dynamics, derive its specific form under the unified framework, and analyze how its state structure and dynamical design affect temporal processing.

A.1 Neuron Models with Linear State Evolution

1. LIF (Burkitt, 2006)

For a LIF neuron with zero resting potential, the continuous-time form is

$$\tau_m \dot{u}(t) = -u(t) + Wx(t),$$

where $x(t)$ is the input and W is the input weight of the neuron. Dividing both sides by τ_m gives

$$\dot{u}(t) = -\frac{1}{\tau_m} u(t) + \frac{1}{\tau_m} Wx(t).$$

Since the membrane potential is the only quantity in LIF that needs to be retained across time, we directly let

$$h(t) = u(t).$$

Therefore, LIF can be written as

$$\begin{aligned}
&\underbrace{\dot{u}(t)}_{\dot{h}} = \underbrace{-\frac{1}{\tau_m}}_{A} \underbrace{u(t)}_{h} + \underbrace{\frac{1}{\tau_m} W}_{B} x(t), \\
&u(t) = \underbrace{1}_{C} \underbrace{u(t)}_{h}, \\
&s(t) = H(u(t) - \vartheta), \qquad u\left(t^{+}\right) = R\left(h\left(t^{-}\right), s(t)\right).
\end{aligned}$$

Thus, LIF corresponds to a one-dimensional state system: $A = -1/\tau_m$ specifies exponential decay of the membrane potential with time constant τ_m; $B = W/\tau_m$ represents input integration and contains the trainable synaptic weight W; $C = 1$ reads out the membrane potential; and R describes soft or hard reset after a spike. The same decay time constant and input-integration coefficient apply throughout the sequence.

2. Spike Response Model (SRM) (Gerstner, 1995; Gerstner et al., 2014)

The SRM with fixed exponential kernels uses two types of traces to retain the effects of past spikes: one for each input and another for the neuron's own spikes. The trace for input j is

$$q_j(t) = \sum_{t_j^f < t} e^{-(t - t_j^f)/\tau_j}.$$

Here, t_j^f denotes the time of an input spike. Each term decays with time constant τ_j, so their sum q_j also decays with this time constant; each new spike increases it by 1. It can therefore be written as an input-driven state update:

$$\dot{q}_j = -\frac{q_j}{\tau_j} + x_j(t),$$

where $x_j(t) = \sum_f \delta(t - t_j^f)$ is the spike train of input j.

The neuron's own firing history is retained in another trace, r:

$$r(t) = \sum_{t^f < t} e^{-(t-t^f)/\tau_r}.$$

Here, t^f denotes the time of a spike generated by the neuron. Like the input traces, r decays between spikes and increases by 1 whenever the neuron fires:

$$\dot{r} = -\frac{r}{\tau_r}, \qquad r(t^+) = r(t^-) + 1 \quad \text{(at a spike)}.$$

The membrane potential is determined jointly by these two types of historical responses. With zero resting potential, the SRM membrane potential is

$$\begin{aligned} u(t) &= \sum_{j=1}^{M} w_j \sum_{t_j^f < t} \epsilon_j(t - t_j^f) + \sum_{t^f < t} \eta(t - t^f) \\ &= \sum_{j=1}^{M} w_j q_j(t) - \kappa r(t). \end{aligned}$$

Here, w_j is the synaptic weight, and $\xi \geq 0$ denotes the time elapsed since a spike. The response to an input spike is $\epsilon_j(\xi) = e^{-\xi/\tau_j}$. The inhibitory response following the neuron's own spike is $\eta(\xi) = -\kappa e^{-\xi/\tau_r}$, where $\kappa > 0$ controls the strength of inhibition.

The membrane potential is thus determined directly by these traces, and the internal states evolving over time are the input traces q_j and the trace r:

$$h(t) = \begin{bmatrix} q_1(t) \\ \vdots \\ q_M(t) \\ r(t) \end{bmatrix}, \qquad x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix}.$$

The SRM with exponential kernels can therefore be written in the following framework form:

$$\underbrace{\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_M \\ \dot{r} \end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix} -\tau_1^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & -\tau_M^{-1} & 0 \\ 0 & \cdots & 0 & -\tau_r^{-1} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_M \\ r \end{bmatrix}}_{h(t)} + \underbrace{\begin{bmatrix} I_M \\ \mathbf{0}^\mathsf{T} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}}_{x(t)},$$

$$u(t) = \underbrace{\begin{bmatrix} w_1 & \cdots & w_M & -\kappa \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_M \\ r \end{bmatrix}}_{h(t)},$$

$$s(t) = H(u(t) - \vartheta), \qquad h(t^+) = R\big(h(t^-), s(t)\big) = \begin{bmatrix} q_1^- \\ \vdots \\ q_M^- \\ r^- + s(t) \end{bmatrix}.$$

From a state-space perspective, the SRM with fixed exponential kernels corresponds to a set of independent real-valued decay states. Choosing the response kernels specifies how these states evolve over time, as reflected in the diagonal entries of A. Input spikes update the corresponding states through B, and C weights these states to form the membrane potential. The trace values evolve with incoming and self-generated spikes, while their decay time constants and readout weights are time-invariant.

3. AD-LIF (BARONIG ET AL., 2025)

The continuous-time form of ad-LIF is

$$\tau_u \dot{u}(t) = -u(t) + I(t) - w(t),$$
$$\tau_w \dot{w}(t) = -w(t) + au(t) + bz(t).$$

In addition to the membrane potential u, ad-LIF introduces an adaptation state w that must be retained across time. The internal state therefore expands from one to two dimensions:

$$h = \begin{bmatrix} u \\ w \end{bmatrix}.$$

In addition to the external input current $I(t)$, the neuron's own spike train $z(t)$ drives the adaptation state. We therefore use the augmented driving vector

$$x = \begin{bmatrix} I \\ z \end{bmatrix}.$$

Thus, ad-LIF can be written in the following form:

$$\underbrace{\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix}}_{\dot{h}} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_u} & -\frac{1}{\tau_u} \\ \frac{a}{\tau_w} & -\frac{1}{\tau_w} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ w \end{bmatrix}}_{h} + \underbrace{\begin{bmatrix} \frac{1}{\tau_u} & 0 \\ 0 & \frac{b}{\tau_w} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} I \\ z \end{bmatrix}}_{x},$$

$$u(t) = \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} u \\ w \end{bmatrix}}_{h},$$

$$s(t) = H\big(u(t) - \vartheta\big), \qquad h(t^+) = R\big(h(t^-), s(t)\big).$$

The learnable time constants τ_u and τ_w parameterize state evolution through A and input integration through B, while $C = [1, 0]$ selects the membrane potential. Adaptation arises from the evolving state w under fixed coefficients governing its interaction with u and its spike-driven updates.

From a state-space perspective, the core of ad-LIF is to extend the single membrane-potential state of LIF into two coupled states, the membrane potential u and the adaptation state w: u drives the accumulation of w, while w in turn suppresses u through negative feedback. The additional state supports history-dependent adaptation through its coupling with the membrane potential and its dependence on spike history. Compared with the single decay mode of LIF, the two-dimensional coupled matrix A enables richer temporal dynamics.

4. TC-LIF (ZHANG ET AL., 2024)

TC-LIF is defined directly in discrete time. It contains a dendritic compartment U^D and a somatic compartment U^S, with the following updates:

$$U^D[t] = U^D[t-1] + \beta_1 U^S[t-1] + I[t] - \gamma S[t-1],$$
$$U^S[t] = U^S[t-1] + \beta_2 U^D[t] - V_{\text{th}} S[t-1],$$
$$S[t] = \Theta\left(U^S[t] - V_{\text{th}}\right),$$

where

$$\beta_1 = -\sigma(c_1) \in (-1, 0), \quad \beta_2 = \sigma(c_2) \in (0, 1).$$

Compared with LIF, TC-LIF adds a dendritic state that must be preserved across time, expanding the internal state from one to two dimensions:

$$h[t] = \begin{bmatrix} U^D[t] \\ U^S[t] \end{bmatrix}.$$

In addition to the external input current $I[t]$, the spike $S[t-1]$ generated at the previous time step directly participates in the updates of both internal states. We therefore use the augmented driving vector

$$x[t] = \begin{bmatrix} I[t] \\ S[t-1] \end{bmatrix}.$$

Because the paper first updates $U^D[t]$ within the same time step and then uses the updated $U^D[t]$ to update $U^S[t]$, substituting the first equation into the second gives

$$U^S[t] = \beta_2 U^D[t-1] + (1 + \beta_1\beta_2)\, U^S[t-1] + \beta_2 I[t] - (\beta_2\gamma + V_{\text{th}})\, S[t-1].$$

Therefore, TC-LIF can be written in the following unified form:

$$\underbrace{\begin{bmatrix} U^D[t] \\ U^S[t] \end{bmatrix}}_{h[t]} = \underbrace{\begin{bmatrix} 1 & \beta_1 \\ \beta_2 & 1+\beta_1\beta_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} U^D[t-1] \\ U^S[t-1] \end{bmatrix}}_{h[t-1]} + \underbrace{\begin{bmatrix} 1 & -\gamma \\ \beta_2 & -(\beta_2\gamma + V_{\text{th}}) \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} I[t] \\ S[t-1] \end{bmatrix}}_{x[t]},$$

$$u[t] = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} U^D[t] \\ U^S[t] \end{bmatrix}}_{h[t]},$$

$$s[t] = H\left(u[t] - V_{\text{th}}\right), \qquad h\left[t^+\right] = R\left(h\left[t^-\right], s[t]\right) = h\left[t^-\right].$$

In TC-LIF, $\beta_1 = -\sigma(c_1)$ and $\beta_2 = \sigma(c_2)$ are determined by the learnable parameters c_1, c_2. The previous spike's recovery terms are already included in the second column of B, so R is the identity in this representation. Inter-compartment interaction and input integration use the same coefficients at each time step, and C always selects the somatic state.

From a state-space perspective, TC-LIF not only extends the one-dimensional state of LIF into two coupled states, but also removes the independent leakage of the two compartments and constrains $\beta_1 < 0$ and $\beta_2 > 0$, such that the state-transition matrix satisfies $\det A = 1$. Its two eigenvalues therefore form a complex-conjugate pair with unit magnitude. As a result, its linear state modes exhibit oscillations without exponential decay, rather than the damped modes found in LIF or general leaky two-dimensional systems. During backpropagation through time, the eigenvalue magnitudes of A^k do not decay exponentially with k, which helps alleviate the vanishing-gradient problem in long sequences.

5. DH-LIF (ZHENG ET AL., 2024)

In addition to the somatic membrane potential of LIF, DH-LIF introduces a dendritic current for each dendritic branch that can preserve information across time. The paper uses the following discrete form:

$$\begin{aligned} u[t+1] &= \beta u[t] + (1-\beta) \sum_{d=1}^{D} i_d[t+1] - u_{\text{th}} o[t] \\ i_d[t+1] &= \alpha_d i_d[t] + (1-\alpha_d)\, I_d[t+1] \\ o[t+1] &= H\left(u[t+1] - u_{\text{th}}\right). \end{aligned}$$

Compared with LIF, DH-LIF adds D dendritic currents that persist across time, expanding the internal state from one to $D+1$ dimensions:

$$h[t] = \begin{bmatrix} i_1[t] \\ \vdots \\ i_D[t] \\ u[t] \end{bmatrix}.$$

Different branches receive their own synaptic inputs, while the previous soma spike $o[t]$ directly participates in the soft reset of the soma. The input to the state is therefore

$$x[t+1] = \begin{bmatrix} I_1[t+1] \\ \vdots \\ I_D[t+1] \\ o[t] \end{bmatrix}.$$

Substituting the dendritic update into the soma update, DH-LIF can be written as

$$\underbrace{\begin{bmatrix} i_1[t+1] \\ \vdots \\ i_D[t+1] \\ u[t+1] \end{bmatrix}}_{h_{[t+1]}} = \underbrace{\begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \alpha_D & 0 \\ (1-\beta)\alpha_1 & \cdots & (1-\beta)\alpha_D & \beta \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} i_1[t] \\ \vdots \\ i_D[t] \\ u[t] \end{bmatrix}}_{h_{[t]}}$$

$$+ \underbrace{\begin{bmatrix} 1-\alpha_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1-\alpha_D & 0 \\ (1-\beta)(1-\alpha_1) & \cdots & (1-\beta)(1-\alpha_D) & -u_{\mathrm{th}} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} I_1[t+1] \\ \vdots \\ I_D[t+1] \\ o[t] \end{bmatrix}}_{x_{[t+1]}},$$

$$u[t+1] = \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} i_1[t+1] \\ \vdots \\ i_D[t+1] \\ u[t+1] \end{bmatrix}}_{h_{[t+1]}},$$

$$\underbrace{o[t+1]}_{s[t+1]} = H\big(u[t+1] - u_{\mathrm{th}}\big), \qquad h[(t+1)^+] = R\left(h[(t+1)^-], s[t+1]\right) = h[(t+1)^-].$$

In DH-LIF, the dendritic and somatic time constants are learnable. However, A, which is formed by α_d and β, the associated B and the $C = [0, \ldots, 0, 1]$ remain unchanged during processing.

From a state-space perspective, the core of DH-LIF is to explicitly introduce a set of real-valued decay states with different time scales. Because the above A is lower triangular, its eigenvalues are $\lambda(A) = \{\alpha_1, \ldots, \alpha_D, \beta\}$. Each dendritic branch therefore corresponds to an independent real decay mode, and different α_d values form different memory time scales. Moreover, these dendritic states are not affected by soma reset and can continue to preserve historical information after multiple neuronal spikes. Compared with ad-LIF/TC-LIF, which enrich dynamics through coupling among a small number of states, DH-LIF expands the temporal basis by introducing multiple heterogeneous decay modes.

6. HETSYN (DENG ET AL., 2025)

HetSyn treats the synaptic current of each input synapse as an independent state. Its continuous-time form is

$$V(t) = \sum_{i=1}^{M} I_i(t) - J(t),$$

$$\frac{\mathrm{d}I_i(t)}{\mathrm{d}t} = -\frac{1}{\tau_{s,i}} I_i(t) + w_i \sum_j \delta\left(t - t_{ji}\right),$$

$$\frac{\mathrm{d}J(t)}{\mathrm{d}t} = -\frac{1}{\tau_J} J(t) + \vartheta \sum_j \delta\left(t - t_{js}\right).$$

Here, I_i is the current of the i-th afferent synapse, and each synapse has its own time constant $\tau_{s,i}$; J is the neuron-level reset current, whose decay is controlled by τ_J. The states propagated across time are

$$h(t) = \begin{bmatrix} I_1(t) \\ \vdots \\ I_M(t) \\ J(t) \end{bmatrix}.$$

Let the spike train of each input synapse be

$$x_i(t) = \sum_j \delta(t - t_{ji}),$$

and the output spike train of the neuron be

$$z(t) = \sum_j \delta\left(t - t_{js}\right).$$

Then the input term is written as

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \\ z(t) \end{bmatrix}.$$

Thus, HetSyn can be directly written in the following form:

$$\underbrace{\begin{bmatrix} \dot{I}_1 \\ \vdots \\ \dot{I}_M \\ \dot{J} \end{bmatrix}}_{\dot{h}} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_{s,1}} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & -\frac{1}{\tau_{s,M}} & 0 \\ 0 & \cdots & 0 & -\frac{1}{\tau_J} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} I_1 \\ \vdots \\ I_M \\ J \end{bmatrix}}_{h} + \underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & w_M & 0 \\ 0 & \cdots & 0 & \vartheta \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_M \\ z \end{bmatrix}}_{x},$$

$$u(t) = \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} I_1 \\ \vdots \\ I_M \\ J \end{bmatrix}}_{h},$$

$$s(t) = H\big(u(t) - \vartheta\big), \qquad h(t^+) = R\big(h(t^-), s(t)\big) = h(t^-).$$

In HetSyn, each synapse's learnable decay coefficient in A and input-integration weight in B specify a time-invariant rule for integrating and retaining its input. The fixed state readout $C = [1, \ldots, 1, -1]$ gives $u(t) = V(t)$. The spike-triggered increment of J is already included through $z(t)$, so R is the identity in this representation.

From a state-space perspective, the state-evolution matrix of HetSyn is purely diagonal, with each synapse corresponding to an independent real-valued decay mode. Different $\tau_{s,i}$ values allow different inputs, after entering the same neuron, to still be retained over different time scales before being aggregated into the membrane potential through $C = [1, \ldots, 1, -1]$. Its richness therefore comes from a large set of independent real-valued modes with different decay rates.

7. D-RF (ZHANG ET AL., 2025)

D-RF starts from the complex oscillatory dynamics of a single RF neuron (Izhikevich, 2001) and extends them to multiple dendritic branches. The i-th dendritic branch is represented by a complex state:

$$z_i(t) = u_i(t) + jv_i(t) \in \mathbb{C}.$$

Here, $u_i(t)$ and $v_i(t)$ are the two real-valued components of the resonant state. Its continuous dynamics are

$$\frac{\mathrm{d}z_i(t)}{\mathrm{d}t} = \left(-\frac{1}{\tau_i} + j\omega_i\right) z_i(t) + \gamma_i I(t),$$

where τ_i controls the decay scale of the branch and ω_i controls its intrinsic oscillation frequency. Meanwhile, D-RF does not use a conventional soft or hard reset, but instead adjusts the threshold according to historical firing activity. The multiple dendritic states form its internal state:

$$h(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix}.$$

Therefore, D-RF can be written in the following form:

$$\underbrace{\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_1} + j\omega_1 & 0 & \cdots & 0 \\ 0 & -\frac{1}{\tau_2} + j\omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{\tau_n} + j\omega_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix}}_{h(t)} + \underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix}}_{B} x(t),$$

$$u(t) = \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix}}_{h(t)},$$

$$s(t) = H\big(u(t) - \vartheta(t)\big), \qquad h(t^+) = h(t^-).$$

In D-RF, the decay rate and oscillation frequency of each mode, as well as the state readout C, are learnable, whereas the input integration B is not learned. From a state-space perspective, the key change in D-RF is that it constructs a set of state modes with different complex eigenvalues:

$$\lambda_i = -\frac{1}{\tau_i} + j\omega_i.$$

The real part $-1/\tau_i$ determines the information decay rate, while the imaginary part ω_i determines the oscillation frequency. Each dendritic branch is therefore a **damped oscillatory mode** with a specific time scale and frequency preference. A single RF neuron branch has only a relatively narrow resonance band, whereas D-RF establishes multiple frequency-selective modes in parallel through different (τ_i, ω_i) values and aggregates them through C, enabling one neuron to cover a broader spectrum and more time scales. In addition, D-RF extends the originally real-valued temporal basis with complex-valued state modes while its dynamical rules also remain fixed during processing.

8. PMSN (CHEN ET AL., 2026)

PMSN starts from multi-compartment interaction and divides a neuron into n adjacently coupled compartments. Each compartment has its own membrane leakage and exchanges state information with neighboring compartments. The generalized multi-compartment dynamics first given in the

paper are

$$\begin{aligned}
\dot{v}^{(1)}(t) &= -\frac{1}{\tau_1} v^{(1)}(t) + \beta_{2,1} v^{(2)}(t) + \gamma_1 I(t), \\
\dot{v}^{(2)}(t) &= \beta_{1,2} v^{(1)}(t) - \frac{1}{\tau_2} v^{(2)}(t) + \beta_{3,2} v^{(3)}(t) + \gamma_2 I(t), \\
&\vdots \\
\dot{v}^{(n)}(t) &= \beta_{n-1,n} v^{(n-1)}(t) - \frac{1}{\tau_n} v^{(n)}(t) + \gamma_n I(t), \\
s(t) &= H\left(v^{(n)}(t) - \vartheta\right).
\end{aligned}$$

Here, τ_i controls the leakage time scale of the i-th compartment, and $\beta_{i,j}$ controls the coupling between adjacent compartments. The final compartment $v^{(n)}$ serves as the spike-generating output compartment. To enable parallel computation, the feedback from the output compartment to the last hidden compartment is removed, i.e., $\beta_{n,n-1} = 0$. Let

$$h(t) = \begin{bmatrix} v^{(1)}(t) \\ v^{(2)}(t) \\ \vdots \\ v^{(n)}(t) \end{bmatrix}.$$

With $x(t) = I(t)$, PMSN can be written as

$$\underbrace{\begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \\ \vdots \\ \dot{v}_{n-1}(t) \\ \dot{v}_n(t) \end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_1} & \beta_{2,1} & 0 & \cdots & 0 \\ \beta_{1,2} & -\frac{1}{\tau_2} & \beta_{3,2} & \cdots & 0 \\ 0 & \beta_{2,3} & -\frac{1}{\tau_3} & \ddots & \vdots \\ \vdots & \vdots & \ddots & -\frac{1}{\tau_{n-1}} & 0 \\ 0 & 0 & \cdots & \beta_{n-1,n} & -\frac{1}{\tau_n} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_{n-1}(t) \\ v_n(t) \end{bmatrix}}_{h(t)} + \underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{bmatrix}}_{B} x(t),$$

$$u(t) = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_{n-1}(t) \\ v_n(t) \end{bmatrix}}_{h(t)},$$

$$s(t) = H\big(u(t) - \vartheta\big), \qquad h(t^+) = R\big(h(t^-), s(t)\big).$$

Here, reset acts only on the final output compartment. Note that after a linear change of basis, PMSN directly trains $A/B/C$ as learnable parameters. Thus, with this model, a fully learnable state-space parameterization has been achieved, but still leaves the state-evolution, input-integration, or state-readout time-invariant.

From a state-space perspective, the core of PMSN is first to extend the single state of LIF into an n-dimensional **coupled state system**. The diagonal terms $-1/\tau_i$ provide the intrinsic leakage time scales of different compartments, while the off-diagonal terms $\beta_{i,j}$ enable continuous information exchange among compartments. Although h_i, τ_i, and $\beta_{i,j}$ can all be real-valued, compartment coupling allows the eigenvalues of A to form complex-conjugate pairs. In other words, interactions among multiple real-valued compartments can naturally produce complex oscillations. After eigendecomposition of A, these coupled compartment states become independent eigenmodes, with different eigenvalues corresponding to different decay rates and oscillation frequencies, thereby forming multiple temporal bases.

9. TV-MCN

$$\underbrace{\begin{bmatrix}\dot{h}_1(t)\\ \dot{h}_2(t)\\ \vdots\\ \dot{h}_n(t)\end{bmatrix}}_{\dot{h}(t)} = \Delta_t \underbrace{\begin{bmatrix}-e^{a_1}+j\omega_1 & 0 & \cdots & 0\\ 0 & -e^{a_2}+j\omega_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & -e^{a_n}+j\omega_n\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}h_1(t)\\ h_2(t)\\ \vdots\\ h_n(t)\end{bmatrix}}_{h(t)} + \Delta_t \underbrace{\begin{bmatrix}b^{\mathrm{R}}_{1,t}+jb^{\mathrm{I}}_{1,t}\\ b^{\mathrm{R}}_{2,t}+jb^{\mathrm{I}}_{2,t}\\ \vdots\\ b^{\mathrm{R}}_{n,t}+jb^{\mathrm{I}}_{n,t}\end{bmatrix}}_{B_t} x(t),$$

$$u(t) = \underbrace{\begin{bmatrix}c^{\mathrm{R}}_{1,t}+jc^{\mathrm{I}}_{1,t}\\ c^{\mathrm{R}}_{2,t}+jc^{\mathrm{I}}_{2,t}\\ \vdots\\ c^{\mathrm{R}}_{n,t}+jc^{\mathrm{I}}_{n,t}\end{bmatrix}^{\mathsf{T}}}_{C_t} \underbrace{\begin{bmatrix}h_1(t)\\ h_2(t)\\ \vdots\\ h_n(t)\end{bmatrix}}_{h(t)},$$

$$s(t) = H\big(u(t)-\vartheta\big), \qquad h\left(t^{+}\right) = R\left(h\left(t^{-}\right), s(t)\right).$$

where

$$\begin{aligned}\Delta_t &= \mathrm{softplus}\left(b_\Delta + W_\Delta q_t\right),\\ q_t &= \mathrm{SiLU}(W_q \mathbf{x}_t),\\ B_t &= W_B \mathbf{x}_t,\\ C_t &= W_C \mathbf{x}_t.\end{aligned}$$

The superscripts R and I denote real and imaginary parts, respectively.

A.2 Neuron Models with Nonlinear State Evolution

1. Hodgkin–Huxley (HH) (Hodgkin & Huxley, 1952)

Starting from membrane capacitance and sodium, potassium, and leak currents, HH describes action-potential generation and recovery through changes in ionic conductances with membrane potential. In addition to the membrane potential V, the model contains a sodium-channel activation variable m, a sodium-channel inactivation variable q, and a potassium-channel activation variable n. Here, the original inactivation variable h is denoted by q to distinguish it from the state vector in the framework. The continuous dynamics are

$$\begin{aligned}C_m\dot{V} &= I - \bar{g}_{\mathrm{Na}} m^3 q (V - E_{\mathrm{Na}}) - \bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}}) - g_{\mathrm{L}}(V - E_{\mathrm{L}}),\\ \dot{m} &= \alpha_m(V)(1-m) - \beta_m(V)m,\\ \dot{q} &= \alpha_q(V)(1-q) - \beta_q(V)q,\\ \dot{n} &= \alpha_n(V)(1-n) - \beta_n(V)n.\end{aligned}$$

Here, C_m is the membrane capacitance; $\bar{g}_{\mathrm{Na}}, \bar{g}_{\mathrm{K}}, g_{\mathrm{L}}$ are conductance parameters; and $E_{\mathrm{Na}}, E_{\mathrm{K}}, E_{\mathrm{L}}$ are the corresponding reversal potentials. The gating transition rates α and β depend on the current membrane potential.

The membrane potential and the three gating variables all evolve over time. We define the state vector and input as

$$h(t) = \begin{bmatrix}V(t)\\ m(t)\\ q(t)\\ n(t)\\ 1\end{bmatrix}, \qquad x(t) = I(t).$$

The final coordinate is fixed at 1. The membrane-potential equation can be written as

$$\dot{V} = -a_V V + b_V + \frac{I}{C_m},$$

where

$$a_V = \frac{\bar{g}_{\mathrm{Na}} m^3 q + \bar{g}_{\mathrm{K}} n^4 + g_{\mathrm{L}}}{C_m},$$
$$b_V = \frac{\bar{g}_{\mathrm{Na}} m^3 q E_{\mathrm{Na}} + \bar{g}_{\mathrm{K}} n^4 E_{\mathrm{K}} + g_{\mathrm{L}} E_{\mathrm{L}}}{C_m}.$$

Each gating equation can likewise be written as $\dot{z} = -(\alpha_z + \beta_z)z + \alpha_z$, where $z \in \{m, q, n\}$. HH can therefore be written in the following framework form:

$$\underbrace{\begin{bmatrix} \dot{V} \\ \dot{m} \\ \dot{q} \\ \dot{n} \\ 0 \end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix} -a_V & 0 & 0 & 0 & b_V \\ 0 & -(\alpha_m + \beta_m) & 0 & 0 & \alpha_m \\ 0 & 0 & -(\alpha_q + \beta_q) & 0 & \alpha_q \\ 0 & 0 & 0 & -(\alpha_n + \beta_n) & \alpha_n \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A(h)} \underbrace{\begin{bmatrix} V \\ m \\ q \\ n \\ 1 \end{bmatrix}}_{h(t)} + \underbrace{\begin{bmatrix} C_m^{-1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B} x(t),$$

$$u(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} V \\ m \\ q \\ n \\ 1 \end{bmatrix}}_{h(t)} = V(t),$$

$$s(t) = H_{\uparrow}(u(t) - \vartheta), \qquad h(t^+) = R\big(h(t^-), s(t)\big) = h(t^-).$$

In this expression, α and β are evaluated at the current $V(t)$; $H_{\uparrow}$ records a spike when the membrane potential crosses the detection threshold from below. Action potentials and repolarization in HH arise naturally from the continuous dynamics, so R is the identity mapping here.

From a state-space perspective, HH consists of the membrane potential and three gating states. The gating states determine the coefficients governing membrane-potential evolution, while the membrane potential in turn affects the evolution of the gating states. Thus, the state-evolution matrix A is itself a function of the state h. In contrast to the fixed decay in LIF, the rates at which the HH states change and the values they approach both vary with the other states. By adding states that regulate one another, HH allows the neuron's time scales and forms of evolution to change with its own state. The input integration B and state readout C remain fixed, respectively injecting current into the membrane potential and reading it out.

2. Izhikevich Neuron (Izhikevich, 2003)

The Izhikevich model describes a neuron using the membrane potential and a recovery variable. A quadratic voltage term produces rapid depolarization, while the recovery variable provides negative feedback. Here, the original recovery variable u is denoted by w to distinguish it from the membrane-potential readout in the framework. The dynamics are

$$\dot{v} = 0.04v^2 + 5v + 140 - w + I,$$
$$\dot{w} = a(bv - w).$$

When $v \geq 30\,\mathrm{mV}$, a spike is recorded and the following updates are applied:

$$v \leftarrow c, \qquad w \leftarrow w + d.$$

Here, a controls the rate of change of the recovery variable, b controls its response to the membrane potential, c is the post-spike membrane potential, and d is the increment in the recovery variable caused by each spike.

The membrane potential v and recovery variable w jointly form the internal state. We define the state vector and input as

$$h(t) = \begin{bmatrix} v(t) \\ w(t) \\ 1 \end{bmatrix}, \qquad x(t) = I(t).$$

Writing the voltage term as $0.04v^2 + 5v = (0.04v + 5)v$ gives the following framework form:

$$\underbrace{\begin{bmatrix}\dot{v}\\ \dot{w}\\ 0\end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix}0.04v+5 & -1 & 140\\ ab & -a & 0\\ 0 & 0 & 0\end{bmatrix}}_{A(h)} \underbrace{\begin{bmatrix}v\\ w\\ 1\end{bmatrix}}_{h(t)} + \underbrace{\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}}_{B} x(t),$$

$$u(t) = \underbrace{\begin{bmatrix}1 & 0 & 0\end{bmatrix}}_{C} \underbrace{\begin{bmatrix}v\\ w\\ 1\end{bmatrix}}_{h(t)} = v(t),$$

$$s(t) = H(u(t) - 30)\,, \qquad h(t^+) = R\big(h(t^-), s(t)\big) = \begin{bmatrix}(1-s)v^- + sc\\ w^- + sd\\ 1\end{bmatrix}.$$

Here, B integrates the input current into the membrane-potential state, and C selects this state as the membrane-potential readout. After a spike, R resets v to c and increases w by d.

From a state-space perspective, the Izhikevich model is a two-dimensional coupled system. The diagonal entry $0.04v + 5$ in $A(h)$, corresponding to the membrane potential, varies with the state. The membrane potential can therefore change the rule governing its own evolution, making this a state-dependent nonlinear model. The recovery-state decay, inter-state coupling, input integration, and state readout are governed by constant coefficients that can not vary over time.

3. DGN (BAI ET AL., 2026)

DGN adds a set of synaptic current states alongside the membrane potential. These states both drive the membrane potential and regulate its decay rate through dynamic conductance. Its continuous dynamics are

$$\tau_s \frac{dD_i(t)}{dt} = -D_i(t) + z_i(t)$$
$$\frac{dV(t)}{dt} = -\left(g_l + \sum_{i=1}^{N} C_i D_i(t)\right) V(t) + \sum_{i=1}^{N} W_i D_i(t).$$

Here, D_i is the time-decaying synaptic current state of the i-th synapse, C_i is the conductance weight that aggregates these states, and W_i is the input-current weight. The internal state should therefore include all synaptic current states and the membrane potential:

$$h(t) = \begin{bmatrix} D_1(t)\\ D_2(t)\\ \vdots\\ D_N(t)\\ V(t)\end{bmatrix},$$

and the input consists of all presynaptic spikes:

$$x(t) = \begin{bmatrix} z_1(t)\\ z_2(t)\\ \vdots\\ z_N(t)\end{bmatrix}.$$

DGN can then be naturally written as

$$\underbrace{\begin{bmatrix}\dot{D}_1(t)\\ \dot{D}_2(t)\\ \vdots\\ \dot{D}_N(t)\\ \dot{V}(t)\end{bmatrix}}_{\dot{h}(t)} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_s} & 0 & \cdots & 0 & 0\\ 0 & -\frac{1}{\tau_s} & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & -\frac{1}{\tau_s} & 0\\ w_1 & w_2 & \cdots & w_N & -\left(g_l + \sum_{i=1}^{N} c_i D_i(t)\right)\end{bmatrix}}_{A(h)} \underbrace{\begin{bmatrix}D_1(t)\\ D_2(t)\\ \vdots\\ D_N(t)\\ V(t)\end{bmatrix}}_{h(t)}$$

$$+ \underbrace{\begin{bmatrix}\frac{1}{\tau_s} & 0 & \cdots & 0\\ 0 & \frac{1}{\tau_s} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \frac{1}{\tau_s}\\ 0 & 0 & \cdots & 0\end{bmatrix}}_{B} \underbrace{\begin{bmatrix}z_1(t)\\ z_2(t)\\ \vdots\\ z_N(t)\end{bmatrix}}_{x(t)},$$

$$u(t) = \underbrace{\begin{bmatrix}0\\ \vdots\\ 0\\ 1\end{bmatrix}^{\mathsf{T}}}_{C} \underbrace{\begin{bmatrix}D_1(t)\\ \vdots\\ D_N(t)\\ V(t)\end{bmatrix}}_{h(t)},$$

$$s(t) = H\big(u(t) - \vartheta\big), \qquad h(t^+) = R\big(h(t^-), s(t)\big).$$

From a state-space perspective, DGN introduces state-dependent evolution in the membrane-potential dimension. Rather than being determined directly by the current input, its state evolution depends on input history stored in the synaptic states $D_i(t)$, whose weighted combination changes the decay coefficient of the membrane potential V. The state matrix should therefore be viewed as $A(h)$. DGN thus forms a nonlinear state system, and its state updates over the full sequence cannot be computed directly in parallel using a fixed convolution or an affine scan. This dynamic modulation is confined to membrane-potential decay, while the remaining synaptic states still evolve with fixed time constants. Meanwhile, B writes inputs into the synaptic states with the fixed coefficient τ_s^{-1}, and C reads out the membrane potential with a fixed mapping.

B Experimental Settings

B.1 Comparison Settings and Task-Level Training Configurations

The experiments include shared-network neuron comparisons and complete-network comparisons. The shared-network comparisons in Tables 2 and 3 use a fully connected network with two layers of 128 neurons for all nine neuron models. The network does not use cross-layer accumulated spike residuals, and the task head reads the activity of the final layer. Input encoders and task heads are configured for each dataset and kept identical across neuron models within the same task. Each method retains its own state variables, synaptic integration, and spike-generation mechanism; DH-LIF, HetSyn, and DGN retain their internal synaptic projections and states.

In the complete-network comparison in Table 4, each method uses its representative architecture. Each method retains the same network across the six synthetic sequence tasks, with only the input and output layers adjusted. All methods use the same data splits and test protocols.

Table 6 lists the task-level training configurations for the shared-network comparisons. Learning rates refer to global base learning rates; method-specific settings are listed separately below. The complete-network comparison uses the same preset training budgets and batch sizes for the six synthetic sequence tasks, with learning rates, optimizers, and architectures specified in Section B.3.

Table 6: Task-level training configurations for the shared-network comparisons.

Task	Preset training budget	Batch size	Base learning rate	Optimizer	Normalization
GSC	200 epochs	128	0.001	AdamW	LayerNorm
SSC	128 epochs	128	0.001	Adam	LayerNorm
SHD	100 epochs	40	0.01	AdamW	LayerNorm
DVS128 Gesture	200 epochs	32	0.005	AdamW	Masked BatchNorm
MAD ICR	200 epochs	128	0.0005	AdamW	LayerNorm
MAD Noisy ICR	200 epochs	128	0.0005	AdamW	LayerNorm
Selective Copying	40,000 steps	64	0.0001	Adam	LayerNorm
Adding	100,000 steps	128	0.001	AdamW	None
STORE–RECALL	4,000 steps	32	0.001	Adam	LayerNorm
Induction	204,800 steps	8	0.001	Adam	BatchNorm

GSC uses a step learning-rate schedule, multiplying the learning rate by 0.1 every 30 epochs. SSC, Selective Copying, and Induction use constant learning rates.

Model selection uses the checkpoint with the best validation performance, measured by the lowest MSE for Adding and the highest task-specific accuracy for the other tasks. Learning rates are adjusted for individual tasks where needed, with the adopted values reported in the configuration tables.

The common linear, embedding, and normalization modules are initialized through their respective `reset_parameters()` methods using a separate shared random stream. Internal neuron parameters follow the initialization scheme of each method.

B.2 Internal State Sizes, Initialization, and Task-Specific Settings

Table 7 summarizes the internal state sizes and baseline initialization schemes for the shared-network comparisons. Table 8 lists the adopted task-specific adjustments; parameters not listed there retain their baseline settings. Here, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and standard deviation σ, and $\mathcal{U}(a, b)$ denotes a uniform distribution over $[a, b]$.

For the baseline neurons, we take the initialization schemes of the respective original methods as starting points; the baseline settings used here are summarized in Table 7. We then adjust selected initialization settings and neuron parameters for individual tasks, while keeping the depth, width, and connectivity of the shared network unchanged. Table 8 lists the adopted adjustments, including task-specific learning rates.

B.3 Architectures and Training Configurations for Complete-Network Comparisons

The complete-network comparison includes PMSN, D-RF, SpikingSSMs, S5-RF, Spiking LMUFormer, and TV-MCN. Their architectures are listed in Table 9. Each network retains its own normalization, residual connections, and internal memory structure. The normalization settings in Table 6 apply only to the shared-network comparisons.

PMSN retains its coupled-spectrum and structured B/C initialization. Ordinary dropout is set to 0 on Selective Copying and STORE–RECALL; the associated spike dropout is also set to 0 on STORE–RECALL. D-RF retains its complex-pole, random complex-readout, and log-uniform timescale initialization. S5-RF initializes its internal step sizes log-uniformly over $[0.004, 0.1]$. Spiking LMUFormer uses a fixed Legendre memory system; its native linear weights use truncated normal initialization with a standard deviation of 0.02, and biases are initialized to 0. TV-MCN uses the

Table 7: Internal state sizes and baseline initialization schemes of the neuron models.

Neuron	Internal state size and initialization
LIF	The decay coefficient is 0.9, and the firing threshold is 1.
ad-LIF	Membrane time constants are initialized from $\mathcal{N}(20, 5^2)$, and adaptation time constants from $\mathcal{N}(200, 5^2)$. The base threshold is 1.6.
TC-LIF	The two raw coupling parameters are initialized to 0, corresponding to an effective value of 0.5 after the sigmoid transformation. The firing threshold is 1, and the dendritic reset coefficient is 0.5.
DH-LIF	Four dendritic branches are used. Membrane parameters are initialized from $\mathcal{U}(0, 4)$ and branch parameters from $\mathcal{U}(2, 6)$. The firing threshold is 1.
HetSyn	Synaptic weights are initialized from $\mathcal{N}(0, 0.01^2)$. Synaptic time constants are initialized to 20 ms, the time step is 1 ms, and the firing threshold is 1.
DGN	The time step is 1, and the synaptic time constant is 20. Conductance and current weights are both initialized from $\mathcal{N}(0.01, 0.005^2)$.
D-RF	A four-dimensional complex state is used. The initial decay rate is 0.5, and frequencies are set to $n\pi$. Time steps are initialized log-uniformly over $[0.001, 0.1]$, and complex-valued readout weights are randomly initialized.
PMSN	Four compartments are used with Coupled-Spectrum initialization. Input integration and state readout use Structured-Physical B and Structured-Modal C initialization, respectively.
TV-MCN	Four compartments are used with a conditioning rank of 32. Initial decay rates are $1, 2, 3, 4$, and frequencies are $0, \pi, 2\pi, 3\pi$. Internal time scales are initialized log-uniformly over $[0.001, 0.1]$. Controller modules use their respective Xavier or uniform initialization schemes.

Table 8: Method- and task-specific configurations for the shared-network comparisons.

Neuron and task	Adopted configuration
LIF – STORE–RECALL	The decay coefficient is changed from 0.9 to 0.99.
ad-LIF – SSC	Membrane time constants are initialized from $\mathcal{N}(5, 1.25^2)$.
ad-LIF – Adding	Membrane time constants are initialized from $\mathcal{N}(5, 1^2)$, and adaptation time constants from $\mathcal{N}(200, 50^2)$. The base threshold is 0.4.
ad-LIF – Induction	Membrane time constants are initialized from $\mathcal{N}(5, 1.25^2)$, and adaptation time constants from $\mathcal{N}(200, 50^2)$. The base threshold is 0.4.
ad-LIF – STORE–RECALL	Membrane time constants are initialized from $\mathcal{N}(5, 1.25^2)$, and adaptation time constants from $\mathcal{N}(200, 50^2)$. The base threshold remains 1.6.
TC-LIF – STORE–RECALL	The initial effective coupling value is changed from 0.5 to 0.125.
DGN – STORE–RECALL	Conductance weights are initialized from $\mathcal{N}(0.001, 0.0005^2)$; current weights retain the initialization $\mathcal{N}(0.01, 0.005^2)$.
PMSN – SHD	The global base learning rate is 0.001.
TV-MCN – SHD	The global base learning rate is 0.002.
TV-MCN – MAD Noisy ICR	The global base learning rate is 0.001.

Table 9: Architectures used in the complete-network comparison.

Method	Architecture
PMSN	Two stages, each containing three spiking layers, with widths of 128 and 256, respectively. Each neuron has four compartments. Average pooling is applied along the feature dimension between stages and at the network output.
D-RF	Six independent residual blocks of width 512, with a complex-state dimension of 8 and a causal convolution at the input. Following the adopted forward implementation, the first block is executed twice, giving six independent sets of block parameters and seven block calls.
SpikingSSMs	Sixteen layers of width 1024 and state dimension 64, with a pre-LayerNorm residual structure. The authors' pretrained SDN modules are loaded and frozen.
S5-RF	Four layers of width 512, with HiPPO spectral initialization divided into 16 blocks. The RF core and LI readout are retained.
Spiking LMUFormer	One native block with hidden width 256, memory size 512, a feedforward expansion factor of 4, and memory time-scale parameter $\theta = 128$.
TV-MCN	Two layers of 128 neurons, with four compartments per neuron. Each layer includes a depthwise causal convolution with kernel size 4. The convolutions are initialized as identity mappings and add 1,280 parameters in total.

same neuron initialization as in the shared-network experiments, with identity initialization for the added convolutions.

For the complete-network comparison, we take the learning rates from each baseline's original configuration as a starting point and make task-specific adjustments where needed. Table 10 lists the adopted global base learning rates for all models. Spiking LMUFormer uses Adam for all tasks. The other five methods use AdamW for MAD ICR, MAD Noisy ICR, and Adding, and Adam for Selective Copying, STORE–RECALL, and Induction. On Induction, PMSN uses the task-uniform Adam configuration with a learning rate of 0.001 and weight decay of 0.

Table 10: Global base learning rates for the complete-network comparison.

Method	MAD ICR	MAD Noisy ICR	Selective Copying	STORE–RECALL	Induction	Adding
PMSN	0.01	0.01	0.01	0.01	0.001	0.01
D-RF	0.005	0.005	0.005	0.005	0.005	0.005
SpikingSSMs	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
S5-RF	0.004	0.004	0.004	0.004	0.004	0.004
Spiking LMUFormer	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
TV-MCN	0.0005	0.001	0.0001	0.001	0.001	0.001

B.4 Evaluation Protocol

All methods use the same data splits and test protocols.

Adding reports mean squared error (MSE) at sequence length $L = 1000$. Tables 3 and 4 report MSE in units of 10^{-1} and 10^{-2}, respectively, rounded to four decimal places after rescaling. For MAD and Selective Copying, token accuracy measures the fraction of target tokens predicted correctly. For 20D STORE–RECALL, bit accuracy measures the fraction of correctly recalled bits.

For Induction, models are trained at sequence length $L = 256$ and evaluated without retraining at $L \in \{64, 128, 256, 512, 1024, 2048, 4096\}$. We report the equally weighted mean accuracy across these seven test lengths and present the accuracy at each length separately.

For the Adding visualization in Figure 5, all nine neuron models use the shared fully connected network and the checkpoints corresponding to their results in Table 3. All panels use the same 512 test sequences at $L = 1000$ and identical axis limits. The dashed line denotes $y = x$. The MSE values in the table are computed over the full test set.

For MAD Noisy ICR, the main comparison uses sequence length $L = 128$ and irrelevant-input proportion $\rho = 0.2$. In Figure 3 (right), each model is trained at $\rho = 0.2$, and the same checkpoint is evaluated at $\rho \in \{0.2, 0.4, 0.6, 0.8\}$ without retraining or checkpoint reselection. Each setting contains 1,280 test sequences, with identical inputs and targets across models. Token accuracy is pooled over supervised recall positions. Shaded 95% confidence intervals are obtained from 2,000 bootstrap resamples of whole test sequences and quantify test-sample uncertainty for each fixed trained model. Data generation follows the MAD protocol at each noise fraction, so the number of content pairs and supervised recall positions also vary with ρ.

For the delayed-recall experiment in Figure 4 (right), we evaluate the checkpoints corresponding to Table 5 on the same 128 test sequences of base length $L = 4096$. We insert $0, 256, 512, 1024,$ or 2048 irrelevant symbols immediately before the recall delimiter, preserving the original input prefix and the 16-token target sequence. All checkpoints are evaluated without retraining. Accuracies are computed on this fixed subset, whereas Table 5 uses the full test set, so the zero-delay accuracies need not match the table values.

B.5 Parameter and Computational Cost Estimation

For Table 4, we report whole-network parameter counts (M) and estimated FLOPs (G/sequence), including the input encoder, network, and task head. Trainable and frozen parameters are included, shared parameters are counted once, and each complex parameter counts as two real-valued scalars. Non-parameter buffers are excluded. Induction costs are averaged equally over the seven evaluation lengths after next-token shifting.

FLOPs estimate dense arithmetic for a single-sequence forward pass in evaluation mode, without spike-sparsity discounts. A real multiply–accumulate counts as two operations; complex multiplication and addition count as six and two, respectively. Real and complex FFTs are estimated as $2.5n \log_2 n$ and $5n \log_2 n$ operations per transform. One-time initialization, embedding lookups, indexing, comparisons, and data movement are excluded.

PMSN, D-RF, SpikingSSMs, and S5-RF are counted by operator tracing, including forward-pass kernel generation and the frozen SDN modules in SpikingSSMs. TV-MCN is traced using the reference recurrence, including causal convolutions, input-dependent coefficient generation, discretization, and state updates. Spiking LMUFormer uses module-based estimates with precomputed kernel spectra, excluding their construction, and assigns six arithmetic operations per LIF element.

For normalization inputs with m elements, r rows, and c channels, the first four counters use $7m+2r$ operations for affine LayerNorm and $2m + 5c$ for inference BatchNorm, with FFT scaling counted separately. The TV-MCN and Spiking LMUFormer counters use $7m$ and $2m$, respectively, without separate FFT-scaling terms. These estimates characterize the arithmetic cost of the evaluated system configurations.

B.6 Parameter Counts of the Shared Network

Table 11 reports the total trainable parameter counts of the two-layer, 128-neuron-per-layer networks used in Tables 2 and 3. Counts include the fully connected projections and neuron-specific parameters, but exclude the input encoder, normalization layers, and task head. Each complex-valued parameter counts as two real-valued scalars.

Relative to time-invariant multi-compartment designs, TV-MCN introduces additional input-dependent projections to generate Δ_t, B_t, and C_t, increasing its parameter count over D-RF and PMSN. Its count remains below those of DH-LIF, HetSyn, and DGN in this configuration. DH-LIF uses a branch-expanded input projection for its four dendritic branches, together with membrane and branch time-constant parameters; the reported count includes all entries of the dense projection tensor, including masked-out entries. HetSyn assigns a learnable weight and decay coefficient to each input–neuron pair. DGN maintains a trace for each presynaptic input and uses separate conductance and current weight matrices to map these traces to each neuron.

Table 11: Trainable parameter counts of the shared two-layer network. K denotes 10^3 parameters. **Bold** and underlining mark the smallest and second-smallest counts, respectively.

Neuron	Trainable Params (K)
LIF (Burkitt, 2006)	**33.024**
ad-LIF (Baronig et al., 2025)	33.536
TC-LIF (Zhang et al., 2024)	33.028
DH-LIF (Zheng et al., 2024)	133.376
HetSyn (Deng et al., 2025)	65.536
DGN (Bai et al., 2026)	65.536
D-RF (Zhang et al., 2025)	37.632
PMSN (Chen et al., 2026)	39.680
TV-MCN	56.064

C Additional Recall Results

Exact-sequence accuracy measures the fraction of examples in which all target tokens are predicted correctly for Selective Copying, MAD ICR, and MAD Noisy ICR. For 20D STORE–RECALL, word accuracy measures the fraction of recalled words with all 20 bits correct. These results use the same checkpoints and test sets as the corresponding main-text tables.

C.1 Comparison with Existing Spiking Neurons

Table 12 reports supplementary exact-sequence and word accuracies for the shared-network comparison in Table 3. Many baseline neurons obtain zero or near-zero exact-sequence accuracy, indicating that correct predictions at individual positions rarely extend to complete target-sequence recovery. Even high token or bit accuracy can conceal incomplete recall: HetSyn achieves 97.44% token accuracy on MAD ICR but only 24.38% exact-sequence accuracy, while DH-LIF achieves 99.27% bit accuracy on STORE–RECALL but 86.27% word accuracy. These stricter metrics reveal limitations in recovering complete sequences or words that are less apparent from per-position accuracy. TV-MCN reaches 99.92% and 99.38% exact-sequence accuracy on MAD ICR and MAD Noisy ICR, respectively, and 98.40% word accuracy on STORE–RECALL.

Table 12: Exact-sequence and word accuracies for the shared-network neuron comparison. All values are in %. **Bold** indicates the best results, and underlining marks nonzero second-best results.

Neuron	Sel. Copy. Exact	MAD ICR Exact	MAD Noisy Exact	STORE–RECALL Word
LIF (Burkitt, 2006)	0.00	0.00	0.00	0.00
ad-LIF (Baronig et al., 2025)	0.00	0.00	0.00	0.00
TC-LIF (Zhang et al., 2024)	0.00	0.00	0.00	0.00
DH-LIF (Zheng et al., 2024)	0.00	0.00	0.00	86.27
HetSyn (Deng et al., 2025)	0.00	24.38	0.31	0.00
DGN (Bai et al., 2026)	0.00	0.00	0.00	0.00
D-RF (Zhang et al., 2025)	0.00	0.00	0.00	78.79
PMSN (Chen et al., 2026)	0.00	0.00	0.00	26.02
TV-MCN	**37.72**	**99.92**	**99.38**	**98.40**

C.2 Comparison with State-of-the-Art Methods

Table 13 reports supplementary exact-sequence and word accuracies for the complete-network comparison in Table 4.

Table 13: Exact-sequence and word accuracies for the complete-network comparison. All values are in %. **Bold** indicates the best results, and underlining marks nonzero second-best results.

Model	Sel. Copy. Exact	MAD ICR Exact	MAD Noisy Exact	STORE–RECALL Word
PMSN (Chen et al., 2026)	0.00	0.00	0.00	63.46
D-RF (Zhang et al., 2025)	0.00	93.13	94.22	**100.00**
SpikingSSMs (Shen et al., 2025)	**75.66**	56.80	25.55	99.82
S5-RF (Huber et al., 2024)	0.00	0.00	0.00	67.38
Spiking LMUFormer (Liu et al., 2024)	0.00	0.00	0.00	0.00
TV-MCN	49.93	**99.92**	**99.69**	99.82

D Supplementary Analyses

D.1 Adding Predictions

Figure 5 compares Adding predictions under the shared network. TV-MCN's predictions follow $y = x$ more closely, consistent with its lower MSE in Table 3.

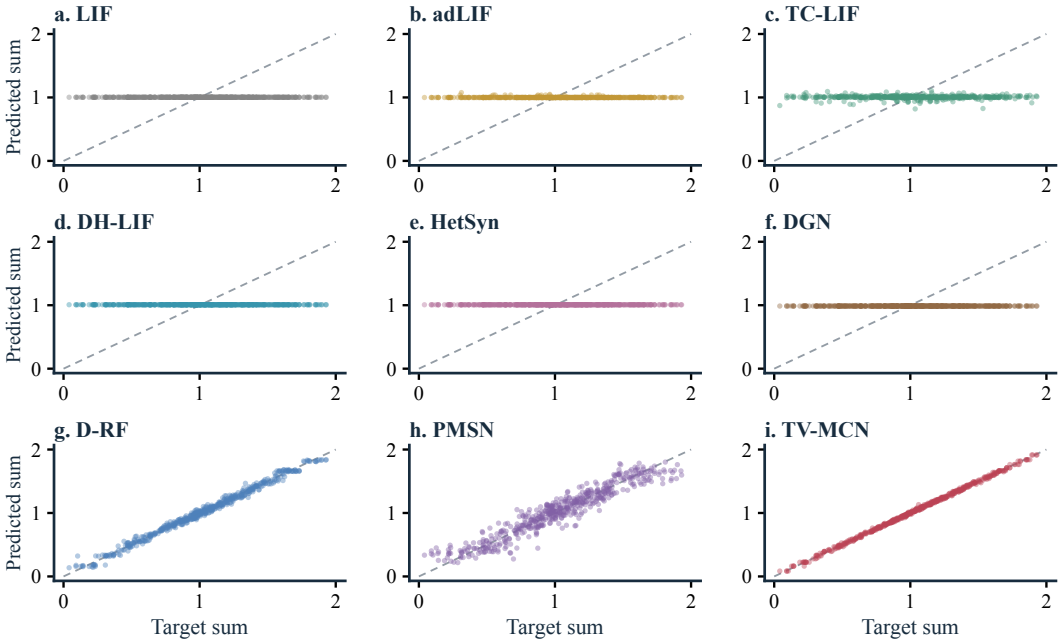


Figure 5: Adding predictions for nine neuron models under the shared two-layer fully connected network with 128 neurons per layer.

D.2 Instruction-Dependent State Evolution

We further examine whether the current instruction changes how the model maintains historical information. In STORE–RECALL, we fix the content sequence and change only the position of the STORE instruction, requiring the model to remember either A or B. Figure 6 shows internal time, the contributions of A and B to the internal states, and the final recall outputs under the two instructions, revealing whether the model selectively retains different content according to the current instruction.

Figure 6 directly illustrates how the current instruction changes neuronal dynamics. When STORE specifies the content to retain, TV-MCN's internal time step decreases markedly, slowing internal-state evolution and promoting information retention. In the full model, the contribution of the selected content, A or B, persists until recall, while the contributions of unselected content and irrelevant inputs gradually weaken. Removing internal-time modulation largely eliminates this distinc-

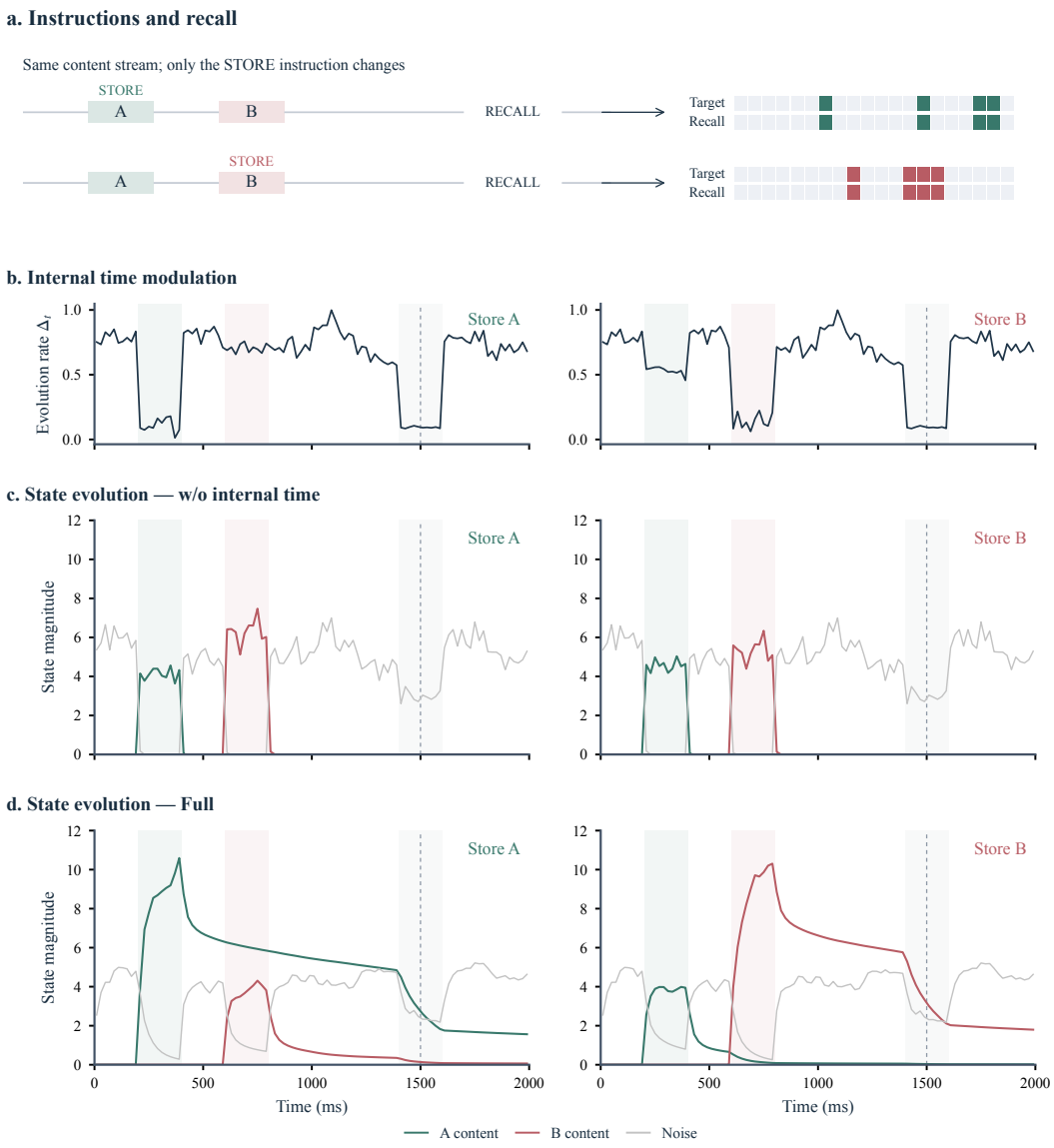


Figure 6: How instructions change information retention. (a) The input content is identical; only the STORE instruction changes, requiring the model to remember either A or B and recall it later. (b) The input-dependent rate of internal-state evolution under the two instructions. (c–d) The model without internal-time modulation and the full TV-MCN are compared to show how strongly A, B, and irrelevant content are retained in the internal states. The left and right columns correspond to remembering A and B, respectively.

tion, and relevant content decays faster. TV-MCN thus adjusts the rate of state evolution according to the current input, selectively maintaining the historical information currently needed and using it at subsequent recall.